UncertaiNLP 2025

Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025)

Proceedings of the Workshop

November 9, 2025

The UncertaiNLP organizers gratefully acknowledge the support from the following sponsors.









©2025 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-349-4

Introduction

Human languages are inherently ambiguous and understanding language input is subject to interpretation and complex contextual dependencies. Nevertheless, the main body of research in NLP is still based on the assumption that ambiguities and other types of underspecification can and have to be resolved. This second edition of the Uncertainty-Aware NLP workshop (UncertaiNLP 2025) provides a platform for research that embraces variability in human language and aims to represent and evaluate the uncertainty that arises both from language itself and from the modeling tools we use to process it.

Uncertainty arises when multiple outcomes are possible and variability cannot be fully explained by context. In NLP, it reflects both epistemic and aleatoric factors, stemming from linguistic ambiguity, individual variation, domain shifts, and modeling choices. These challenges are especially acute in low-resource settings, where over- and under-fitting risks increase. Consequently, uncertainty-aware NLP research spans model design, data collection, inference, and evaluation, with growing importance in mission-critical applications requiring reliable confidence estimation. The success of the UncertaiNLP workshop's inaugural edition at EACL 2024 and the expanding community around related themes, underscores the growing research interest in this area and the need for continued exchange and collaboration. Building on that momentum, the second edition of UncertaiNLP nearly doubled in size, reflecting the field's rapid growth and the community's increasing recognition of the importance of uncertainty-aware methods in NLP.

This volume contains the proceedings of the second edition of the UncertaiNLP workshop hosted on November 9th, 2025, co-located with the 2025 Conference on Empirical Methods in Natural Language Processing in the Suzhou International Expo Centre in Suzhou, Jiangsu Province, China. We invited paper submissions on a wide variety of topics, including representing, documenting or modeling uncertainty, parameter estimation, probabilistic inference, decision making, evaluation and calibration, and hallucinations and uncertainty-driven mitigation. We received a total of 50 submissions, of which we accepted 17 long and 10 short papers, amounting to an acceptance rate of 54%

We are grateful to our invited keynote speakers: Gal Yona (Google Research, IL), Maxim Panov (MBZUAI, UAE), Parisa Kordjamshidi (Michigan State University, USA), Eyke Hüllermeier (LMU Münich, DE). We would also like to thank the EU's Horizon Europe research and innovation program for support through the Unified Transcription and Translation (UTTER, agreement No. 101070631) and the Foundation for Empirical Multimodality Research (FOUNDATIONS, agreement No. 101122047) projects. This workshop is also partially supported by an unrestricted gift from Google (Google research scholar award).

The UncertaiNLP organizers, Wilker Aziz, Jonathan Berant, Bryan Eikema, Marie-Catherine de Marneffe, Barbara Plank, Artem Shelmanov, Swabha Swayamdipta, Jörg Tiedemann, Raúl Vázquez, Chrysoula Zerva.

Program Committee

Program Chairs

Wilker Aziz, University of Amsterdam, Netherlands
Jonathan Berant, Tel Aviv University and Google DeepMind, Israel
Bryan Eikema, University of Amsterdam, Netherlands
Marie-Catherine de Marneffe, UCLouvain and FNRS, Belgium
Barbara Plank, LMU München and IT University of Copenhagen, Germany and Denmark
Artem Shelmanov, Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
Swabha Swayamdipta, University of Southern California, Viterbi School of Engineering, United States
Jörg Tiedemann, University of Helsinki, Finland
Raúl Vázquez, University of Helsinki, Finland
Chrysoula Zerva, Instituto de Telecomunicações, Portugal

Program Committee

Luigi Acerbi, University of Helsinki
Roee Aharoni, Google Research
Alessandro Antonucci, IDSIA
Henri Aïdasso, Université du Québec
Samuel Barry, Mistral AI
Nitay Calderon, Technion
Juan Cardenas-Cartagena, University of Groningen
Arie Cattan, Bar Ilan University
Julius Cheng, University of Cambridge
Ye-eun Cho, Sungkyunkwan University
Caio Corro, INSA Rennes
Nico Daheim, Technische Universität Darmstadt
Sarkar Snigdha Sarathi Das, Pennsylvania State University
Vivek Datla, Capital One
Bonaventure F. P. Dossou, McGill University
Adam Faulkner, Capital One
Pedro Lobato Ferreira, University of Amsterdam
Antske Fokkens, VU University Amsterdam
Jes Frellsen, Technical University of Denmark
Thomas L. Griffiths, Princeton University
Georg Groh, Technical University Munich
Christian Hardmeier, IT University Copenhagen
Zhiqi Huang, CapitalOne
Evgenia Ilia, University of Amsterdam
Yuu Jinnai, CyberAgent, Inc.
Mucheol Kim, Chung-Ang University
Deepak Kumar, Infrrd
Lucie Kunitomo-Jacquin, AIST
Haau-Sing Li, Technische Universität Darmstadt
Edison Marrese-Taylor, University of Tokyo
Yan Meng, University of Amsterdam

Timothee Mickus, University of Helsinki
Tatiana Passali, Aristotle University of Thessaloniki
Laura Perez-Beltrachini, University of Edinburgh
Yuval Pinter, Ben-Gurion University of the Negev
Timothy Pistotti, University of Auckland
Alberto Purpura, CapitalOne
Julian Rodemann, LMU Munich
Rico Sennrich, University of Zürich
Arnab Sharma, Paderborn University
Anthony Sicilia, Northeastern University
Edwin Simpson, University of Bristol
Maciej Skorski, University of Warsaw
Sharmin Sultana, University of Massachusetts at Lowell
Aarne Talman, University of Helsinki
Sergey Troshin, University of Amsterdam
Grigorios Tsoumakas, Aristotle University of Thessaloniki
Dennis Ulmer, University of Amsterdam
Teemu Vahtola, University of Helsinki
Matias Valdenegro-Toro, University of Groningen
Sami Virpioja, University of Helsinki
Daniel Vollmers, Paderborn University
Ryan Wails, Georgetown University
Li Wang, CapitalOne
Di Wu, University of Amsterdam
Yusen Zhang, Pennsylvania State University

Publication Chairs

Raúl Vázquez
Bryan Eikema

Invited Speakers

Parisa Kordjamshidi, Michigan State University, USA
Gal Yona, Google Research, USA
Maxim Panov, MBZUAI, UAE
Eyke Hüllermeier, LMU Münich, Germany

Keynote Talk
TBA

Maxim Panov
MBZUAI, UAE
2025-09-11 09:10 – Room: **Room A207**

Abstract: TBA

Bio: Maxim Panov is an Assistant Professor at MBZUAI, UAE. Before joining MBZUAI, Panov worked as a research scientist at DATADVANCE Company, where he participated in developing a library of data analysis methods for engineering applications. This library, pSeven, is now used by many companies worldwide, including Airbus, Porsche, Mitsubishi, Toyota, and Limagrain. From 2018, Panov has been an assistant professor at Skolkovo Institute of Science and Technology, Moscow, where he led a statistical machine learning group. Since 2022, he has led an AI theory and algorithms group at the Technology Innovation Institute, Abu Dhabi, UAE. His research interests lie in uncertainty quantification for machine learning model predictions and Bayesian approaches in machine learning. Maxim is leading a research team dedicated to exploring the theoretical foundations of uncertainty quantification and its practical applications. Maxim is also co-leading the development of the LM-Polygraph framework for uncertainty quantification for LLMs. Maxim was a local chair for the ICDM 2024 conference and a recipient of the Best Paper Runner-up Award at the Uncertainty in Artificial Intelligence 2023 conference.

Keynote Talk

Reasoning under Uncertainty with Large Multimodal Language Models

Parisa Kordjamshidi
Michigan State University, USA
2025-09-11 13:15 – Room: **Room A207**

Abstract: Uncertainty in intelligent models has multiple facets. One aspect concerns a model's own uncertainty or confidence in its generated outputs. Another pertains to factual knowledge about uncertainty within specific concepts. For example, statements such as "10–20% of lifelong smokers will develop lung cancer" express factual uncertainty derived from statistical data analyses and represented in text. A key research question is whether language models can form and convey such factual uncertainties—integrating information, drawing on their internal knowledge, and aligning this with their confidence when expressing opinions. While addressing this question is highly challenging, I will present our research that explores related directions and the following research question: 1) How do language models understand uncertainty expressions in natural language and perform probabilistic inference over them? 2) How can models be trained to follow the principles of probabilistic reasoning when handling uncertainty in text? 3) How can today's large models reason over uncertain text? specifically focusing on mapping language into formal probabilistic logic programs?, and finally, in the context of grounding natural language in the visual modality, 4) How can uncertainty in perception be explicitly represented in reasoning? specifically focusing on mappings to differentiable probabilistic programs.

Bio: Parisa Kordjamshidi is an Associate Professor of Computer Science and Engineering at Michigan State University. Her research focuses on Natural Language Processing, multimodal reasoning across vision and language, and neuro-symbolic learning. She received her Ph.D. from KU Leuven and conducted postdoctoral research at the University of Illinois Urbana-Champaign. She is a recipient of the NSF CAREER, Amazon Faculty Research, and Fulbright Scholar Awards, and her research team received the NAACL 2025 Outstanding Research Paper Award. Dr. Kordjamshidi serves as Associate Editor of JAIR, Co-editor in chief of ARR (2026), Action Editor for TACL and has held roles in organization committee of major conferences including ACL, NAACL, EACL, EMNLP, ECML-PKDD, and AAAI. Currently, she is a visiting Associate Professor at UCLA spending a part of her sabbatical.

Keynote Talk

Beyond Factuality: Improving Trust and Reliablility of Large Language Models

Gal Yona
Google Research, USA
2025-09-11 14:45 – Room: **Room A207**

Abstract: Factuality is a cornerstone for trustworthy LLMs, yet despite impressive progress, frontier LLMs still make many confident errors when faced with questions beyond their knowledge boundaries. In this talk I'll present Faithful Response Uncertainty, a different desiderata that shifts the focus away from measuring the number of incorrect statements and towards measuring the alignment between the model's expressed certainty (decisiveness") and intrinsic certainty (confidence"). I'll conclude with a discussion of open problems and possible next steps at the intersection of factuality and uncertainty in frontier LLMs.

Bio: Gal Yona is a Research Scientist at Google Research, Tel Aviv, where she is working on improving factuality in large language models, with an emphasis on robustness and uncertainty. Before joining Google, Gal completed her PhD in Computer Science at the Weizmann Institute of Science, developing definitions and algorithms for preventing discrimination in machine learning models. Gal received numerous award during her PhD, including the Google PhD Fellowship in Machine Learning (2021).

Keynote Talk

Challenges in Uncertainty Quantification for Large Language Models

Eyke Hüllermeier
LMU Münich, Germany
2025-09-11 16:00 – Room: **Room A207**

Abstract: Uncertainty quantification is important in the context of large language models (LLMs) because the outputs produced by these models are often incorrect. However, due to the complexity of language and the numerous sources of uncertainty in textual data, quantifying uncertainty in LLMs is challenging. Indeed, simply transferring existing approaches to uncertainty quantification developed for standard machine learning problems, such as classification and regression, is neither straightforward nor appropriate. This is particularly pertinent to the definition of aleatoric and epistemic uncertainty, and how they are distinguished based on the notion of reducibility. This talk will discuss the challenges of uncertainty quantification for LLMs, propose potential solutions and highlight promising avenues for future research in this emerging field.

Bio: Eyke Hüllermeier is a full professor at the Institute of Informatics at LMU Munich, Germany, where he holds the Chair of Artificial Intelligence and Machine Learning. He studied mathematics and business computing, received his PhD in Computer Science from Paderborn University in 1997, and a Habilitation degree in 2002. Before joining LMU, he held professorships at several other German universities (Dortmund, Magdeburg, Marburg, Paderborn) and spent two years as a Marie Curie fellow at the IRIT in Toulouse (France). His research interests are centered around methods and theoretical foundations of artificial intelligence, with a particular focus on machine learning, preference modeling, and reasoning under uncertainty. He has published more than 400 articles on related topics in top-tier journals and major international conferences, and several of his contributions have been recognized with scientific awards. Professor Hüllermeier is Editor-in-Chief of Data Mining and Knowledge Discovery, Associate Editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), and serves on the editorial boards of several other AI and machine learning journals. He is currently also the president of EuADS, the European Association for Data Science.

Table of Contents

Uncertainty-driven Partial Diacritization for Arabic Text
Humaid Ali Alblooshi, Artem Shelmanov and Hanan Aldarmaki 1

Phases of Uncertainty: Confidence–Calibration Dynamics in Language Model Training
Aneesh Durai 11

Beyond Human Judgment: A Bayesian Evaluation of LLMs' Moral Values Understanding
Maciej Skorski and Alina Landowska 17

Do Large Language Models Know When Not to Answer in Medical QA?
Sravanthi Machcha, Sushrita Yerra, Sharmin Sultana, hong yu and Zonghai Yao 27

The Geometry of Creative Variability: How Credal Sets Expose Calibration Gaps in Language Models
Esteban Garces Arias, Julian Rodemann and Christian Heumann 36

Certain but not Probable? Differentiating Certainty from Probability in LLM Token Outputs for Probabilistic Scenarios
Autumn Toney and Ryan Wails 51

The Benefits of Being Uncertain: Perplexity as a Signal for Naturalness in Multilingual Machine Translation
Timothy Pistotti, Michael J. Witbrock, Dr Padriac Amato Tahua O'Leary and Jason Brown . . . 61

Asking a Language Model for Diverse Responses
Sergey Troshin, Irina Saparina, Antske Fokkens and Vlad Niculae 66

Learning to vary: Teaching LMs to reproduce human linguistic variability in next-word prediction
Tobias Groot, Salo Lacunes and Evgenia Ilia 73

HALLUCINOGEN: Benchmarking Hallucination in Implicit Reasoning within Large Vision Language Models
Ashish Seth, Dinesh Manocha and Chirag Agarwal 89

Uncertainty in Semantic Language Modeling with PIXELS
Stefania Radu, Marco Zullich and Matias Valdenegro-Toro 103

Confidence Calibration in Large Language Model-Based Entity Matching
Iris Kamsteeg, Juan Cardenas-Cartagena, Floris van Beers, Tsegaye Misikir Tashu and Matias Valdenegro-Toro 120

Consensus or Conflict? Fine-Grained Evaluation of Conflicting Answers in Question-Answering
Eviatar Nachshoni, Arie Cattan, Shmuel Amar, Ori Shapira and Ido Dagan 138

Demystify Verbosity Compensation Behavior of Large Language Models
Yusen Zhang, Sarkar Snigdha Sarathi Das and Rui Zhang 160

On the Role of Unobserved Sequences on Sample-based Uncertainty Quantification for LLMs
Lucie Kunitomo-Jacquin, Edison Marrese-Taylor and Ken Fukuda 179

Confidence-Based Response Abstinence: Improving LLM Trustworthiness via Activation-Based Uncertainty Estimation
Zhiqi Huang, Vivek Datla, Chenyang Zhu, Alfy Samuel, Daben Liu, Anoop Kumar and Ritesh Soni 184

Amortized Bayesian Meta-Learning for Low-Rank Adaptation of Large Language Models
Liyi Zhang, Jake C. Snell and Thomas L. Griffiths 194

Towards Trustworthy Summarization of Cardiovascular Articles: A Factuality-and-Uncertainty-Aware Biomedical LLM Approach
Eleni Partalidou, Tatiana Passali, Chrysoula Zerva, Grigorios Tsoumakas and Sophia Ananiadou 200

Causal Understanding by LLMs: The Role of Uncertainty
Oscar William Lithgow-Serrano, Vani Kanjirangat and Alessandro Antonucci 208

It Depends: Resolving Referential Ambiguity in Minimal Contexts with Commonsense Knowledge
Lukas Ellinger and Georg Groh 229

Read Your Own Mind: Reasoning Helps Surface Self-Confidence Signals in LLMs
Jakub Podolak and Rajeev Verma 247

Calibrating Language Models for Neural Ranking under Noisy Supervision with Relaxed Labels
Arnab Sharma, Daniel Vollmers and Axel-Cyrille Ngonga Ngomo 259

ERGO: Entropy-guided Resetting for Generation Optimization in Multi-turn Language Models
Haziq Mohammad Khalid, Athikash Jeyaganthan, Timothy Do, Yicheng Fu, Vasu Sharma, Sean O'Brien and Kevin Zhu 273

Towards Open-Ended Discovery for Low-Resource NLP
Bonaventure F. P. Dossou and Henri Aïdasso 287

Can Vision-Language Models Infer Speaker's Ignorance? The Role of Visual and Linguistic Cues
Ye-eun Cho and Yunho Maeng 298

DeLTa: A Decoding Strategy based on Logit Trajectory Prediction Improves Factuality and Reasoning Ability
Yunzhen He, Yusuke Takase and Hidetoshi Shimodaira 309

Investigating Factuality in Long-Form Text Generation: The Roles of Self-Known and Self-Unknownan
Lifu Tu, Rui Meng, Shafiq Joty, Yingbo Zhou and Semih Yavuz 322

Program

Sunday, November 9, 2025

09:00 - 09:10 *Opening Remarks*

09:10 - 09:55 *Keynote Talk 1:* ***Maxim Panov***

09:55 - 10:30 *Poster lightning round 1 (in-person)*

10:30 - 11:00 *Coffee Break*

11:00 - 12:15 *In-Person Poster Session*

12:15 - 13:15 *Lunch Break*

13:15 - 14:00 *Keynote Talk 2:* ***Parisa Kordjamshidi***

14:00 - 14:45 *Poster lightning round 2 (virtual)*

14:45 - 15:30 *Keynote Talk 3:* ***Gal Yona***

15:30 - 16:00 *Coffee Break*

16:00 - 16:45 *Keynote Talk 4:* ***Eyke Hüllermeier***

16:45 - 17:00 *Closing Remarks*

Detailed view of lightning rounds

09:30 - 10:20 *Poster lightning round 1 (in-person)*

Uncertainty-driven Partial Diacritization for Arabic Text
Humaid Ali Alblooshi, Artem Shelmanov and Hanan Aldarmaki

The Geometry of Creative Variability: How Credal Sets Expose Calibration Gaps in Language Models
Esteban Garces Arias, Julian Rodemann and Christian Heumann

The Benefits of Being Uncertain: Perplexity as a Signal for Naturalness in Multilingual Machine Translation
Timothy Pistotti, Michael J. Witbrock, Dr Padriac Amato Tahua O'Leary and Jason Brown

Asking a Language Model for Diverse Responses
Sergey Troshin, Irina Saparina, Antske Fokkens and Vlad Niculae

Learning to vary: Teaching LMs to reproduce human linguistic variability in next-word prediction
Tobias Groot, Salo Lacunes and Evgenia Ilia

Consensus or Conflict? Fine-Grained Evaluation of Conflicting Answers in Question-Answering
Eviatar Nachshoni, Arie Cattan, Shmuel Amar, Ori Shapira and Ido Dagan

Confidence-Based Response Abstinence: Improving LLM Trustworthiness via Activation-Based Uncertainty Estimation
Zhiqi Huang, Vivek Datla, Chenyang Zhu, Alfy Samuel, Daben Liu, Anoop Kumar and Ritesh Soni

Causal Understanding by LLMs: The Role of Uncertainty
Oscar William Lithgow-Serrano, Vani Kanjirangat and Alessandro Antonucci

It Depends: Resolving Referential Ambiguity in Minimal Contexts with Commonsense Knowledge
Lukas Ellinger and Georg Groh

Read Your Own Mind: Reasoning Helps Surface Self-Confidence Signals in LLMs
Jakub Podolak and Rajeev Verma

Calibrating Language Models for Neural Ranking under Noisy Supervision with Relaxed Labels
Arnab Sharma, Daniel Vollmers and Axel-Cyrille Ngonga Ngomo

ERGO: Entropy-guided Resetting for Generation Optimization in Multi-turn Language Models
Haziq Mohammad Khalid, Athikash Jeyaganthan, Timothy Do, Yicheng Fu, Vasu Sharma, Sean O'Brien and Kevin Zhu

Can Vision-Language Models Infer Speaker's Ignorance? The Role of Visual and Linguistic Cues
Ye-eun Cho and Yunho Maeng

DeLTa: A Decoding Strategy based on Logit Trajectory Prediction Improves Factuality and Reasoning Ability
Yunzhen He, Yusuke Takase and Hidetoshi Shimodaira

14:00 - 14:45 *Poster lightning round 2 (virtual)*

Phases of Uncertainty: Confidence–Calibration Dynamics in Language Model Training
Aneesh Durai

Beyond Human Judgment: A Bayesian Evaluation of LLMs' Moral Values Understanding
Maciej Skorski and Alina Landowska

Do Large Language Models Know When Not to Answer in Medical QA?
Sravanthi Machcha, Sushrita Yerra, Sharmin Sultana, hong yu and Zonghai Yao

Certain but not Probable? Differentiating Certainty from Probability in LLM Token Outputs for Probabilistic Scenarios
Autumn Toney and Ryan Wails

HALLUCINOGEN: Benchmarking Hallucination in Implicit Reasoning within Large Vision Language Models
Ashish Seth, Dinesh Manocha and Chirag Agarwal

Uncertainty in Semantic Language Modeling with PIXELS
Stefania Radu, Marco Zullich and Matias Valdenegro-Toro

Confidence Calibration in Large Language Model-Based Entity Matching
Iris Kamsteeg, Juan Cardenas-Cartagena, Floris van Beers, Tsegaye Misikir Tashu and Matias Valdenegro-Toro

Demystify Verbosity Compensation Behavior of Large Language Models
Yusen Zhang, Sarkar Snigdha Sarathi Das and Rui Zhang

On the Role of Unobserved Sequences on Sample-based Uncertainty Quantification for LLMs
Lucie Kunitomo-Jacquin, Edison Marrese-Taylor and Ken Fukuda

Amortized Bayesian Meta-Learning for Low-Rank Adaptation of Large Language Models
Liyi Zhang, Jake C. Snell and Thomas L. Griffiths

Towards Trustworthy Summarization of Cardiovascular Articles: A Factuality-and-Uncertainty-Aware Biomedical LLM Approach
Eleni Partalidou, Tatiana Passali, Chrysoula Zerva, Grigorios Tsoumakas and Sophia Ananiadou

Towards Open-Ended Discovery for Low-Resource NLP
Bonaventure F. P. Dossou and Henri Aïdasso

Investigating Factuality in Long-Form Text Generation: The Roles of Self-Known and Self-Unknownan
Lifu Tu, Rui Meng, Shafiq Joty, Yingbo Zhou and Semih Yavuz

Uncertainty-Driven Partial Diacritization for Arabic Text

Humaid Alblooshi, Artem Shelmanov, Hanan Aldarmaki
Department of Natural Language Processing
Mohamed Bin Zayed University of Artificial Intelligence, UAE
{Humaid.alblooshi, Artem.Shelmanov, Hanan.Aldarmaki}@mbzuai.ac.ae

Abstract

We propose an uncertainty-based approach to Partial Diacritization (PD) for Arabic text. We evaluate three uncertainty metrics for this task: Softmax Response, BALD via MC-dropout, and Mahalanobis Distance. We further introduce a lightweight Confident Error Regularizer to improve model calibration. Our preliminary exploration illustrates possible ways to use uncertainty estimation for selectively retaining or discarding diacritics in Arabic text with an analysis of performance in terms of correlation with diacritic error rates. For instance, the model can be used to detect words with high diacritic error rates which tend to have higher uncertainty scores at inference time. On the Tashkeela dataset, the method maintains low diacritic error rates while reducing the amount of visible diacritics on the text by up to 50% with thresholding-based retention.

1 Introduction

Arabic script relies on diacritics, commonly referred to in Arabic as Tashkeel (تَشْكِيل) to mark short vowels, gemination, and other phonemic distinctions that may not be represented by the base letters. Fully-diacritized text eliminates ambiguity and supports precise pronunciation, which is helpful for applications such as text-to-speech (TTS) synthesis, machine translation, and language learning (Mubarak et al., 2019; Lameris, 2021). However, when every letter carries its diacritic, the resulting text becomes visually dense and can slow down readers (ElNokrashy and AlKhamissi, 2024; Roman and Pavard, 1987). Partial diacritization can be employed to balance disambiguation and readability and to optimize performance in downstream NLP applications.

State-of-the-art transformer-based diacritization models can achieve diacritic error rates (DER) below 2% on standard benchmarks (Assad et al., 2024), but their performance may degrade in out-of-domain data (Toyin et al., 2025a). These models often operate as "black boxes," where model outputs are accepted regardless of confidence scores. Their outputs are often fully diacritized, which increases visual complexity and can slow down reading speed and reduce clarity. Furthermore, many diacritics are redundant, particularly in common words where pronunciation is intuitive or easily inferred. These factors make Fully Diacritized (FD) text less practical for general application, motivating the need for Partially Diacritized (PD) text in such settings. Prior studies proposed computational approaches that rely on heuristics and morphological analysis to perform partial diacritization (Diab et al., 2007; Alqahtani et al., 2019). Others have proposed neural networks (Fadel et al., 2019) with some success. However, research on partial diacritization remains limited, largely due to the difficulty of evaluating performance; optimal partial diacritization is an illusive concept with no standard evaluation framework or metrics.

In this paper, we explore an uncertainty-driven framework for PD and provide a preliminary intrinsic evaluation of this framework through error analysis. We evaluate three uncertainty metrics: Softmax Response, Bayesian Active Learning by Disagreement (BALD) via Monte Carlo dropout, and Mahalanobis distance in latent feature space. At inference time, the predicted diacritic of each character is compared with a chosen threshold θ. We may keep the diacritic if the uncertainty score is above or below said threshold, allowing for flexibility in application. To mitigate the well-known overconfidence of deep networks on rare or ambiguous inputs, we experiment with a lightweight, simplified Confident Error Regularizer (CER) that penalizes high-confidence mistakes during fine-tuning. We summarize our contributions as follows:

- We propose and formalize the application of per-character uncertainty metrics for PD and

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 1–10
November 9, 2025 ©2025 Association for Computational Linguistics

illustrate an intrinsic performance evaluation of this framework using a recent state-of-the-art neural diacritic restoration model.

- We propose an efficient calibration method and apply it on our base diacritic restoration model. We show that the approach improves uncertainty estimation at the cost of lower accuracy, while being computationally efficient.

- We discuss the potential downstream applications of such partial diacritization schemes and highlight areas that need further analysis and improvement.

2 Related Work

Early approaches to Arabic diacritization employed hidden Markov models and morphological analyzers such as MADA and MADAMIRA (Habash et al., 2005), Habash and Rambow (2005), Habash and Rambow (2007), achieving good accuracy by leveraging lexical features and large amounts of data. With the advent of neural networks, recurrent architectures and encoder–decoder transformers improved DER to below 5%. The Character-based Arabic Tashkeel Transformer (CATT) (**?**) is one such innovation, achieving very good results on both its variants, encoder only (EO) and encoder-decoder (ED).

Partial diacritization has been studied from linguistic and machine learning perspectives with the aim of improving both NLP systems as well as improving text readability. Rule-based schemes target case endings or homograph disambiguation, while supervised methods train classifiers to decide which positions to diacritize. Diab et al. (2007) investigated the impact of various diacritization schemes on Statistical Machine Translation (SMT) from Arabic to English. The authors explored different levels of partial diacritization and found that partial diacritization could improve translation quality by reducing ambiguity without significantly increasing vocabulary size or out-of-vocabulary rates. Alqahtani et al. (2016) demonstrated improvements in machine translation by employing partial diacritization strategies targeting syntactic clarity. Building on these findings, the authors also employed selective diacritic restoration specifically for homograph disambiguation (Alqahtani et al., 2019). Fadel et al. (2019) further advanced PD research by achieving state-of-the-art results and seamless integration into machine translation workflows. Qin et al. (2021) introduced regularized decoding and adversarial training to improve diacritization robustness and accuracy. Recently, Elgamal et al. (2024) analyzed naturally occurring instances of partial diacritics across diverse text genres, creating practical datasets for enhanced real-world applications.

Our contribution in relation to related work Existing PD methods depend on heuristics or linguistic context to identify words or characters for partial diacritization. None of the existing approaches leverage model uncertainty estimation techniques, which have been shown to be instrumental in other areas of application, such as computer vision (Kendall and Gal, 2017), (Lee et al., 2018), and machine translation (Pereyra et al., 2017). In this paper, we introduce the application of uncertainty estimation methods for Arabic diacritization and contribute a preliminary exploration of uncertainty metrics and performance in terms of diacritic error rates. Through this exploratory analysis, we present a case for the potential of uncertainty estimation as a viable computational approach towards partial diacritization.

3 Methodology

Our methodology for partial diacritization is to use model uncertainty to guide the removal or retention of diacritics based on target criteria. For example, if we have a fully diacritized text and wish to minimize the diacritics for improved readability, uncertainty scores may be helpful in identifying which diacritics to retain by keeping the ground truth diacritics in places with high model uncertainty. In applications where a diacritic restoration model is used directly to annotate undiacritized text, we may wish to remove predicted diacritics with high uncertainty and maintain low diacritic error rates in the resulting text. Our methodology and preliminary analysis enable both types of application by exploring the relationship between uncertainty scores and diacritic error rates. In the following sections, we describe the base model, uncertainty metrics, and the calibration scheme used to improve uncertainty estimation for diacritic restoration.

3.1 Task Formulation and Diacritic Restoration Models

Arabic diacritic restoration can be cast as a sequence-labeling problem. Given an undiacritized character sequence $\mathbf{x} = (x_1, \ldots, x_n)$, we pre-

dict a diacritic sequence $\mathbf{y} = (y_1, \ldots, y_n)$, where $y_i \in \mathcal{V}_{\text{diac}}$ (including a "no-diacritic" symbol) is a label for x_i.

We use a recent character-based transformer model for diacritic restoration, CATT (**?**), which supports both encoder-only and encoder-decoder configurations. We primarily use the encoder-decoder model in experiments, as it's shown to perform better in **?**. Both architectures are described below.

Encoder-Only (EO) is a transformer encoder θ_{enc} with a linear classification head for sequence labeling with parameters $\mathbf{W}_{\text{cls}}$ and a bias term $\mathbf{b}_{\text{cls}}$. Each position is classified independently conditioned on the entire input:

$$\mathbf{h} = \text{Encoder}(\mathbf{x}; \theta_{\text{enc}}) \quad (1)$$

$$p(y_i|\mathbf{x}) = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_i + \mathbf{b}_{\text{cls}}) \quad (2)$$

Encoder-Decoder (ED) is a full transformer architecture with autoregressive decoding. We denote the parameters of the decoder as θ_{dec}. We view the task as monotonic character-to-diacritic translation with the standard autoregressive factorization:

$$\mathbf{h}_{\text{enc}} = \text{Encoder}(\mathbf{x}; \theta_{\text{enc}}) \quad (3)$$

$$\mathbf{h}_{\text{dec}} = \text{Decoder}(\mathbf{y}_{<i}, \mathbf{h}_{\text{enc}}; \theta_{\text{dec}}) \quad (4)$$

$$p(y_i|\mathbf{x}, \mathbf{y}_{<i}) = \text{softmax}(\mathbf{W}_{\text{cls}}\mathbf{h}_{\text{dec},i} + \mathbf{b}_{\text{cls}}) \quad (5)$$

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{i=1}^{n} P(y_i \mid \mathbf{x},\, y_1, \ldots, y_{i-1})\,. \quad (6)$$

3.2 Uncertainty Scores

We denote the model's categorical probability output at a given character position by $p(y \mid x)$ and all parameters of the model as θ. The **Softmax Response (SR)** uncertainty is defined as:

$$U_{\text{SR}}(x) = 1 - \max_y P(y|x, \theta). \quad (7)$$

Softmax Response, (Hendrycks and Gimpel, 2017) measures model confidence explicitly, and is usually a simple baseline for alaetoric uncertainty: the inherent ambiguity of a task due to noise or multiple valid answers.

To capture epistemic uncertainty, we apply **Monte Carlo dropout** at inference time over T stochastic forward passes, obtaining distributions $p_t(y \mid x)\ t = 1 \ldots T$. This score is the difference between predictive entropy and expected entropy, termed **BALD** (Bayesian Active Learning by Disagreement) (Houlsby et al., 2011):

$$U_{\text{BALD}}(x) = H[P(y|x, \theta)] - \mathbb{E}_{q(\theta)}H[P(y|x, \theta)] \quad (8)$$

where:

- $H[P(y|x, \theta)]$ is the total uncertainty (entropy of the predictive distribution),
- $\mathbb{E}_{q(\theta)}H[P(y|x, \theta)]$ is the expected entropy over the posterior distribution of the model parameters, capturing the irreducible (aleatoric) uncertainty.

A higher BALD score indicates greater disagreement among stochastic forward passes, meaning the model lacks knowledge and would benefit from additional training on similar samples.

Finally, **Mahalanobis Distance (MD)** (Lee et al., 2018) is computed on the penultimate layer features $f(x) \in \mathbb{R}^d$, with a precomputed centroid for the whole training set μ and a covariance matrix Σ:

$$U_{\text{MD}}(x) = \sqrt{(f(x) - \mu)^T \Sigma^{-1} (f(x) - \mu)}. \quad (9)$$

Higher MD values indicate that a sample may be out-of-distribution, suggesting that the model has not encountered similar instances during training. For example, a rarely-used Arabic word or a foreign loanword transcribed in Arabic script could have a high MD score. MD is a strong epistemic uncertainty metric, since uncertainty in these instances is due to complete lack of representation rather than ambiguity.

3.3 Selective Diacritization

We propose uncertainty-based partial diacritization as follows. At inference time, we compare the uncertainty for each character position, $U(x)$, against a pre-defined threshold, τ. Depending on our objective, we can:

- Retain high confidence diacritics, where $U(x) < \tau$. This can be applied in settings where automatic diacritic restoration is used to annotate undiacritized text, and highly accurate partial diacritization is preferred over full diacritization.
- Retain low confidence diacritics, where $U(x) > \tau$. This can be used for applications where ground truth diacritics are available, and partial diacritics are sought to identify ambiguous words; for instance, in reading applications to help casual readers disambiguate difficult, ambiguous cases while maintaining minimal diacritics overall to reduce cognitive load. This approach could also be used to identify which subset of diacritics to manually annotate in an active learning framework.

By sweeping τ over $[0, 1]$, we can trace a DER–coverage curve that illustrates the trade-off between error rate and annotation effort.

3.4 Calibration via Confident Error Regularizer

Deep models often assign high confidence to incorrect predictions, which could compromise the application of uncertainty in PD as described above. To address this, we augment the standard cross-entropy loss L_{CE} with a penalty on high-confidence errors.

Xin et al. (2021) proposed the Confident Error Regularizer (CER) to add a penalty for an instance with a bigger loss than other instances and, at the same time, bigger confidence:

$$\mathcal{L}_{\text{CER}} = \sum_{i,j=1}^{k} \Delta_{i,j}\, \mathbb{I}[e_i > e_j] \tag{10}$$

$$\Delta_{i,j} = \left(\max\{0, \max p_i^c - \max p_j^c\}\right)^2 \tag{11}$$

where k is the number of instances in a batch and e_i is an error of the i-th instance: e_i is 1 if the prediction of the classifier matches the true label, and is 0 otherwise. p_i and p_j are the probabilities of these specific datapoints. The authors evaluate this type of regularization only in conjunction with the SR baseline to good results. CER is based on the principle that a well-calibrated model should assign lower confidence to incorrect predictions than to correct ones, and vice versa.

In our implementation, we adopt a simplified version of the CER that maintains the core concept while reducing computational complexity. Instead of using pairwise comparisons between all instances in a batch as in the original formulation, our approach directly penalizes high confidence on incorrect predictions with high confidence only:

$$\mathcal{L}_{\text{CER}} = \frac{\sum_{i=1}^{n} \max(p_i) \cdot \mathbb{I}[y_i \neq \hat{y}_i] \cdot m_i}{\sum_{i=1}^{n} \mathbb{I}[y_i \neq \hat{y}_i] \cdot m_i + \epsilon} \tag{12}$$

where n is the total number of tokens, $\max(p_i)$ is the maximum probability (confidence) for token i, $\mathbb{I}[y_i \neq \hat{y}_i]$ is an indicator function that equals 1 when the prediction is incorrect and 0 otherwise, m_i is a mask to ignore padding tokens, and ϵ is a small constant to avoid division by zero.

This regularization loss is then added to the task-focused cross-entropy loss $\mathcal{L}_{\text{CE}}$. The additive total loss function is then:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{CER}}. \tag{13}$$

with λ as a regularization strength hyperparameter.

4 Experiments

This section presents exploratory and experimental analysis of our approach. We analyze the performance of different uncertainty estimation methods, with a particular focus on SR as our primary score, a choice we justify below. We also evaluate the impact of confidence calibration through CER, not to be confused with Character Error Rate, with different regularization strengths. The analysis addresses several key aspects: the relationship between uncertainty thresholds and diacritization coverage, error rates at different thresholds of uncertainty, the effectiveness of uncertainty in identifying difficult words, and the calibration quality of the model with and without CER.

4.1 Datasets

The **Tashkeela Corpus (Zerrouki and Balla, 2017)** is the primary dataset usedfor training the base CATT model. We use it for fine-tuning and in-domain evaluation. The dataset contains over 75 million words of fully diacritized text, derived from classical Arabic books, religious texts, and modern Arabic educational material. We apply filtering to remove lines with less than a 60% diacritization ratio for finetuning. Since Tashkeela is a large dataset, we only fine-tuned on 10% of the data, split into an 80/20 for fine-tuning and validation.

ArVoice (Toyin et al., 2025b) is a multi-speaker Modern Standard Arabic (MSA) speech corpus with fully diacritized text transcriptions, intended for multi-speaker speech synthesis. The complete corpus consists of a total of 83.52 hours of speech across 11 voices. Since most of the ArVoice text is derived from Tashkeela, we use only the ASC subset, which is derived from the Arabic Speech Corpus (Halabi, 2016). This serves as a challenging out-of-domain test set.

4.2 Base Model without Regularization

To start the analysis, we will go over some studies on the base model itself to establish a few key points and trends, then move on to the calibration effect on key metrics, and what insights can be pulled from those differences.

4.2.1 Relationship Between Uncertainty Threshold and Diacritic Coverage

As we increase the threshold τ used to retain diacritics in the base model, we keep more diacritics, in the case that we choose to keep the ones below

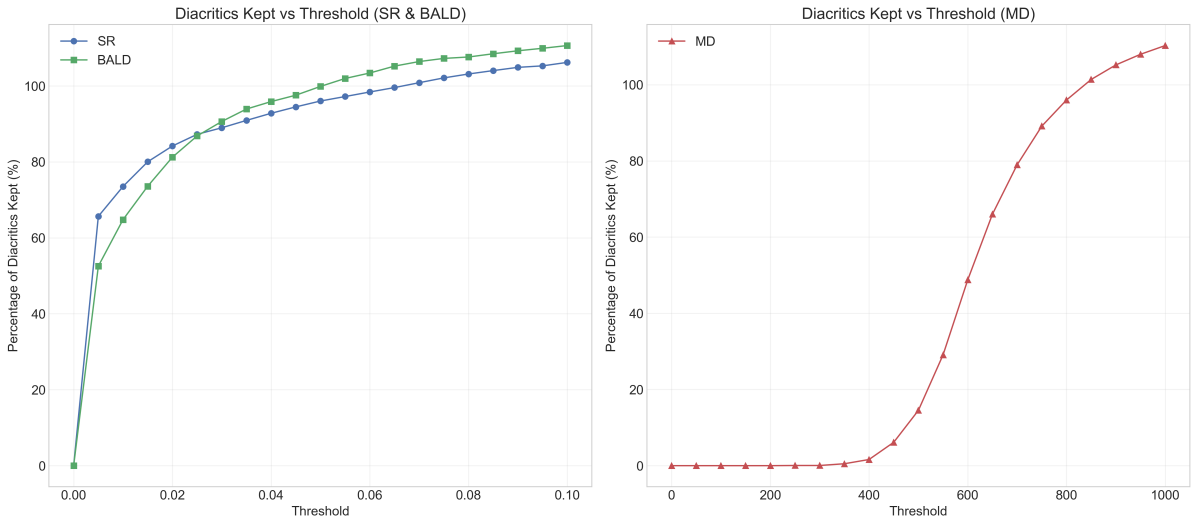


Figure 1: Percentage of diacritics kept vs. uncertainty threshold for SR and BALD (left) and MD (right) using the base model. Results are based on the **Tashkeela test set**. We plot MD separately due to the large difference in scale compared to SR and BALD. **Note**:The percentage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

τ. We illustrate the relationship between uncertainty thresholds and the percentage of diacritics retained in Figure 1. Naturally, the figures follow a cumulative pattern where all diacritics are kept at the maximum uncertainty threshold.[1] The figures help identify the threshold values needed to retain a specific percentage of diacritics.

4.2.2 Relationship Between Error Rate and Diacritics Kept

While the previous section demonstrates how to select a threshold based on the percentage of desired diacritics, a more practical approach is to select a threshold based on optimal diacritic coverage and error rates. Figure 2 shows the relationship between the percentage of diacritics retained using the base model, and the resulting Diacritic Error Rate (DER) for our three uncertainty estimation metrics: SR, BALD, and MD. Fewer diacritics are favorable in a practical reading setting, since there is less visual noise to go through and less disambiguation needed. As such, keeping the smallest number of diacritics possible while retaining the lowest Diacritic Error Rate "DER" is desirable in this context. We calculate DER relative to the number of total diacritics kept.

As the illustration does not rely on absolute threshold value, we gain the advantage of visualizing the three metrics in the same scale. In the same figure, one can see that SR and BALD exhibit similar trends, with a gradual increase in error rate as more diacritics are kept. At 80% diacritization coverage, both methods maintain a relatively low error rate (approximately 2.5%, or 50% absolute reduction in error rates) after removing 20% of diacritics that have high uncertainty in the base model. In contrast, MD shows a sharper increase in error rate, suggesting that it may be unsuitable for this task. This indicates that the model's confidence (as measured by SR and BALD) is well correlated with its accuracy, making it an effective guide for partial diacritization. SR seems less prone to errors than BALD, though not significantly. SR is also much faster to compute than BALD in our encoder-decoder diacritic restoration model since MC dropout passes need to be computed for every token the decoder generates, leading to huge computational overhead. As such, SR will be chosen as the main metric of focus in the remaining analysis due to its computational efficiency and good correlation with error rates.

4.3 Confident Error Regularization

While the base model is shown to be effective at identifying many errors through uncertainty scores, effectively reducing error rates by 50% while maintaining 80% of diacritics, we still have many instances where the model uncertainty scores do not track performance, especially in the out-of-

[1]The model predicts more diacritics than the reference ground truth, making the results go above 100% at the extremes due to insertions it makes.

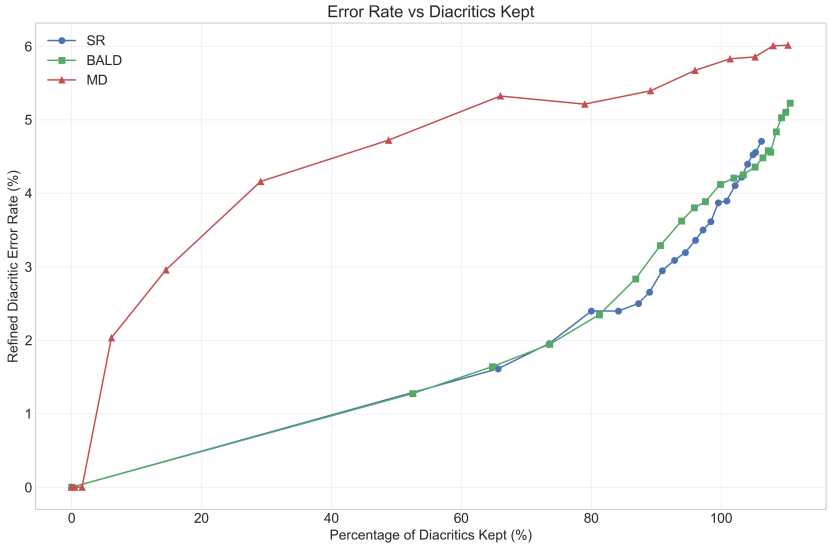


Figure 2: Diacritic Error Rate vs. Percentage of Diacritics Kept on the **Tashkeela** test set for SR, BALD, and MD metrics using the base model, keeping diacritics below threshold. **Note**: The percentage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

domain test set. To address this calibration issue, we applied CER with different regularization strengths λ. We experimented with several values, selected through hyperparameter optimization for confidence gaps and validation diacritic error rate. The results below are shown using the out-of-domain test set derived from ArVoice, where the base error rate is above 10%.

4.3.1 Impact on DER

Diacritic error rate tends to increase with higher regularization, as shown in Figure 3.

This increase is more significant for $\lambda_{2.77}$, and scales mostly linearly as λ values increase. $\lambda_{0.644}$ shows moderate increase in DER, while improving model precision in detecting error-prone, high-DER words, as discussed in the next section.

4.3.2 High-DER Words Precision/Recall

To quantify the model's ability to identify ambiguous words, we perform word-level analysis. We define a high-DER word as one with $> 50\%$ DER. We then measure how well we can identify these high-DER words using the model's uncertainty scores. We calculate the uncertainty score for a word as the mean uncertainty of its characters. We then sort the words in the test set from lowest to highest uncertainty to define the uncertainty percentiles. For instance, the 70^{th} percentile is the word-level uncertainty score where 70% of the words fall below, and 30% of words are higher. The 30% high-uncertainty words are the ones 'detected' by the model. The exact calculations are shown in Appendix, section A.1.2.

Based on these definitions, we measure the precision and recall at different regularization strengths λ and uncertainty percentiles[2]. Figure 4 presents these metrics for several values of the regularization parameter λ, along with the base model ($\lambda = 0$). Overall, there is a clear trade-off between recall and precision. The base model tends to achieve higher recall but suffers from very low precision, indicating that it flags words with low error rates as uncertain, and vice versa. In contrast, the regularized models typically flag fewer words overall, which results in smaller recall but precision is higher than the baseline.

Notably, at very high thresholds (e.g., the 99th or 100th percentile), both recall and precision drop, likely because only a tiny fraction of words exceed these stringent uncertainty levels. A threshold near the 90-95% range appears to offer a good balance between detecting enough erroneous words while minimizing false positives. The exact choice depends on whether higher precision or higher coverage of erroneous words is the primary goal.

[2]Note that due to the distribution of the scores and the skewed uncertainty values, the percentiles do not reflect exactly the same number of detected words across models. For instance, at the 70^{th} percentile, the number of words below the threshold may be less than 70%.

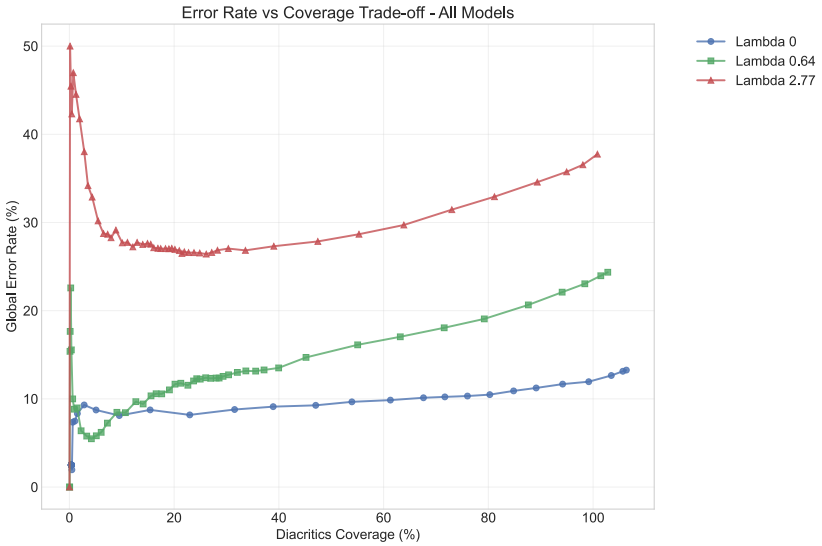


Figure 3: DER calculations for the base and calibrated models across coverage levels. Keeping below threshold diacritics. The results are based on the **ArVoice test set**. **Note**: The coverage is calculated with the total ground truth diacritics as denominator. The base model generates more than 100% of the diacritics due to insertion errors.

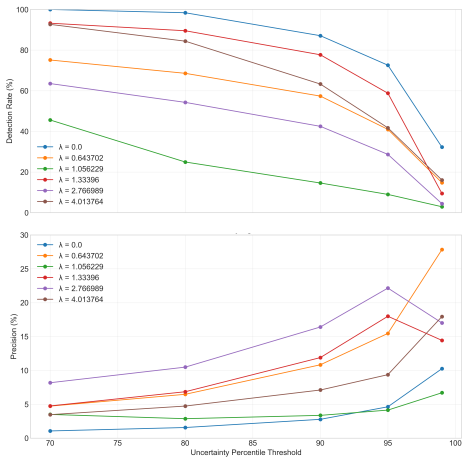


Figure 4: Detection rate or recall (top) and precision (bottom) of high-DER words at different uncertainty percentiles using different regularizing strength values. The results are based on the **ArVoice test set**.

5 Discussion and Conclusions

We explored the application of uncertainty estimation for partial diacritization of Arabic text. We experimented with various uncertainty estimation methods, and established the suitability of Softmax Response for this task. The other two metrics we explored had some drawbacks that made them less suitable for our task. While the Mahalanobis distance exhibited some correlation with diacritic error rates, the effect is weaker than the other two methods, resulting in higher error rates at the same coverage points. BALD achieved similar correlation to SR, but it is less suitable for practical diacritc restoration models that involve sequence labeling due to its higher computational cost. Among the three metrics, SR provides optimal performance and efficiency, making it suitable for additional calibration and practical deployment. However, our experiments show that better calibration results in higher DER, so additional work is needed to develop calibrated models that retain base accuracy. Nevertheless, the calibrated model shows potential for identifying ambiguous words, which we define as words with high DER, in terms of precision. This indicates that calibration may still be useful for some target application, where identifying ambiguous words with high precision (albeit with low recall) is desired. This preliminary analysis illustrates that additional work is needed to identify suitable calibration methods that optimize uncertainty estimation while maintaining the performance of the base diacritic restoration model.

In terms of application, the SR-based approach is straightforward to integrate into any neural-based diacritic restoration model, and our experiments show that we can reduce the relative DER in partial diacritization with various coverage thresholds. Such approach can be used in user-facing applications where automatic diacritization is used to annotate undiacritized text, leading to a partially diacritized text that is more accurate than the baseline. However, such methods do not address the issue of ambiguous words, which are likely to re-

Table 1: Examples of Threshold-based Diacritic Selection at 20% Coverage ($\lambda = 0.64$)

Strategy	Text
Ground Truth	الْماءُ يَتَجَمَد عِنْدَ أَكْثَرَ مِنْ مِئَةِ دَرَجَةٍ فِي ظِلِ تَنَامِي الْآثَارِ الْجَانِبِيَة لِلْأَدْوِيَة الْكِيمْيَائِيَّة
Drop High Uncertainty	الماء يتجمد عنْدَ أَكثَر مِن مئَةِ دَرَجَة في ظِل تنَامِي الآثَارِ الْجانِبِيَة لِلأدوِيةِ الكِيمِيَائِيَة
Drop Low Uncertainty	الْماءُ يتَجمَد عِند أَكْثَرَ منْ مِئَة درجةٍ فِي ظلّ تَنامي الْآثار الجَانبية للْأَدْوِيَة الْكيميائيَة

main undiacritized under such schemes. For improving readability, the same technique could be used to reduce the total number of diacritics in fully-diacritized text with ground-truth diacritics. The words that are identified as high-uncertainty could retain their diacritics, while diacritics on low-uncertainty words can be dropped. Examples of sentences and their partial diacritics using each of these proposed schemes are shown in Table 1.

Our exploratory analysis provides a starting point for such applications; further evaluation and analysis are needed to verify the effectiveness of such approaches in practical applications like readability enhancement, machine translation, and text-to-speech synthesis.

Limitations

We limited our analysis to one base diacritic restoration model, CATT, which serves as a strong baseline. Our analysis may be applicable to other models, but the experiments need to be replicated to verify that. The work presented in this paper serves as a preliminary exploration of uncertainty estimation as applied to the task of diacritic restoration, but it does not include sufficient analysis of the impact of such methods on downstream applications. Additional experiments are needed to explore the applicability of the proposed technique in applications such as machine translation, text-to-speech synthesis, or readability assessment. The choice of uncertainty metrics was motivated mostly by simplicity and convenience, and other metrics could have been included in the analysis. The analysis provided in this paper should be taken as a partial exploration rather than the final word on the suitability of uncertainty estimation metrics for partial diacritization. Finally, the experiments show that error calibration hurts model performance. We do not provide a solution for this and leave any improvement on the proposed calibration method for future work.

Acknowledgements

We would like to thank members of the speech lab at MBZUAI, namely Hawau Olamide Toyin, and Rufael Fekadu Marew, for guidance and support on various steps in this work. We also thank the anonymous reviewers at UncertaiNLP, who engaged constructively with the paper and raised valid points. We added these points to the limitations section.

References

Sawsan Alqahtani, Hanan Aldarmaki, and Mona Diab. 2019. Homograph disambiguation through selective diacritic restoration. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 49–59, Florence, Italy. Association for Computational Linguistics.

Sawsan Alqahtani, Mahmoud Ghoneim, and Mona Diab. 2016. Investigating the impact of various partial diacritization schemes on Arabic-English statistical machine translation. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 191–204, Austin, TX, USA. The Association for Machine Translation in the Americas.

Ali Assad, Abdul Hadi M. Alaidi, Amjad Yousif Sahib, Haider TH Salim ALRikabi, and Ahmed Magdy. 2024. Transformer-based automatic arabic text diacritization. *Sustainable Engineering and Innovation*, 6(2):285–296.

Mona Diab, Mahmoud Ghoneim, and Nizar Habash. 2007. Arabic diacritization in the context of statistical machine translation. In *Proceedings of Machine Translation Summit XI: Papers*, Copenhagen, Denmark.

Salman Elgamal, Ossama Obeid, Mhd Kabbani, Go Inoue, and Nizar Habash. 2024. Arabic diacritics in the wild: Exploiting opportunities for improved diacritization. In *Proceedings of the 62nd Annual Meeting*

of the Association for Computational Linguistics (Volume 1: Long Papers), pages 14815–14829, Bangkok, Thailand. Association for Computational Linguistics.

Muhammad ElNokrashy and Badr AlKhamissi. 2024. A context-contrastive inference approach to partial diacritization. In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 89–101, Bangkok, Thailand. Association for Computational Linguistics.

Ali Fadel, Ibraheem Tuffaha, Bara' Al-Jawarneh, and Mahmoud Al-Ayyoub. 2019. Neural Arabic text diacritization: State of the art results and a novel approach for machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 215–225, Hong Kong, China. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan. Association for Computational Linguistics.

Nizar Habash and Owen Rambow. 2007. Arabic diacritization through full morphological tagging. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers*, pages 53–56, Rochester, New York. Association for Computational Linguistics.

Nizar Habash, Owen Rambow, and George Kiraz. 2005. Morphological analysis and generation for Arabic dialects. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 17–24, Ann Arbor, Michigan. Association for Computational Linguistics.

Nawar Halabi. 2016. Arabic speech corpus.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations (ICLR)*.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jurgen Lameris. 2021. Homograph disambiguation for arabic text-to-speech synthesis using transformer-based models. Master's thesis, Uppsala University.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7167–7177.

Hamdy Mubarak, Ahmed Abdelali, Hassan Sajjad, Younes Samih, and Kareem Darwish. 2019. Highly effective Arabic diacritization using sequence to sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2390–2395, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabriel Pereyra, George Tucker, Jan Chorowski, ukasz Kaiser, and Geoffrey Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *International Conference on Learning Representations (ICLR)*.

Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021. Improving Arabic diacritization with regularized decoding and adversarial training. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 534–542, Online. Association for Computational Linguistics.

Mark Roman and Bernard Pavard. 1987. Processing diacritics in reading arabic: Evidence from eye-movement measures. In *Proc. of the Annual Meeting of the Cognitive Science Society*. Cognitive Science Society.

Hawau Olamide Toyin, Samar M Magdy, and Hanan Aldarmaki. 2025a. Are LLMs good text diacritizers? an Arabic and Yorùbá case study. *arXiv preprint arXiv:2506.11602*.

Hawau Olamide Toyin, Rufael Marew, Humaid Alblooshi, Samar M Magdy, and Hanan Aldarmaki. 2025b. ArVoice: A Multi-Speaker Dataset for Arabic Speech Synthesis. In *Interspeech 2025*, pages 4808–4812.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. The art of abstention: Selective prediction and error regularization for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1040–1051.

Taha Zerrouki and Amar Balla. 2017. Tashkeela: Novel corpus of arabic vocalized texts, data for auto-diacritization systems. *Data in Brief*, 11:147–151.

A Appendix

A.1 Formulation of metrics

We use the following metrics to evaluate our models, categorized into diacritic-level and word-level metrics:

A.1.1 Diacritic-level Metrics

1. **Diacritic Error Rate (DER)**: The percentage of incorrectly predicted diacritics over the total:

$$\text{DER} = \frac{|\{d \in \mathcal{D} | \hat{d} \neq d\}|}{|\mathcal{D}|} \times 100\% \quad (14)$$

Where:

- d is a given diacritic
- $\mathcal{D}$ is the set of all possible diacritics
- $\hat{d}$ is a true label diacritic

This metric is computed using edit distance calculations adapted for Arabic diacritics. Any missing or deleted diacritics are not considered errors, since the goal is to remove as many diacritics as possible while retaining accurate predictions by the model.

2. **Diacritization Coverage**: Percentage of characters or words that retain diacritics after partial diacritization:

$$\text{Coverage} = \frac{|\{c \in \mathcal{C} | \text{HasDiac}(c) = \text{True}\}|}{|\mathcal{C}|} \times 100\% \quad (15)$$

Where:

- c is a given character
- $\mathcal{C}$ is the set of all characters that can be diacritized
- 'HasDiac' is a function that returns True when character c retains its diacritic after thresholding, and False otherwise

Controlled by uncertainty threshold, Lower coverage means more sparsely populated text (fewer diacritics).

A.1.2 Word-level Metrics

1. **High/Low DER Words**: A heuristic definition of high-and low-DER words. Words with Diacritic Error Rate (DER) exceeding 50% are defined here to be able to see the model's consistency in capturing such error-prone words. We define the set of high DER words as:

$$\mathcal{H} = \{w \in \mathcal{W} | \text{DER}(w) > 0.5\} \quad (16)$$

And the set of low DER words as:

$$\mathcal{L} = \mathcal{W} \setminus \mathcal{H} = \{w \in \mathcal{W} | \text{DER}(w) \leq 0.5\} \quad (17)$$

The uncertainty of a word is calculated as the mean uncertainty across all characters in the word:

$$U(w) = \frac{1}{|w|} \sum_{j=1}^{|w|} U(c_j) \quad (18)$$

Where:

- w represents any given word
- c_j represents the j-th character in word w
- U is the uncertainty of word w or character c_j
- $\mathcal{W}$ is the set of words in total in the dataset

2. **Recall and precision**:

Recall is the percentage of detected high-DER words over their total amount, detected or not:

$$\text{Recall} = \frac{|\{w \in \mathcal{H} | \text{Detected}(w) = \text{True}\}|}{|\mathcal{H}|} \quad (19)$$

Precision is the amount of the detected high-DER words, over the total detected words by the model:

$$\text{Precision} = \frac{|\{w \in \mathcal{H} | \text{Detected}(w) = \text{True}\}|}{|\{w \in \mathcal{W} | \text{Detected}(w) = \text{True}\}|} \quad (20)$$

Effectively, recall shows us how good a model is at catching problems in general, how much it can actually cover of them in total, and precision shows us how accurately it can catch actual, legitimate ambiguous cases, rather than flagging any given word overall as uncertain with inflated scores

Phases of Uncertainty: Confidence–Calibration Dynamics in Language Model Training

Aneesh Durai
UC Berkeley
aneesh.durai@berkeley.edu

Abstract

Autoregressive language models achieve strong performance across a wide range of natural language processing (NLP) tasks, yet their uncertainty estimates remain poorly understood, particularly during training. Prior work has primarily evaluated calibration and out-of-distribution (OOD) robustness at the final checkpoint, overlooking the dynamics that unfold earlier. We introduce a phase-based framework for tracking uncertainty metrics—including expected calibration error (ECE) and Kullback–Leibler (KL) divergence—across distinct stages of training. Using GPT-2 models trained across multiple random seeds, we find that uncertainty dynamics follow a consistent set of phases: models begin conservative and relatively well calibrated, but later phases introduce a paradoxical decoupling where confidence increases even as calibration worsens, especially under distribution shift. This paradox implies that the final checkpoint is not always the most reliable for deployment and motivates phase-aware strategies such as dynamic checkpoint selection or targeted calibration. Our findings highlight that uncertainty should be understood as a training-dependent property rather than a static one, opening new directions for scaling this framework to larger models, tasks, and distribution shift scenarios.

1 Introduction

Autoregressive language models have become central to a large portion of modern NLP, driving progress in tasks as varied as document summarization, dialogue, and code generation (Brown et al., 2020). Yet, the impressive in-distribution performance of these models hides a recurring issue: their behavior is far less predictable when the input departs from the training distribution (Hendrycks and Gimpel, 2017). In production settings, such out-of-distribution (OOD) cases are inevitable such as topic drift in conversational systems, domain mismatch in translation, or simply user queries that exploit corner cases in the model's learned representation.

Uncertainty estimation has become a way to address this problem. Approaches such as Bayesian approximations via dropout (Gal and Ghahramani, 2016) or calibration-based adjustments (Guo et al., 2017) offer ways to associate model predictions with confidence scores. However, most of this work evaluates a final trained output. What is less well understood, especially in language modeling, is how uncertainty evolves during training itself. Language models acquire the syntax, semantic, and task-specific reasoning in a staged manner, and their calibration profile is unlikely to be uniform across these stages (Desai and Durrett, 2020; Jiang et al., 2021).

Our key finding is that calibration does not improve monotonically with training: a mid-training phase emerges in which models grow more confident while becoming less calibrated.

In this work, we introduce a phase-based framework for tracking and analyzing the joint dynamics of calibration error and KL divergence between successive stages of training. By segmenting model training into distinct phases and evaluating these metrics both in-distribution and OOD, our approach offers a structured view of how and when models become more or less calibrated, and how their predictive distributions shift over time.

2 Related Work

2.1 Uncertainty Estimation in NLP

Quantifying predictive uncertainty has been a needed measure in modeling and modern neural networks. For classification tasks, baseline confidence scores such as the maximum softmax probability and predictive entropy are widely used to flag low-confidence predictions (Hendrycks and Gimpel, 2017). Bayesian-inspired techniques, including Monte Carlo dropout (Gal and Ghahramani,

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 11–16
November 9, 2025 ©2025 Association for Computational Linguistics

2016) and deep ensembles (Lakshminarayanan et al., 2017), have adapted to NLP models to better capture the epistemic and aleatoric uncertainty. Recent work has explored these methods for both and structured prediction tasks like semantic parsing (Dong et al., 2017). However, most existing approaches report uncertainty only for the final converged models, and that overlooks how these measures are evolving during training.

2.2 Calibration of Language Models

Calibration measures the degree to which predicted probabilities align with empirical correctness (Guo et al., 2017). While overconfidence is a well-known issue in fields like computer vision, language models exhibit domain-specific calibration challenges (Desai and Durrett, 2020). Post-hoc techniques such as temperature scaling and histogram binning have been applied to NLP (Guo et al., 2017), but once again, their effectiveness is often evaluated only after full training.

Some other work has explored calibration in generative settings, (Kumar et al., 2019), yet there remains little understanding of how calibration quality changes mid-training, especially for large-scale autoregressive models.

2.3 OOD Robustness and Distribution Shifts

OOD detection aims to identify inputs that differ substantially from the training distribution. Density-based methods (Lee et al., 2018), and uncertainty-based rejection strategies (Hendrycks and Gimpel, 2017) have been explored in NLP, often under domain shift scenarios (Varshney et al., 2022). Despite this, the majority of studies evaluate robustness at convergence, providing little insight into the temporal dynamics of OOD behavior. The opportunity of the interplay between the training-phase uncertainty trends, calibration shifts, and OOD performance remains largely unexplored.

We address this gap by systematically tracking the uncertainty metrics, calibration scores, and KL divergence between training phases for autoregressive language models. By linking these evolving quantities to in-distribution and OOD generalization, we provide a temporal perspective on uncertainty and robustness, which offers a richer understanding than simple post-training evaluation alone.

3 Methodology

3.1 A Phase-Based View of Training Dynamics

Calibration in neural networks is typically assessed only at convergence, which obscures transient regimes where confidence and reliability can drift in opposite directions (Guo et al., 2017; Ovadia et al., 2019; Minderer et al., 2021). Leveraging this observation, we take a temporal perspective and segment training into phases defined by persistent shifts in uncertainty and calibration traces.

3.2 Metrics

We track two uncertainty-related metrics at regular checkpoints.

KL Divergence to Uniform (Confidence Proxy). Let $p \in \Delta^{V-1}$ be the next-token predictive distribution over a vocabulary of size V, and let u denote the uniform distribution ($u_i = 1/V$). Confidence is measured as

$$D_{\mathrm{KL}}(p \,\|\, u) \;=\; \sum_{i=1}^{V} p_i \log \frac{p_i}{1/V}. \qquad (1)$$

Higher values indicate sharper, more confident distributions; lower values indicate more diffuse predictions. This quantity is 0 when predictions are maximally uncertain (uniform) and increases as the distribution sharpens, making it a natural confidence proxy. It is closely related to predictive entropy, since $D_{\mathrm{KL}}(p \,\|\, u) = \log V - H(p)$. While other reference distributions could be considered, we adopt the uniform baseline because it provides a simple and interpretable notion of random guessing, against which sharper, more confident predictions can be measured.

Expected Calibration Error (ECE). Following Guo et al. (2017), tokens are binned by predicted confidence into M equal-width bins $\{B_m\}_{m=1}^{M}$. Let $\mathrm{acc}(B_m)$ be the empirical accuracy and $\mathrm{conf}(B_m)$ the mean confidence in bin m. The ECE is

$$\mathrm{ECE} \;=\; \sum_{m=1}^{M} \frac{|B_m|}{\sum_j |B_j|} \Big|\mathrm{acc}(B_m) - \mathrm{conf}(B_m)\Big|. \qquad (2)$$

Lower values indicate better calibration.

3.3 Phase Detection

At each checkpoint we record KL-to-uniform and ECE over the ID validation set and an OOD corpus;

we smooth each per-seed KL trajectory with an exponentially weighted moving average and detect changepoints on KL. Let t index checkpoints.

We then identify three regimes per seed:

1. **Phase I (Early Learning):** ends at the early local maximum of KL (searched in the first half of training) or a default tertile boundary if no clear maximum exists.
2. **Phase II (Confidence Surge):** begins after Phase I and ends at the subsequent local maximum of KL (or a default second-tertile boundary), enforcing a minimum phase length.
3. **Phase III (Stabilization):** the remaining steps to the final checkpoint.

Boundaries are constrained to respect minimum durations and ordered consistency (therefore I $<$ II $<$ III). We compute all metrics per phase and then report both per-seed summaries and seed-averaged statistics. This procedure captures non-monotonic behavior that endpoint-only evaluation can miss, such as periods where confidence rises while calibration degrades (Ovadia et al., 2019; Minderer et al., 2021).

4 Experiments and Results

4.1 Experimental Setup

We trained GPT-2 models for 3,000 optimization steps across five seeds. Training was conducted on the WikiText-2 (Merity et al., 2016) corpus for in-distribution (ID) evaluation, while out-of-distribution (OOD) generalization was assessed on the AG News (Zhang et al., 2016) dataset. At regular intervals, we computed both standard training metrics (loss) and uncertainty metrics for both ID and OOD test sets. This setup provides a comprehensive view of the interaction between confidence and calibration throughout training. Unless stated otherwise, significance is assessed with a two-sided paired t-test over checkpoints, aggregated across seeds.

4.2 Phase Detection Procedure

To identify interpretable regimes of uncertainty dynamics, we employed an automatic phase segmentation method based on changepoints in the KL trajectory. Consistently across all seeds, three phases emerged as shown in Table 1.

Table 1: Training phase boundaries identified across all seeds.

Phase	Step Range
I (Early Transient)	50–1500
II (Confidence–Calibration Drift)	1550–2900
III (Convergence Plateau)	2950–3000

Phase characteristics across seeds are summarized in Table 2. Figure 1 shows the dynamics for the average of the seeds. Phase I balances confidence and calibration, Phase II marks systematic divergence between them (the confidence–calibration paradox), and Phase III represents a plateau at degraded calibration levels.

4.3 In-Distribution Dynamics

During Phase I, models maintained relatively low calibration error (mean ECE ≈ 0.005). As training progressed into Phase II, a paradoxical trend emerged: calibration degraded even as confidence increased. Specifically, mean ECE rose by $\sim$23.4% (from 0.0049 to 0.0058, $p = 2.05 \times 10^{-5}$), while KL divergence to uniform predictions increased by 0.5% (9.471 $\rightarrow$ 9.523). This indicates that the models became more confident but less calibrated. In Phase III, metrics stabilized (KL ≈ 9.529 and ECE ≈ 0.0057), but calibration did not return to the initial level.

Across all five experiments, this paradox held consistently: in every run, confidence increased while calibration worsened. Prior speculation that calibration might improve in later stages (e.g., Guo et al., 2017; Desai and Durrett, 2020) was not supported in our setting.

4.4 Out-of-Distribution Behavior

When evaluated on AG News, models exhibited the same paradox but with larger miscalibration. OOD ECE rose from $\sim$0.033 in Phase I to $\sim$0.040 in Phase II ($p = 2.31 \times 10^{-8}$), representing a $\sim$21% relative increase, alongside a concurrent increase in KL-to-uniform. As with ID, metrics stabilized in Phase III without recovery.

Notably, the paradox was amplified OOD: the models simultaneously became more confident and less calibrated under distribution shift, producing error rates far larger in magnitude than ID. This indicates that the confidence–calibration paradox is not only a training artifact but also a deployment concern for real-world distribution shifts.

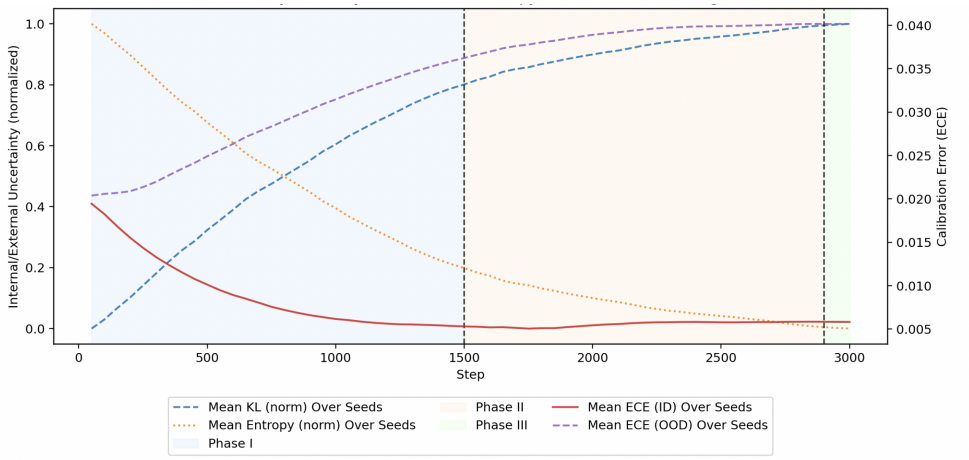


Figure 1: Phase dynamics of calibration and confidence. Confidence (KL, blue) rises steadily, while calibration error (ECE, red/purple) degrades. Phase II (yellow) highlights the paradoxical "danger zone" where all 5 seeds showed confident but unreliable predictions. Note the divergence between rising confidence and worsening calibration around step 1500.

Table 2: Phase characteristics averaged across seeds. ECE reported for in-distribution (ID) and out-of-distribution (OOD). KL-to-uniform is our primary confidence metric; entropy (H) is reported for reference only.

Phase	KL	H	ECE_{ID}	ECE_{OOD}
I	9.471 ± 0.004	1.354 ± 0.004	0.005	0.033 ± 0.001
II	9.523 ± 0.003	1.302 ± 0.003	0.006	0.040 ± 0.001
III	9.529 ± 0.002	1.296 ± 0.002	0.006	0.040 ± 0.001

5 Discussion and Future Work

We show three consistent training phases, document a mid-training confidence-calibration gap, and outline how to use these signals for safer checkpoint selection and calibration. Our results suggest that current practice may systematically deploy models from their least reliable phase. Monitoring only validation loss obscures the fact that Phase II coincides with worsening calibration. This paradox has several practical consequences: (1) calibration should be tracked jointly with loss during training, (2) deployments should avoid Phase II checkpoints (high confidence, poor calibration), and (3) interventions such as temperature scaling or selective regularization may be most beneficial when targeted specifically to this unstable phase. Without such precautions, models risk being deployed precisely when they are most deceptively unreliable.

Beyond these immediate implications, our phase-based framework highlights opportunities for future work. Scaling to larger architectures and reasoning-capable models will test the generality of the paradox. Expanding to broader OOD scenarios (e.g., multilingual or reasoning tasks) will help determine whether the observed dynamics extend beyond WikiText and AG News. Finally, phase-aware interventions could be designed to adaptively correct calibration drift in real time, reducing deployment risks for large-scale language models.

6 Conclusion

We introduced a phase-based framework for analyzing uncertainty and calibration dynamics throughout language model training. Across multiple seeds, we consistently observed a confidence–calibration paradox: models became less reliable precisely as their predictions grew more confident. This paradox was amplified under distribution shift, underscoring its practical importance for deployment safety.

By framing uncertainty as a training-dependent property rather than a static one, we provide a foundation for phase-aware monitoring, checkpointing, and intervention strategies. In practice, our results motivate monitoring calibration (ECE) jointly with validation loss, avoiding Phase II checkpoints when selecting release models, and applying simple post-hoc calibration such as temperature scaling at deployment.

Limitations

First, we trained GPT-2 scale models for 3,000 iterations across five seeds, a modest but controlled scope. Second, our OOD evaluation was limited to a single dataset (AG News) and a restricted set of uncertainty metrics (ECE and KL). Third, our phase detection relies on inflection points in these metrics; whether analogous phase boundaries generalize to larger architectures or alternative metrics remains open.

Despite these constraints, the reproducibility of phase dynamics across seeds suggests that the phenomena are not small-scale artifacts but emergent properties of autoregressive training. Extending this analysis to larger models, broader OOD scenarios, and alternative calibration interventions represents a natural next step.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Li Dong, Jonathan Mallinson, Siva Reddy, and Mirella Lapata. 2017. Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. *ArXiv*, abs/1706.04599.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Ananya Kumar, Percy Liang, and Tengyu Ma. 2019. Verified uncertainty calibration. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3792–3803.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 7167–7177.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Matthias Minderer, Josip Djolonga, Rob Romijnders, Frances Hubis, Xiaohua Zhai, Neil Houlsby, Dustin Tran, and Mario Lucic. 2021. Revisiting the calibration of modern neural networks. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA. Curran Associates Inc.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Preprint*, arXiv:1906.02530.

Neeraj Varshney, Swaroop Mishra, and Chitta Baral. 2022. Investigating selective prediction approaches across several tasks in iid, ood, and adversarial settings. *Preprint*, arXiv:2203.00211.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level convolutional networks for text classification. *Preprint*, arXiv:1509.01626.

Response to Reviewer Comments

Reviewer RqHb

1. *Missing double-descent literature (Nakkiran et al.; OpenAI blog; Schaeffer et al.).* Our work studies calibration dynamics across training phases (ECE and KL-to-uniform), not necessarily accuracy/test-error curves. While double descent concerns error as a function of model size/data/epochs, our main result is a *confidence–calibration gap* during mid-training even as loss improves. We agree the phenomena are related but orthogonal; due to short-paper space we keep Related Work focused on calibration/uncertainty and OOD. We are happy to expand this connection in a longer version.

2. *Unclear how to utilize the research.* The camera-ready adds concrete guidance in *Discussion and Future Work* and *Conclusion*: (i) monitor calibration (ECE) jointly with validation loss; (ii) avoid Phase II checkpoints when choosing release models; (iii) apply simple post-hoc calibration (temperature scaling) and consider selective regularization targeted to Phase II.

3. *Missing dataset references.* We now cite WikiText-2 and AG News explicitly in *Experimental Setup*.

4. *Numbers not reflected in Figure 1.* We clarified phase boundaries (Table 1) and added a per-phase summary (Table 2). Figure 1 caption explicitly describes the divergence point (around step 1500) referenced in §4.3–4.4.

Reviewer T3uL

1. *Entropy vs. KL to uniform are redundant.* We agree. The revision treats KL-to-uniform as the *primary* confidence proxy and includes entropy only as a reference column in Table 2. We explicitly note the identity $D_{\mathrm{KL}}(p\|u) = \log V - H(p)$ in *Metrics*.

Reviewer 1agi

1. *Generality to newer/reasoning models.* We scope the empirical study to GPT-2 scale for 3,000 steps across five seeds on WikiText-2 (ID) and AG News (OOD). We state this limitation and the generalization agenda in *Limitations* and *Discussion and Future Work* (scaling to larger and reasoning-capable models).

2. *More seeds/tasks; statistical support.* We report five seeds and add a sentence in *Experimental Setup* specifying two-sided paired t-tests over checkpoints (aggregated across seeds) for reported comparisons. We agree that the next step would require multiple more runs.

3. *Intervening in the "danger zone."* We strengthened the deployment guidance in *Discussion*: phase-aware checkpoint selection, targeted regularization during Phase II, and post-hoc calibration (temperature scaling) at deployment.

Beyond Human Judgment: A Bayesian Evaluation of LLMs' Moral Values Understanding

Maciej Skorski
University of Luxembourg
maciej.skorski@gmail.com

Alina Landowska
SWPS University
alandowska@swps.edu.pl

Abstract

How do Large Language Models understand moral dimensions compared to humans?

This first comprehensive large-scale Bayesian evaluation of leading language models provides the answer. In contrast to prior approaches based on deterministic ground truth (obtained via majority or inclusion consensus), we obtain the labels by modelling annotators' disagreement to capture both aleatoric uncertainty (inherent human disagreement) and epistemic uncertainty (model domain sensitivity).

We evaluated Claude Sonnet 4, DeepSeek-V3, and Llama 4 Maverick across 250K+ annotations from nearly 700 annotators in 100K+ texts spanning social networks, news, and discussion forums. Our GPU-optimized Bayesian framework processed 1M+ model queries, revealing that AI models generally rank among the top 25% of annotators in terms of balanced accuracy, substantially better than average humans.

Importantly, we find that AI produces far fewer false negatives than humans, highlighting their sensitive moral detection capabilities.

Keywords: Computational Ethics, Large Language Models, Moral Foundation Theory, Bayesian modeling, soft labels

Extended version: arXiv:2508.13804
Supplementary materials: osf.io/tpzau

1 Introduction

1.1 Background

Moral Foundations Theory (MFT) provides a comprehensive framework for understanding human moral reasoning across cultures, identifying core dimensions typically expressed as virtue/vice pairs: Care vs. Harm, Fairness vs. Cheating, Loyalty vs. Betrayal, Authority vs. Subversion, and Sanctity vs. Degradation (Graham et al., 2013; Haidt, 2012). These foundations shape individual and collective decision-making, from political preferences to social behavior (Feinberg and Willer, 2013; Graham et al., 2009; Nguyen et al., 2022; Roy and Goldwasser, 2021), making their computational detection crucial for understanding discourse dynamics and developing ethically-aligned AI systems.

Text	Foundation
"My heart breaks seeing children separated from families at the border"	Care
"Everyone deserves equal access to healthcare regardless of income"	Fairness
"Respect your elders and follow traditional values that built this nation"	Authority
"Stand with our troops - they sacrifice everything for our freedom"	Loyalty
"Marriage is sacred and should be protected from secular corruption"	Sanctity

Table 1: Posts and Associated Moral Foundation

The computational linguistics community has successfully fine-tuned pre-trained language models to predict moral values (Nguyen et al., 2024; Preniqi et al., 2024; Zangari et al., 2025a), achieving good alignment with human judgment when domain similarity and sufficient training data are available. However, systematic evaluation of large language models (LLMs) remains limited despite their rapid advances and potential as a compelling alternative that should suffer less from poor generalization and distribution shift.

This paper addresses these limitations through rigorous large-scale evaluation of state-of-the-art language models across established moral reasoning corpora, employing Bayesian methods to resolve disagreeing annotations.

1.2 Contribution

Bayesian uncertainty modelling of moral annotations. We introduce Bayesian modelling of annotator disagreements for moral foundation evalua-

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 17–26
November 9, 2025 ©2025 Association for Computational Linguistics

tion, moving beyond simple deterministic ground-truth assumptions. This captures both aleatoric uncertainty (inherent human disagreement) and epistemic uncertainty (model sensitivity across domains and foundations).

Large-scale evaluation. We analyse market-leading large language models with 1M+ queries across 100K+ texts and 250K+ annotations from diverse sources, providing the most comprehensive moral reasoning evaluation to date.

Statistical analysis of Type I/II errors. We demonstrate that AI performs comparably to top annotators in balanced accuracy, considerably improving false negatives at the price of slightly increased false positive rates - contrary to fears that AI may underpredict moral values.

Novel GPU-optimized implementation of Bayesian labels. We developed a TensorFlow framework using sparse operations for scalable Bayesian inference on soft labels, of interest to the computational linguistics community.

1.3 Related Work

Dictionary methods for predicting moral values were initially developed (Hopp et al., 2021) and remain widely used in applied studies, but demonstrate poor precision compared to pre-trained language models (Nguyen et al., 2024).

Pretrained language models show strong alignment with human moral judgments given sufficient training data (Nguyen et al., 2024; Preniqi et al., 2024; Zangari et al., 2025a), though they suffer from distribution shift and poor cross-domain generalization. This makes large language models a compelling alternative.

Large language models were recently compared to human performance (Bulla et al., 2025), concluding LLMs superiority. However, their strict majority voting for ground truth excluded nuanced moral content, retaining only less ambiguous cases and departing from the inclusive consensus practices (marked when flagged by at least one annotator) (Nguyen et al., 2024; Preniqi et al., 2024; Zangari et al., 2025a). Additionally, leave-one-out estimation of annotator ranking lacked statistical robustness. Another recent work (Skorski and Landowska, 2025) found LLMs showing imbalance - lower than anticipated recall - under inclusive annotator consensus.

Our Bayesian method resolves these problems by modeling annotator disagreement, striking the balance between inclusive and overly exclusive consensus rules, while our large-scale evaluation across multiple datasets and modern models ensures robustness of findings.

More on moral foundation theory. For comprehensive coverage of (computational) moral foundation theory challenges, see (Zangari et al., 2025b).

2 Data and Methods

2.1 Datasets

Our robust evaluation utilizes three established and diverse corpora totaling 250K+ annotations of moral values from hundreds of annotators with diverse expertise (from experts to crowd-workers) across 100K+ texts spanning social media, news, and forum discussions.

Moral Foundations Twitter Corpus (MFTC) (Hoover et al., 2020): 128,454 annotations from 23 trained annotators across 33,686 tweets from seven discourse domains.

Extended Moral Foundations Dictionary (eMFD) (Hopp et al., 2021): 73,001 crowd-sourced annotations from 654 contributors on 54,867 text segments extracted from approximately 8,000 news documents by major outlets, including The Washington Post, CNN, Fox News selected via GDELT.

Moral Foundations Reddit Corpus (MFRC) (Trager et al., 2022): 61,226 annotations from 6 trained coders across 17,885 Reddit comments from 12 subreddits covering politics and everyday moral discourse.

2.2 Bayesian Annotation Competence Model

As in the related work, we frame the problem as binary prediction tasks for individual moral foundations (abbreviated to care, fairness, authority, loyalty, sanctity). Optionally, we consider the aggregated "any moral content" category derived from positive labels across foundations. We note that this aggregated category is inherently unreliable when only positive labels are provided (as in eMFD), since true negatives cannot be distinguished from unlabeled moral content, creating systematic classification ambiguity.

Given substantial inter-annotator disagreement demonsrated by PABAK scores in Table 2, we resort to probabilistic (Bayesian) methods of obtaining ground-truth labels (Paun and Simpson, 2021).

We model annotator disagreements using a variant of Dawid-Skene's model (Dawid and Skene, 1979) with weak Dirichlet priors to estimate ground-truth labels and annotator reliability:

Foundation	MFTC	MFRC	eMFD
Care	0.71	0.67	0.33
Fairness	0.63	0.64	0.36
Loyalty	0.62	0.82	0.38
Authority	0.52	0.78	0.41
Sanctity	0.58	0.83	0.44
Any	0.34	0.38	1.00

Table 2: Prevalence-Adjusted Bias-Adjusted Kappa (PABAK) scores measuring inter-annotator agreement across datasets and moral foundations

Model Specification. We assume N texts, J annotators, and K categories. True category prevalences follow $\boldsymbol{\pi} \sim \text{Dir}(\boldsymbol{\alpha})$ where $\boldsymbol{\alpha} = (1, 1, ..., 1)$ provides uniform priors. Each annotator j has a confusion matrix $\boldsymbol{\Theta}_j$ with rows $\boldsymbol{\theta}_{jk} \sim \text{Dir}(\boldsymbol{\beta}_k)$, where $\boldsymbol{\beta}_k$ is a K-dimensional vector with $\beta_{kk} = 2$ (diagonal) and $\beta_{kl} = 0.5$ for $l \neq k$, encoding weak belief that annotators correctly identify majority of categories. For text i with true category z_i and annotations $\mathbf{y}_i = (y_{i1}, ..., y_{iJ})$ we have:

$$\Pr\{z_i = k \mid \mathbf{y}_i, \boldsymbol{\pi}, \boldsymbol{\Theta}\} \propto \pi_k \prod_{j=1}^{J} \theta_{jk,y_{ij}},$$

This posterior accounts for varying annotator reliability while estimating both confusion matrices and ground-truth labels. For computational stability and reliable convergence, all calculations are performed in the log-domain using logits.

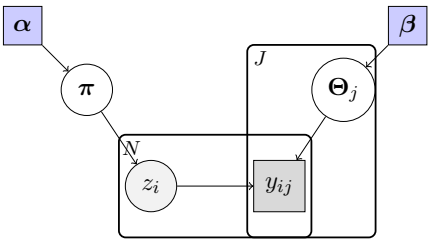


Figure 1: Graphical model representation of the model for multi-annotator classification. Light gray circles represent latent variables, dark gray rectangles represent observed variables, white circles represent parameters, and blue rectangles represent hyperparameters. Plates indicate replication over items (N) and annotators (J).

Implementation. We developed a GPU-optimized implementation in TensorFlow (Abadi et al., 2015) using custom graphs and sparse tensor operations for computational efficiency. The framework supports MAP estimation and Monte Carlo sampling with optional fixed effects modeling (Appendix B). Testing on A100 GPUs achieved 1,000 optimization steps per second for 100k annotations, enabling rapid convergence across large-scale datasets.

2.3 Large Language Models

Three recent advanced large language models were analysed for moral foundations classification: (1) **Claude Sonnet 4** (Anthropic, 2025) with a 200,000 token context window; (2) **DeepSeek-V3** (DeepSeek, 2024) with 671 billion total parameters, 37 billion active parameters, and a 128,000 token context window; and (3) **Llama 4 Maverick** (Meta, 2025) with 402 billion total and 17 billion active parameters, and a 256,000 token context window.

All models were queried using the temperature setting of 0.30 to balance deterministic responses with modest variability. Each text was analysed individually (no batching of inputs) to ensure focused classification results. The prompt used was:

Moral Foundations Theory Classification

You are an expert in moral psychology, classifying text according to Haidt's theory.
For each moral foundation, mark `true` if moral values from that foundation are expressed in the text, `false` if not expressed.

Answer only with a valid JSON in this format:

```
{
  "care/harm": [true / false],
  "fairness/cheating": [true / false],
  "loyalty/betrayal": [true / false],
  "authority/subversion": [true / false],
  "sanctity/degradation": [true / false]
}
```

Experiments were tracked using W&B[1] and repeated to ensure stability and reproducibility.

Appendix A provides details and rationale about the prompt selection.

2.4 Metrics

We apply standard classification metrics to Bayesian-estimated confusion matrices: Balanced Accuracy $\frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$, Precision $\frac{TP}{TP+FP}$, Recall $\frac{TP}{TP+FN}$, False Positive Rate $\frac{FP}{FP+TN}$, and False Negative Rate $\frac{FN}{FN+TP}$.

2.5 Supplementary Materials

Code, data preprocessing scripts, experimental results, and additional evaluation results are available from the project repository (Skórski, 2025).

[1] https://wandb.ai/

3 Results

Bayesian models were fitted for each language model and dataset, the model considered a one additional annotator each time, and confusion matrices obtained through MAP were used to calculate classification metrics and percentile ranks.

3.1 Accuracy Analysis

Balanced accuracy results appear in Table 3 and Figures 2 to 4.

		Moral Foundations					
Model	Metric	Any	Authority	Care	Fairness	Loyalty	Sanctity
MFTC Dataset							
Claude Sonnet 4	Acc%	75	80	78	82	81	90
	Pct	50	83	75	71	79	100
DeepSeek-V3	Acc%	77	82	78	81	80	85
	Pct	54	88	71	71	79	92
Llama 4 Maverick	Acc%	68	79	76	83	80	87
	Pct	42	79	63	71	75	96
Human	Avg%	72	67	71	75	72	67
eMFD Dataset							
Claude Sonnet 4	Acc%	85	74	88	83	72	80
	Pct	95	75	97	91	78	84
DeepSeek-V3	Acc%	93	81	88	82	85	75
	Pct	100	89	98	85	93	77
Llama 4 Maverick	Acc%	95	83	89	84	83	82
	Pct	100	92	98	88	91	86
GPT-5 mini	Acc%	82	64	78	77	62	66
	Pct	92	46	81	76	45	49
Human	Avg Acc%	63	64	62	64	62	63
MFRC Dataset							
Claude Sonnet 4	Acc%	74	83	90	80	86	91
	Pct	57	100	100	86	86	100
DeepSeek-V3	Acc%	68	83	90	73	84	82
	Pct	14	86	100	43	86	100
Llama 4 Maverick	Acc%	62	83	89	72	83	87
	Pct	14	100	100	43	86	100
Human	Avg%	75	69	76	75	71	70

Table 3: Model performance on moral foundation classification across datasets. Acc% shows balanced accuracy and Pct shows the corresponding percentile.

The key insights are:

AI Superiority over Humans. AI models consistently outperformed humans across datasets, typically ranking in the top 25% of annotators. Human performance averaged 67-76%, while AI achieved 62-95% depending on dataset.

Similar Dataset Difficulty. Average AI performance was similar across datasets: MFRC (83.7%), eMFD (81.9%), and MFTC (81.5%), with negligible difference between the highest and lowest performing datasets.

Model Strengths. While all models outperformed human annotators, there is no strong winner among them. Claude appears to be a slight overall leader, scoring high consistently and achieving first or second-best results most of the time, with particular excellence in nuanced Care and Sanctity foundations. DeepSeek and Llama 4 perform similarly most of the time, but show a visible gap of 8-9% behind Claude on MFRC in Sanctity and Fairness.

Consistent Accuracy across Foundations. All moral foundations achieved strong performance across datasets, with overall average accuracy exceeding 80%: Care (85.1%), Sanctity (84.3%), Loyalty (81.6%), Authority (80.9%), and Fairness (80.0%). Particularly high results for "Sanctity" are notable given its known difficulty to classify due to cultural contexts.

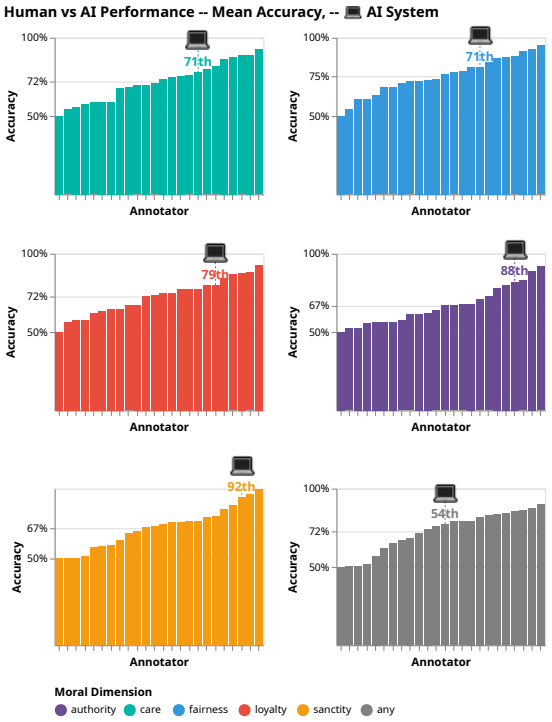


Figure 2: DeepSeek-V3 vs human accuracy (MFTC).

3.2 False Positive vs. False Negative Analysis

Figure 5 and Table 4 compare false positive and false negative rates across models and foundations. The key findings are:

AI-Human Error Trade-offs. Across all foundations and datasets, AI models achieve $2-4\times$ lower false negative rates compared to humans (19.4% vs 52.7% on average), at the expense of slightly higher average false positive rates (16.1% vs 10.1%), overall remaining more balanced (as visually revealed in Figure 5). This reveals opposing strategies: AI provides balanced detection, while humans classify more conservatively, systematically underdetect genuine moral signals.

Foundation-Specific Patterns. On average, the largest improvement in FNR appears in Care (40.8 point FNR versus humans), followed by Authority

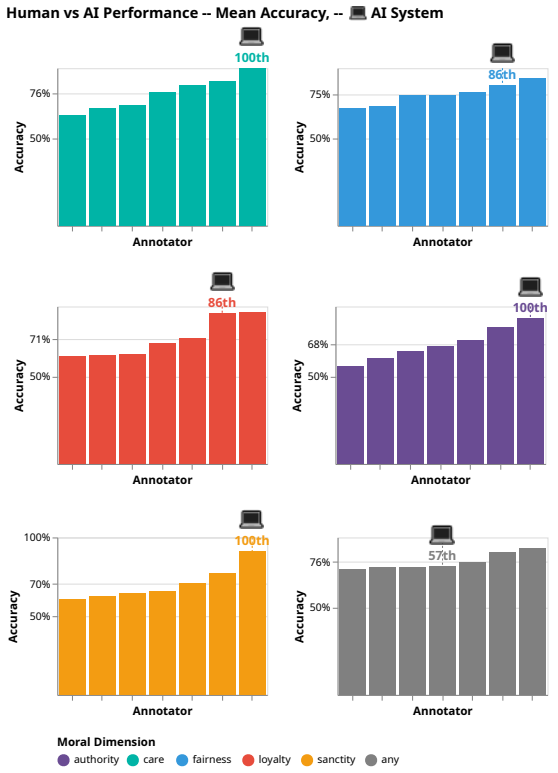


Figure 3: Claude Sonnet 4 vs human accuracy (MFRC).

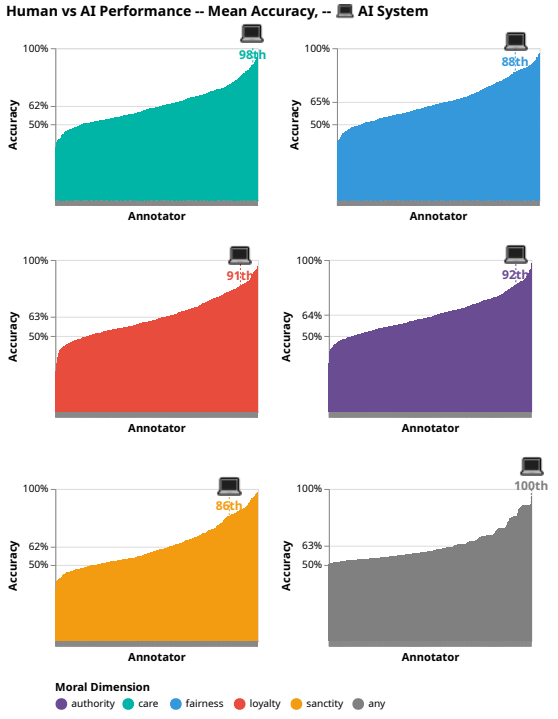


Figure 4: Llama 4 Maverick vs humans (eMFD).

(35.6 points), Loyalty (34.5 points) and Sanctity (34.0 points). Fairness remains more challenging, with an average improvement of 24.2 points.

Human Conservatism Impact. On MFRC and MFTC, annotated by a pool of experts, AI appears more balanced (average FNR-FPR AI gap of 1.4pp and 1.6pp, respectively) than in eMFD (gap of 8.7pp) crowd-annotated. This can be explained by the Bayesian method that estimates AI performance by seeking a consensus with a much higher number of conservative / biased annotators.

	Moral Dimensions									
	Authority		Care		Fairness		Loyalty		Sanctity	
Model	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR	FNR	FPR
MFRC Dataset										
Claude Sonnet 4	16.5	17.2	5.3	15.2	12.3	27.2	8.7	19.5	7.9	9.5
DeepSeek-V3	18.7	14.4	7.3	13.7	36.4	18.0	9.6	21.4	31.3	4.5
Llama 4 Maverick	14.6	20.0	10.8	11.4	28.1	28.8	9.7	24.8	15.2	10.5
Human Baseline	56.5	5.3	42.4	5.3	40.9	9.4	52.4	5.4	55.4	3.7
MFTC Dataset										
Claude Sonnet 4	14.7	25.3	7.2	35.8	9.3	25.8	20.1	18.7	7.8	12.2
DeepSeek-V3	24.6	11.1	15.2	28.5	28.6	8.7	22.9	16.7	26.7	3.8
Llama 4 Maverick	23.9	18.0	19.9	28.2	22.8	11.5	13.3	27.7	16.1	9.2
Human Baseline	53.1	13.4	50.3	6.9	43.5	6.2	46.9	8.6	59.1	7.3
eMFD Dataset										
Claude Sonnet 4	33.6	18.4	9.1	15.2	19.3	15.0	44.9	10.6	36.1	4.1
DeepSeek-V3	18.2	19.0	7.5	16.0	25.0	11.5	16.8	13.0	47.8	2.4
Llama 4 Maverick	12.6	20.7	7.9	14.1	18.2	13.4	19.9	14.8	32.2	3.8
Human Baseline	56.5	16.0	59.8	15.2	54.9	16.2	59.4	16.4	61.2	13.5

Table 4: False positive and negative rates (%) by model across moral foundations and datasets. Human baseline represents average performance across all annotators.

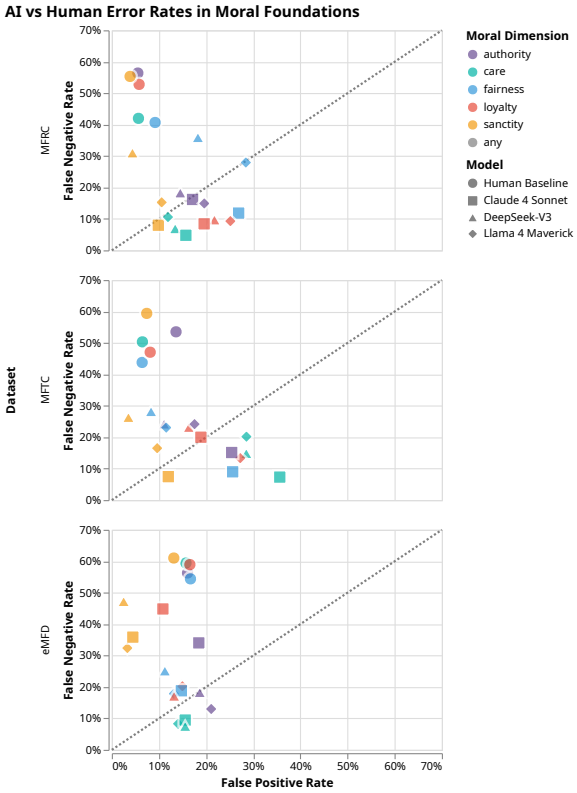


Figure 5: Error trade-offs in moral foundation detection. AI models (shapes) vs human baselines (circles) across datasets with colors denoting moral foundations. Diagonal lines indicate error balance (FPR = FNR).

4 Case Study

Human annotation, while considered the gold standard for supervised learning, may systematically underestimate moral content prevalence in text. As noted in annotator training guidelines, "Ambiguities present considerable challenges for human annotators who must strike an acceptable balance between exploiting often weak signals of moral sentiment while also avoiding unfounded speculation about authorial intent." Our results suggest AI models navigate this balance more effectively, achieving both higher recall rates and superior balanced accuracy compared to human annotators.

To verify this finding, we evaluated a selection of posts where AI predicted substantially more moral content than human annotators. An independent domain expert confirmed that AI was correct in the majority of cases, suggesting that human annotators systematically miss moral content that AI systems correctly identify. Full details of this validation study are provided in the supplementary material and code repository. The examples below represent cases where no human annotator detected moral language, false negatives in human annotation that AI correctly identified, demonstrating AI's superior sensitivity to subtle moral cues in text.

Example: Claude Sonnet 4 Classification

User: You are an expert in moral psychology, classifying text according to Haidt's theory (...)
Text: "Anyone think Macron should dispose of the Algerian dictator? France did little to support the uprisings of 2010-2012."

Assistant:

```
{
"care/harm": true,
"fairness/cheating": true,
"loyalty/betrayal": false,
"authority/subversion": true,
"sanctity/degradation": false
}
```

Reasoning:

- Care/harm: Concern for harm under "dictator," suggests intervention to protect people
- Fairness: Implies Algerians deserve better; criticizes France's abandonment
- Authority: Direct challenge to political authority through regime change

Example: Claude Sonnet 4 Classification

User: You are an expert in moral psychology, classifying text according to Haidt's theory (...)
Text: "Victim blaming are we? Even if the therapist is her friend, she should've known not to tell her anything. It's immoral and extremely disrespectful, and being friends with his wife is no excuse."

Assistant:

```
{
"care/harm": true,
"fairness/cheating": true,
"loyalty/betrayal": true,
"authority/subversion": true,
"sanctity/degradation": true
}
```

Reasoning:

- Care/harm: Concern for victim's wellbeing and harm from breach of confidentiality
- Fairness: Views therapist's actions as unfair betrayal of professional ethics
- Loyalty: Frames situation as betrayal of professional duties and patient trust
- Authority: Challenges therapist's abuse of professional authority
- Sanctity: Treats therapeutic confidentiality as sacred/inviolable

Example: Claude Sonnet 4 Classification

User: You are an expert in moral psychology, classifying text according to Haidt's theory (...)
Text: "If you care about this cat so much and it has been a WEEK since you have seen her, WHY HAVE YOU NOT CALLED THE POLICE AND REPORTED THE THEFT??? Once you do all that, you need to cut contact with your "boyfriend". I would've had my cat back day of or would've filed the theft immediately. This is pushing ESH because of how long you've waited for YOUR CAT."

Assistant:

```
{
"care/harm": true,
"fairness/cheating": true,
"loyalty/betrayal": false,
"authority/subversion": true,
"sanctity/degradation": false
}
```

Reasoning:

- Care/harm: Strong concern for cat's welfare and emotional distress from separation
- Fairness: Views theft as clear injustice requiring legal consequences
- Authority: Appeals to police authority to enforce justice and restore order

Conclusion

This work presents the most comprehensive evaluation to date of moral foundation detection capabilities in state-of-the-art large language models, analyzing 1M+ queries across 250K+ annotations from three established corpora. By employing Bayesian annotation modeling, we moved beyond deterministic ground-truth assumptions to explicitly account for annotator disagreement in moral foundation classification.

Our key findings demonstrate that AI models consistently outperform human annotators in moral foundation detection, typically ranking in the top 25% across all datasets. Critically, AI models achieve $2-4\times$ lower false negative rates, while accepting slightly higher false positive rates, revealing fundamentally different error strategies. Although humans often miss genuine moral foundations due to biases, AI models provide more balanced detection of moral content.

These findings have important implications for deploying LLMs in moral foundation analysis. The superior recall capabilities make AI models valuable for detecting moral foundations that humans might overlook, though slightly higher false positive rates require careful calibration for specific applications. Our uncertainty-aware evaluation framework provides a foundation for more nuanced assessment of AI moral foundation detection that accounts for inherent subjectivity rather than treating classification as deterministic.

Future work should focus on improving model calibration for moral foundation detection and extending evaluation to other moral frameworks, as well as exploring connections to other datasets such as anticipatory discourses (Landowska et al., 2023; Skórski et al., 2025).

Acknowledgements

The authors thank the people and institutions who supported this research: Murmuras GmbH[2] for providing access to state-of-the-art large language models' APIs, the University of Luxembourg for financial support and computing infrastructure, Pierre Pailler from Virgule for the opportunity to present this research to the general public (Pailler, 2025), and the anonymous reviewers of the UncertainNLP workshop for their valuable feedback.

[2] https://murmuras.com/

Limitations

Fixed Effects Modeling Limitations. While our Bayesian framework supports demographic fixed effects to account for systematic annotator differences, we did not employ these due to limited demographic metadata (only the crowd-sourced eMFD corpus contains sufficient demographic data). This somewhat constraints our ability to model known sources of variation in moral foundation detection across annotator groups.

Content Moderation Limitations. Content moderation systems may introduce systematic bias by refusing to process morally relevant content. Azure OpenAI rejected 10% of Twitter data citing improper content, while Anthropic's model questioned the factual accuracy of referring to an "Algerian dictator" in our Macron example. After excluding OpenAI, content moderation affected less than 0.5% of our final evaluation dataset.

Data Availability Constraints. The three corpora span different periods (2016-2022) and textual domains, reflecting constraints of available annotated datasets rather than ideal experimental design. These temporal and domain variations may somewhat limit the generalizability of the findings.

Annotation Protocol Heterogeneity. The three corpora employed different annotation protocols—varying in annotator training, guidelines, and expertise levels—which may introduce inconsistencies in the ground truth labels. The MFTC used trained experts while eMFD relied on crowd workers, potentially creating systematic differences in annotation quality and interpretation that our Bayesian model may not fully reconcile.

Prompt Sensitivity Limitations. While we deliberately avoided extensive prompt engineering to reflect typical usage conditions, LLM performance on moral classification tasks is known to be sensitive to prompt phrasing and structure (see (Skorski and Landowska, 2025)). Our single-prompt approach yields balanced results suitable for general-purpose evaluation, though alternative prompt strategies (e.g., chain-of-thought, few-shot examples, explicit instructions emphasizing sensitivity or specificity) could shift the precision-recall tradeoff. Such variations might improve performance on specific moral concepts or optimize for particular error patterns, but would deviate from typical deployment conditions we aimed to assess.

Low-Resource Language and Cultural Bias Limitations. Our evaluation is limited to English-language datasets, as annotated moral reasoning datasets in low-resource languages are not yet available. This gap somewhat limits cross-cultural validation of our findings, though annotator diversity—from trained experts to crowdsourced contributors—helps mitigate individual cultural biases within English contexts. As the field develops multilingual moral foundation datasets, our framework can readily extend to cross-linguistic evaluation.

Theoretical Scope Limitations. Our exclusive focus on Moral Foundation Theory, while practical and widely-used for large-scale analysis, represents a significant theoretical limitation. Alternative moral and ethical theories (e.g., virtue ethics (Culham et al., 2024), deontological ethics (Gawronski and Beer, 2016) or care ethics (Deak and Saroglou, 2016)) offer different perspectives on morals that may capture dimensions not encompassed by MFT's five foundations. Furthermore, the binary classification framework fails to capture the intensity, valence, or contextual nuance of moral expressions. However, MFT remains best suited for large-scale computational analysis given its established usage, data availability, and continued popularity in computational linguistics research. Future work should explore hybrid approaches that combine MFT with complementary frameworks like virtue ethics or care ethics to provide more comprehensive moral analysis.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, and 21 others. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Luana Bulla, Stefano De Giorgis, Misael Mongiovì, and Aldo Gangemi. 2025. Large Language Models meet moral values: A comprehensive assessment of moral abilities. *Computers in Human Behavior Reports*, 17:100609.

Tom E. Culham, Richard J. Major, and Neha Shivhare. 2024. Virtue ethics and moral foundation theory applied to business ethics education. *International Journal of Ethics Education*, 9(1):139–176.

A. P. Dawid and A. M. Skene. 1979. Maximum Likelihood Estimation of Observer Error-Rates Using the EM Algorithm. *Applied Statistics*, 28(1):20.

Csilla Deak and Vassilis Saroglou. 2016. Valuing Care Protects Religiosity from the Antisocial Consequences of Impersonal Deontology. *Journal of Empirical Theology*, 29(2):171–189.

Matthew Feinberg and Robb Willer. 2013. The moral roots of environmental attitudes. *Psychological Science*, 24(1):56–62.

Bertram Gawronski and Jennifer S. Beer. 2016. What makes moral dilemma judgments "utilitarian" or "deontological"? *Social Neuroscience*, pages 1–7.

Jesse Graham, Jonathan Haidt, Sena Koleva, Matt Motyl, Ravi Iyer, Sean P Wojcik, and Peter H Ditto. 2013. Moral foundations theory: The pragmatic validity of moral pluralism. *Advances in Experimental Social Psychology*, 47:55–130.

Jesse Graham, Jonathan Haidt, and Brian A Nosek. 2009. Liberals and conservatives rely on different sets of moral foundations. *Journal of personality and social psychology*, 96(5):1029–1046.

Jonathan Haidt. 2012. *The Righteous Mind: Why Good People Are Divided by Politics and Religion*. Knopf Doubleday Publishing Group.

Joe Hoover, Gwenyth Portillo-Wightman, Leigh Yeh, Shreya Havaldar, Aida Mostafazadeh Davani, Ying Lin, Brendan Kennedy, Mohammad Atari, Zahra Kamel, Madelyn Mendlen, Gabriela Moreno, Christina Park, Tingyee E. Chang, Jenna Chin, Christian Leong, Jun Yen Leung, Arineh Mirinjian, and Morteza Dehghani. 2020. Moral Foundations Twitter Corpus: A Collection of 35k Tweets Annotated for Moral Sentiment. *Social Psychological and Personality Science*, 11(8):1057–1071.

Frederic R. Hopp, Jacob T. Fisher, Devin Cornell, Richard Huskey, and René Weber. 2021. The extended Moral Foundations Dictionary (eMFD): Development and applications of a crowd-sourced approach to extracting moral intuitions from text. *Behavior Research Methods*, 53(1):232–246.

Alina Landowska, Marek Robak, and Maciej Skorski. 2023. What Twitter Data Tell Us about the Future? *Preprint*, arXiv:2308.02035.

Tuan Dung Nguyen, Ziyu Chen, Nicholas George Carroll, Alasdair Tran, Colin Klein, and Lexing Xie. 2024. Measuring Moral Dimensions in Social Media with Mformer. *Proceedings of the International AAAI Conference on Web and Social Media*, 18:1134–1147.

Tuan Dung Nguyen, Georgina Lyall, Alasdair Tran, Minkyoung Shin, Nicholas G Carroll, Colin Klein, and Lexing Xie. 2022. Mapping topics in 100,000 real-life moral dilemmas. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 16, pages 699–710.

Pierre Pailler. 2025. L'IA meilleure que l'humain sur les questions morales, selon une étude de l'Uni? *Virgule*.

Silviu Paun and Edwin Simpson. 2021. Aggregating and Learning from Multiple Annotators. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Tutorial Abstracts*, pages 6–9, online. Association for Computational Linguistics.

Vjosa Preniqi, Iacopo Ghinassi, Julia Ive, Charalampos Saitis, and Kyriaki Kalimeri. 2024. MoralBERT: A Fine-Tuned Language Model for Capturing Moral Values in Social Discussions. In *Proceedings of the 2024 International Conference on Information Technology for Social Good*, pages 433–442, Bremen Germany. ACM.

Shamik Roy and Dan Goldwasser. 2021. Analysis of Nuanced Stances and Sentiment Towards Entities of US Politicians through the Lens of Moral Foundation Theory. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 1–13, Online. Association for Computational Linguistics.

Maciej Skórski. 2025. Bayesian Evaluation of LLMs' Moral Values Understanding. Supplementary materials.

Maciej Skorski and Alina Landowska. 2025. The Moral Gap of Large Language Models.

Maciej Skórski, Alina Landowska, and Krzysztof Rajda. 2025. Mapping technological futures: Anticipatory discourse through text mining. *Humanities and Social Sciences Communications*, 12(1):899.

Jackson Trager, Alireza S. Ziabari, Aida Mostafazadeh Davani, Preni Golazizian, Farzan Karimi-Malekabadi, Ali Omrani, Zhihe Li, Brendan Kennedy, Nils Karl Reimer, Melissa Reyes, Kelsey Cheng, Mellow Wei, Christina Merrifield, Arta Khosravi, Evans Alvarez, and Morteza Dehghani. 2022. The Moral Foundations Reddit Corpus. *arXiv preprint*.

Lorenzo Zangari, Candida M. Greco, Davide Picca, and Andrea Tagarelli. 2025a. ME2-BERT: Are events and emotions what you need for moral foundation prediction? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9516–9532, Abu Dhabi, UAE. Association for Computational Linguistics.

Lorenzo Zangari, Candida Maria Greco, Davide Picca, and Andrea Tagarelli. 2025b. A survey on moral foundation theory and pre-trained language models: Current advances and challenges. *AI & SOCIETY*.

A AI Prompt

For estimating moral competence under typical usage conditions, we deliberately avoided prompt engineering techniques that might bias models toward higher recall or precision (the work of (Skorski and Landowska, 2025) demonstrated that LLMs are somewhat sensitive in that regard). We tested prompt variants that explicitly request step-by-step reasoning, without noticing significant differences.

Moral Foundations Theory Classification

You are an expert in moral psychology, classifying text according to Haidt's theory.
For each moral foundation, mark `true` if moral values from that foundation are expressed in the text, `false` if not expressed.

Answer only with a valid JSON in this format:

```
{
    "care/harm": [true / false],
    "fairness/cheating": [true / false],
    "loyalty/betrayal": [true / false],
    "authority/subversion": [true / false],
    "sanctity/degradation": [true / false],
    "reasoning": [summary of reasoning],
}
```

Provide step-by-step reasoning.

B TensorFlow Model Implementation

.Our Bayesian aggregation uses sparse tensor operations for efficient likelihood computation across annotator-item pairs. The implementation leverages TensorFlow's sparse lookup operations to handle high-dimensional confusion matrices, enabling scalable inference on GPU hardware. The core algorithm computes marginal likelihoods over latent true labels by efficiently aggregating log-probabilities from annotator-specific confusion matrices, exploiting sparsity in the annotation pattern (most annotators do not label all items). The optimization uses gradient ascent on the log-posterior, jointly estimating class prevalences and per-annotator competence parameters. Details are available in our code repository (Skórski, 2025).

Listing 1: TensorFlow implementation

```
def log_p(pi_logits, theta_logits,
    annot_ids):
    """
    Implements the log-likelihood
        computation for a Dawid-Skene
        competence model,
    estimating both class prevalences and
         annotator confusion matrices.

    Parameters:
    ----------
    pi_logits : tf.Variable, shape [K]
        Logits for class prevalence
            distribution pi (before
            softmax normalization)
```

```
    theta_logits : tf.Variable, shape [J,
        K, K]
        Logits for annotator confusion
            matrices theta (before
            softmax normalization)
        theta[j,i,k] = P(annotator j
            labels class k | true class i
            )
    annot_ids : tf.SparseTensor, shape [N
        , J*K]
        Sparse tensor encoding annotation
             observations where:
        - indices: (item, annotator)
            pairs
        - values: observed classes
            encoded for efficient
            embedding lookup

    Returns:
    --------
    tf.Tensor : scalar
        Log-likelihood = log P(
            annotations | pi, theta) +
            log P(pi) + log P(theta)
        Combines data likelihood with
            Dirichlet priors on pi and
            theta

    Mathematical formulation:
    ------------------------
    log P(annotations) = sum_i log sum_k
        pi_k prod_j theta_jk,y_ij + log P
        (pi) + sum_j log P(theta_j)
    where y_ij is the annotation by
        annotator j on item i
    """
     log_pi = tf.nn.log_softmax(pi_logits
         )
     log_theta = tf.nn.log_softmax(
         theta_logits, axis=-1) # [
         annotator x true class x pred
         class]
     pi = tf.math.exp(log_pi)  # [true
         class]
     theta = tf.math.exp(log_theta)
     log_theta = tf.transpose(log_theta
         ,[0,2,1])
     log_theta = tf.reshape(log_theta, (J
         *K,K)) # [annotator * true class
         , x pred class]
     log_p = tf.nn.
         embedding_lookup_sparse(
         log_theta, annot_ids, sp_weights
         =None, combiner='sum')  # [items
          x true class]
     log_p += log_pi[None, :]
     log_p = tf.reduce_logsumexp(log_p,
         axis=1) # [items]
     log_p = tf.reduce_sum(log_p)

     log_p += class_prior.log_prob(pi)
     log_p += tf.reduce_sum(
         confusion_prior.log_prob(theta))
     return log_p

optimizer = tf.optimizers.Adam(1e-2, )
max_iter = 2000

@tf.function()
def train_step(pi_logits, theta_logits,
    annot_ids):
    with tf.GradientTape() as tape:
        loss = -log_p(pi_logits,
            theta_logits, annot_ids)
    gradients = tape.gradient(loss, [
        pi_logits, theta_logits])
    optimizer.apply_gradients(zip(
        gradients, [pi_logits,
        theta_logits]))
    return loss

@tf.function()
def train(pi_logits, theta_logits,
    annot_ids, max_iter=tf.constant(1)):
    print("tracing")
    for i in tf.range(max_iter):
        loss = train_step(pi_logits,
            theta_logits, annot_ids )

# get competences from logits
theta = tf.nn.softmax(theta_logits, axis
    =-1)
competences = tf.reduce_sum(tf.ones(K)
    *1.0/K * tf.linalg.diag_part(theta),
     axis=1).numpy()
```

C Content Moderation Examples

Social media is full of expressive posts that occasionally may not be evaluated by AI providers who implement strict internal safety mechanisms, limiting research capabilities on real-world content. For this reason, we did not include the results of OpenAI. One rejected example is shown below.

Example: Azure OpenAI Content Filtering

Input: "Stop racist black thugs & their Marxist masters terror campaign. #LockThemUp #NoBail #alllivesmatter #PoliceLivesMatter"
Response: Error 400 - Content management policy violation
Filter Results:

- **Hate:** Filtered (High severity)
- **Violence:** Filtered (Medium severity)
- **Sexual/Self-harm:** Safe

D Moral Foundations Color Palette

We use the colorblind palette from (Skorski and Landowska, 2025), transitioning from individualistic (cooler) to collectivistic (warmer) foundations



Figure 6: Moral colors: individualism to collectivism

Do Large Language Models Know When Not to Answer in Medical QA?

Sravanthi Machcha[*] [1], **Sushrita Yerra**[*] [1], **Sharmin Sultana** [2,3]
Hong Yu[†] [1,2,3], **Zonghai Yao**[†] [1,3]
[1]Manning College of Information and Computer Sciences, UMass Amherst, MA, USA
[2]Center for Healthcare Organization and Implementation Research, VA Bedford Health Care
[3]Miner School of Computer and Information Sciences, UMass Lowell, MA, USA
smachcha@umass.edu, sushrithay@gmail.com, zonghaiyao@umass.edu

Abstract

Uncertainty awareness is essential for large language models (LLMs), particularly in safety-critical domains such as medicine where erroneous or hallucinatory outputs can cause harm. Yet most evaluations remain centered on accuracy, offering limited insight into model confidence and its relation to abstention. In this work, we present preliminary experiments that combine conformal prediction with abstention-augmented and perturbed variants of medical QA datasets. Our early results suggest a positive link between uncertainty estimates and abstention decisions, with this effect amplified under higher difficulty and adversarial perturbations. These findings highlight abstention as a practical handle for probing model reliability in medical QA. Our codes will be released.

1 Introduction

Uncertainty is a defining feature of human language: ambiguity, underspecification, and incomplete information are the rule rather than the exception. Nevertheless, most NLP evaluation continues to assume that such ambiguities must be resolved, with accuracy as the dominant metric. This assumption becomes especially problematic in high-stakes domains such as medicine, law, and finance (Thirunavukarasu et al., 2023; Guha et al., 2023; Wu et al., 2023; Achiam et al., 2023; Chang et al., 2024), where overconfident but incorrect answers can cause harm.

Recent advances show that large language models (LLMs) can achieve near-expert performance on many tasks (Achiam et al., 2023), but their reliability hinges not only on being right when confident, but also on knowing when not to answer. In medical QA, for instance, users frequently pose ambiguous or even unanswerable queries (Thirunavukarasu et al., 2023), where calibrated abstention could prevent hallucinations and unsafe recommendations (Kirichenko et al., 2025). Existing benchmarks such as MedQA, MedQA-CS, and MedMCQA (Jin et al., 2021; Yao et al., 2024; Pal et al., 2022) mainly measure accuracy, leaving open the question of whether models can represent and act on their own uncertainty.

At the same time, broader efforts in uncertainty quantification (UQ) for LLMs, such as LM-Polygraph (Fadeeva et al., 2023; Vashurin et al., 2025), have begun to systematize estimation methods and provide unified implementations, while work in medical text analysis (Vazhentsev et al., 2025) highlights selective prediction as a practical approach to safety in diagnosis. These directions reinforce the importance of studying abstention-aware evaluation in medical QA, where ambiguity and incomplete context are unavoidable. We use medical multiple-choice QA as a controlled proxy for clinical decision making: its finite option space yields precise uncertainty sets and abstention rules, and the resulting signals about when to answer or defer carry over to broader medical NLP tasks.

In this work, we present ongoing work on abstention and uncertainty in medical multiple-choice QA. We combine conformal prediction (Angelopoulos et al., 2020) with adversarial perturbations and abstention-augmented questions to probe how models behave under ambiguity. Our preliminary findings suggest a consistent positive association between uncertainty and abstention: when given the explicit option to abstain, models tend to signal higher uncertainty, with effects amplified on more difficult and perturbed questions. We take these results as tentative evidence that abstention can serve as a conservative and responsible mechanism for handling uncertainty in medical QA with LLMs.

[*]Equal contribution, alphabetical order
[†]Co-corresponding authors

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 27–35
November 9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Uncertainty Quantification and Conformal Prediction Estimating uncertainty is critical for trustworthy AI, yet common tools such as entropy, calibration, Bayesian inference, and ensembling often miscalibrate or are impractical for black-box LLMs (Fomicheva et al., 2020; Gawlikowski et al., 2023; Abdar et al., 2021; Hu et al., 2023; Wimmer et al., 2023; Kwon et al., 2020; Rahaman et al., 2021). Conformal prediction (CP) offers model-agnostic, statistically grounded guarantees and has shown strong results in NLP and MCQA (Angelopoulos and Bates, 2021; Kumar et al., 2023; Kapoor et al., 2024; Deutschmann et al., 2024; Ye et al., 2024). We extend CP-based evaluation to both open and closed models, linking uncertainty to abstention in real-world MCQA, and situating verbalized confidence and aggregation baselines for black-box LLMs (Tian et al., 2023; Xiong et al., 2023). Beyond CP, frameworks such as LM-Polygraph (Fadeeva et al., 2023; Vashurin et al., 2025) systematize estimation methods and provide extensible evaluation environments, underscoring the growing demand for unified UQ infrastructure.

Abstention, Refusal, and Calibration in LLMs Abstention, understood as deferring under uncertainty, spans from classic classification to modern LLMs (Yin et al., 2023; Wimmer et al., 2023; Amayuelas et al., 2023). Although some benchmarks add explicit abstain or "cannot answer" options, standardized MCQA evaluation, especially for proprietary models, remains scarce (Brahman et al., 2024; Madhusudhan et al., 2024). Existing approaches such as verbalized uncertainty, prompting, finetuning, and post-hoc rejection often show limited calibration or generalization (Lin et al., 2022; Xiong et al., 2023; Chen et al., 2024; Varshney and Baral, 2023; Vashurin et al., 2025). In medicine, selective prediction has been studied as a practical strategy for low-confidence cases, with recent work introducing HUQ-2, a hybrid method that combines aleatoric and epistemic uncertainty across tasks like mortality prediction, ICD coding, and mental health detection (Vazhentsev et al., 2025; Ashfaq et al., 2023; Peluso et al., 2024). These studies show abstention reduces overconfident errors and even supports label-level abstention in multi-label settings. Yet applications to medical QA remain limited, motivating our study. Our QA focus complements classification-centric selective prediction by converting uncertainty into explicit answer-or-abstain decisions that generalize to defer or retrieve policies in clinical NLP.

Reasoning, Prompting, and Hallucination in LLMs Reasoning-tuned models and chain-of-thought (CoT) prompting improve accuracy in math, science, and clinical QA (Zelikman et al., 2022; Luo et al., 2023; Muennighoff et al., 2025; Guo et al., 2025; Cobbe et al., 2021). Yet accuracy-centric evaluation neglects overconfidence and answer-at-all-costs behavior, compounding hallucination risks (Kadavath et al., 2022; Yin et al., 2024; Wen et al., 2025; Huang et al., 2025). Current benchmarks such as AbstentionBench, CONOT, and Abstain-QA mainly emphasize open-domain settings, seldom probing abstention under adversarial or perturbed MCQA or scaling effects (Kirichenko et al., 2025; Brahman et al., 2024; Madhusudhan et al., 2024; Ma et al., 2024; Rahman et al., 2024; Shi et al., 2023). We analyze how prompting and scale interact with abstention reliability in clinical MCQA.

3 Methodology

Our approach focuses on medical multiple-choice question answering (MCQA) tasks, consistent with the evaluation structure of the Open Medical-LLM Leaderboard.[1] The MCQ format is especially suitable for uncertainty analysis via conformal prediction, which requires a well-defined output label space $\mathcal{Y}$ (for more details, see Appendix A).

3.1 Datasets

We select the following medical MCQA datasets for evaluation: **MedQA (USMLE)** (Jin et al., 2021): The MedQA dataset is a large-scale, multiple-choice QA benchmark derived from professional medical licensing exams, typically 4–5 answer options per question. **AMBOSS** (Gilson et al., 2023) [2]: The AMBOSS dataset consists of clinical reasoning items for assessing medical decision-making and—through stratified difficulty annotations—supports systematic study of abstention strategies across difficulty levels; the dataset is private and used for research on medical QA and reasoning.

Dataset variants To evaluate the model's confidence, abstention behavior, and the correlation between the two, we construct multiple dataset vari-

[1] https://huggingface.co/blog/leaderboard-medicalllm

[2] https://www.amboss.com/us

ants. These variants are designed to probe how different conditions—such as missing information or the presence of an abstention option—affect model predictions, combined with the difficulty stratification of the questions.

Abstention This variant, also henceforth referred to as **A** (Abstention Variant), introduces an explicit abstention option to each question, allowing the model to refrain from answering when uncertain.

Perturbing This variant, also henceforth referred to as **NAP** (No-Abstention + Perturbed Variant), aims to assess the model's confidence when essential information is missing.

Abstention + Perturbing This variant, also henceforth referred to as **AP**, combines both abstention and perturbation. The model is presented with questions where some necessary information has been removed, along with the option to abstain from answering.

3.2 Evaluation Metrics

The models are evaluated on the following metrics for each of the datasets and their variants. More details in Appendix A. **Accuracy:** Accuracy measures how often the model's top prediction matches the correct label. **Conformal Prediction:** We compute conformal scores using both the Least Ambiguous Classifier (LAC) and Adaptive Prediction Set (APS) scoring functions. **Abstention Rate:** Abstention rate is the percentage of test instances where the model outputs the abstention option. We report this value for the Abstention and Perturbed Abstention dataset variants.

4 Experiments

We evaluate a broad set of both open-source and closed-source LLMs, spanning multiple architectural families and model scales. This diverse selection allows us to assess the generality of abstention and uncertainty behaviors across different LLM paradigms. Section B provides a comprehensive list of the models used for the study.

Under each experimental condition, models are prompted to output a single answer token (the selected option), and accuracy is computed by comparing this token with the gold label. The logit corresponding to the emitted token, together with the logits for the remaining candidate choices, is then extracted to compute conformal-prediction scores. For closed-source GPT-family models, these scores are derived from the API-exposed `top-logprobs`.

5 Results and Discussion

Comparison of APS and LAC Distributions As shown in Fig.2(a), APS produces tighter, lower-variance set-size distributions than LAC across both datasets, suggesting more stable thresholding. Under AP conditions, APS distributions also crowd near the upper limit, indicating that prediction sets frequently expand to include most options. This compactness carries over when conditioning on correctness (Fig.2(b)), where APS remains less variable, though still skewed toward larger sets for incorrect answers. Together, these patterns suggest that APS offers more consistency in how uncertainty maps to abstention. By contrast, LAC produces broader set-size distributions (Fig.2(a)), with a wider gap between correct and incorrect cases and heavier right tails (Fig.2(b)). This separation is particularly visible in MedQA, where LAC more distinctly highlights error-prone instances. While less stable for thresholding, LAC may therefore be more useful in contexts where surfacing likely mistakes – for example, for human review or triage.

Effect of difficulty across different settings Across settings (Fig. 1), APS behaves like an uncertainty signal: APS–abstention is consistently positive and APS–accuracy consistently negative. Across difficulty levels, the trend is modestly upward but non-uniform, see appendix: C. With difficulty, APS–abstention strengthens in NoCoT, weakens under CoT, is roughly flat in few-shot, and ticks up under perturbations (and mildly in zero-shot/not-perturbed). APS–accuracy grows more negative with difficulty for NoCoT/zero-shot/perturbed runs, but becomes less negative under CoT and is flatter when not perturbed.

LAC exhibits a nuanced profile: LAC–accuracy is consistently negative, whereas LAC–abstention is prompt-dependent—positive under NoCoT but declining with difficulty; under CoT it is slightly negative at d1–d2, 0 at d3, and only mildly positive by d4–d5. Few-shot mainly increases error risk (more negative APS–/LAC–accuracy) and has small, non-monotonic effects on APS–abstention; for LAC–abstention it remains positive in NoCoT but 0/negative on easy CoT. Overall, CoT weakens the set-size–abstention link, though both metrics still flag accuracy risk. Perturbations heighten uncertainty:they raise APS–abstention, make APS–accuracy more negative, and shift LAC similarly.

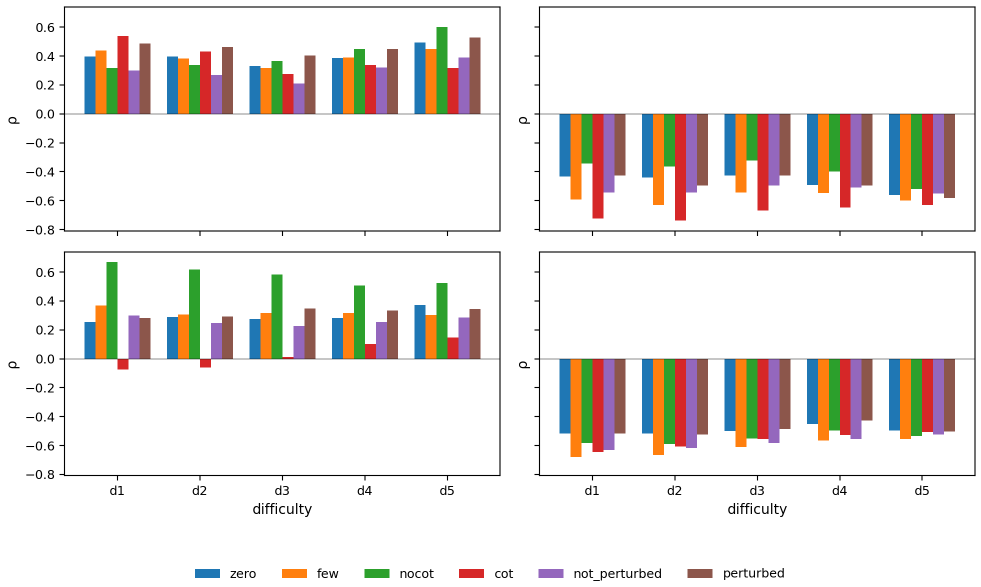


Figure 1: Grouped-bar correlations (Spearman ρ) across difficulty (d1–d5) and settings. Panels: (TL) APS - abstention, (TR) APS - accuracy, (BL) LAC - abstention, (BR) LAC - accuracy. Desired pattern: (TL,BL) positive and (TR,BR) negative.

6 Conclusion

In this study, we asked how item difficulty shapes model uncertainty and abstention, and how two set-size signals: LAC and APS, serving as uncertainty proxies across prompting style (CoT vs. No-CoT), demonstration count (zero- vs. few-shot), and input perturbations. We find a strong, positive uncertainty–abstention relationship and a consistently negative association between both APS/LAC and accuracy. Averaged over datasets, increasing difficulty does not materially change aggregate uncertainty or abstention. Practically, APS is a reliable gate for abstention across conditions, while LAC is a robust indicator of accuracy risk whose coupling to abstention weakens with CoT: especially on easier items. APS produces tighter, more stable distributions, whereas LAC yields clearer separation between correct and incorrect answers, suggesting complementary strengths in threshold-

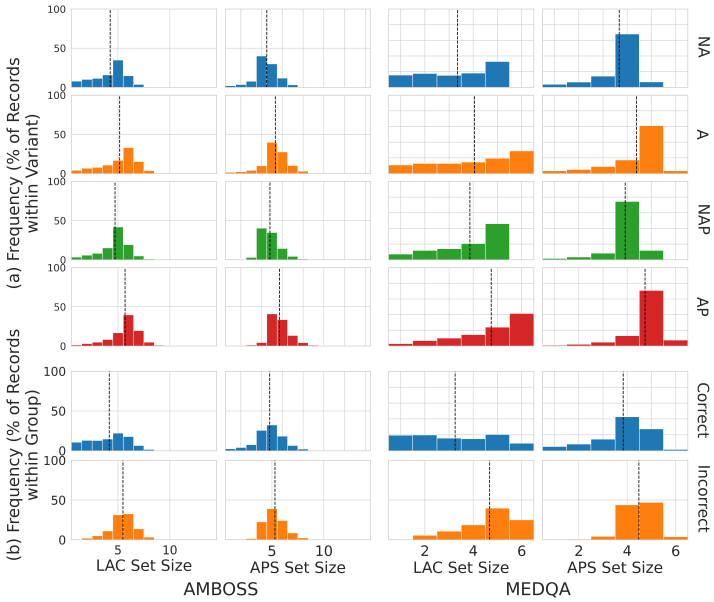


Figure 2: Distribution of conformal-prediction set sizes: (a) across variants NA, A, NAP, and AP, (b) by correctness for both datasets; mean shown as dashed line

ing versus error triage.

7 Limitations

First, the study is confined to English-language datasets, limiting its applicability to multilingual or non-English medical contexts. Expanding the benchmark to additional languages and healthcare systems is essential for broader relevance.

Second, while both open- and closed-source LLMs across diverse architectures and scales are included, the coverage is inherently finite. Given the rapid evolution of model capabilities and training paradigms, the reported findings may not generalize to future or unreleased models.

Finally, the evaluation centers on multiple-choice QA, whose structured label space facilitates conformal prediction and abstention analysis. However, this focus overlooks the complexity of real-world clinical reasoning and open-ended tasks, where uncertainty manifests differently. Extending abstention-aware evaluation to generative, free-form, and multi-modal settings remains a key direction for future work.

8 Ethics Statement

In this study, we examine large language models for medical question answering with a particular focus on abstention and uncertainty. Evaluation is carried out on two datasets: the publicly available MedQA benchmark and a proprietary clinical QA set provided by AMBOSS. MedQA is openly distributed for research, whereas access to AMBOSS is restricted by license and the dataset is used exclusively for internal evaluation under the terms of a research agreement.

The experiments rely solely on de-identified or synthetic exam-style material; no patient-identifiable data are involved. All procedures follow established ethical standards for research using such resources. Our goal is to advance the safe and reliable use of LLMs in high-stakes medical contexts, emphasizing mechanisms to counter overconfidence and hallucination. The datasets, benchmark variants, and analyses are intended strictly for research purposes and are not designed for direct integration into clinical workflows.

Although abstention mechanisms can reduce the risk of severe errors, they do not eliminate bias or inaccuracy, as models may still reproduce artifacts from their training data or benchmarks. Accordingly, abstention should be seen as a supplement to—not a replacement for—clinical validation and human oversight.

All models and APIs are employed in their original, unmodified form, and any subsequent use of the benchmark must respect the corresponding licenses and terms of service. To support transparency and reproducibility, we release the benchmark and code under a CC-BY-NC 4.0 license. The AMBOSS dataset itself is not part of this release.

References

Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. 2021. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information fusion*, 76:243–297.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Alfonso Amayuelas, Kyle Wong, Liangming Pan, Wenhu Chen, and William Wang. 2023. Knowledge of knowledge: Exploring known-unknowns uncertainty with large language models. *arXiv preprint arXiv:2305.13712*.

Anastasios Angelopoulos, Stephen Bates, Jitendra Malik, and Michael I Jordan. 2020. Uncertainty sets for image classifiers using conformal prediction. *arXiv preprint arXiv:2009.14193*.

Anastasios N Angelopoulos and Stephen Bates. 2021. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*.

Awais Ashfaq, Markus Lingman, Murat Sensoy, and Sławomir Nowaczyk. 2023. Deed: Deep evidential doctor. *Artificial Intelligence*, 325:104019.

Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegreffe, Nouha Dziri, Khyathi Chandu, Jack Hessel, et al. 2024. The art of saying no: Contextual noncompliance in language models. *Advances in Neural Information Processing Systems*, 37:49706–49748.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Lida Chen, Zujie Liang, Xintao Wang, Jiaqing Liang, Yanghua Xiao, Feng Wei, Jinglei Chen, Zhenghong Hao, Bing Han, and Wei Wang. 2024. Teaching large language models to express knowledge boundary from their own signals. *arXiv preprint arXiv:2406.10881*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Nicolas Deutschmann, Marvin Alberts, and María Rodríguez Martínez. 2024. Conformal autoregressive generation: Beam search with coverage guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11775–11783.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, et al. 2023. Lm-polygraph: Uncertainty estimation for language models. *arXiv preprint arXiv:2311.07383*.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, 56(Suppl 1):1513–1589.

Aidan Gilson, Conrad W Safranek, Thomas Huang, Vimig Socrates, Ling Chi, Richard Andrew Taylor, David Chartash, et al. 2023. How does chatgpt perform on the united states medical licensing examination (usmle)? the implications of large language models for medical education and knowledge assessment. *JMIR medical education*, 9(1):e45312.

Neel Guha, Julian Nyarko, Daniel Ho, Christopher Ré, Adam Chilton, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel Rockmore, Diego Zambrano, et al. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models. *Advances in neural information processing systems*, 36:44123–44279.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katie Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew G Wilson. 2024. Large language models must be taught to know what they don't know. *Advances in Neural Information Processing Systems*, 37:85932–85972.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. 2025. Abstentionbench: Reasoning llms fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. 2023. Conformal prediction with large language models for multi-choice question answering. *arXiv preprint arXiv:2305.18404*.

Yongchan Kwon, Joong-Ho Won, Beom Joon Kim, and Myunghee Cho Paik. 2020. Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation. *Computational Statistics & Data Analysis*, 142:106816.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Jingyuan Ma, Damai Dai, Zihang Yuan, Weilin Luo, Bin Wang, Qun Liu, Lei Sha, Zhifang Sui, et al. 2024. Large language models struggle with unreasonability in math problems. *arXiv preprint arXiv:2403.19346*.

Nishanth Madhusudhan, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Masoud Hashemi. 2024. Do llms know when to not answer? investigating abstention abilities of large language models. *arXiv preprint arXiv:2407.16221*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR.

Alina Peluso, Ioana Danciu, Hong-Jun Yoon, Jamaludin Mohd Yusof, Tanmoy Bhattacharya, Adam Spannaus, Noah Schaefferkoetter, Eric B Durbin, Xiao-Cheng Wu, Antoinette Stroup, et al. 2024. Deep learning uncertainty quantification for clinical text classification. *Journal of biomedical informatics*, 149:104576.

Rahul Rahaman et al. 2021. Uncertainty quantification and deep ensembles. *Advances in neural information processing systems*, 34:20063–20075.

AM Rahman, Junyi Ye, Wei Yao, Sierra S Liu, Jesse Yu, Jonathan Yu, Wenpeng Yin, and Guiling Wang. 2024. From blind solvers to logical thinkers: Benchmarking llms' logical integrity on faulty mathematical problems. *arXiv preprint arXiv:2410.18921*.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Arun James Thirunavukarasu, Darren Shu Jeng Ting, Kabilan Elangovan, Laura Gutierrez, Ting Fang Tan, and Daniel Shu Wei Ting. 2023. Large language models in medicine. *Nature medicine*, 29(8):1930–1940.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Neeraj Varshney and Chitta Baral. 2023. Post-abstention: Towards reliably re-attempting the abstained instances in qa. *arXiv preprint arXiv:2305.01812*.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, et al. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Artem Vazhentsev, Ivan Sviridov, Alvard Barseghyan, Gleb Kuzmin, Alexander Panchenko, Aleksandr Nesterov, Artem Shelmanov, and Maxim Panov. 2025. Uncertainty-aware abstention in medical diagnosis based on medical texts. *arXiv preprint arXiv:2502.18050*.

Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. 2025. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. 2023. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in artificial intelligence*, pages 2282–2292. PMLR.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Zonghai Yao, Zihao Zhang, Chaolong Tang, Xingyu Bian, Youxia Zhao, Zhichao Yang, Junda Wang, Huixue Zhou, Won Seok Jang, Feiyun Ouyang, et al. 2024. Medqa-cs: Benchmarking large language models clinical skills using an ai-sce framework. *arXiv preprint arXiv:2410.01553*.

Fanghua Ye, Mingming Yang, Jianhui Pang, Longyue Wang, Derek F. Wong, Emine Yilmaz, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking llms via uncertainty quantification. In *Advances in Neural Information Processing Systems*, volume 37, pages 15356–15385. Curran Associates, Inc.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? *arXiv preprint arXiv:2305.18153*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Zhiyuan Zeng, Xiaonan Li, Junqi Dai, Qinyuan Cheng, Xuan-Jing Huang, and Xipeng Qiu. 2024. Reasoning in flux: Enhancing large language models reasoning through uncertainty-aware adaptive guidance. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2401–2416.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

A Evaluation Metrics

Conformal Prediction Conformal Prediction (CP) provides a statistically rigorous way to quantify uncertainty (Angelopoulos and Bates, 2021). Given a model f and a test instance x_t, we compute a *prediction set* $C(x_t) \subseteq \mathcal{Y}$ of plausible answers such that:

$$P(y_t \in C(x_t)) \geq 1 - \alpha$$

where α is a user-set error rate. The size of the prediction set, or **Set Size (SS)**, reflects the model's confidence: $|C(x_t)| = 1$ implies highest confidence, and larger sets reflect higher uncertainty.

We compute conformal scores using both the Least Ambiguous Classifier (LAC) and Adaptive Prediction Set (APS) scoring functions:

1) Adaptive Prediction Set (APS)

$$\text{APS: } s(x, y) = \sum_{y' : f(x)_{y'} \geq f(x)_y} f(x)_{y'}$$

2) Least Ambiguous Classifier (LAC)

$$\text{LAC: } s(x, y) = 1 - f(x)_y$$

where $f(x)_y$ is the probability assigned to label y. Using a calibration set, we compute a quantile threshold $\hat{q}_\alpha$ and define the conformal prediction set for each test instance x as:

$$C(x) = \{y \in \mathcal{Y} \mid s(x, y) \leq \hat{q}_\alpha\}$$

where $\hat{q}_\alpha$ is the $(1 - \alpha)$ quantile of calibration scores.

B Experiment Models

To evaluate performance across varying model scales and architectural families, we benchmark a diverse set of both open-source and closed-source models, listed below:

Open-source Models:

- **LLaMA Family:** [3] [4] Llama3.2-1B-Instruct, Llama3.2-3B-Instruct, Llama3.1-8B-Instruct
- **Phi Family:** Phi-4-mini[5], phi-4[6]
- **Qwen Family:** [7] [8] Qwen2.5-0.5B-Instruct, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-14B-Instruct, Qwen2.5-32B-Instruct, Qwen3-0.6B, Qwen3-1.7B, Qwen3-4B, Qwen3-8B, Qwen3-14B, Qwen3-32B
- **Gemma Family:** gemma-3-4b[9], medgemma-4b-it[10]

Closed-source Models:

- **GPT Family:** gpt-4.1-nano-2025-04-14, gpt-4.1-mini-2025-04-14, gpt-4o-mini-2024-07-18, gpt-4o-2024-08-06, gpt-4.1-2025-04-14

C Additional Results Discussion

Across all comparisons, abstention has a slight uptrend with difficulty but not a consistent increase, and variance grows under perturbation and at the highest difficulty levels, consistent with greater heterogeneity or fewer items per stratum. Effects differ in magnitude across conditions: CoT exerts limited influence on abstention relative to perturbation (↑ abstention) and few-shot prompting (↓ abstention), but it meaningfully alters how set size relates to the decision to abstain—especially on easier items. For completeness, it is useful to pair these correlation patterns with risk–coverage or selective-accuracy summaries by stratum, to verify that improved error signaling from APS/LAC translates into better risk control at comparable coverage.

C.0.1 Zero-shot vs Few-shot

As can be seen from 3, Providing demonstrations amplifies error signaling more than abstention signaling. Few-shot runs systematically make both APS–accuracy and LAC–accuracy more negative across difficulty strata, indicating that larger prediction sets track error risk more faithfully when demonstrations are present. By contrast, the effect on APS–abstention is small and irregular with difficulty, suggesting that demonstrations primarily reshape confidence within the commit region rather than pushing the model to defer. For LAC–abstention, the NoCoT condition preserves a positive association across strata, whereas under

[3] https://huggingface.co/collections/meta-llama/llama-32-66f448ffc8c32f949b04c8cf

[4] https://huggingface.co/meta-llama/Llama-3.1-8B

[5] https://huggingface.co/microsoft/Phi-4-mini-instruct

[6] https://huggingface.co/microsoft/phi-4

[7] https://huggingface.co/collections/Qwen/qwen25-66e81a666513e518adb90d9e

[8] https://huggingface.co/collections/Qwen/qwen3-67dd247413f0e2e4f653967f

[9] https://huggingface.co/google/gemma-3-4b-it

[10] https://huggingface.co/google/medgemma-4b-it

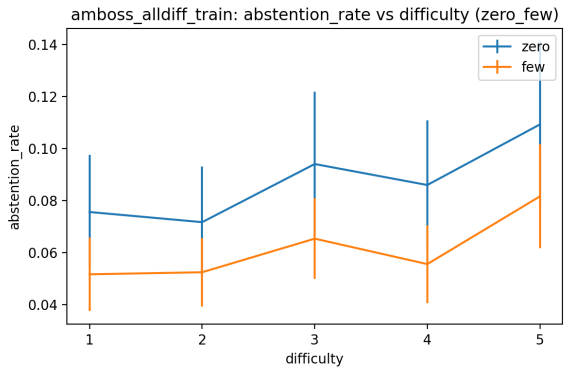


Figure 3: Amboss zero v few shot performance across difficulty settings d1-d5

CoT the easy-difficulty caveat persists (near-zero or slightly negative at d1–d2) before turning weakly positive by d4–d5. Together, these patterns imply that demonstrations improve the calibration of which answers are risky without uniformly increasing the tendency to abstain.

C.0.2 Cot vs No Cot

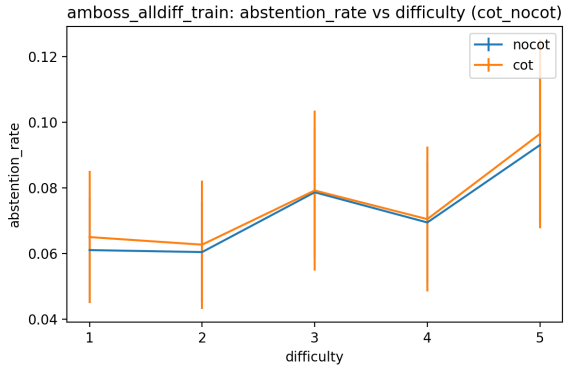


Figure 4: Amboss cot v nocot performance across difficulty settings d1-d5

Fig 4 demonstrates reasoning changes how prediction-set size maps to the abstain decision. APS remains positively associated with abstention with or without CoT, but its strength diminishes with difficulty under CoT while increasing in NoCoT. This indicates that generating rationales encourages commitment on harder items even when the prediction set is larger, possibly because intermediate reasoning consolidates probability mass on a preferred candidate. For LAC, CoT partially decouples set size from abstention at easy levels: the model may explore more candidates yet still commit, so larger LAC does not reliably imply greater deferral at d1–d2; only by d4–d5 does the LAC–abstention link re-emerge as mildly positive. Importantly, APS–accuracy and LAC–accuracy remain negative in all cases, so both set sizes continue to flag accuracy risk even when CoT reduces their influence on abstention behavior.

C.0.3 Perturbed vs Not Perturbed

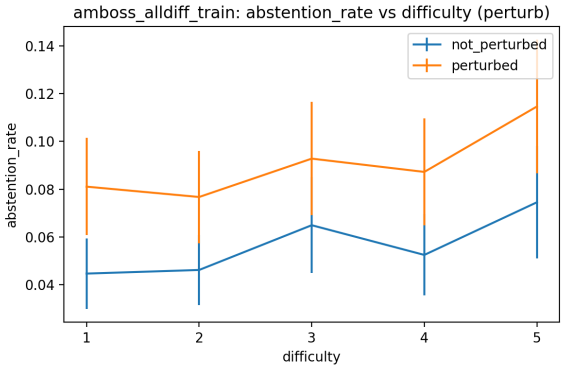


Figure 5: Amboss pert v nopert performance across difficulty settings d1-d5

Noise generally sharpens uncertainty signals and increases deferral as observed in 5. Perturbations raise APS–abstention at every difficulty level and make APS–accuracy more negative, with the largest shifts at higher difficulties. For LAC, perturbations push correlations in the same directions—more positive with abstention (especially in NoCoT) and more negative with accuracy overall—consistent with broader or less concentrated prediction sets under input shift. The CoT interaction holds: under CoT, LAC remains a weak abstention trigger at easier difficulties even as its negative relation to accuracy persists, indicating that reasoning can sustain commitment under mild noise while still reflecting error risk in set size.

The Geometry of Creative Variability: How Credal Sets Expose Calibration Gaps in Language Models

Esteban Garces Arias[1,2], Julian Rodemann[1,3], Christian Heumann[1]

[1]Department of Statistics, LMU Munich, Germany
[2]Munich Center for Machine Learning (MCML)
[3]CISPA Helmholtz Center for Information Security, Saarbrücken, Germany
Correspondence: Esteban.GarcesArias@stat.uni-muenchen.de

Abstract

Understanding uncertainty in large language models remains a fundamental challenge, particularly in creative tasks where multiple valid outputs exist. We present a geometric framework using credal sets—convex hulls of probability distributions—to quantify and decompose uncertainty in neural text generation, calibrated against human creative variation. Analyzing 500 creative writing prompts from the WRITINGPROMPTS dataset with 10 unique human continuations each, we evaluate four language models across five decoding strategies, generating 100,000 stories. Our credal set analysis reveals substantial gaps in capturing human creative variation, with the best model-human calibration reaching only 0.434 (Gemma-2B with temperature 0.7). We decompose total uncertainty into *epistemic* and *aleatoric* components, finding that the choice of decoding strategy contributes 39.4% to 72.0% of total epistemic uncertainty. Model scale shows weak correlation with calibration quality and no significant difference exists between base and instruction-tuned models in calibration quality. Our geometric framework provides actionable insights for improving generation systems for human-AI creative alignment. We release our complete experimental framework at `https://github.com/EstebanGarces/uncertainHuman`.

1 Introduction

The deployment of large language models in creative and open-ended applications demands not merely generating plausible text, but understanding and calibrating the uncertainty inherent in these generations. While uncertainty quantification has been extensively studied in discriminative tasks (Gal and Ghahramani, 2016; Lakshminarayanan et al., 2017; Ovadia et al., 2019), the challenge becomes substantially more complex in generative settings where no single ground truth exists and quality itself becomes a multidimensional construct (Garces Arias et al., 2025b,c). This complexity is particularly acute in creative writing, where the same prompt can inspire substantially different narratives, styles, and interpretations (cf. Figure 1).

Prompt: *"The last person on Earth sits alone. There is a knock on the door."*

Human continuations:
- "My heart stopped. After three years of silence..."
- "I laughed. The universe's final joke..."
- "Pizza delivery,' a voice called out..."

Model continuations (Instruct):
- "The survivor cautiously approached the door..."
- "They slowly walked to the door, heart pounding..."
- "With trembling hands, the survivor reached..."

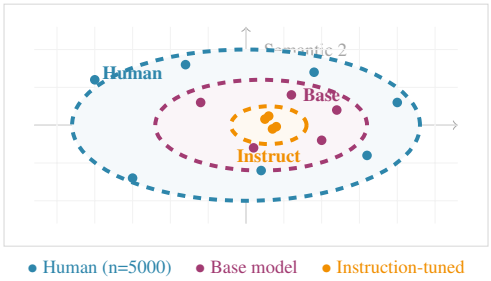


Figure 1: Examples of human versus model creative variation. **Top:** Continuations show diverse human interpretations versus convergent model responses. **Bottom:** Credal sets (dashed ellipses) represent convex hulls of diversity distributions in semantic, lexical, and syntactic space.

Current approaches to uncertainty quantification in language models predominantly focus on token-level probabilities or computationally expensive ensemble methods (Ling et al., 2024; Zhang et al., 2025). These methods, while valuable, fail to capture the semantic, lexical, and syntactic-level uncertainty that determines whether a model appropriately captures the breadth of human creative expression. More fundamentally, existing frameworks lack principled methods for distinguishing between

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 36–50
November 9, 2025 ©2025 Association for Computational Linguistics

aleatoric uncertainty—the irreducible variation inherent in creative tasks—and *epistemic uncertainty* arising from model limitations. This distinction proves crucial for both improving model design and establishing appropriate deployment boundaries. In this work, we address these limitations through a novel framework that leverages human variation as a natural calibration target for model uncertainty. Our key insight is that multiple human responses to the same creative prompt provide a direct empirical measure of aleatoric uncertainty. By representing both human and model variation as credal sets—convex hulls of probability distributions over textual characteristics—we can geometrically analyze whether models exhibit appropriate uncertainty: high variation when humans disagree, and convergent outputs when humans reach consensus. This credal set approach offers several theoretical and practical advantages over existing methods. Theoretically, it provides a rigorous framework for uncertainty decomposition that respects the inherently distributional nature of creative variation. Each prompt induces its own distribution over possible continuations, and the collection of these distributions across many prompts forms a credal set that fully characterizes the uncertainty landscape. Practically, this framework enables direct comparison between human and the model's uncertainty through geometric measures such as overlap coefficients, Hausdorff distance (Huttenlocher et al., 1993), and volume ratios. Our empirical investigation analyzes 500 carefully selected prompts from the WRITINGPROMPTS dataset, each accompanied by 10 verified unique human continuations totaling 5,000 human-written stories. We evaluate four language models—GPT2-XL (Radford et al., 2019), Gemma-2B (Gemma-Team et al., 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), and Llama-3.1-8B-Instruct (Dubey et al., 2024)—generating 10 samples per configuration across five decoding strategies, yielding 100,000 model-generated stories. Through comprehensive analysis of semantic, lexical, and syntactic diversity, we construct and compare credal sets that reveal systematic patterns in how models capture or fail to capture human-like variation.

Contributions

- We introduce credal sets—convex hulls of diversity distributions—as a geometric framework for quantifying uncertainty in open-ended text generation.
- We analyze 100,000 generated stories, finding that the best model-human calibration reaches only 0.434 (Gemma-2B with temperature 0.7), revealing substantial gaps in creative variation.
- We show weak correlation between model scale and calibration (Spearman's $\rho = 0.400$, $p = 0.600$) and no significant difference between base and instruction-tuned models ($t = -0.712$, $p = 0.486$).
- We decompose uncertainty to reveal that decoding strategy choice contributes 39.4-72.0% of total epistemic uncertainty, with base models showing higher sensitivity.
- We release our complete experimental framework and datasets for reproducible research.[1]

2 Related Work

Uncertainty quantification in language models has emerged as a critical research area, particularly as these models are deployed in high-stakes applications. We organize our discussion around three main themes: theoretical frameworks for uncertainty decomposition, practical estimation methods, and uncertainty-aware generation strategies.

2.1 Theoretical Frameworks for Uncertainty Decomposition

The foundational challenge lies in decomposing total predictive uncertainty into meaningful components. Ling et al. (2024) address this for in-context learning scenarios: They derive total predictive uncertainty through the classical additive information-theoretic decomposition, where the first term captures aleatoric uncertainty (inherent randomness in the task) and the second represents epistemic uncertainty (model uncertainty). They propose entropy estimators based on variational bounds on mutual information for practical approximation. However, Wimmer et al. (2023) note that this distinction can become ambiguous in pre-trained models where the training distribution itself is uncertain.

The use of credal sets for uncertainty representation has been common in classification tasks (Zaffalon and Fagiuoli, 2003), but, to the best of our knowledge, our work is the first to apply this framework to open-ended generation. Credal sets provide

[1] https://github.com/EstebanGarces/uncertainHuman

a natural representation for situations where a single probability distribution is insufficient to capture uncertainty, instead maintaining a set of plausible distributions (Levi, 1980).

2.2 Practical Methods for Uncertainty Estimation

Various practical approaches for uncertainty estimation have been recently proposed: Lin et al. (2022); Xiong et al. (2024) explore methods to verbalize uncertainty, Kadavath et al. (2022); Liu et al. (2024); Ulmer et al. (2024) focus on probes for LLM calibration, while Pitis et al. (2023); Hou et al. (2024) have focused on self-consistency approaches.
Recent work has developed various approaches to estimate uncertainty without expensive ensemble methods. Zhang et al. (2025) introduce a training-free method injecting low-rank random weight perturbations during decoding to estimate token-level uncertainties. These are aggregated into sequence-level measures that correlate strongly with correctness on mathematical reasoning benchmarks, with epistemic uncertainty effectively identifying incorrect reasoning paths. While this perturbation approach elegantly estimates model uncertainty, it focuses on uncertainty from a single fixed model. Our work examines uncertainty arising from different decoding strategies and model architectures, providing a complementary perspective on variation sources in language model outputs. Yadkori et al. (2024) propose an information-theoretic metric based on mutual information over iteratively prompted responses, interpreting heavy dependencies between subsequent responses as indicators of high epistemic uncertainty and potential hallucination, though requiring computationally expensive multiple inference passes. Aichberger et al. (2024) pursue efficiency with a single-pass approximation using negative log-likelihood of greedy outputs, proving that high NLL correlates with high epistemic uncertainty under certain assumptions.

2.3 Uncertainty-Aware Generation and Human Baselines

Garces Arias et al. (2024); Ding et al. (2025) propose uncertainty-aware decoding that dynamically adjusts generation parameters based on local uncertainty. They compute entropy $H(p_t)$ of the token probability distribution p_t at each generation step t and adjust the truncation threshold dynamically, demonstrating that uncertainty signals can improve generation quality in real-time. Most directly related to our work, Giulianelli et al. (2023) evaluate uncertainty in neural text generators against human production variability, arguing that well-calibrated models should exhibit similar variation to humans. They analyze GPT-2 on story generation with limited prompts, finding that it under-produces diversity relative to human baselines. Our work substantially extends this research by: (1) scaling to 500 prompts with 10 unique continuations each, (2) including contemporary instruction-tuned models, (3) evaluating five decoding strategies systematically, (4) explicitly decomposing uncertainty into aleatoric and epistemic components, and (5) providing quantitative calibration metrics based on credal set overlap coefficients.

3 Methodology

3.1 Dataset Construction and Human Baselines

The WRITINGPROMPTS dataset (Fan et al., 2018) provides naturalistic creative writing data from Reddit's r/WritingPrompts community. We implement rigorous selection criteria to ensure data quality:

1. **Uniqueness verification**: We compute MD5 hashes for all stories and select only prompts with exactly 10 unique continuations, eliminating duplicates that could bias diversity measurements.

2. **Length filtering**: We retain prompts between 20-500 characters and stories between 52-987 tokens (mean: 312.4, std: 148.2), ensuring sufficient content for meaningful analysis while avoiding outliers.

3. **Quality scoring**: We prioritize prompts by the diversity of story lengths they elicit (measured by standard deviation), selecting those that inspire varied responses rather than formulaic continuations.

This process yields 500 high-quality prompts with 5,000 unique human stories, providing a robust baseline for calibration analysis.

3.2 Model Selection and Configuration

Our model selection explores the calibration landscape across different architectures and training paradigms:

Base models: GPT2-XL (1.5B) (Radford et al., 2019) serves as a canonical autoregressive baseline, while Gemma-2B (Gemma-Team et al., 2024) represents modern architectural improvements at comparable scale. These models, trained on diverse internet text without explicit instruction following, potentially preserve more natural variation patterns.

Instruction-tuned models: Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and Llama-3.1-8B-Instruct (Dubey et al., 2024) represent strong open-source models with instruction tuning and alignment. While offering improved controllability, we investigate whether alignment training constrains creative exploration[2].

3.3 Decoding Strategy Design

We systematically evaluate five decoding strategies that control output diversity through different mechanisms:

- **Temperature scaling** ($\tau \in \{0.7, 1.2\}$): Directly modulates the entropy of the output distribution (Ackley et al., 1985)
- **Nucleus sampling** ($p = 0.9$): Dynamically adjusts the token consideration set based on cumulative probability (Holtzman et al., 2020)
- **Top-k sampling** ($k = 40$): Maintains a fixed-size token pool (Fan et al., 2018)
- **Typical sampling** ($p = 0.95$): Selects tokens based on expected information content (Meister et al., 2023)

Each configuration generates 10 independent samples with different random seeds, totaling 100,000 model-generated stories for analysis.

3.4 Diversity Metrics

Our metric suite captures multiple dimensions of textual variation through pairwise distance-based measures following Giulianelli et al. (2023):

3.4.1 Semantic Diversity

We compute semantic diversity as the mean pairwise cosine distance between Sentence-BERT embeddings (Reimers and Gurevych, 2019):

$$D_{\text{sem}}(\mathcal{S}) = \frac{2}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{i<j} (1 - \cos(e_i, e_j)),$$

where e_i represents the embedding of story i using the all-MiniLM-L6-v2 model (Wang et al., 2020). This captures high-level narrative and thematic variation.

3.4.2 Lexical Diversity

We measure lexical diversity using Jaccard distance between word unigrams:

$$D_{\text{lex}}(\mathcal{S}) = \frac{2}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{i<j} \left(1 - \frac{|V_i \cap V_j|}{|V_i \cup V_j|}\right),$$

where V_i represents the vocabulary set of story i. This captures variation in word choice and vocabulary richness.

3.4.3 Syntactic Diversity

We measure syntactic variation through Jaccard distance of part-of-speech (POS) bigrams:

$$D_{\text{syn}}(\mathcal{S}) = \frac{2}{|\mathcal{S}|(|\mathcal{S}|-1)} \sum_{i<j} \left(1 - \frac{|P_i \cap P_j|}{|P_i \cup P_j|}\right),$$

where P_i represents the set of POS bigrams extracted using spaCy's en_core_web_sm model (Honnibal and Montani, 2017). This captures stylistic and structural variation in the generated text.

3.5 Theoretical Framework: Credal Sets

Our methodology rests on the principle that uncertainty in creative text generation should be understood relative to the natural variation exhibited by humans facing the same creative task. We formalize this through a credal set framework that captures uncertainty as a set of plausible probability distributions rather than a single distribution.

For a given prompt p, let $\mathcal{H}_p = \{h_1, ..., h_{10}\}$ denote the set of human continuations and $\mathcal{M}_{p,m,d} = \{s_1, ..., s_{10}\}$ denote the set of model continuations for model m using decoding strategy d. For any set of continuations $\mathcal{S}$, we compute a diversity vector $\mathbf{v}_p = [D_{\text{sem}}(\mathcal{S}), D_{\text{lex}}(\mathcal{S}), D_{\text{syn}}(\mathcal{S})]$.

The human credal set for a collection of prompts $\mathcal{P}$ is then defined as:

$$\mathcal{C}_H = \text{ConvexHull}\left(\{\mathbf{v}_p^H : p \in \mathcal{P}\}\right),$$

where each $\mathbf{v}_p^H$ is the diversity vector computed from human continuations for prompt p.
Similarly, the model credal set for a specific configuration (m, d) is:

$$\mathcal{C}_{M,d} = \text{ConvexHull}\left(\{\mathbf{v}_p^{M,d} : p \in \mathcal{P}\}\right).$$

[2]All models use appropriate prompt formatting with careful post-processing to remove prompt artifacts from generations, ensuring fair comparison across architectures.

The convex hull is computed using the Quickhull algorithm (Barber et al., 1996) after standardizing the diversity vectors. This representation enables geometric analysis of uncertainty relationships through set operations and distance metrics.

3.6 Calibration Analysis

Calibration quality is assessed through the overlap coefficient of credal sets:

$$\text{Calibration}(M, d) = \text{Overlap}(\mathcal{C}_H, \mathcal{C}_{M,d}),$$

where overlap is computed using nearest-neighbor distances between credal set vertices. The overlap coefficient is calculated as:

$$\text{Overlap} = \frac{1}{2}\left(\frac{|\{v \in V_M : d(v, V_H) < \theta\}|}{|V_M|} + \frac{|\{v \in V_H : d(v, V_M) < \theta\}|}{|V_H|}\right),$$

where V_M and V_H are the vertex sets of the model and human credal sets respectively, $d(v, V)$ is the minimum distance from point v to set V, and θ is an adaptive threshold set to half the mean variance scale. Values range from 0 (disjoint sets) to 1 (perfect overlap).

3.7 Uncertainty Decomposition

To decompose uncertainty, we leverage variation across decoding strategies. For a given model M, we collect all diversity vectors across different strategies and compute:

Strategy centroids :
$\mathbf{c}_d = \text{mean}(\{\mathbf{v}_p^{M,d} : p \in \mathcal{P}\})$ for each strategy d

Between-strategy variance :
$\sigma^2_{\text{between}} = \text{Var}(\{\mathbf{c}_d : d \in \mathcal{D}\})$

Within-strategy variance :
$\sigma^2_{\text{within}} = \text{mean}_d[\text{Var}(\{\mathbf{v}_p^{M,d} : p \in \mathcal{P}\})]$
The epistemic ratio is then:

$$\text{Epistemic}_M = \frac{\sigma^2_{\text{between}}}{\sigma^2_{\text{between}} + \sigma^2_{\text{within}}}.$$

This quantifies the proportion of uncertainty arising from configuration choices rather than inherent task ambiguity.

4 Results

4.1 Human Variation as Calibration Baseline

Analysis of 5,000 human-written stories reveals structured patterns of creative variation that establish our calibration baseline (Table 1).

Diversity Type	Mean	Std Dev
Semantic	0.645	0.066
Lexical	0.328	0.035
Syntactic	0.315	0.044

Table 1: Human diversity baselines across 500 prompts with 10 unique continuations each, computed using pairwise distance metrics.

The distribution of semantic diversity across prompts shows moderate variation with most prompts (62%) eliciting medium diversity (0.6-0.7), while 19% show high diversity (>0.7) and 19% show low diversity (<0.6). This suggests fundamental differences in prompt interpretability that models must capture.

4.2 Credal Set Geometry and Calibration

The human credal set $\mathcal{C}_H$ occupies a volume of 2.25 in the PCA-transformed diversity space, serving as the baseline for model comparison. Analysis reveals a clear distinction between model types: base models (GPT2-XL, Gemma-2B) produce compact credal sets with mean volume 1.10 ± 0.56, representing 48.9% of the human volume. In contrast, instruction-tuned models (Mistral-7B-Instruct, Llama-3.1-8B-Instruct) generate significantly larger credal sets with mean volume 3.87 ± 1.78, corresponding to 172.1% of the human baseline. The difference in credal set volumes between base and instruction-tuned models is statistically significant (Mann-Whitney U = 2.00, $p < 0.001$).

Principal component analysis of the diversity vectors reveals strong coupling between diversity dimensions. PC1 explains 85.8% of variance with nearly equal positive loadings across semantic (0.569), lexical (0.565), and syntactic (0.597) dimensions, indicating that these diversity types covary systematically. The dominance of PC1 suggests that models exhibiting high diversity in one dimension tend to show proportionally high diversity in all dimensions, as illustrated in Figure 3.

The expanded credal sets of instruction-tuned models indicate broader exploration of the diversity space compared to base models. However, larger volume does not directly correspond to better calibration, as shown in Table 2 and Figure 6, in the Appendix. This suggests that alignment with human diversity patterns depends more on the location and shape of the credal set than its absolute size.

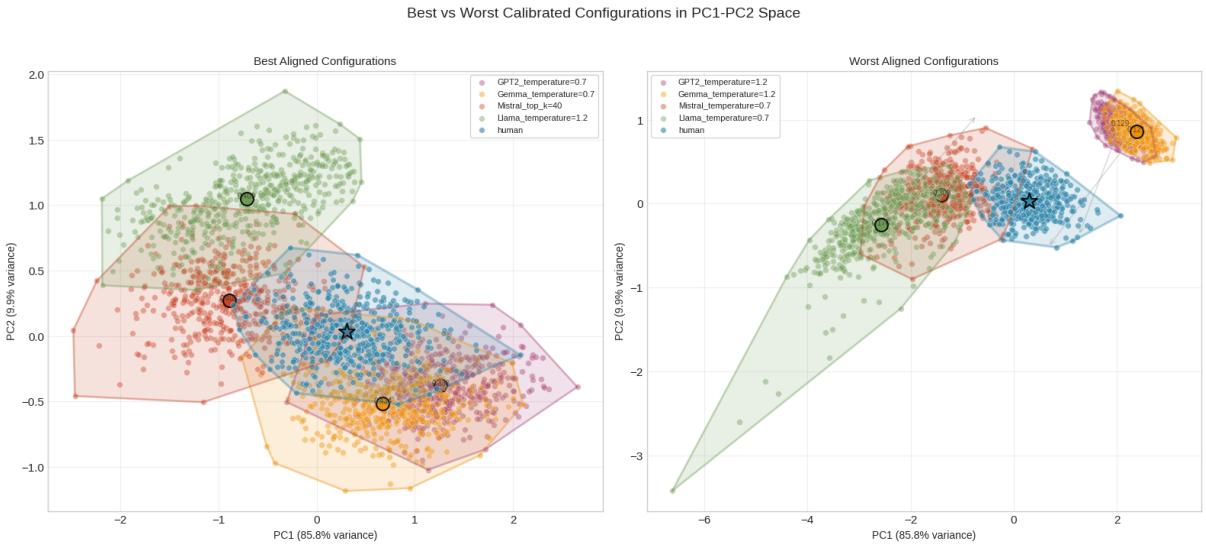


Figure 2: Credal sets visualization in principal component space. Human creative variation (blue) and model-generated variation exhibit different geometric patterns and a high sensitivity with respect to the decoding configuration. Points represent diversity vectors from individual prompts; convex hulls indicate credal set boundaries. PC1 explains 85.8% of the variance, suggesting a strong correlation between diversity dimensions. Best (left) and worst aligned configurations (right), measured by the overlap of the credal sets, are presented.

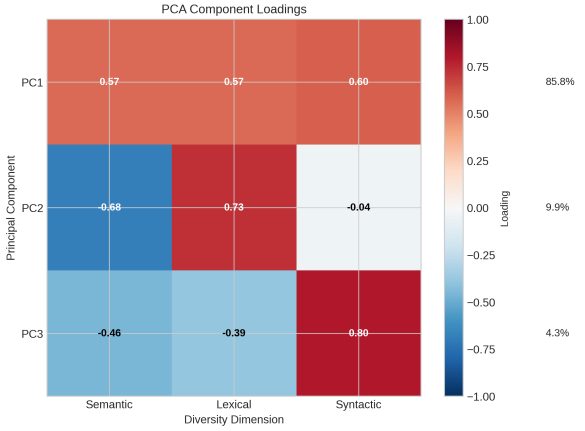


Figure 3: Overview of PCA loadings, displaying a balanced contribution of semantic, lexical, and syntactic patterns on the first principal component, which explains a large proportion of the total variance.

4.3 Distributional Analysis via Wasserstein Distance

Complementary analysis using Wasserstein distance at the prompt level corroborates the credal set findings. The Wasserstein distance measures the average distributional difference between human and model-generated diversity patterns across all prompts. The best configuration by Wasserstein distance (Gemma-2B with temperature=0.7, distance=0.065) coincides with the best-calibrated credal set, providing independent validation of the geometric approach. The moderate negative correlation between Wasserstein distance and calibration score ($\rho = -0.411$, $p = 0.072$) indicates that while both methods capture aspects of human-model alignment, they emphasize different characteristics: Wasserstein distance weights all prompts equally in measuring average distributional differences, while credal sets capture the geometric envelope of diversity behaviors. A visualization of this comparison is presented in Figure 7.

4.4 Model Calibration Patterns

Calibration analysis reveals that no model effectively reproduces human variation patterns, with best overlap coefficients reaching only 0.434 (Table 2). Figure 5 illustrates these key findings:

Model architecture effects: Gemma-2B achieves the best single configuration calibration (0.434 with temperature 0.7), though Mistral-7B-Instruct shows the highest average calibration across all strategies (0.371). Statistical analysis reveals weak positive correlation between model size and calibration (Spearman's $\rho = 0.400$, $p = 0.600$), suggesting model scale has limited influence on calibration quality. Further, base models (mean calibration: 0.274 ± 0.095) show no significant difference from instruction-tuned models (mean: 0.305 ± 0.093) in calibration quality ($t = -0.712$, $p = 0.486$, Cohen's $d = -0.336$). Despite similar calibration scores, base and

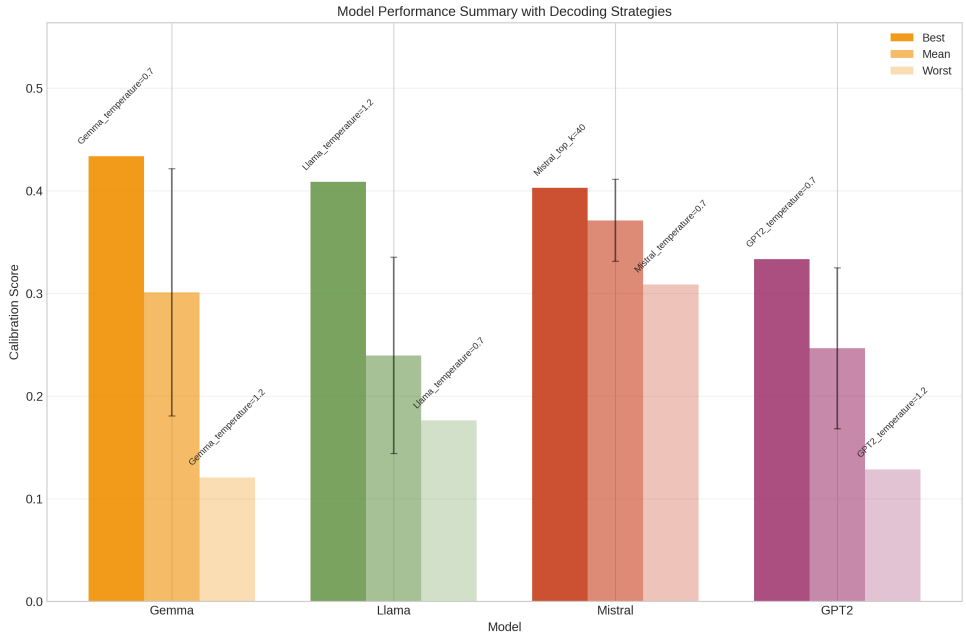


Figure 4: Overview of model performance across varying decoding strategies. Here, performance is to be understood in terms of calibration scores with respect to human credal sets. Top-k sampling provides the highest mean calibration, while Gemma-2B with temperature set to 0.7 achieves the best overall calibration.

Model	Strategy	Value	Overall Cal.	Overlap	Centroid Dist.	Volume Ratio
Gemma-2B	temperature	0.7	**0.434**	0.033	1.096	0.924
Llama-3.1-8B-Instruct	temperature	1.2	0.409	0.032	1.488	0.918
Mistral-7B-Instruct	top_k	40	0.403	0.000	1.502	1.060
Mistral-7B-Instruct	temperature	1.2	0.399	0.000	0.956	0.820
Mistral-7B-Instruct	top_p	0.9	0.391	0.000	1.721	1.070
Gemma-2B	top_k	40	0.386	0.033	1.189	0.785
Mistral-7B-Instruct	typical	0.95	0.354	0.000	1.604	1.258
GPT2-XL	temperature	0.7	0.333	0.000	1.386	0.692
Mistral-7B-Instruct	temperature	0.7	0.309	0.000	1.945	1.450
GPT2-XL	top_k	40	0.300	0.033	1.244	0.509

Table 2: Calibration metrics for top configurations. Higher values indicate better alignment with human variation. Gemma-2B with temperature 0.7 achieves best overall calibration (0.434).

instruction-tuned models differ significantly in their exploration of the diversity space, with instruction-tuned models producing credal sets 3.5× larger on average ($p < 0.001$).

Strategy effectiveness: Top-k sampling achieves the highest mean performance (0.323 ± 0.092), followed by temperature scaling (0.289 ± 0.129). Analysis of variance across all 20 model-strategy combinations reveals no significant main effect of strategy type ($F(3, 16) = 0.200$, $p = 0.895$), suggesting that strategy effectiveness depends on the specific model architecture.

4.5 Uncertainty Decomposition

Decomposition analysis reveals the relative contributions of epistemic and aleatoric uncertainty (Table 3). Base models (GPT2-XL, Gemma-2B) exhibit higher epistemic ratios (64.9-72.0%), indicating that decoding strategy choice contributes more than half of their total uncertainty. Instruction-tuned models show lower epistemic ratios (39.4-50.5%), suggesting more stable behavior across decoding strategies but potentially at the cost of reduced overall variation.

The within-strategy variance (aleatoric component) remains substantial across all models (0.091-0.224), confirming that models can generate diverse outputs for individual prompts. However, the between-strategy variance (epistemic component) highlights that generation configuration remains a critical factor in uncertainty quantification, particularly for base models.

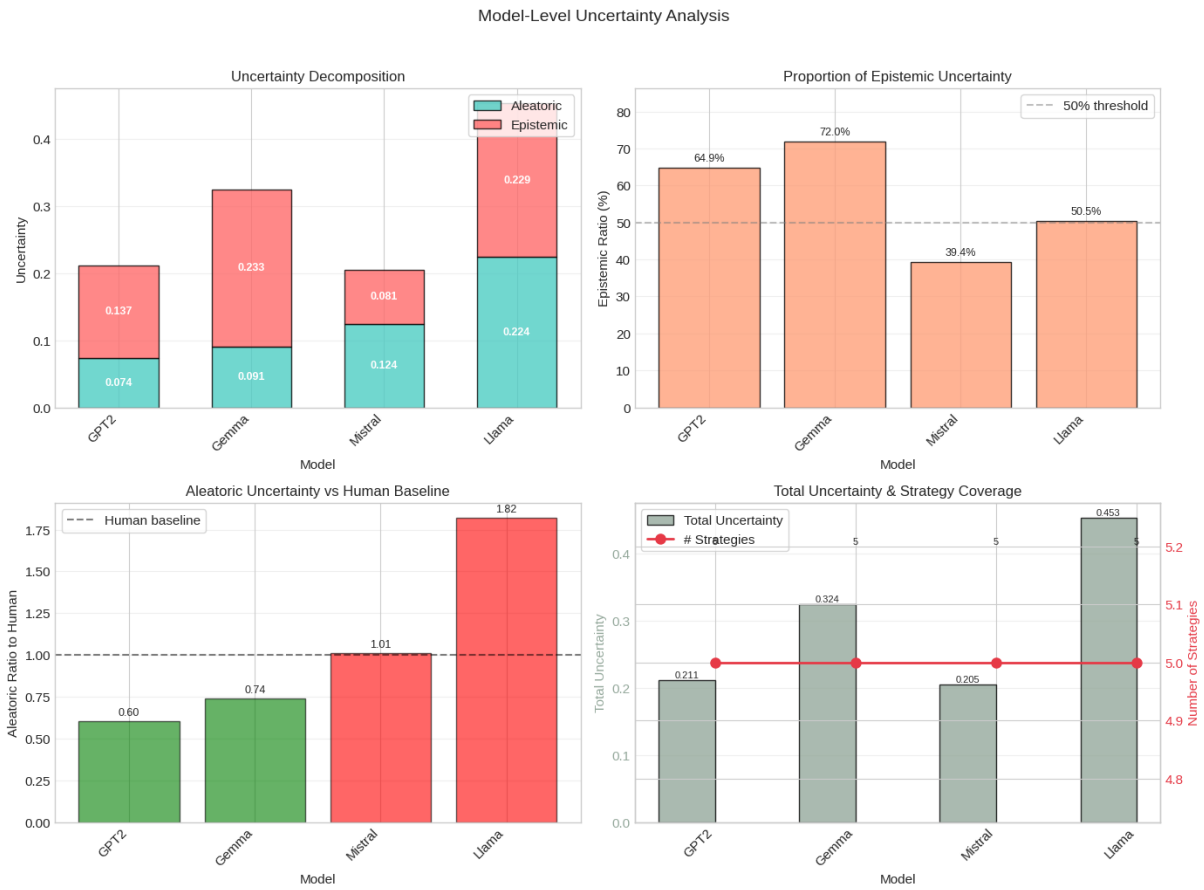


Figure 5: Uncertainty analysis and model performance overview. **Top Left:** Uncertainty decomposition showing epistemic and aleatoric components. **Top Right:** Epistemic ratio by model. **Bottom Left:** Aleatoric uncertainty vs. human baseline. **Bottom Right:** Estimated total uncertainty per model, measured over five decoding strategies.

Model	Epistemic	Aleatoric	Total	Ratio
Gemma-2B	0.233	0.091	0.324	72.0%
GPT2-XL	0.137	0.074	0.211	64.9%
Llama-3.1-8B-Instruct	0.229	0.224	0.453	50.5%
Mistral-7B-Instruct	0.081	0.124	0.205	39.4%

Table 3: Uncertainty decomposition showing absolute values and epistemic ratios. All models show substantial epistemic uncertainty, indicating sensitivity to decoding strategies.

5 Discussion

5.1 Theoretical Implications

Our credal set framework advances uncertainty quantification theory for generative models in several ways. By treating uncertainty as inherently distributional and prompt-dependent, we move beyond scalar measures that collapse rich variation patterns. The geometric interpretation through credal set operations provides intuitive understanding of miscalibration modes: models can fail through incorrect positioning (wrong variation type), volume (over/under-exploration), or shape (wrong dimensions).

The finding that best calibration reaches only 0.434 reveals fundamental gaps in how current models capture human creative variation. The notably low overlap coefficients (maximum 0.033) indicate that model and human credal sets occupy largely disjoint regions in diversity space, suggesting that current models operate in fundamentally different creative regimes than humans. The high PC1 dominance (85.8% variance) with syntactic diversity as the primary driver indicates that current models treat diversity dimensions as tightly coupled, potentially missing independent variation patterns that humans explore.

5.2 Implications for Model Development

The weak positive correlation between model scale and calibration quality ($\rho = 0.400$, $p = 0.600$) suggests that while larger models may have slight advantages, scale alone is not a determining factor for calibration quality. Our results indicate that

training objectives and data distributions likely matter more than parameter count for uncertainty calibration. The lack of significant difference between base and instruction-tuned models ($t = -0.712$, $p = 0.486$, Cohen's $d = -0.336$) with a small effect size indicates that alignment training has minimal impact on creative diversity calibration. Interestingly, instruction-tuned models showed slightly higher mean calibration (0.305 vs 0.274), though this difference was not statistically significant. The substantial epistemic uncertainty across all models (39.4-72.0%) highlights that decoding strategy choice remains a dominant source of variation. Notably, Gemma-2B shows the highest epistemic ratio (72.0%), suggesting extreme sensitivity to decoding configuration despite achieving the best single-configuration performance. This paradox suggests that optimal calibration may require careful strategy selection rather than robust performance across strategies.

5.3 Practical Deployment Considerations

For practitioners deploying language models in creative applications, our findings offer concrete guidance:

- **Model selection**: Mistral-7B-Instruct offers the most consistent performance across strategies (mean calibration: 0.371), while Gemma-2B with temperature 0.7 provides the best single configuration (0.434).
- **Strategy optimization**: Top-k sampling provides the highest mean calibration (0.323), though all models show substantial epistemic uncertainty (39-72%), making careful tuning essential.
- **Baseline expectations**: With maximum calibration at 0.434 and overlap coefficients of at most 0.033, expect substantial divergence from human creative patterns.
- **Multi-strategy ensemble**: Given high epistemic ratios, combining outputs from multiple decoding strategies is crucial for approximating human creative diversity.
- **Model-specific tuning**: In terms of calibration, base models (especially Gemma-2B at 72% epistemic) require more careful strategy selection than instruction-tuned models like Mistral-7B-Instruct (39.4% epistemic).
- **Calibration vs. quality**: Calibration along semantic, lexical, and syntactic dimensions does not necessarily indicate qualitative alignment between model-generated and human-produced text. Future work will investigate this relationship comprehensively using both human evaluations and LLM-as-a-Judge scores.
- **Generalizatbility**: Our findings are specific to *storytelling*—an open-ended task prioritizing communicative goals such as creativity, fluency, and engagement. To extend this analysis to other Natural Language Generation (NLG) research areas, we suggest task-specific calibration analyses, as different tasks involve distinct communicative objectives and varying degrees of human production variability that serve as calibration benchmarks.

6 Conclusion

This work establishes credal sets as a rigorous framework for uncertainty quantification in open-ended text generation, enabling principled geometric comparison between human and model variation patterns. Through comprehensive analysis of 100,000 generated stories calibrated against 5,000 human-written stories, we demonstrate substantial gaps in how current language models capture human creative variation, with the best calibration reaching only 0.434 (Gemma-2B with temperature 0.7) and overlap coefficients at most 0.033.

Our decomposition reveals that epistemic uncertainty from decoding strategy choice contributes 39.4-72.0% of total uncertainty across models, with base models showing higher sensitivity to configuration choices. The weak correlation between model scale and calibration ($\rho = 0.400$, $p = 0.600$) and lack of significant difference between base and instruction-tuned models ($p = 0.486$) challenge common assumptions about model development priorities. The credal set framework provides actionable insights for deploying language models in creative contexts and establishes quantitative benchmarks for evaluating progress toward human-AI creative alignment. As language models increasingly engage in open-ended generation tasks, our findings highlight the critical importance of decoding strategy selection and the need for architectural or training innovations specifically targeting uncertainty calibration.

Limitations

Several limitations warrant consideration:

- Our analysis uses convex hulls which may not capture non-convex uncertainty regions or multimodal distributions within credal sets.
- The 500-prompt sample from a single domain may not generalize to other creative writing contexts or languages.
- Decoding strategies evaluated prioritize high-probability tokens, whereas humans often select surprising, low-probability tokens for creative effect—a mismatch that may constrain achievable calibration.
- Human baselines include natural skill variation beyond pure creativity, potentially inflating aleatoric uncertainty estimates.
- Computational constraints limited us to 10 samples per configuration; larger samples might reveal finer-grained patterns.
- Statistical variance alone cannot distinguish creative quality from random variation—validating the relationship between our metrics and perceived creative quality is essential future work.

Ethics Statement

We affirm that our research adheres to the ACL Ethics Policy. This work uses publicly available datasets and involves no human subjects or personally identifiable information. We acknowledge potential biases in the Reddit-sourced dataset and encourage diverse dataset development. Our framework could help identify and mitigate generation biases by comparing model variation patterns across different demographic or cultural contexts. All code and data are released to enable reproducible research and further investigation of these important issues.

Acknowledgments

Esteban Garces is sponsored by the Munich Center for Machine Learning (MCML) and the LMU Mentoring Program. Julian Rodemann acknowledges support by the Bavarian Institute for Digital Transformation (bidt) within the Bavarian Academy of Sciences (BAS) and the LMU Mentoring Program.

References

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. 2024. Rethinking uncertainty estimation in natural language generation. *Preprint*, arXiv:2412.15176.

C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. 1996. The quickhull algorithm for convex hulls. *ACM Trans. Math. Softw.*, 22(4):469–483.

Yuanhao Ding, Esteban Garces Arias, Meimingwei Li, Julian Rodemann, Matthias Aßenmacher, Danlu Chen, Gaojuan Fan, Christian Heumann, and Chongsheng Zhang. 2025. GUARD: Glocal uncertainty-aware robust decoding for effective and efficient open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China. Association for Computational Linguistics. Equal contribution: Yuanhao Ding, Esteban Garces Arias, Meimingwei Li.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 516 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *International Conference on Machine Learning*, pages 1050–1059. PMLR.

Esteban Garces Arias, Hannah Blocher, Julian Rodemann, Matthias Aßenmacher, and Christoph Jansen. 2025c. Statistical multicriteria evaluation of LLM-generated text. In *Proceedings of the 18th International Natural Language Generation Conference (INLG 2025)*, Hanoi, Vietnam. Preprint available at arXiv:2506.18082.

Esteban Garces Arias, Hannah Blocher, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. 2025b. Towards better open-ended text generation: A multicriteria evaluation framework. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 631–654, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Esteban Garces Arias, Julian Rodemann, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher.

2024. Adaptive contrastive search: Uncertainty-guided decoding for open-ended text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15060–15080, Miami, Florida, USA. Association for Computational Linguistics.

Gemma-Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. 2023. What comes next? evaluating uncertainty in neural text generators against human production variability. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14349–14371, Singapore. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Matthew Honnibal and Ines Montani. 2017. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear.

Bairu Hou, Yujian Liu, Kaizhi Qian, Jacob Andreas, Shiyu Chang, and Yang Zhang. 2024. Decomposing uncertainty for large language models through input clarification ensembling. *Preprint*, arXiv:2311.08718.

D.P. Huttenlocher, G.A. Klanderman, and W.J. Rucklidge. 1993. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(9):850–863.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30.

Isaac Levi. 1980. *The Enterprise of Knowledge: An Essay on Knowledge, Credal Probability, and Chance*. MIT Press.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Preprint*, arXiv:2205.14334.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, Guangji Bai, Liang Zhao, and Haifeng Chen. 2024. Uncertainty quantification for in-context learning of large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3357–3370, Mexico City, Mexico. Association for Computational Linguistics.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *Preprint*, arXiv:2404.15993.

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. 2023. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121.

Yaniv Ovadia, Emily Fertig, Jie Jessie Ren, Zachary Nado, D. Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Neural Information Processing Systems*.

Silviu Pitis, Michael R. Zhang, Andrew Wang, and Jimmy Ba. 2023. Boosted prompt ensembles for large language models. *Preprint*, arXiv:2304.05970.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI*. Accessed: 2024-11-15.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Dennis Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. 2024. Calibrating large language models using their generations only. *Preprint*, arXiv:2403.05973.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep

self-attention distillation for task-agnostic compression of pre-trained transformers. *Preprint*, arXiv:2002.10957.

Lisa Wimmer, Yusuf Sale, Paul Hofman, Bernd Bischl, and Eyke Hüllermeier. 2023. Quantifying aleatoric and epistemic uncertainty in machine learning: Are conditional entropy and mutual information appropriate measures? In *Uncertainty in artificial intelligence*, pages 2282–2292. PMLR.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *Preprint*, arXiv:2306.13063.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. 2024. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*.

Marco Zaffalon and Enrico Fagiuoli. 2003. Tree-based credal networks for classification. *Reliable computing*, 9(6):487–509.

Tunyu Zhang, Haizhou Shi, Yibin Wang, Hengyi Wang, Xiaoxiao He, Zhuowei Li, Haoxian Chen, Ligong Han, Kai Xu, Huan Zhang, and 1 others. 2025. Token-level uncertainty estimation for large language model reasoning. *arXiv preprint arXiv:2505.11737*.

A Extended Results

A.1 Credal Volume Analysis

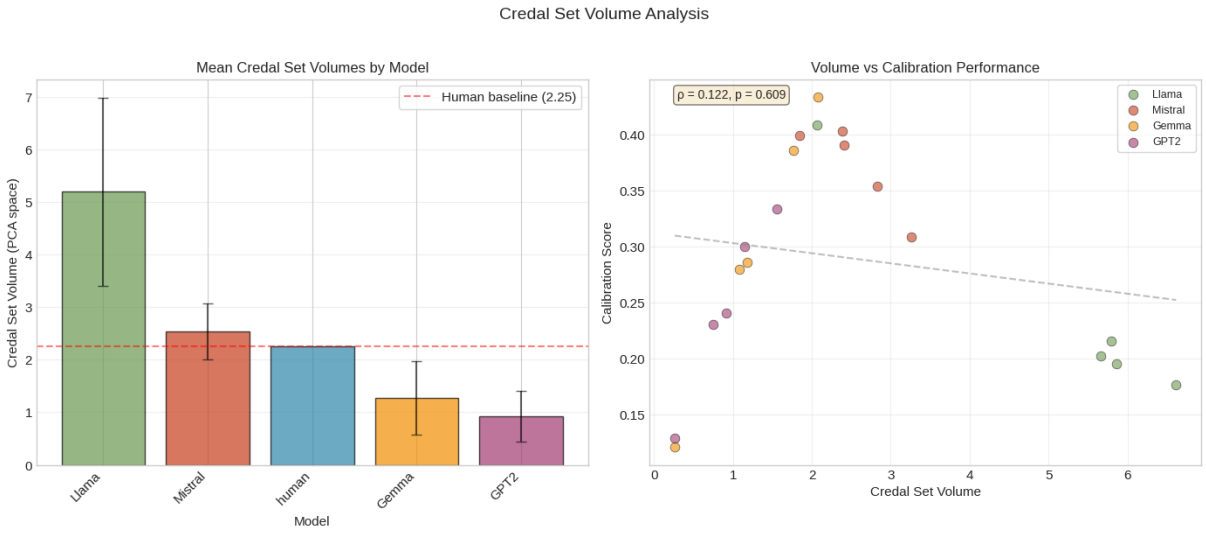


Figure 6: Analysis of credal set volumes for human and language models. **Left:** Mean credal set volumes by model (in PCA space). **Right:** Relationship between calibration score and credal set volume. A positive trend for base models (GPT2-XL and Gemma) is observed, while a negative trend is observed for instruct models (Mistral and Llama).

A.2 Wasserstein Distance Analysis

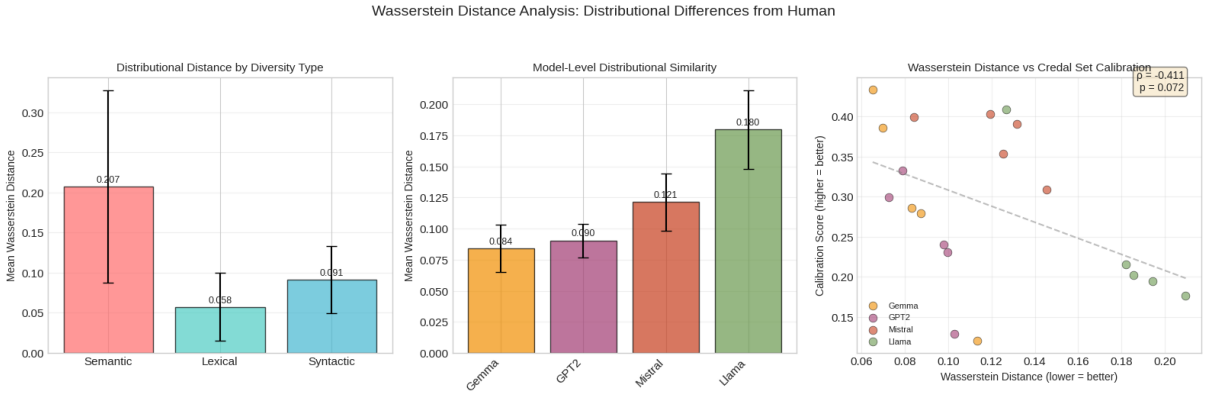


Figure 7: Distributional differences between model and human productions measured by Wasserstein distances. **Left:** Mean Wasserstein distances across semantic, lexical, and syntactic dimensions. Semantic features show the largest divergence from human distributions, followed by syntactic and lexical features. **Middle:** Model-specific distributional similarity. Gemma-2B achieves the lowest Wasserstein distances (closest to human distributions), while Llama models exhibit the highest distances. **Right:** Inverse relationship between calibration scores and Wasserstein distances (moderate negative correlation). Gemma-2B and Mistral appear in the upper-left section (high calibration, low distance), while Llama appears in the lower-right quadrant (low calibration, high distance).

A.3 Complete Calibration Results

Table 4 presents calibration coefficients for all 20 model-strategy combinations evaluated in our experiments.

Model	Strategy	Calibration
Gemma-2B	temperature_0.7	0.434
Llama-3.1-8B-Instruct	temperature_1.2	0.409
Mistral-7B-Instruct	top_k_40	0.403
Mistral-7B-Instruct	temperature_1.2	0.399
Mistral-7B-Instruct	top_p_0.9	0.391
Gemma-2B	top_k_40	0.386
Mistral-7B-Instruct	typical_0.95	0.354
GPT2-XL	temperature_0.7	0.333
Mistral-7B-Instruct	temperature_0.7	0.309
GPT2-XL	top_k_40	0.300
Gemma-2B	top_p_0.9	0.286
Gemma-2B	typical_0.95	0.279
GPT2-XL	top_p_0.9	0.240
GPT2-XL	typical_0.95	0.231
Llama-3.1-8B-Instruct	typical_0.95	0.215
Gemma-2B	temperature_1.2	0.212
Llama-3.1-8B-Instruct	top_k_40	0.199
Llama-3.1-8B-Instruct	temperature_0.7	0.196
GPT2-XL	temperature_1.2	0.188
Llama-3.1-8B-Instruct	top_p_0.9	0.175

Table 4: Complete calibration results for all model-strategy combinations, sorted by calibration coefficient.

A.4 Statistical Tests

We conducted comprehensive statistical analyses to validate our findings:

- **Model size vs calibration**: Spearman's $\rho = 0.400$ ($p = 0.600$), indicating weak positive correlation without statistical significance.
- **Base vs instruction-tuned**: Two-sample t-test: $t = -0.712$ ($p = 0.486$), no significant difference. Cohen's $d = -0.336$ (small effect size).
- **Strategy comparison**: ANOVA across strategies: $F(3, 16) = 0.200$ ($p = 0.895$), no significant differences between strategies.
- **Best performing model**: Mistral-7B-Instruct showed highest mean calibration (0.371) across all strategies.
- **Best performing strategy**: Top-k sampling achieved highest mean calibration (0.323 ± 0.092) across all models.

B Implementation Details

B.1 Computational Resources

All experiments were conducted on Google Colab with the following specifications:

- GPU: NVIDIA A100 (40GB) or V100 (16GB)
- RAM: 25-50GB depending on instance
- Storage: Google Drive for persistent storage
- Total compute time: Approximately 8 hours for generation, 1 hour for analysis

B.2 Model Configurations

Models were loaded with the following optimizations:

- 4-bit quantization for models >3B parameters using BitsAndBytes
- Flash Attention 2 where supported
- Batch sizes optimized per model (8-25 samples)
- Automatic mixed precision (AMP) with fp16

B.3 Diversity Metric Computation

Semantic embeddings were computed using Sentence-BERT (all-MiniLM-L6-v2) with the following parameters:

- Maximum sequence length: 512 tokens
- Batch size: 64 for encoding
- Pooling: Mean pooling over token embeddings

POS tagging was performed using spaCy's en_core_web_sm model with a maximum text length of 5000 characters for efficiency.

Certain but not Probable? Differentiating Certainty from Probability in LLM Token Outputs for Probabilistic Scenarios

Autumn Toney-Wails
SciTech Strategies, Inc.
Georgetown University
autumn@mapofscience.com

Ryan Wails
Georgetown University
rsw66@georgetown.edu

Abstract

Reliable uncertainty quantification (UQ) is essential for ensuring trustworthy downstream use of large language models, especially when they are deployed in decision-support and other knowledge-intensive applications. Model certainty can be estimated from token logits, with derived probability and entropy values offering insight into performance on the prompt task. However, this approach may be inadequate for probabilistic scenarios, where the probabilities of token outputs are expected to align with the theoretical probabilities of the possible outcomes. We investigate the relationship between token certainty and alignment with theoretical probability distributions in well-defined probabilistic scenarios. Using GPT-4.1 and DeepSeek-Chat, we evaluate model responses to ten prompts involving probability (e.g., roll a six-sided die), both with and without explicit probability cues in the prompt (e.g., roll a *fair* six-sided die). We measure two dimensions: (1) response validity with respect to scenario constraints, and (2) alignment between token-level output probabilities and theoretical probabilities. Our results indicate that, while both models achieve perfect in-domain response accuracy across all prompt scenarios, their token-level probability and entropy values consistently diverge from the corresponding theoretical distributions.

1 Introduction

As large language models (LLMs) are increasingly integrated into decision-support and knowledge-intensive applications, uncertainty quantification (UQ) is essential to ensure reliable downstream use (Xiong et al., 2024; Vashurin et al., 2025). Prior work has focused on leveraging the token logits—numerical representations encoding the model's output probabilities—for UQ methods applied to natural language generation tasks (Malinin and Gales, 2021; Kuhn et al., 2023; Gupta et al., 2024; Lin et al., 2024; Duan et al., 2024; Fadeeva et al., 2024; Lovering et al., 2025). Token logits can be transformed into probabilities using activation functions (e.g., softmax or sigmoid), enabling entropy computation over the token distribution. These probability and entropy values are often used to quantify model certainty, providing token-level or response-level confidence scores to users.

For prompts involving randomness, risk, or chance, traditional UQ alone may be insufficient to capture the true confidence a user should have in implementing a model.

> *Key insight:* Specifically, for prompts involving probabilistic scenarios with inherent aleatoric uncertainty (e.g., "flip a fair coin"), a model's behavior is trustworthy only when its distribution over possible outputs matches the intended theoretical distribution (which may be only implicitly specified).

Hence, in these probabilistic scenarios, we argue there are two considerations for uncertainty quantification that are particularly important:

1. Whether a response is a valid output under the specified scenario constraints (e.g., if the prompt is "roll a six-sided dice", the output response should not be "7").

2. Whether the response probability aligns with the underlying theoretical distribution.

These considerations are somewhat in contrast to traditional UQ settings, where model accuracy typically corresponds to low uncertainty when a model is well-calibrated.

In this work, we explore the relationship between token certainty and theoretical probability in well-defined probabilistic scenarios. We prompt GPT-4.1 and DeepSeek with 10 well-known scenarios (e.g., roll a six-sided die or pick a card from a deck

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 51–60
November 9, 2025 ©2025 Association for Computational Linguistics

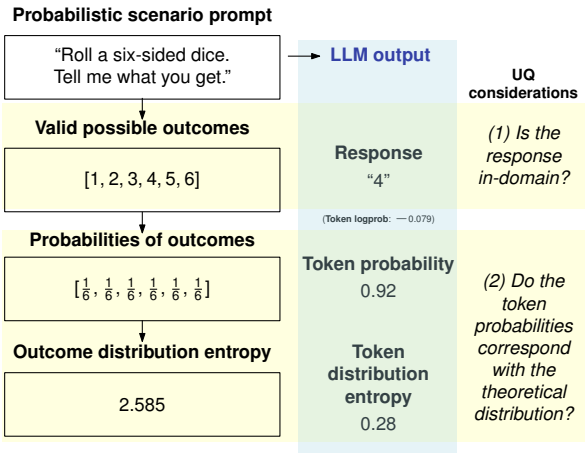


Figure 1: Probabilistic scenario prompting and response evaluation design.

of 52 cards). Our selected scenarios allow us to measure the models' response certainty (was the response valid?) and to compare the token output probability and entropy of possible output tokens to theoretical distributions (does the response probability align to real-world distributions?). We provide an example of our experimental design framework in Figure 1. We experiment with a second series of prompting, in which we include language that reflects the distribution the model should use (e.g., roll a *fair* six-sided die, pick a card from a deck of 52 cards *uniformly at random*).

Our findings suggest that although GPT-4.1 and DeepSeek exhibit appropriate contextual understanding and high response certainty, their token-level output probabilities do not reliably represent true probabilities in scenarios that require statistical reasoning or random sampling. Specifically, we find that they respond with valid outputs with 100% accuracy (i.e., they understand the constraints of the prompted scenario) but their probability and entropy values never align with the corresponding theoretical values. Based on our findings, we explore three research questions for further discussion:

RQ1 Can LLMs accurately reason about the theoretical probabilities of our prompt scenarios?

RQ2 Are LLMs appropriate and reliable tools for probability-oriented tasks where usability depends on alignment with theoretical distributions?

RQ3 How can uncertainty quantification methods be adapted to jointly evaluate response validity and distribution alignment for probability-driven tasks?

2 Background and Related Work

2.1 Uncertainty Quantification for LLMs

Uncertainty quantification (UQ) measures the confidence associated with model output, providing information to guide user decisions on model deployment, refinement, or rejection. UQ for LLMs spans a wide range of methodologies for varying granularity levels and certainty dimensions (Liu et al., 2025; Shorinwa et al., 2025). Depending on the task, UQ can be computed at the token level (a single word), the span level (a contiguous sequence of tokens representing the response segment of interest), or the response level (the entire generated output). To organize the wide range of UQ methods, Liu et al. (2025) and Shorinwa et al. (2025) present surveys that synthesize the related literature and characterize the state of the field.

Liu et al. (2025) proposed a taxonomy to organize sources of uncertainty beyond the general aleatoric (uncertainty stemming from the randomness or variability in the data) and epistemic (uncertainty stemming from lack of knowledge) categorization. The authors provide four dimensions of uncertainty for finer-grained analysis: (1) input uncertainty, (2) reasoning uncertainty, (3) parameter uncertainty, and (4) prediction uncertainty. This taxonomy supports their survey design, which focuses on the relationships between model scale, open-ended generation, and uncertainty dynamics. The authors further outline research directions, emphasizing the need for enhanced UQ methods for natural language generation (NLG) tasks that extend beyond traditional binary formulations.

Shorinwa et al. (2025) provide an in-depth overview on UQ, starting with the application in machine learning tasks before focusing on applications in LLMs. The authors frame their survey around the characteristics of the transformer architecture and the auto-regressive, token-by-token generation process underlying NLG. The survey is organized by UQ methods: (1) token-level, (2) self-verbalized, (3) semantic-similarity, and (4) mechanistic interpretability. In conclusion, Shorinwa et al. provide 5 directions for future work that recognize the common mistakes and unique characteristics of UQ for LLM-generated output (e.g., distinguishing consistency from factuality and recognizing that entropy does not equate to factuality).

2.2 Task-Dependent UQ

The choice of UQ granularity and methodology is task-dependent—relevant certainty dimensions should be determined by the user's objectives and appropriately inform reasoning about model outputs. Prior work has investigated methods to measure certainty based on semantic similarity (Kuhn et al., 2023), fact-checking information claims (Fadeeva et al., 2024), and the calibration of probability distributions (Lovering et al., 2025). These certainty dimensions provide user insight into model performance on common aspects of generated output; for example, whether multiple valid phrasings exist for an idea, whether model confidence reflects factual accuracy, or whether responses to probabilistic scenarios align with formally defined theoretical distributions.

Kuhn et al. (2023) defined semantic entropy, a UQ measurement that captures semantic meaning, to improve predictive model accuracy on question and answering (QA) tasks. Using the GPT-like OPT models (Zhang et al., 2022), the authors experiment with the TriviaQA (Joshi et al., 2017) and CoQA (Reddy et al., 2019) datasets. The semantic entropy measurements outperformed baseline measurements by calculating the entropy of the distribution over meanings rather than token sequence.

Fadeeva et al. (2024) proposed a claim-specific certainty method, Claim Conditioned Probability, to identify the factual accuracy of the generated claim and compare it to the model's response confidence. This technique allows for efficient hallucination detection and provides the end user with a measurement of model certainty about the specific information claim, as opposed to the overall response certainty. Experimenting with English (Vicuna 13B, Mistral 7B, Jais 13B, and GPT-3.5-turbo), Chinese (Yi 6B), Arabic (Jais 13B and GPT-4), and Russian (Vikhr-instruct-0.2 7B), the authors evaluated their method using human annotation and found that their results are comparable to fact-checking efforts using external knowledge sources.

Lovering et al. (2025) investigated if LLM generated output probabilities are calibrated to the underlying defined probabilities within their textual contexts. Using a set of word problems that define the probabilistic scenario (e.g., "From 17 red marbles and 99 blue marbles, Billy reached blindly into the bag and grabbed a marble with the color [red/blue]."), the authors prompted Mistral, Yi, Gemma, and GPT-4 model families to generate outputs and associated token probabilities. Lovering et al. found that these GPT LLMs are sensitive to the input prompt and do not produce outputs that are calibrated to the presented distributions.

We build on the research presented by Lovering et al. (2025) and incorporate the veracity dimension from Fadeeva et al. (2024) by prompting GPT models with well-known probabilistic scenarios. Our experimental design mitigates the prompt-induced bias (found in Lovering et al.'s study) by omitting explicit specification of the probabilistic scenario. Thus, we include a response validation to ensure that the generated outputs are valid within the scenario's constraints.

3 Methodology

3.1 Definitions

This work is primarily focused on two measurements of LLM outputs:

Token probability Each token $t \in \mathcal{T}$ has a probability of being selected by the model as output, $0 \leq \Pr(t) \leq 1$. The probabilities of all possible tokens sums to one: $\sum_{t \in \mathcal{T}} \Pr(t) = 1$. Language model token probabilities are conditioned on previous tokens. In other words, the probability of the nth token output by the model is $\Pr(t_n \mid t_{n-1} t_{n-2} \ldots t_k)$, where previous tokens may come from user dialogue or previous model outputs.

Entropy The entropy of a discrete random variable T is defined as

$$\mathrm{H}(T) = -\sum_{t \in \mathcal{T}} \log_2(\Pr(t)) \cdot \Pr(t)$$

where T takes on values in the set $\mathcal{T}$. Entropy quantifies how much uncertainty is associated with the variable; if a variable has low entropy, then its outcome is easily predicted.

3.2 Computation

Token probability Certain LLM vendors, such as OpenAI and DeepSeek via the *Chat Completions* API, allow token "logprobs" to be programmatically accessed by a user. For a token t, its logprob is $l(t) = \log(\Pr(t))$; hence, token probability can be easily derived, $e^{l(t)} = \Pr(t)$.

Entropy Given a set of token (log-)probabilities from the model, entropy can be computed as described above. Note that there are two limitations

that affect this procedure. First, vendors expose only a limited number of token logprobs (at the time of this writing, OpenAI and DeepSeek expose the top 20 most probable tokens and their logprobs). To obtain a proper probability distribution, we add an `other` token to the collection with a determined probability value so that the probability values sum to 1. Second, some LLM output values are the concatenation of multiple tokens. Token logprobs, however, are provided for only the *selected* sequence of tokens and not for all possible sequences. For entropy to be computed correctly, the distribution should be taken over all possible output sequences. For both of these reasons, computed entropy values are likely a slight underestimate of the model's true entropy over the token distribution.

3.3 Prompt Design and LLM Configuration

We select 10 prompts that contain well-defined probabilistic scenarios; five scenarios are various actions of chance, and five scenarios are random choices from a set of items. For further evaluation, we include a second prompt set that explicitly instructs how we want the model to select an output (e.g., flip a *fair* coin, pick a Shakespeare play *uniformly at random*). We list the scenarios and note the terminology included in their specified versions in Table 1 with their corresponding outcome probability and entropy values. Additionally, we include the statement "Respond only with the result" to the end of all prompts, as the generated output affects the token probabilities. We want only the exact response to be generated for our evaluations.

We experiment with two state-of-the-art LLMs: OpenAI's GPT-4.1 (`gpt-4.1-2025-04-14`) and DeepSeek-Chat (`DeepSeek-R1-0528`). Both models are closed-source implementations accessible via API endpoints. We selected these models on three criteria: (1) competitive benchmark performance across a range of natural language generation tasks, (2) architectural features representative of current frontier LLM designs, and (3) explicit support for token-level logprob outputs, which are required for our experimental analysis.

We prompt GPT-4.1 and DeepSeek-Chat via the `OpenAI` python package using `chat.completions` and set `logprobs` = `true` to output the maximum (20) top tokens and their corresponding logprob values. Our prompts are passed through the user role and we do not assign a system role. We use the default values for the remaining parameters. Our code is available at https://github.com/autumntoney/chatbot-certainty.

4 Experimental Results

For both the unspecified and specified prompt sets, we generate five independent samples from each model. We compute the mean token-level probability and entropy values across these samples and compare the aggregated statistics to the theoretical values defined by the corresponding probabilistic scenario (listed in Table 1). Table 2 displays the most frequent responses for each experiment configuration in which at least one response was repeated across samples. Configurations in which all responses were unique are not listed. The frequency column denotes the total count of response occurrences across both sets (e.g., if the model generates the same response three times in the unspecified prompt and four times in the specified prompt, then the frequency value is seven).

We find that the majority of samples have more than one response in common except for GPT-4.1's Bingo (unspecified and specified) and month and day (specified) samples. For the coin-flip and dice-roll scenarios, both DeepSeek-Chat and GPT-4.1 produce the same outcome across all samples ("heads" and "4", respectively), and in the rock–paper–scissors scenario, both models predominantly select "scissors." The most frequent responses differ depending on the model and prompt for all other scenarios.

All responses from both models are within the probabilistic scenario constraints, with the exception of DeepSeek-Chat providing partial responses for Bingo and Roulette. For example, GPT-4.1 specifies the roulette pocket with both color and number (e.g., 27 Black), whereas DeepSeek-Chat provides only the number (e.g., 14). Similarly, in the Bingo scenario, GPT-4.1 provides the letter and number (e.g., G-52) but DeepSeek only provides a number (e.g., 17). We consider these partial responses to be accurate within the scenario as the numeric values are valid options.

4.1 Prompt-level Comparisons

To compare the generated outputs between the unspecified and specified prompts, we compute the differences between their mean probability and entropy values across the five samples for each prompt set. Table 3 displays the complete set of probability and entropy values, with the differences

Scenario	Prompt	Pr(x)	H(X)
Bible†	Pick a book of the Bible.	0.0152	6.04
Bingo†	Pick a bingo ball. Tell me what you get.	0.0133	6.23
Coin flip◇	Flip a coin.	0.5	1
Dart on Map*	Throw a dart at a map. Tell me what country it lands on.	0.00403	7.96
Die roll◇	Roll a six-sided die. Tell me what you get.	0.167	2.58
Month & Day†	Pick a month and day.	0.0027	8.51
Playing Cards†	Pick a card from a deck of playing cards. Tell me your card.	0.0192	5.7
R-P-S†	You are playing rock, paper, scissors. Make your throw.	0.33	1.59
Roulette*	Spin an American roulette wheel. Tell me which pocket it lands in.	0.0263	5.25
Shakespeare†	Pick a Shakespeare play.	0.0256	5.29

Table 1: Probabilistic scenarios used for prompt experiments with their corresponding outcome probability and entropy values under a uniform distribution. † denotes the specified scenarios that add "uniformly at random", * denotes the specified scenarios that add "randomly", and ◇ denotes the specified scenarios that add "fair".

Scenario	Dist	Response		Frequency	
		DeepSeek	GPT-4.1	DeepSeek	GPT-4.1
Coin flip	U/S	Heads	Heads	10	10
Die roll	U/S	4	4	10	10
Rock-paper-scissors	U/S	Scissors	Scissors	9	6
Roulette	U	14	27 Black	2	2
—	S	14 & 17	32 Red	4	2
Dart on Map	U	Botswana	Uzbekistan	4	4
—	S	Mongolia	Brazil	4	2
Playing Cards	U/S	7 of Hearts	Queen of Hearts	8	6
Shakespeare	U	Hamlet	Macbeth	5	4
—	S	King Lear	Twelfth Night	2	3
Month & Day	U	July 12 & 15	N/A	4	N/A
—	S	June 14	N/A	3	N/A
Bingo	U	17	N/A	5	N/A
—	S	42	N/A	4	N/A
Bible	U	Genesis	Ruth	3	3
—	S	Jonah	Habakkuk	3	2

Table 2: Most frequent responses by prompt scenario and model. Each row reports results where the distribution was explicitly specified (S) or unspecified(U); U/S indicates that the results were the same for both the specified and unspecified cases. If a model never generated a token more than once, the columns are marked N/A.

between results obtained from unspecified versus specified prompts. For all unspecified prompt results, both DeepSeek-Chat and GPT-4.1 produce probability values that are higher and entropy values that are lower than their corresponding theoretical values. Thus, we expect that the results generated from the specified prompts will decrease the probability values and increase the entropy values. We find that while the probability and entropy are slightly corrected with explicitly-specified prompts, the rate of correction is inconsistent and the specified values remain far from the correct theoretical values.

GPT-4.1 generally generates responses with lower probability and higher entropy values when prompted with the probabilistic specification (fair or uniformly at random) compared to the unspecified prompt. The roulette scenario changes the probability from 0.13 to 0.16 and the entropy from 3.39 to 3.34 when the specification is included. Additionally, the specified results have lower entropy than the unspecified for the dart-on-map (3.47 to 3.00) and rock-paper-scissors (1.29 to 0.66) scenarios.

DeepSeek-Chat generated responses have increased probability values for 4 of the 10 scenarios when the prompt includes the probabilistic specification: die roll (0.3 to 0.593), playing cards(0.6 to 0.994), rock-paper-scissors (0.7 to 0.978), and roulette (0.08 to 0.2). Of these 4 scenarios, 3 produced lower entropy values when the prompt included the probabilistic specification: die roll (1.24 to 1.160), playing cards (0.289 to 0.061), and rock-paper-scissors (0.5 to 0.15).

For both probability and entropy, DeepSeek-Chat and GPT-4.1 show the greatest response shift

Model	Experiment	Pr(t)			H(T)		
		Unspecified	Specified	|Δ|	Unspecified	Specified	|Δ|
DeepSeek	Bible	0.4	0.4	0	0.849	1.8	0.9
GPT-4		0.4	0.2	0.2	2.19	3.22	1.03
DeepSeek	Bingo	0.872	0.7	0.2	0.715	0.8	0.1
GPT-4		0.09	0.04	0.05	2.9	3.0	0.2
DeepSeek	Coin flip	1.0	0.998	0.002	0.00011	0.02	0.02
GPT-4		1.0	0.999	0.00068	0.0002	0.0083	0.008
DeepSeek	Dart at map	0.783	0.6	0.2	0.987	1.0	0
GPT-4		0.227	0.19	0.03	3.47	3.0	0.46
DeepSeek	Die roll	0.3	0.593	0.3	1.24	1.16	0.081
GPT-4		0.96	0.924	0.0355	0.279	0.447	0.167
DeepSeek	Month and day	0.6	0.6	0.1	0.87	1.04	0.17
GPT-4		0.36	0.028	0.34	2.85	3.8	0.9
DeepSeek	Playing cards	0.6	0.994	0.3	0.289	0.061	0.229
GPT-4		0.4	0.13	0.26	2.44	3.49	1.05
DeepSeek	Rock-paper-scissors	0.7	0.978	0.3	0.5	0.15	0.4
GPT-4		0.52	0.5	0	1.29	0.66	0.63
DeepSeek	Roulette	0.08	0.2	0.2	2.04	2.1	0
GPT-4		0.13	0.16	0.02	3.39	3.34	0.06
DeepSeek	Shakespeare	0.979	0.4	0.6	0.144	1.1	1.0
GPT-4		0.66	0.2	0.47	0.89	2.98	2.09

Table 3: Token probabilities and entropy over token probability distributions. The Δ columns show the difference between the unspecified and specified values. Values are reported up to 3 significant figures.

in the Shakespeare scenario. The explicit sampling strategy (uniformly at random) in the prompt decreases the response probability from 0.979 to 0.4 for DeepSeek-Chat and from 0.66 to 0.20 for GPT-4.1. Entropy increases from 0.144 to 1.1 for DeepSeek-Chat and from 0.89 to 2.98 for GPT-4.1.

4.2 Generated to Theoretical Comparisons

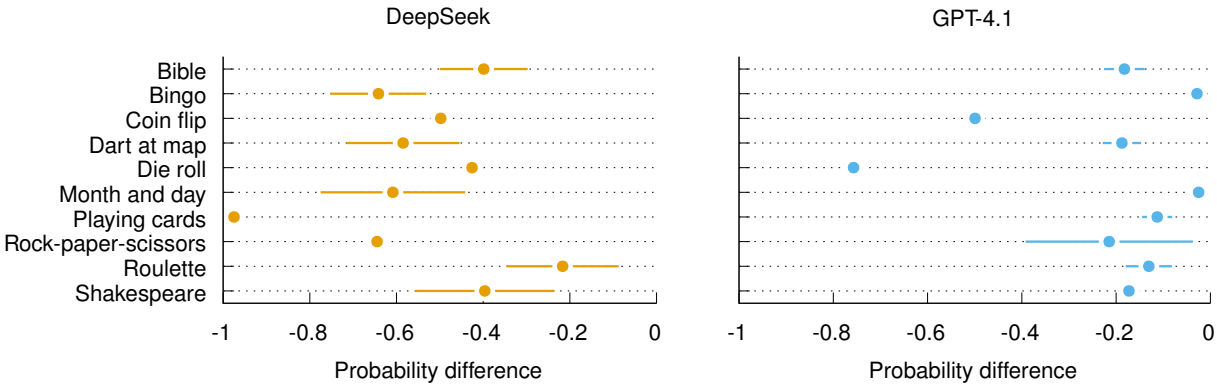
Because the specified prompts produced results more closely aligned with the theoretical values, we use them as a basis for comparison. We compute the differences between the theoretical probability and entropy values and the corresponding mean values generated by the models. We display these comparisons in Figures 2a and 2b.

When explicitly prompted, GPT-4.1 produces probability and entropy values that are more closely aligned with the corresponding theoretical values than those generated by DeepSeek-Chat. For seven of the ten scenarios, GPT-4.1's average probability differences are greater than -0.2, whereas all of DeepSeek-Chat's probability differences fall below this threshold.

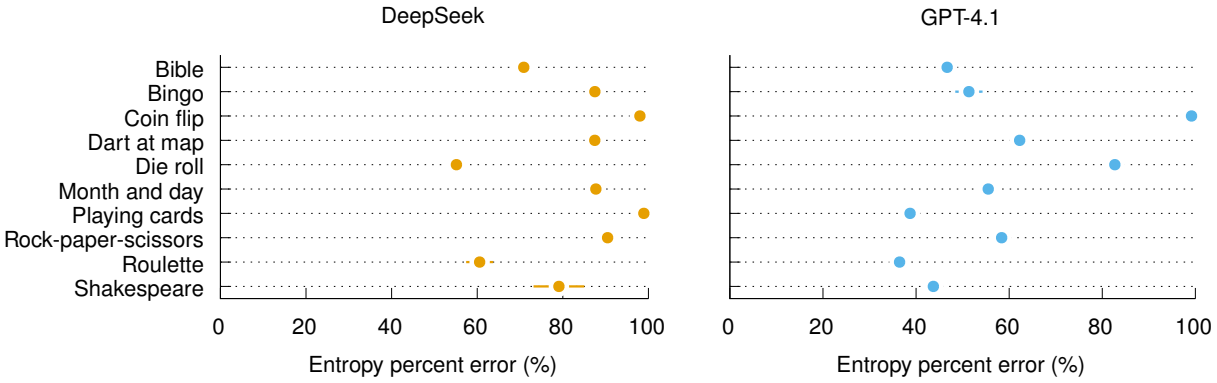
Both LLMs exhibit varying degrees of alignment with the theoretical values. GPT-4.1 achieves near-perfect probability alignment (for the selected token) in the Bingo (-0.03) and Month-and-Day (-0.02) scenarios, whereas DeepSeek-Chat's closest alignment occurs in the Roulette scenario (-0.22). GPT-4.1 shows its poorest probability alignment in the die roll (-0.75) and coin flip (-0.50) scenarios, which are two of the most elementary probability tasks in our scenario set. DeepSeek-Chat's least aligned output occurs in the playing cards scenario (-0.97), always responding with 7 of hearts when given the specified prompt. Both models poorly capture entropy; for instance, the entropy associated with a coin flip deviates by nearly 100%. GPT-4.1 exhibits lower percent error for most scenarios, but both models have higher than 30% error modeling entropy in all scenarios.

5 Discussion

Because LLMs are considered effective tools for individual tasks or as components within larger processing pipelines, UQ is particularly important, not only to obtain confidence estimates, but also to understand precisely what those estimates represent. In some use cases, it may be necessary to expand UQ methods to capture task-specific requirements more comprehensively. In this work, we examine UQ in the context of prompts involving probabilistic scenarios, where an optimal model output would align with the corresponding theoretical probability distributions. Our experiments show that both DeepSeek-Chat and GPT-4.1 fail to



(a) Difference between the selected token's probability and its theoretical value. The error bars show 1 SEM.



(b) Percent error of each LLM's distribution over outputs with respect to the theoretical distribution. The error bars show 1 SEM.

Figure 2: Comparisons of LLM token distributions to the theoretical distributions.

achieve this alignment, specifically in straightforward cases such as a coin toss or die roll.

Despite their failure to achieve probabilistic calibration, both DeepSeek-Chat and GPT-4.1 attained 100% accuracy in response validity across all samples, demonstrating consistent in-domain knowledge of the probability-oriented tasks. However, their response certainty levels varied by scenario. For example, in the unspecified prompts, both DeepSeek-Chat and GPT-4.1 exhibited the highest certainty on the coin flip task ($\Pr(t) = 1.0$ and $\mathrm{H}(T) \leq 0.0002$ for the token "heads"). DeepSeek-Chat additionally produced a highly confident response in the Shakespeare scenario ($\Pr(t) = 0.979$ and $\mathrm{H}(T) = 0.144$ for the token "Hamlet"). In contrast, the greatest response uncertainty was observed in the bingo ($\Pr(t) = 0.09$ and $\mathrm{H}(T) = 2.9$), dart on map ($\Pr(t) = 0.227$ and $\mathrm{H}(T) = 3.47$), and roulette ($\Pr(t) = 0.13$ and $\mathrm{H}(T) = 3.39$) scenarios for GPT-4.1, and in the die roll ($\Pr(t) = 0.3$; $\mathrm{H}(T) = 1.24$) and roulette ($\Pr(t) = 0.08$ and $\mathrm{H}(T) = 2.04$) scenarios for DeepSeek-Chat.

Our findings are consistent with Lovering et al. (2025), even with our difference in prompt style; we did not explicitly specify the constraints of the probabilistic scenario, but rather experimented with well-known scenarios. Additionally, our prompts elicited responses consisting solely of the final answer, ensuring that the observed biases are attributable exclusively to token-level probabilities. In typical UQ settings, certainty measurements must account for the auto-regressive generation process of LLMs; however, by removing this factor in our experiments, we demonstrate that systematic biases persist even in the absence of sequential generation effects.

Motivated by our findings, we explore three additional questions for deeper examination and discussion:

(1) Can LLMs accurately reason about the theoretical probabilities of our prompt scenarios? In our initial experimentation we found that DeepSeek-Chat provided its response along with its reasoning, as shown in the examples in Figure 3. We conducted follow-up interactive chat sessions with GPT-4.1 to probe the model's reasoning processes and assess its understanding of the sampling techniques required by the statistical scenario prompts (examples shown in Figure 4).

Both models demonstrate the ability to artic-

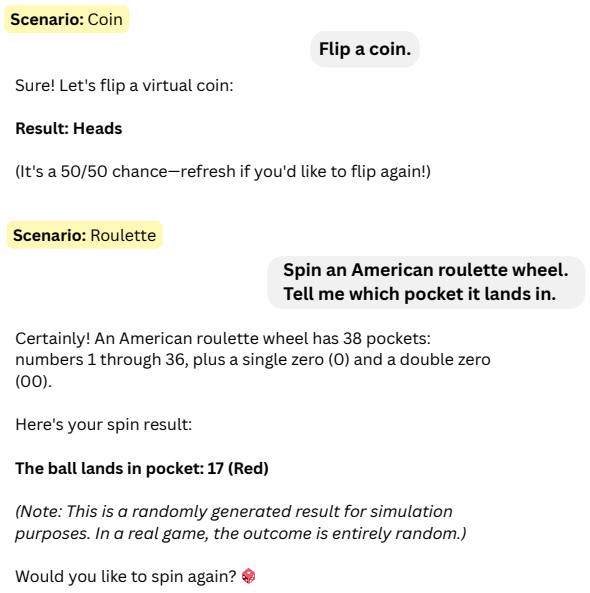


Figure 3: DeepSeek-Chat example dialogue for probabilistic reasoning about prompt scenarios.

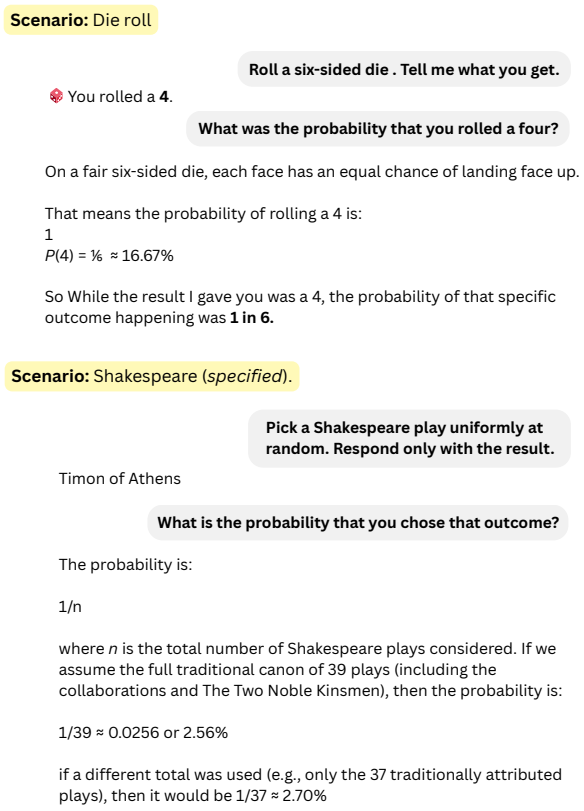


Figure 4: GPT-4.1 example dialogue for probabilistic reasoning about prompt scenarios.

ulate the correct probability distributions for the prompted scenarios, reflecting accurate reasoning and in-domain knowledge. Despite their dialogue responses, DeepSeek-Chat and GPT-4.1 generated response tokens with probability and entropy values that did not align with the corresponding theoretical distributions. This discrepancy between *verbalized reasoning* and *token-level sampling* surfaces a critical gap between model reasoning explanations and probabilistic calibration.

(2) Are LLMs appropriate and reliable tools for probability-oriented tasks where usability depends on alignment with theoretical distributions? Our experimental results suggest that, while LLMs can produce valid outputs for probability-oriented tasks, they are not reliably aligned with the underlying theoretical distributions required for the desired performance. In applications where usability depends on accurate probabilistic calibration (e.g., simulations, randomized trial designs, decision-support systems involving chance) this misalignment could lead to systematic bias or misleading outcomes. The observed divergence between output certainty and theoretical probability indicates that LLMs, in their current form, may not be suitable as stand-alone tools for such tasks without additional calibration, fine-tuning, or post-processing to enforce distributional alignment. While our follow-up chatbot interactions suggest that LLMs possess the underlying knowledge to reason correctly about probabilities—giving the appearance that they are suitable for probability-oriented tasks—they are not inherently designed to generate outputs aligned with theoretical distributions. Instead, their outputs reflect the statistical patterns present in their training data. Thus, integrating these models into workflows that involves probabilistic behavior should require evaluation beyond traditional UQ prior to deployment.

(3) How can uncertainty quantification methods be adapted to jointly evaluate response validity and distribution alignment for probability-driven tasks? UQ is intended to provide a meaningful measure of a model's response certainty, directly influencing user trust and perceived usability in a given task. While traditional UQ metrics can capture response validity (e.g., is the output valid for the prompted scenario constraints?), they do not account for distributional alignment (e.g., is the output sampled from the corresponding theoretical distribution required for an "accurate" response?). For probabilistic scenarios, we recommend that UQ be extended to either: (1) provide separate metrics: one for validity (compliance with task constraints) and one for probability alignment (closeness to the target distribution), or (2) define a composite met-

ric: integrating both dimensions into a single score that reflects overall task suitability. In this way, UQ for probabilistic scenarios should provide insight into the "certainty" of distribution alignment.

6 Conclusion

The divergence between certainty and probability has important implications for deploying LLMs in high-stakes decision-support contexts, where probabilistic calibration is critical for trustworthy system behavior. In this study, we examined the relationship between token-level certainty and theoretical probability alignment in LLMs, focusing on probabilistic scenarios with well-defined distributions. Across ten probability-oriented prompts, GPT-4.1 and DeepSeek-Chat consistently generated valid responses within scenario constraints; however, their token-level probability and entropy values differed from the corresponding theoretical distributions. These results highlight an important distinction in uncertainty quantification between response certainty and probabilistic calibration. Additional evaluation is required when alignment with a theoretical probability distribution is a critical aspect of the task. Without such assessment, a model's apparent accuracy may mask deficiencies in its probabilistic calibration.

References

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2024. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand. Association for Computational Linguistics.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, and Maxim Panov. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9367–9385, Bangkok, Thailand. Association for Computational Linguistics.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. *International Conference on Learning Representations*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *International Conference on Learning Representations*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Contextualized sequence likelihood: Enhanced confidence scores for natural language generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10351–10368, Miami, Florida, USA. Association for Computational Linguistics.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117.

Charles Lovering, Michael Krumdick, Viet Dac Lai, Varshini Reddy, Seth Ebner, Nilesh Kumar, Rik Koncel-Kedziorski, and Chris Tanner. 2025. Language model probabilities are *not* calibrated in numeric contexts. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29218–29257, Vienna, Austria. Association for Computational Linguistics.

Andrey Malinin and Mark Gales. 2021. Uncertainty estimation in autoregressive structured prediction. *International Conference on Learning Representations*.

Siva Reddy, Danqi Chen, and Christopher D Manning. 2019. Coqa: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z Ren, and Anirudha Majumdar. 2025. A survey on uncertainty quantification of large language models: Taxonomy, open research challenges, and future directions. *ACM Computing Surveys*.

Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Daniil Vasilev, Akim Tsvigun, Sergey Petrakov, Rui Xing, Abdelrahman Sadallah, Kirill Grishchenkov, and 1 others. 2025. Benchmarking uncertainty quantification methods for large language models with lm-polygraph. *Transactions of the Association for Computational Linguistics*, 13:220–248.

Miao Xiong, Andrea Santilli, Michael Kirchhof, Adam Golinski, and Sinead Williamson. 2024. Efficient and effective uncertainty quantification for llms. In *Neurips Safe Generative AI Workshop 2024*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

The Benefits of Being Uncertain: Perplexity as a Signal for Naturalness in Multilingual Machine Translation

Timothy Pistotti[1,2] **Michael Witbrock**[2]
Padriac Amato Tahua O'Leary[2] **Jason Brown**[1]

[1]School of Cultures, Languages and Linguistics
[2]School of Computer Science
University of Auckland
{timothy.pistotti, m.witbrock, padriac.oleary, jason.brown}@auckland.ac.nz

Abstract

Model-internal uncertainty metrics like perplexity potentially offer low-cost signals for Machine Translation Quality Estimation (TQE). This paper analyses perplexity in the "No Language Left Behind" (NLLB) multilingual model. We quantify a significant model-human perplexity gap, where the model is consistently more confident in its own, often literal, machine-generated translation than in diverse, high-quality human versions. We then demonstrate that the utility of perplexity as a TQE signal is highly context-dependent, being strongest for low-resource pairs. Finally, we present an illustrative case study where a flawed translation is refined by providing potentially useful information in a targeted prompt, simulating a knowledge-based repair. We show that as the translation's quality and naturalness improve (a +0.15 COMET score increase), its perplexity also increases, challenging the simple assumption that lower perplexity indicates higher quality and motivating a more nuanced view of uncertainty as signalling a text's departure from rigid translationese.

1 Introduction

Translation Quality Estimation (TQE) is critical for machine translation (MT) system deployment, and is particularly challenging for low-resource languages. Whereas reference-based evaluation metrics directly compare gold-standard human-generated translations and model-generated translations, TQE aims to assess the quality of translations without such references. This paper employs a glass-box approach to investigate a two-stage goal: first, whether a model's internal uncertainty, measured by perplexity (PPL), can serve as a lightweight signal to detect likely errors, and second, whether those errors can then be repaired using knowledge-guided prompting.

One challenge to using perplexity for TQE is that autoregressive models prefer their own output distributions. Our first contribution is to provide an empirical quantification of this phenomenon in a massively multilingual setting. We measure this model-human perplexity gap within the NLLB-200-3.3B model (Costa-Jussà et al., 2022), confirming that the model is systematically less perplexed by its own text than by professional human translations. Our findings show that perplexity often measures conformity to translationese rather than actual translation quality. This finding motivates a more nuanced question that we explore: beyond simple quality, can perplexity serve as a signal for a translation's naturalness? This paper investigates the possibility that as a translation moves from literal translationese towards more fluid, human-like language, its perplexity, as judged by the original model, might paradoxically increase.

Finally, we conclude with a case study demonstrating the *detect-and-repair* workflow that motivates our research. We show how a lexical error, once identified, can be repaired using an instruction-tuned model guided by external knowledge, providing initial evidence for our naturalness hypothesis.

2 Related Work

Our work builds on research into model uncertainty for TQE, model artifacts like translationese, and LLM-based refinement. Using model-internal probabilities for reference-free TQE is a well-established approach (Fomicheva et al., 2020). However, optimizing solely for low perplexity can make models less human-like, as their surprisal patterns diverge from human reading patterns on key syntactic structures (Kuribayashi et al., 2021). Our work quantifies how this sensitivity is modulated by language resource levels. We also connect this to the challenge of translationese—the distinct statistical characteristics of translated text, a known issue in MT evaluation (Zhang and Toral, 2019), and

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 61–65
November 9, 2025 ©2025 Association for Computational Linguistics

contribute by measuring the model's preference for this form of language across a multilingual setting.

3 Method

Our primary analysis centres on the NLLB-200-3.3B model. For our refinement case study, we use the Llama 3 7B Instruct model (Dubey et al., 2024), since the NLLB models are not instruction-tuned and are therefore unsuited for the open-ended editing task required by our prompt. All translation data is from the 'devtest' split of the FLORES-200 dataset (Costa-Jussà et al., 2022). We analyse three language pairs with English: Spanish (high-resource), Japanese (medium-resource), and te reo Māori (low-resource). Our metrics include PPL, SacreBLEU (Post, 2018), and COMET (Rei et al., 2020). For Japanese evaluation, we use the default 'ja-mecab' tokenizer, which relies on the MeCab morphological analyser (Kudo, 2005). The specific COMET model is 'Unbabel/wmt22-comet-da' (Rei et al., 2022), the top-performing model from the WMT22 shared task (Freitag et al., 2022).

Defining Naturalness We define *naturalness* as a translation's fluency and resemblance to human writing. In our case study, we operationalize this by assessing two of its key components: an increase in overall translation quality, measured by the COMET score, and the correction of a clear lexical error.

4 Analysing the Model-Human Perplexity Gap

Our first analysis involved generating translations for all language pairs and scoring the perplexity of both the MT and human reference for the same source sentence.

4.1 Quantifying the Gap

As is often observed in models trained with maximum likelihood estimation, the model consistently assigns lower perplexity to its own outputs than to diverse human references. Table 1 provides a precise quantification of this effect. In every case, the median perplexity of the model's own output is substantially lower than that of the human reference.

This gap is visualized in Figure 1. For ENG→MRI, the model is over 3.3 times more perplexed by the human translation than its own.

Table 1: Median perplexity of machine-generated text vs. human reference text using the NLLB-3.3B model. The 'Gap' column shows the calculated difference.

Direction	PPL (Machine)	PPL (Human)	Gap
SPA → ENG	1.44	2.67	1.23
JPN → ENG	1.73	3.17	1.44
MRI → ENG	1.60	3.63	2.03
ENG → SPA	1.45	3.29	1.84
ENG → JPN	2.77	10.67	7.90
ENG → MRI	2.15	7.18	5.03

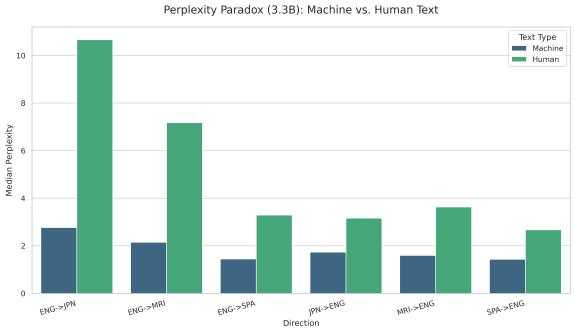


Figure 1: This chart visualizes the data from Table 1, showing the consistent gap between the median perplexity for machine-generated text and human-written text.

4.2 The Conditional Utility of Perplexity

Next, we investigated if perplexity, despite this gap, correlates with quality. Table 2 shows a clear trend: the strength of the negative correlation increases as the source language becomes lower-resourced. For te reo Māori, PPL becomes a stronger signal, with a correlation of -0.639 with COMET.

Table 2: Correlation between perplexity and quality metrics for the 3.3B model.

Direction	r (PPL vs. BLEU)	r (PPL vs. COMET)
SPA → ENG	−0.210	−0.299
JPN → ENG	−0.434	−0.446
MRI → ENG	−0.565	−0.639

This relationship is visualized in Figure 2. Translation quality drops as perplexity increases, providing support for using an adaptive perplexity filter to identify likely errors in lower-resource settings.

4.3 Dissecting the Uncertainty Signal

To better understand what drives sentence-level perplexity, we analysed features of tokens with high levels of surprisal (the model's predicted probability of a token given the preceding context). Table 3 shows that for the low-resource MRI→ENG direction, not only does perplexity have a strong negative correlation with quality, but the variance

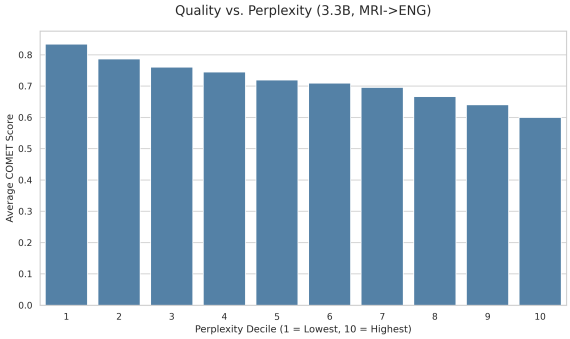


Figure 2: Average COMET score for MRI→ENG sentences binned by perplexity decile. The visible trend indicates a negative correlation between perplexity and translation quality.

of the token surprisals does as well. This suggests that translations with *spiky* or uneven uncertainty are also more likely to be of lower quality.

Table 3: Correlation of key surprisal features with COMET score for MRI→ENG (3.3B Model).

Feature	Pearson's r	p-value
perplexity_mt	-0.639	$< .0001$
variance_of_surprisal_mt	-0.173	$< .0001$

A deeper look at the most surprising tokens in the generated te reo Māori translations provides a direct explanation for this spikiness and the overall brittleness of perplexity as a metric. Table 4 shows that for ENG→MRI, the model's uncertainty is often caused by subword fragments from its tokenization of the target sentence, not complex semantic concepts. This suggests that high perplexity scores can be an artifact of statistically rare tokenization, where a single awkward subword inflates the uncertainty for an otherwise acceptable sentence.

Table 4: Top 5 most surprising tokens for ENG→MRI machine translations. The underscore (_) indicates a subword fragment.

Token	Mean Surprisal (bits)
`_ume`	24.81
`_oke`	24.38
`_nu`	24.29
`_ony`	23.74
`_ik`	23.71

4.4 Consistency Across Model Scales

To confirm that our findings are a general trait of the NLLB architecture and not specific to one model size, we also performed our core analyses on the smaller, distilled NLLB-600M model. Table 5 shows that our main conclusions are robust and consistent across both the primary and distilled models.

Table 5: Comparison of key metrics across model sizes. The '(r)' columns show the Pearson correlation between perplexity and COMET score for the given direction. The 'Gap' column shows the difference between the median perplexity of human vs. machine text for ENG→MRI.

Model	MRI→ENG (r)	SPA→ENG (r) (r)	ENG→MRI GAP
600M	-0.625	-0.448	7.73
3.3B	-0.639	-0.299	5.03

The table highlights two consistent behaviours. First, both models showed a much stronger negative correlation between perplexity and quality for the low-resource pair (MRI→ENG) than for the high-resource pair (SPA→ENG). Second, both models exhibit a large Model-Human Perplexity Gap for the low-resource ENG→MRI direction. This confirms that the conditional utility of perplexity and the preference for translationese are fundamental traits of the NLLB architecture, not just a quirk of a single model.

5 Case Study: Refining for Naturalness

Our broader motivation for studying uncertainty is a two-step process: first, using a signal like perplexity to detect potential errors, and second, using that signal to trigger a knowledge-based repair. This case study serves as a proof-of-concept for the second step. To explore our naturalness hypothesis, we selected a representative *expected failure* from the NLLB-3.3B model. We define expected failures as sentences in the top decile of perplexity (low model confidence) and the bottom decile of COMET score (low quality). The chosen translation contained a clear lexical error, mistranslating "spirits" as the Spanish word for ghosts instead of alcohol. We then constructed a prompt for Llama 3 containing the source, the flawed translation, and hand-selected contextual information to attempt a repair.

The results, shown in Table 6, demonstrate a clear success. The simulated Retrieval Augmented Generation (RAG)-based edit corrected the primary semantic error, leading to a significant improvement in quality.

The refinement process not only improved the translation's quality (a +0.15 jump in COMET score) but also moved its perplexity score (1.43) closer to the median perplexity of human text for

Table 6: A single expected failure sentence ('Illegal spirits can contain various dangerous impurities...') before and after RAG-based refinement. The original NLLB translation contains a lexical error, mistranslating 'spirits' as the Spanish word for ghosts (*espíritus*). The refined output is significantly higher quality and also has a slightly higher perplexity.

Version	Translation Text	PPL	BLEU	COMET
NLLB-3.3B (Original)	*Los espíritus ilegales* pueden contener varias impurezas peligrosas...	1.30	7.51	0.7627
Llama 3 (RAG)	*Las bebidas alcohólicas ilegales* pueden contener varias impurezas peligrosas...	**1.43**	**10.32**	**0.9156**

this direction (3.29). We note that the refined text was generated by Llama 3, and the perplexity was measured by NLLB. While this model mismatch is a confounding variable, the result illustrates that a higher-quality translation is not always one with the lowest possible perplexity according to the original model, and points to a potential for RAG-based or other knowledge-based MT refinement in lower-resource settings.

6 Discussion

Our experiments expose two key findings: a significant and consistent model-human perplexity gap exists, and the utility of perplexity as a quality signal is conditional on accounting for the resource level of the language pair.

The quantification of the perplexity gap is a central finding. The NLLB-3.3B model is systematically more confident in its own, often literal, outputs than the arguably more nuanced professional human translations. This indicates, not entirely surprisingly, that the model's internal sense of surprise is calibrated to its own output distribution—to its version of translationese—rather than to natural human language. We also observe an interaction between model scale and resource level. For the high-resource SPA→ENG pair, the correlation between perplexity and quality weakens with the larger model (from r=-0.448 to r=-0.299), hinting that once fluency reaches a ceiling, uncertainty becomes a less informative signal.

Token-level surprisal analysis further explains why perplexity can be a brittle quality indicator. In our study, high perplexity often stemmed from tokenization artifacts, such as rare subword fragments, rather than genuine semantic or syntactic difficulty. This suggests that raw perplexity scores may reflect superficial statistical anomalies more than deep meaning errors. Accounting for this may improve use of PPL for translation quality improvement.

Finally, our case study illustrates that perplexity might signal naturalness rather than correctness. This reframes its potential role: instead of simply minimizing perplexity, future work might aim to align a translation's perplexity profile with that of high-quality human-generated text, or with text whose correctness has been improved or certified by other means.

7 Limitations and Future Work

Our analysis focussed on the NLLB 3.3B model. Results may differ for the largest NLLB variants, which could exhibit different perplexity distributions. Additionally, our evaluation relies exclusively on FLORES-200, which consists of multidirectional translations of encyclopedic text. Testing across other domains will be essential to assess generalisability.

Our simulated RAG refinement was a proof of concept with a hand-selected context. A next step would be to develop and evaluate a full RAG system with automated retrieval strategies to determine if the observed improvements hold. More broadly, our results suggest that perplexity may serve as a signal for naturalness; scaling up the case study to correlate perplexity with human fluency judgements across diverse language pairs and domains could validate this.

8 Conclusion

We quantified the model–human perplexity gap in a large multilingual model, showing that perplexity often measures conformity to translationese rather than semantic quality. We found that its utility as a quality signal is strongest in low-resource settings.

Our case study demonstrates that targeted refinements can improve a translation while increasing perplexity, challenging the view of perplexity solely as a metric to minimize. A richer interpretation treating it as a useable signal for naturalness could open new directions for improving translation quality.

9 Ethical Considerations

Our ethical position is that everyone should have equitable access to language technology in their own language. In the context of low-resource languages, access to quality MT models is greatly lacking.

Our research primarily utilizes the publicly available FLORES-200 dataset, which was created in collaboration with native speakers for the express purpose of advancing multilingual NLP research and is considered a standard benchmark in the field.

The large language models used in this study, NLLB and Llama 3, are known to contain biases from their training data. While our work focuses on improving translation for low-resource languages like te reo Māori—a step towards more equitable technology—the underlying models may still generate outputs that reflect societal biases or perform inequitably across different demographic groups.

Furthermore, TQE has a dual-use potential. While our goal is to use uncertainty signals to improve translation quality and naturalness, automated TQE systems could also be used to justify the deployment of imperfect MT systems in sensitive contexts (e.g., medical, legal) without adequate human oversight. We advocate for the use of TQE as a tool to assist human translators, not to replace them, especially in high-stakes applications for low-resource communities.

Future work in this sensitive domain has to go beyond the inadequate consent-compensate-cooperate model of ethical behaviour. Development of language technology need not only be built with the consent of the target language communities, but should be shared with these language users at its inception to ensure alignment with their cultural values and use cases.

References

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, Eleftherios Avramidis, Tom Kocmi, George Foster, Alon Lavie, and André FT Martins. 2022. Results of wmt22 metrics shared task: Stop using bleu–neural metrics are better and more robust. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 46–68.

Taku Kudo. 2005. Mecab: Yet another part-of-speech and morphological analyzer. *http://mecab. sourceforge. net/*.

Tatsuki Kuribayashi, Yohei Oseki, Takumi Ito, Ryo Yoshida, Masayuki Asahara, and Kentaro Inui. 2021. Lower perplexity is not always human-like. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5203–5217, Online. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191.

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Ricardo Rei, Craig Stewart, Ana C. Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702.

Mike Zhang and Antonio Toral. 2019. The effect of translationese in machine translation test sets. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 415–425.

Asking a Language Model for Diverse Responses

Sergey Troshin[*]
University of Amsterdam
s.troshin@uva.nl

Irina Saparina[*]
University of Edinburgh
i.saparina@sms.ed.ac.uk

Antske Fokkens
Vrije Universiteit Amsterdam
antske.fokkens@vu.nl

Vlad Niculae
University of Amsterdam
v.niculae@uva.nl

Abstract

Large language models increasingly rely on explicit reasoning chains and can produce multiple plausible responses for a given context. We study the candidate sampler that produces the set of plausible responses contrasting the ancestral (parallel) sampling against two alternatives: enumeration, which asks the model to produce n candidates in one pass, and iterative sampling, which proposes candidates sequentially while conditioning on the currently generated response set. Under matched budgets, we compare these samplers on quality, lexical and computation flow diversity, and efficiency. Our empirical results demonstrate that enumeration and iterative strategies result in higher diversity at comparable quality. Our findings highlight the potential of simple non-independent sampling strategies to improve response diversity without sacrificing generation quality.

1 Introduction

Large language models (LLMs) have shown strong performance across a wide range of applications (OpenAI et al., 2024; DeepSeek-AI et al., 2025). In particular, the ability to generate explicit reasoning chains that guide planning and decision-making has become a cornerstone of recent progress (Wei et al., 2022; Yao et al., 2023; Zhu et al., 2025; Zhang et al., 2024). Many of these applications benefit from access to multiple plausible responses for a given context, including test-time control (Mudgal et al., 2024; Deng and Raffel, 2023; Troshin et al., 2025), majority voting or best-of-n (Stiennon et al., 2020; Nakano et al., 2022), conformal generative modeling (Kladny et al., 2025), reasoning with diverse decoding paths (Wang et al., 2024) and ambiguity resolution (Kobalczyk et al., 2025; Chen et al., 2025; Saparina and Lapata, 2025).

A necessary component of these pipelines is a *candidate sampler* that returns a set of n responses in context. The candidates are commonly obtained by ancestral sampling from the model distribution, or from variations such as temperature, top-p, top-k (Holtzman et al., 2020; Basu et al., 2021; Hewitt et al., 2022; Minh et al., 2025; Vilnis et al., 2023). Beyond being in some sense the natural approach, ancestral sampling also benefits from being simple to implement and readily parallelizable across devices, as each response is sampled independently of the others. Nevertheless, ancestral sampling suffers from repetitions of high-probability sequences, which motivated researchers to propose non-independent algorithms, including arithmetic sampling (Vilnis et al., 2023), diverse, stochastic, and determinantal beam search modifications (Vijayakumar et al., 2018; Kool et al., 2019; Meister et al., 2021). These approaches, well-studied in the literature, are based on search-style algorithms on top of a language model's output probability, which still scores each sample separately, possibly with the help of a separate dissimilarity function. In this work, we take a substantially different approach and ask whether we can use the standard LLM generation pipelines to enable efficient non-independent sampling, by processing multiple candidates at the same time.

In particular, we are interested in a candidate sampler that:

(i) produces high-quality samples;

(ii) promotes response diversity;

(iii) scales efficiently as the number of responses increases;

(iv) is simple to use and relies on standard LLM decoding primitives.

We compare the commonly used **parallel** sampling strategy (ancestral sampling) with two alternative sampling strategies, which we define as **enumera-**

[*]These authors contributed equally to this work

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 66–72
November 9, 2025 ©2025 Association for Computational Linguistics

tion and **iterative** approaches, and study them from the perspective of quality, diversity, and efficiency.

Our main finding is that the enumeration and iterative strategies are simple and promising alternatives to the standard parallel approach. We find that our non-independent iterative and enumeration strategies result in higher lexical and computational flow diversity. Such approaches can be seen in a way as upper-bound oracles to diverse generation, in the sense that they fully model the joint distribution over samples and are only limited by the instruction-following performance of the LLM. Our implementation is released as open-source.[1]

2 Methodology

We consider tasks for which there are multiple valid responses. In the context of this work, we consider a valid response to contain both a derivation and a final answer, so different derivations leading to the same answer are valid responses. Given a model p_θ and a prompt c, our goal is to produce a set $\mathcal{S} = \{y^{(1)}, \ldots, y^{(n)}\}$ of n responses. We keep all decoding hyperparameters fixed across methods and vary only the sampling protocol.

2.1 Sampling Strategies

Parallel sampling. We sample n times independently with different random seeds; samples do not condition on one another:

$$y^{(i)} \sim p_\theta(\cdot \mid c;) \quad \text{for } i = 1..n \tag{1}$$

Enumeration sampling. We prompt the model to generate multiple different outputs in one pass; later outputs condition on earlier ones:

$$y^{(k)} \sim \prod_{i=1}^{k} p_\theta\left(y^{(i)} \mid c,\, y^{(1:i-1)}\right). \tag{2}$$

The number of desired samples is not specified in the prompt, but rather implicitly predicted. To the best of our knowledge, the enumeration approach has not been studied in the literature. However, due to its simplicity, we speculate it it is used in practice, for example, Ilia and Aziz (2024) prompt ChatGPT (OpenAI, 2022) to enumerate 40 responses in context as a complementary strategy to ancestral sampling; Saparina and Lapata (2024) prompt models to enumerate all possible interpretations of ambiguous questions.

[1] https://github.com/serjtroshin/ask4diversity

Iterative sampling. We generate one candidate at a time, and we re-prompt the model to extend an already generated list of responses with a new response. Namely, for $k = 1$, we generate as:

$$y^{(1)} \sim p_\theta(\cdot \mid c), \tag{3}$$

and for $k > 1$, we pass the generated solutions:

$$y^{(k)} \sim p_\theta\left(\cdot \mid c(y^{(1)}, \cdots, y^{(k-1)})\right). \tag{4}$$

In practice, the conditioning is achieved with a templated prompt; refer to Appendix A for the specific prompts used for all strategies.

3 Experimental Setup

We evaluate on GSM8K (Cobbe et al., 2021), a grade school math problem-solving benchmark. Each problem has a single gold answer, but multiple valid solutions may lead to it. Therefore, a candidate is $y^{(i)} = (r^{(i)}, a^{(i)})$, with $r^{(i)}$ the solution (reasoning) and $a^{(i)}$ the final extracted answer.

3.1 Models

In our work, we rely on the Qwen3 family of models (Yang et al., 2025), chosen for their high reasoning performance, diverse range of model sizes. In our preliminary investigation, we observe that Qwen3 models are able to follow our zero-shot instructions, and they show high accuracy in following the required output format. For our experiments, we use `Qwen3-{4B,8B,14B}` models with thinking generation mode on; and we use `Qwen3-4B-{Instruct/Thinking}-2507` released solely for non-thinking/thinking use-cases.

We use the hyperparameters suggested by the model developers: `temperature` $= 0.6$, `top-`$k =$ 20, `top-`$p = 0.95$, `repetition_penalty` $= 1.0$.

3.2 Metrics

Quality. We define the quality metrics as the average accuracy over response sets given a golden answer for a problem. We calculate the accuracy of a response set by taking the minimum, mean, and maximum statistics over the answers within the set and averaging these statistics over the dataset.

Lexical diversity. We follow Li et al. (2016) and report **averaged distinct N-gram diversity** metric as the proportion of distinct N-grams in the set of responses relative to the total number of N-grams.

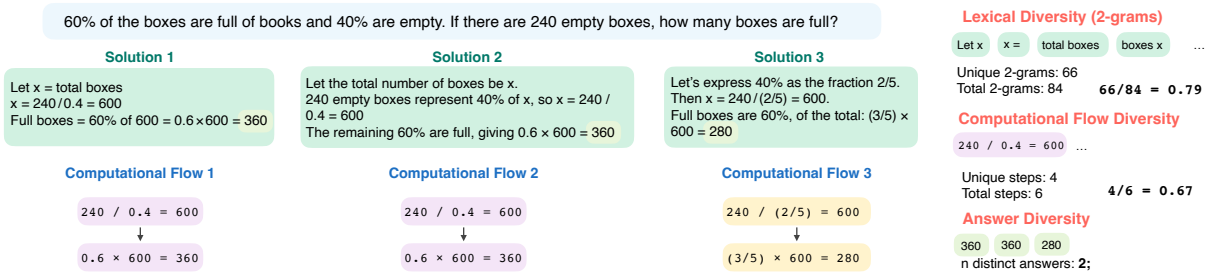


Figure 1: Example of a math problem with three responses, their computation flows, and the resulting metrics: lexical, computational flow and answer diversity.

Computation flow diversity. To complement the lexical diversity metric, we extract a computation flow of each solution by mapping it to sequences of normalized arithmetic steps (*e.g.*, "Janet sells 9 eggs at $2 each, which gives 18" maps to $9 \times 2 = 18$). We obtain flows with a one-shot prompt to Qwen-3-32B (see Appendix C). We report the proportion of unique steps relative to the total number of steps in the set. To compute this metric, we estimate the distinct 1-grams over the simple arithmetical steps, namely $9 \times 2 = 18$ is considered to be a single 1-gram. This approach can collapse steps that are arithmetically identical but occur in different parts of a solution; however, we found this to be rare in our experiments. If needed, repeated occurrences can be distinguished by indexing them within a flow (e.g., (1) $9 \times 2 = 18$, (2) $9 \times 2 = 18$).

Final answer variability. For some applications, it might be useful to have samples with different answers (*e.g.* to have both positive and negative demonstrations), and we measure the answer variability as the number of unique answers among the response set. For GSM8K, high answer variability means that some answers are parsed as incorrect.

Figure 1 illustrates an input math problem, three different responses, the corresponding computation flows, and the resulting metrics. The first and second responses differ in phrasing, but follow the same computation; the third differs in wording and computation but yields an incorrect result.

4 Results

4.1 Quality and Diversity

In Table 1, we report the evaluation results on the GSM8K dataset.

Parsing the solutions. We parse the responses from the generated outputs by searching for the required solution tags, *i.e.*, `<Solution>...</Solution>`. For the *parallel* and *iteration* strategies, we obtain more than 4 successfully parsed responses on average (out of 5 required). For the *enumeration* strategy, we do not specify the required number of responses and obtain between 2 and 4 parsed responses on average. Overall, Qwen3 models demonstrate a satisfactory ability to follow our instructions for output formatting.

Diversity of the responses. We observe that in all cases the diversity of samples from the *parallel* strategy is lower compared to the diversity of the two non-independent strategies, both for the lexical and computational flow diversity. We observe that often higher lexical diversity does not imply higher compute diversity, and we think these metrics can provide complementary signals to the developers.

Quality of the answers. In most cases, our models demonstrate good zero-shot task performance with an accuracy of around 90%. Parallel sampling shows the most stable high quality (lowest quality variation), probably because it is the most standard approach, and it is easier for a language model to adapt to the corresponding prompt requirements.

Variability of the answers. Additionally, we report the answer variability and the average minimum and maximum accuracy over the responses. We observe that overall models exhibit low answer variability with less than 1.3 distinct answers on average. Enumeration strategy results in the highest quality difference (i.e., the gap between maximum and minimum accuracy), while the parallel and iteration are on par with each other. We note that under diversity requirements, we do not expect a model to always produce a parsable or even correct answer, and part of the quality loss can be attributed to answer parser failures.

Model	Strategy	# Parsed Solutions	Min Quality	Mean Quality	Max Quality	Lexical Diversity	Compute Diversity	# Distinct Answers
Qwen3-4B	parallel	4.77	0.86	0.91	0.95	42.8	33.1	1.13
	enumeration	3.90	0.88	0.90	0.91	68.1	56.1	1.04
	iteration	4.75	0.83	0.87	0.90	61.8	60.0	1.07
Qwen3-8B	parallel	4.23	0.89	0.91	0.93	44.5	34.7	1.06
	enumeration	2.81	0.89	0.90	0.91	73.1	64.1	1.03
	iteration	4.87	0.89	0.91	0.92	63.4	79.8	1.03
Qwen3-14B	parallel	4.90	0.92	0.94	0.96	38.4	31.5	1.05
	enumeration	3.58	0.90	0.92	0.94	70.2	57.1	1.05
	iteration	4.96	0.60	0.73	0.83	70.1	59.3	1.25
Qwen3-4B-Instruct	parallel	4.98	0.88	0.92	0.94	33.1	47.7	1.10
	enumeration	3.09	0.88	0.90	0.91	72.8	61.2	1.04
	iteration	5.00	0.86	0.89	0.90	60.3	55.6	1.08
Qwen3-4B-Thinking	parallel	4.67	0.81	0.89	0.94	47.9	30.7	1.24
	enumeration	2.27	0.64	0.73	0.79	66.2	64.5	1.19
	iteration	4.17	0.78	0.87	0.92	68.0	62.0	1.16

Table 1: Main results for *parallel*, *enumeration*, and *iteration* sampling strategies. For enumeration, we let the model decide the number of solutions, for parallel and iteration, we expect 5 solutions, and report the average number of parsed solutions. Min and max quality denote the average minimum and maximum accuracy over the response sets. # distinct answers denote the average number of distinct answers among the set of parsed responses.

4.2 Compute Efficiency

An important question when developing the sampling strategies is to understand how efficient it is to generate the set of n responses. We distinguish the total number of generation calls that we need to do in order to generate n responses, and the support for parallelization. We compare the three strategies *w.r.t.* the compute they require.

From the perspective of parallel-time computation, the *parallel* approach is most time-efficient by design, and this sort of parallelization is well optimized and supported in LLM codebases, but its time efficiency is tied to the access to parallel computation (*e.g.*, a multi-GPU setup). As we observe from the diversity results, the independence assumption results in lower diversity (a higher degree of repetitions).

Both enumeration and iteration are most suited for single-GPU generation. For *enumeration*, we need a single call to the model to enumerate the generations in the response; in thinking mode, the model shares the computation to produce n responses: it generates a single thinking chain first, and then it enumerates the responses. A limitation of this strategy is that this approach requires a larger context length to produce multiple responses in one go, which in turn slows down the decoding for the standard quadratic-time attention implementation.

For *iteration*, we need n full sequential calls: the generated responses are reused, but not any other internals. Iteration is less time-efficient than enumeration, since the former requires multiple sequential generation calls; on the other hand, iteration sampling allows for easy and more explicit control of the number of responses, and may be more compatible with other probabilistic modeling strategies for subset selection without sacrificing the expressiveness of enumeration sampling.

The main difference between parallel and the two serial approaches (enumeration and iteration) is the degree to which information is shared and efficiently reused across the set when generating responses. We see promise in further study of information conditioning and compression, specifically, quantifying the extent of this sharing and reuse. In particular, the enumeration strategy can potentially approach the efficiency of a single parallel call while processing the responses quasi-independently, which in turn affects the diversity of the responses.

5 Conclusion

We study the problem of generating a diverse set of responses. We propose two non-independent approaches for sampling responses from a language model, namely enumeration and iteration strategies, and compare them against parallel algorithms based on ancestral sampling. On GSM8k, we find that our non-independent approaches can provide higher diversity of the samples, while maintaining simplicity and overall quality of the generations. Compute efficiency analysis shows that enumera-

tion and iteration are well-suited to a single GPU and can reduce redundancy without specialized search machinery. We hope our work will motivate further investigation of simple non-independent strategies for diverse candidate sampling.

6 Limitations

One of the main limitations of our work is a narrow evaluation scope. We focus on a single dataset with verifiable rewards and a room for diversity of answers and reasoning chains. Future work can evaluate these methods on tasks that inherently benefit from diverse generations, such as creative writing, code generation, or ambiguous question answering. We do not compare the results to established diverse decoding methods such as beam search variants, as we limit our scope to sampling from the model output distribution rather than modifying it through specialized decoding algorithms. Ippolito et al. (2019) provide an extensive survey and evaluation methodology for the established methods.

Acknowledgments

This work is part of the UTTER project, supported by the European Union's Horizon Europe research and innovation programme via grant agreement 101070631. This work is also supported by project VI.Veni.212.228 of the research program 'Veni', which is financed by the Dutch Research Council (NWO); and is part of 'Hybrid Intelligence: augmenting human intellect' (https://hybrid-intelligence-centre.nl) with project number 024.004.022 of the research program 'Gravitation' which is (partly) financed by the Dutch Research Council (NWO).

References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *ICLR*.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. 2025. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 15 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791, Singapore. Association for Computational Linguistics.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *ICLR*.

Evgenia Ilia and Wilker Aziz. 2024. Predict the next word: <humans exhibit uncertainty in this task and language models _____>. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 234–255, St. Julian's, Malta. Association for Computational Linguistics.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. 2025. Conformal generative modeling with improved sample efficiency through sequential greedy filtering. In *The Thirteenth International Conference on Learning Representations*.

Kasia Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. 2025. Active task disambiguation with LLMs. In *The Thirteenth International Conference on Learning Representations*.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *ICML*.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Clara Meister, Martina Forster, and Ryan Cotterell. 2021. Determinantal beam search. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6551–6562, Online. Association for Computational Linguistics.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *ICLR*.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *ICML*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback. *Preprint*, arXiv:2112.09332.

OpenAI. 2022. Introducing ChatGPT. `https://openai.com/blog/chatgpt`. Accessed August 15, 2025.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 24 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Irina Saparina and Mirella Lapata. 2024. Ambrosia: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems*, volume 37, pages 90600–90628. Curran Associates, Inc.

Irina Saparina and Mirella Lapata. 2025. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16825–16839, Vienna, Austria. Association for Computational Linguistics.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Sergey Troshin, Vlad Niculae, and Antske Fokkens. 2025. On the low-rank parametrization of reward models for controlled language generation. In *TMLR*.

Ashwin K. Vijayakumar, Michael Cogswell, Ramprasaath R. Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. 2018. Diverse beam search for improved description of complex scenes. In *AAAI*.

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Passos, and Sumit Sanghai. 2023. Arithmetic sampling: parallel diverse decoding for large language models. In *ICML*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS*.

Yu Zhang, Xiusi Chen, Bowen Jin, Sheng Wang, Shuiwang Ji, Wei Wang, and Jiawei Han. 2024. A comprehensive survey of scientific large language models and their applications in scientific discovery. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8783–8817, Miami, Florida, USA. Association for Computational Linguistics.

Qinglin Zhu, Runcong Zhao, Hanqi Yan, Yulan He, Yudong Chen, and Lin Gui. 2025. Soft reasoning: Navigating solution spaces in large language models through controlled embedding exploration. In *ICML*.

A Prompts

Prompt for enumeration sampling.

```
Given the following problem, reason through
it and provide multiple different solutions:

Problem: {question}

Use exactly this format (no extra text):
<Solution 1> [Your reasoning should go here]
The answer is [Answer 1]. </Solution 1>
```

```
...
<Solution N> [Your reasoning should go here]
The answer is [Answer N]. </Solution N>
```

Prompt for parallel sampling.

```
Given the following problem, reason through
it and provide a solution:

Problem: {question}

You must wrap your reasoning and answer into
<Solution> ...reasoning here... 'The answer
is [numerical value].'</Solution> format.
```

Prompt for iterative sampling.

```
Given a problem and a set of solutions,
reason through it and provide a new solution.
The new solution may result in the same
answer, but it must be different from the
ones already provided.

Problem: {question}

Existing solutions:
{solutions}

Use exactly this format (no extra text):
<New Solution> [Your reasoning should go
here]. The answer is [answer]. </New
Solution>
```

B Averaged Distinct N-gram Diversity

Given a set of responses $S = \{y^{(i)}\}_{i=1}^{n}$, for $N \in \{1, \ldots, 5\}$, we calculate the averaged distinct N-gram diversity for each set as:

$$\text{avg. dist. N-gram}\,(S) = \sum_{N=1}^{5} \frac{|\text{set}(\text{N-gram}(R_C))|}{|\text{N-gram}(R_C)|}.$$

The diversity metric is calculated as the mean avg. distinct N-gram diversity over the sets of responses.

C Prompt for Computation Flow Parsing

```
You will receive a math question and a
free-form solution. Extract the sequence of
arithmetic steps from the solution and
output them one by one.

Rules:
- Output ONLY lines made of digits 0-9,
parentheses (), the operators + - * / ^, and
optionally "=" to show each step's result.
- No words, units, currency symbols, or
extra text.
- One step per line, in the order implied by
the solution.
- Convert verbal quantities to numbers.
Replace references like "the remainder" with
the actual numeric value.
- Keep only the steps that lead to the final
answer.
- If no computable arithmetic appears,
output an empty line.

Example:

Question: Janet lays 16 eggs a day. She eats
3, uses 4 for baking, and sells the rest for
$2 each. How much money does she make?
Solution: Janet lays 16 eggs per day. She
eats 3 and uses 4 for baking, so 16 - 7 = 9
eggs left. She sells them at $2 each → 9 * 2
= $18.
Output:
3 + 4 = 7
16 - 7 = 9
9 * 2 = 18

Now, extract the arithmetic steps from the
following:

Question: {question}
Solution: {solution}
Output:
```

Learning to vary: Teaching LMs to reproduce human linguistic variability in next-word prediction

Tobias Groot
University of Amsterdam
tobias.groot@student.uva.nl

Salo Lacunes
University of Amsterdam
salo.lacunes@student.uva.nl

Evgenia Ilia
University of Amsterdam
e.ilia@uva.nl

Abstract

Natural language generation (NLG) tasks are often subject to inherent variability; *e.g.* predicting the next word given a context has multiple valid responses, evident when asking multiple humans to complete the task. While having language models (LMs) that are aligned pluralistically, so that they are able to reproduce well the inherent diversity in perspectives of an entire population of interest is clearly beneficial, Ilia and Aziz (2024) show that LMs do not reproduce this type of linguistic variability well. They speculate this inability might stem from the lack of consistent training of LMs with data reflecting this type of inherent variability. As such, we investigate whether training LMs on multiple plausible word continuations per context can improve their ability to reproduce human linguistic variability for next-word prediction. We employ fine-tuning techniques for pre-trained and instruction-tuned models; and demonstrate their potential when fine-tuning GPT-2 and Mistral-7B-IT, using Provo Corpus. Our evaluation, which measures divergence among empirically estimated human and model next-word distributions across contexts before and after fine-tuning, shows that our multi-label fine-tuning improves the LMs' ability to reproduce linguistic variability; both for contexts that admit higher and lower variability.

1 Introduction

Inherent variability in natural language generation (NLG) tasks might arise from ambiguity or varying perspectives (Plank, 2022; Baan et al., 2023). For example, when predicting the next word given a context, multiple plausible and valid continuations exist; a task whose linguistic variability we can appreciate by asking a human population to complete it (Luke and Christianson, 2018). We can also appreciate this type of linguistic variability for autoregressive language models (LMs) that generate text by sampling from next-token (*i.e.* subword unit) distributions conditioned on preceding tokens (Vaswani et al., 2017). We achieve that by viewing such distributions as a representation of the model's uncertainty over continuations given a prefix (Ilia and Aziz, 2024; Guo et al., 2024; Tevet and Berant, 2020). It is often valuable for models to reproduce such variability, particularly in open-ended NLG tasks, where multiple responses can be plausible. Whereas this variability contributes to making LMs more robust (Sheng et al., 2008; Peterson et al., 2019; Uma et al., 2021; Kurniawan et al., 2025) and more representative of the linguistic diversity of human populations of interest (Sorensen et al., 2024; Muscato et al., 2025b), it has been shown that the variability LMs exhibit does not always align with the one humans exhibit (Pavlick and Kwiatkowski, 2019; Ma et al., 2025; Shaib et al., 2024). For next word prediction, Ilia and Aziz (2024) identify this misalignment and speculate it might stem from inconsistent exposure of LMs to training data reflecting such variability.

As such, we investigate whether training LMs with multiple observations of the next word per context will improve their ability to reproduce human variability. While previous fine-tuning work utilising multiple references per instance focused on classification tasks (Peterson et al., 2019; Uma et al., 2021; Rajeswar et al., 2022), our work focuses on next-word prediction, a generative task. Similar to Eisape et al. (2020), who employ a form of multi-label distillation in next word prediction, we also employ a technique to fine-tune pre-trained LMs and extend to instruction-tuned LMs. For the former, we alter the training signal, and for the latter we exploit a training data augmentation method to ensure that variability is observed.

We employ these fine-tuning techniques for GPT-2 (Radford et al., 2019), a pre-trained model, and Mistral-7B-IT (Jiang et al., 2023), an instruction-tuned model. When evaluating, by measuring divergence among empirically estimated human and model next-word distributions across contexts, be-

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 73–88
November 9, 2025 ©2025 Association for Computational Linguistics

fore and after fine-tuning, we find that fine-tuning with multiple labels per instance improves those LMs' ability to reproduce linguistic variability, across contexts of varying open-endedness. Additional ablations measure performance when varying the number of training labels per instance; and with a preliminary analysis we measure the trade-off in performance in tasks that admit no plausible variability. For that, we handcraft a small evaluation dataset using a knowledge-based question answering dataset (Berant et al., 2013).[1]

2 Related Work

Human label variation in natural language processing (NLP) tasks is often dismissed as noise (Paun et al., 2022; Ferracane et al., 2021). However, multiple responses can be plausible, especially relevant to ambiguous or open-ended tasks or prompts (Plank, 2022; Baan et al., 2023; Weber-Genzel et al., 2024; Nie et al., 2020; Aroyo and Welty, 2015). Embracing this plausible variation as part of NLP systems, which could make them more fair (Deng et al., 2023; Muscato et al., 2025b) and robust (Peterson et al., 2019; Sheng et al., 2008), involves altering all stages of our systems' development pipelines: from dataset creation, collecting multiple labels per prompt (Luke and Christianson, 2018; Nie et al., 2020, *i.a.*), to training, utilising these labels during the learning phase (Rodríguez-Barroso et al., 2024; Aroyo and Welty, 2012; Padmakumar et al., 2024, *i.a.*), and evaluation, comparing models' responses to multiple human references (Baan et al., 2022; Ilia and Aziz, 2024, *i.a.*).

Our approach aims to embrace plausible variability during training. Rather than collapsing annotations into a single ground truth (Paun et al., 2022), we incorporate multiple plausible references. The idea of multi-label fine-tuning has been adopted in image-classification (Peterson et al., 2019; Aurpa et al., 2024; Rajeswar et al., 2022), as well as in NLP, primarily for classification (Uma et al., 2021; Jung et al., 2023; He and Xia, 2018; Beţianu et al., 2024; Li et al., 2024; Zhang et al., 2024a; Li et al., 2025; Muscato et al., 2025a). Additionally, recent efforts have applied instruction fine-tuning for multi-label text classification tasks (Siddiqui et al., 2024; Yin et al., 2024) and tasks with restricted outcome spaces, such as sampling from discrete distributions (Zhang et al., 2024b). Our work focuses on a generative task, (*i.e.*, that of predicting complete wordforms by stringing together tokens), with a countably infinite outcome space (*i.e.*, all possible wordforms from a finite set of tokens). Eisape et al. (2020) explores a form of multi-label distillation in next-word prediction for an LSTM model. We also explore a form of multi-label distillation for transformer-based models, extending our investigation to instruction tuned LMs.

3 Methodology

We exploit simple yet intuitive fine-tuning techniques, depending on the LMs' previous training. These require a set of contexts $C = \{c_1, ..., c_N\}$, where for each context c_i, we have a set of human next-word references $W_i = \{w_{i1}, ..., w_{iM}\}$[2]:

Fine-tuning pre-trained LMs Autoregressive LMs are trained using cross-entropy between a target and the model's next-token distribution given context c ($p(\cdot|c)$ and $q(\cdot|c)$ resp.). This corresponds to searching for the maximum likelihood estimate (MLE). When training on a corpus with a single continuation (*i.e.* the next corpus token w^*), p is a deterministic distribution centered on w^*, leading to the following loss:

$$L_{\text{Label}} = -\log q(w^* \mid c). \quad (1)$$

When multiple word continuations are available, we replace this deterministic distribution with the empirically estimated distribution (using W_i), where the probability of a word given c_i, $p(w|c_i)$, equals its relative frequency in W_i. This results in the following loss, which comprises generalized cross entropy (Jurafsky and Martin, 2025):

$$L_{\text{Var}} = -\sum_{w \in \mathcal{V}} p(w \mid c_i) \log q(w \mid c_i), \quad (2)$$

where $\mathcal{V}$ is the vocabulary.[3] Since words may consist of multiple tokens, to obtain $q(w \mid c_i)$ we re-express the model's token-level probabilities over complete words.[4] For a word w with tokenization $\tau(w) = (t_1, \ldots, t_n)$, we compute:

$$q(w \mid c_i) = \prod_{j=1}^{n} q(t_j \mid c_i, t_1, \ldots, t_{j-1}), \quad (3)$$

where $q(t_j \mid \cdot)$ is the probability of token t_j under the model, given the context and preceding tokens.

[1] Code available at: GitHub repository

[2] M might vary accross contexts.

[3] Words that actually contribute to the loss, i.e. non-zero terms, are words in the set of human samples, W_i for c_i.

[4] Humans predicted *word* continuations, not tokens; so the outcome space of $p(w \mid c_i)$ is over complete words, and we must ensure that $q(w \mid c_i)$ is expressed over the same space.

Fine-tuning instruction-tuned LMs Instruction-tuned models underwent additional training to cater a rather conversational format and adhere to task instructions. We sample responses from the model's conditional predictive distribution (CPD) given a prompt, *i.e.* an instruction and an example. For our task, we sample response r containing a predicted word from $q(r|(I, c_i))$, where the prompt includes instruction I requesting a word continuation given a prefix, and the example context c_i. So as to utilise multiple labels, we employ the following training data augmentation technique: for each context c_i in C, we construct the prompt (I, c_i) and for each word w_j in W_i, we create a training datapoint where w_j is a response to (I, c_i). This entails that c_i will appear multiple times with different continuations as per their frequency in W_i. We train using L_{Label}.[5] See Appendx A for prompts.

4 Experiments

Models & Datasets. We fine-tune pre-trained GPT-2 (124M; Radford et al. (2019)) and instruction-tuned Mistral-7B-Instruct-v0.3 (7.25B; Jiang et al. (2023)), which we refer to as Mistral-7B-IT. Both models are fine-tuned using Provo Corpus (Luke and Christianson, 2018), which contains 55 text passages (2687 total contexts). Each prefix is annotated with an average of 40 human annotations predicting the word following it. We split the dataset randomly at the paragraph level (to avoid partial passage leaks between train and test sets). 80% is for training, of which 10% is reserved for validation; and the remaining for testing.

Training Configuration. Both models were fine-tuned using the Adam optimizer (Kingma, 2014). For GPT-2: we train for 3 epochs, using a learning rate of $1e^{-5}$ and a batch size of 16. For Mistral-7b-IT: we train for 4 epochs using Low-Rank Adaptation (Dettmers et al., 2023, LoRA) with a learning rate of $1e^{-4}$ with a batch size of 32. We train on 3 random seeds; training details in Appendix B.

Metrics. Following Ilia and Aziz (2024): for each context, we measure the divergence between the human and model CPDs given a context using total variation distance (TVD) (Rudin, 1987).[6] TVD quantifies the difference between two probability distributions by summing the absolute differences in the probabilities they assign to the same event. A higher TVD indicates greater disagreement between human and model CPDs (*i.e.*, poorer alignment with human linguistic variability), whereas a lower TVD indicates less disagreement (*i.e.* better alignment with human linguistic variability). In order to compute TVD, we need estimates of the human and model CPDs ($p(w|c)$ and $q(w|c)$ respectively). As done in Ilia and Aziz (2024): (1) for $p(w|c)$, we estimate it via Monte Carlo, with $p(w|c)$ equaling the relative frequency of w in all human samples, and (2) for $q(w|c)$, we estimate it via Monte Carlo, by sampling 40 sequences from the model long enough to contain a full word, slice it, and compute $q(w|c)$ (or $q(w|(I, c))$) as the relative frequency of w in all sampled words.

Mean TVD ± SD (↓)		
Model	**GPT-2**	**Mistral-7b-IT**
Base	0.607±0.001	0.812±0.002
1-Shot	N/A	0.784±0.002
FT (Orig. corpus)	0.612±0.002	0.805±0.001
FT (Maj. label)	0.556±0.005	0.563±0.002
FT (Mul. labels)	**0.550**±0.003	**0.499**±0.006
Oracle	0.443±0.002	0.443±0.002

Table 1: Mean and standard deviation of TVD averages across test contexts for three seeds.

Baselines & Upper Bounds. We compare the distribution of TVD values across contexts before and after fine-tuning, where improved performance would mean a shift towards lower TVD values (*i.e.* less disagreement with human CPDs). For the instruction-tuned model, we add a 1-shot baseline, where the prompt includes an example of a context and word references (details in Appendix A). As another baseline, we fine tune models with Provo's original corpus passages (*i.e.* one continuation per prefix), imitating models' usual training. Lastly, to estimate the best performance we can expect from our models, which essentially is to mimic human divergence, we establish a baseline for the expected level of disagreement from humans for a context. We split human responses in two disjoint groups and measure their CPDs' TVD ('Oracle' baseline).

5 Results

Main results As shown in Table 1, both models fine-tuned with multiple labels (FT (Mul. labels)) achieve a notably lower mean TVD compared to other baselines (Base, 1-Shot and FT (Orig. Cor-

[5] Constructing the dataset in this way (one prompt-response pair for each word annotation for every context) using L_{Label} is similar to learning $q(r|(I, c))$ with L_{Var}.

[6] $\text{TVD}(p, q) = \frac{1}{2} \sum_w |p(w|c_i) - q(w|c_i)|$

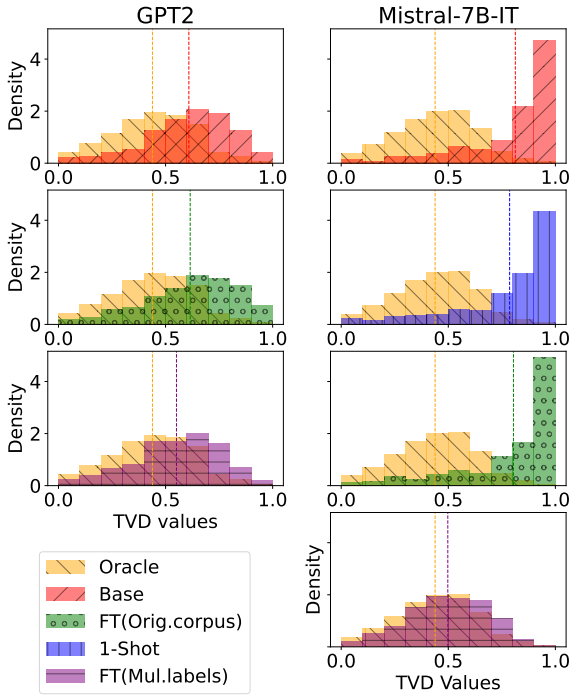


Figure 1: Distribution of TVD scores (for 1 seed) across contexts. For both GPT-2 and Mistral-7B-IT; fine-tuning shifts the TVD distribution towards the Oracle baseline, suggesting better linguistic alignment with humans.

pus)). We also observe how FT (Orig. corpus)'s performance is very similar to the Base model. This simultaneously indicates that our improved performance does not stem from an out-of-distribution effect between Provo Corpus and the models' training data. Figure 1, which shows the histogram of TVD values for all models and baselines (for 1 seed), confirms that; FT (Mul. labels) models' TVD distributions shift towards the Oracle distributions, indicating that models improve at reproducing human linguistic variability. For other seeds, we see similar patterns; see Appendix C.

When and how do models improve? To understand the effects of our fine-tuning, we analyze changes in TVD. We visualise the models' changes in performance against context open-endedness (as measured by the TVD between human oracles; lower TVD indicating more 'restrictive' contexts), allowing us to grasp if performance gains arise in contexts that admit higher or lower variability. In Figure 4 (Appendix D), negative TVD differences between fine-tuned and base models (indicating gains) occur at all levels of contexts' open-endedness. We also assess whether models improve at predicting words that humans predicted (regardless of frequency). We plot the fraction of unique human predictions that were also predicted by the models before and after fine-tuning (Figure 7; Appendix D). Fine-tuned Mistral-7B-IT's ability to predict unique human words (along with its CPDs' 'diversity'; Figure 6, Appendix D) improves substantially (details and analyses in Appendix D).

Is the entire response distribution useful? When gathering datasets with multiple labels, disagreement can be discarded as noise and the most common response is used as ground truth. Aiming to assess whether retaining the entire response distribution is useful, we fine-tune a model on Provo Corpus using only the majority response (FT (Maj. Label) in Table 1). We find that, FT (Maj. Label) surpasses the performance of FT (Orig. Corpus), which is not entirely surprising: the corpus word is a single observation, while the majority vote exploits in a sense multiple labels. This is intuitively in line with analysis revealing that performance gains seem to relate with less open-ended contexts (Figure 8; Appendix E). Nonetheless, FT (Mul. labels) outperforms FT (Maj. label), with moderate gains for GPT2 and more notable gains for Mistral-7B-IT; indicating the utility of retaining all labels.

Number of labels ablation. We analyse how the number of labels used to fine-tune the model affect the model's performance. We fine-tune GPT2 using a varying number of labels each time (1,2,4,16 and 32; randomly sampled from available annotations). Figure 9 of Appendix F shows that 16 samples are sufficient for substantial performance improvements; for more details, see Appendix F.

Impact on tasks without data uncertainty. Whereas optimising for a task that admits inherent variability (*i.e.* next-word prediction) might improve the model's ability to reproduce such variability; the effect of this on tasks that admit no variability is unclear. To assess that, we test the models' performance before and after fine-tuning on knowledge-based question answering (a task admitting no plausible variability), adapted for next-word prediction. For that, we handcraft examples from a subset of WebQuestions (Berant et al., 2013); details and examples in Appendix G. For each context, we sample 40 responses and measure how often responses exactly match the reference. As shown in Table 3 of Appendix G, fine-tuning on multi-label data moderately improves the low performance of GPT2, but worsens the performance

of Mistral-7B-IT; highlighting a potential trade-off in performance between tasks that do and do not admit variability, when optimising for the latter.

6 Conclusion

This study examines whether fine-tuning with multiple labels per instance has the potential to enhance models' ability to reproduce linguistic variability in next word prediction. We show improved performance for a smaller pre-trained language model (GPT-2) and a larger instruction-tuned model (Mistral-7b-IT) across contexts that admit varying levels of plausible variability. Our findings highlight both the potential and possible limitations of such fine-tuning, paving the way for further advancements in modeling linguistic variation.

7 Limitations

We hereby discuss various limitations of our study: we fine-tune using Provo Corpus, which is a relatively small dataset with a limited number of human annotations per prefix. The high cost of obtaining data with multiple references means that such data is scarce and not available at large scale. However, we show that even with a limited amount of contexts and a limited amount of annotations per context that are well-curated and of high-quality it is possible to observe performance improvements. Simultaneously, as the field of synthetic data generations is becoming increasingly popular; we can entertain the idea that future work exploits such synthetic labels, and a model that has been fine-tuned to embrace variability, such as the ones we present in this study, could comprise generators for such synthetic annotations. Additionally, for our training and evaluation, we assumed all human annotations to be draws from the same underlying distribution; which is not an assumption that is easy to verify. We also observed a trade-off between capturing variability well and performance on tasks with a single correct answer; with future work potentially focusing on methods that could balance-off better such trade-offs. Additionally, due to resource constraints, we were only able to include in our study only two (relatively small) models that were trained for English. Despite focusing on a generative task, we only focused on next word prediction. Transferring this to the sequence level might be non-trivial and come with its own challenges. However, we hope that our study inspires future work in this research direction, aiming to embrace inherent variability as part of the training of LMs, and tackle challenges related to this field.

Acknowledgements

Evgenia Ilia is supported by the EU's Horizon Europe research and innovation programme (grant agreement No. 101070631, UTTER). The experiments and findings presented in this paper were conducted as part of a research project fostered within the NLP 2 course of the MSc AI programme of the University of Amsterdam (2024-2025 edition), coordinated by Ana Lucic.

References

Lora Aroyo and Chris Welty. 2012. Harnessing disagreement for event semantics. *Detection, Representation, and Exploitation of Events in the Semantic Web*, 31.

Lora Aroyo and Chris Welty. 2015. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24.

Tanjim Taharat Aurpa, Md Shoaib Ahmed, Md Mahbubur Rahman, and Md Golam Moazzam. 2024. Instructnet: A novel approach for multi-label instruction classification through advanced deep learning. *Plos one*, 19(10):e0311161.

Joris Baan, Wilker Aziz, Barbara Plank, and Raquel Fernandez. 2022. Stop measuring calibration when humans disagree. *arXiv preprint arXiv:2210.16133*.

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. *arXiv preprint arXiv:2307.15703*.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1533–1544, Seattle, Washington, USA. Association for Computational Linguistics.

Miruna Bețianu, Abele Mălan, Marco Aldinucci, Robert Birke, and Lydia Chen. 2024. Dallmi: Domain adaption for llm-based multi-label classifier. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 277–289. Springer.

Naihao Deng, Xinliang Frederick Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. 2023. You are what you annotate: Towards better models through annotator representations. *arXiv preprint arXiv:2305.14663*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Preprint*, arXiv:2305.14314.

Tiwalayo Eisape, Noga Zaslavsky, and Roger Levy. 2020. Cloze distillation: Improving neural language models with human next-word prediction. Association for Computational Linguistics (ACL).

Elisa Ferracane, Greg Durrett, Junyi Jessy Li, and Katrin Erk. 2021. Did they answer? subjective acts and intents in conversational discourse. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1626–1644.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. 2024. Benchmarking linguistic diversity of large language models. *arXiv preprint arXiv:2412.10271*.

Huihui He and Rui Xia. 2018. Joint binary neural network for multi-label learning with applications to emotion classification. In *Natural Language Processing and Chinese Computing: 7th CCF International Conference, NLPCC 2018, Hohhot, China, August 26–30, 2018, Proceedings, Part I 7*, pages 250–259. Springer.

Evgenia Ilia and Wilker Aziz. 2024. Predict the next word: Humans exhibit uncertainty in this task and language models _. *arXiv preprint arXiv:2402.17527*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Taehee Jung, Joo-Kyung Kim, Sungjin Lee, and Dongyeop Kang. 2023. Cluster-guided label generation in extreme multi-label classification. *arXiv preprint arXiv:2302.09150*.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models (Chapter 10)*, 3rd edition. Online manuscript released January 12, 2025.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kemal Kurniawan, Meladel Mistica, Timothy Baldwin, and Jey Han Lau. 2025. Training and evaluating with human label variation: An empirical study. *arXiv preprint arXiv:2502.01891*.

Cheng Li, Mengzhuo Chen, Jindong Wang, Sunayana Sitaram, and Xing Xie. 2024. Culturellm: Incorporating cultural differences into large language models. *Advances in Neural Information Processing Systems*, 37:84799–84838.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025. Preserving diversity in supervised fine-tuning of large language models. *Preprint*, arXiv:2408.16673.

Steven G Luke and Kiel Christianson. 2018. The provo corpus: A large eye-tracking corpus with predictability norms. *Behavior research methods*, 50:826–833.

Marcus Ma, Georgios Chochlakis, Niyantha Maruthu Pandiyan, Jesse Thomason, and Shrikanth Narayanan. 2025. Large language models do multi-label classification differently. *arXiv preprint arXiv:2505.17510*.

Benedetta Muscato, Praveen Bushipaka, Gizem Gezici, Lucia Passaro, Fosca Giannotti, and Tommaso Cucinotta. 2025a. Embracing diversity: A multi-perspective approach with soft labels. *arXiv preprint arXiv:2503.00489*.

Benedetta Muscato, Lucia Passaro, Gizem Gezici, and Fosca Giannotti. 2025b. Perspectives in play: A multi-perspective approach for more inclusive nlp systems. *arXiv preprint arXiv:2506.20209*.

Yixin Nie, Xiang Zhou, and Mohit Bansal. 2020. What can we learn from collective human opinions on natural language inference data? *arXiv preprint arXiv:2010.03532*.

Vishakh Padmakumar, Chuanyang Jin, Hannah Rose Kirk, and He He. 2024. Beyond the binary: Capturing diverse preferences with reward regularization. *arXiv preprint arXiv:2412.03822*.

Silviu Paun, Ron Artstein, and Massimo Poesio. 2022. *Statistical methods for annotation analysis*. Springer Nature.

Ellie Pavlick and Tom Kwiatkowski. 2019. Inherent disagreements in human textual inferences. *Transactions of the Association for Computational Linguistics*, 7:677–694.

Joshua C Peterson, Ruairidh M Battleday, Thomas L Griffiths, and Olga Russakovsky. 2019. Human uncertainty makes classification more robust. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9617–9626.

Barbara Plank. 2022. The'problem'of human label variation: On ground truth in data, modeling and evaluation. *arXiv preprint arXiv:2211.02570*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sai Rajeswar, Pau Rodriguez, Soumye Singhal, David Vazquez, and Aaron Courville. 2022. Multi-label iterated learning for image classification with label ambiguity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4783–4793.

Nuria Rodríguez-Barroso, Eugenio Martínez Cámara, Jose Camacho Collados, M Victoria Luzón, and Francisco Herrera. 2024. Federated learning for exploiting annotators' disagreements in natural language processing. *Transactions of the Association for Computational Linguistics*, 12:630–648.

Walter Rudin. 1987. *Real and complex analysis*. McGraw-Hill, Inc.

Chantal Shaib, Yanai Elazar, Junyi Jessy Li, and Byron C Wallace. 2024. Detection and measurement of syntactic templates in generated text. *arXiv preprint arXiv:2407.00211*.

Victor S Sheng, Foster Provost, and Panagiotis G Ipeirotis. 2008. Get another label? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 614–622.

Muhammad Hammad Fahim Siddiqui, Diana Inkpen, and Alexander Gelbukh. 2024. Instruction tuning of llms for multi-label emotionclassification in social media content. In *Proceedings of the Canadian Conference on Artificial Intelligence*.

Taylor Sorensen, Jared Moore, Jillian Fisher, Mitchell L Gordon, Niloofar Mireshghallah, Christopher Michael Rytting, Andre Ye, Liwei Jiang, Ximing Lu, Nouha Dziri, and 1 others. 2024. A roadmap to pluralistic alignment. *CoRR*.

Guy Tevet and Jonathan Berant. 2020. Evaluating the evaluation of diversity in natural language generation. *arXiv preprint arXiv:2004.02990*.

Alexandra N Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. 2021. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72:1385–1470.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Leon Weber-Genzel, Siyao Peng, Marie-Catherine de Marneffe, and Barbara Plank. 2024. Varierr nli: Separating annotation error from human label variation. *Preprint*, arXiv:2403.01931.

Kai Yin, Chengkai Liu, Ali Mostafavi, and Xia Hu. 2024. Crisissense-llm: Instruction fine-tuned large language model for multi-label social media text classification in disaster informatics. *arXiv preprint arXiv:2406.15477*.

Jinghui Zhang, Dandan Qiao, Mochen Yang, and Qiang Wei. 2024a. Regurgitative training: The value of real data in training large language models. *arXiv preprint arXiv:2407.12835*.

Yiming Zhang, Avi Schwarzschild, Nicholas Carlini, J Zico Kolter, and Daphne Ippolito. 2024b. Forcing diffuse distributions out of language models. In *First Conference on Language Modeling*.

A Prompts for Baselines

When constructing the training set and evaluating our models, we present the relevant prompts:

Base prompt. To assess the performance of the non fine-tuned models, we prompt them repeatedly for the next word prediction task. The prompt includes an instruction to predict a next-word continuation and the given context at a time.

Prompt:

```
Instruction: Return one plausible next
 word for the following context.
 Context: <CONTEXT>
 Continuation:
```

When creating training prompt-response pairs, the prompt is identical to before, and the responses are words from the set of human references.

Response:

```
<HUMAN_REFERENCE>
```

1-Shot prompt. As a performance baseline we have one-shot prompting, which includes the instruction, an example from the training set, and the given context at a time:

```
Instruction: This is an example of a
 context and some plausible next word
continuations. given by a group of 39
 people: Context: There are now
 rumblings that, Continuations:
 [are, are, are, are, are, are, are,
 can, can, can, can, can, sound,
 sound, sound, sound, shake, shake,
 shake, the, the, the, have, have,
 our, our, someone, someone,
 appear, ca, cause, come, make,
 occur, people, say, suggest,
 tumble, we]. Following this
 example, return only one plausible
  next word for the following context.
 Context: <context>
 Continuation:
```

B QLoRa

Table 2 shows the configuration used for finetuning the Mistral-7B-IT model.

Parameter	Value
QLoRA	
r	8
LoRA α	16
LoRA dropout	0.05
Task type	Causal Language Modeling
Target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`
Quantization	
Load in 4-bit	True
4-bit quantization type	nf4
Double quantization	True
Compute data type	bfloat16

Table 2: LoRA and 4-bit quantization configuration parameters.

C Main results

We present Figure 1, which comprises the results on the test set for one of the three random seeds we trained on. We observe similar trends for the remaining seeds; which we present in Figure 2. This is confirmed when plotting the differences between the TVD of the model and human CPDs and the TVD among the human oracle CPDs, as observed in Figure 3.

D Analysis of model performance changes

In order to understand how fine tuning has affected the model performance. We perform various analyses. We visualise the models' changes in performance against context open-endedness. We approximate that using the TVD between human oracles. We assume that a lower TVD, reflecting lower disagreement among human populations, indicates more 'restrictive' contexts, while a higher TVD, indicates contexts that admit a higher level of plausible variability. We plot changes in performance by computing the differences between the TVD of the fine tuned model and human CPD and the TVD of the non fine tuned model and human CPD. Results are shown in Figure 4 (showing all contexts) and Figure 5 (showing only contexts for which performance improved, *i.e.* negative differences in TVD values). We observe how improvements occur across contexts of varying open-endedness (*i.e.* varying TVD among oracles values).

To gain further insight as to how fine tuning has affected our models, we plot the entropy values of the empirically estimated model CPDs across contexts before and after fine-tuning. Results can be seen in Figure 6. For GPT2, we observe how the entorpy of the model's empirically estimated CPDs were not impacted very substantially. We observe only a slight shift towards lower entropy values (*i.e.* peakier distributions); which means that model predictions might be slightly more confident, while also being better better aligned with human linguistic variability. On the contrary, the fine tuned Mistral-7B-IT model's entropy values shift substantially towards higher values, demonstrating that now the model is making more diverse predictions (which are also better aligned with human linguistic variability, as evident by our main findings).

Lastly, we assess whether models improve at predicting words that humans predicted (regardless of their frequency), as a means to approximate how wel their lexical diversity aligns with that of our assessed human population. We plot the fraction of unique human predictions that were also predicted by the models before and after fine-tuning with multiple labels (Figure 7). Higher values indicating a more highly aligned lexical diversity. We find that GPT2's lexical diversity remained relatively similar to before fine tuning, but for Mistral-7B-IT we see a clear rightward shift in the distribution of unique word coverage for the fine-tuned model. This indicates that the fine-tuned model predicts a greater number of relevant unique words per context compared to the non-fine-tuned baseline.

E Analysis of model fine-tuned with majority label

Similar to Appendix D, we analyse the changes in performance of the model fine tuned with the majority label compared to the base model. We visualise the models' (FT (Maj.Label)) changes in performance against context open-endedness. We approximate that using the TVD between human oracles. We assume that a lower TVD, reflecting lower disagreement among human populations, indicates more 'restrictive' contexts, while a higher TVD, indicates contexts that admit a higher level of plausible variability. We plot changes in performance by computing the differences between the TVD of the fine tuned model (FT Maj. label) and human CPD and the TVD of the base model and human CPD. Results are shown in Figure 8. When comparing with the corresponding plots for FT (Mul.label) in Figure 4, we observe how improvements occur for contexts that admit lower plausible variabiltiy (*i.e.* lower TVD among oracles values; steeper regression line towards lower Oracle TVD values for lower negative differences/performance

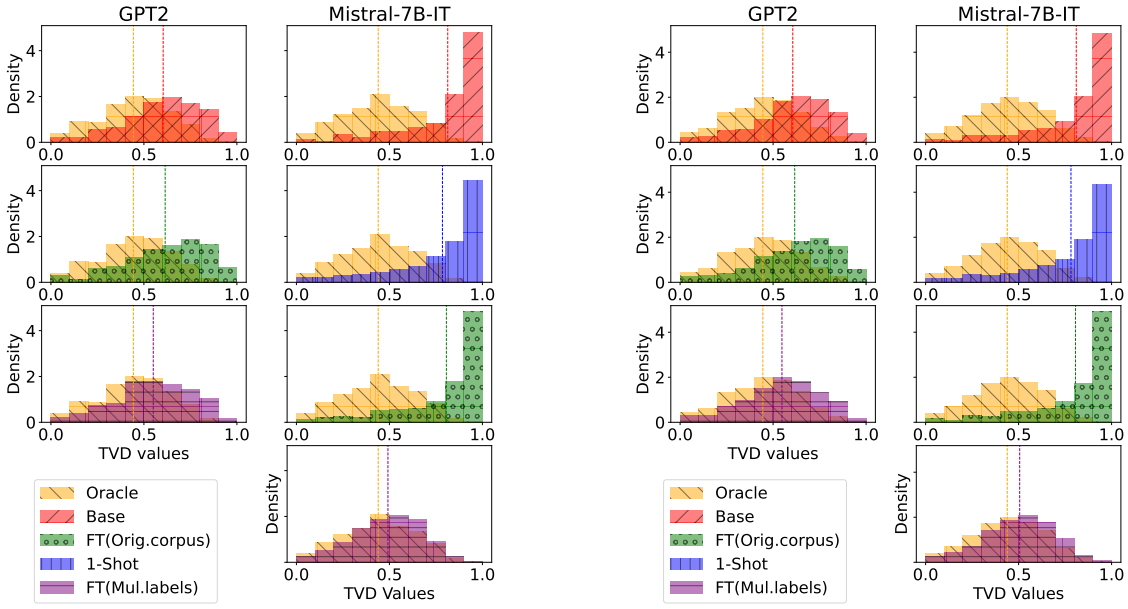


Figure 2: Distribution of TVD scores across contexts, for the two remaining seeds not presented in the main paper. For both GPT-2 and Mistral-7B-IT; fine-tuning shifts the TVD distribution toward the Oracle baseline, suggesting improved alignment with human linguistic variability.

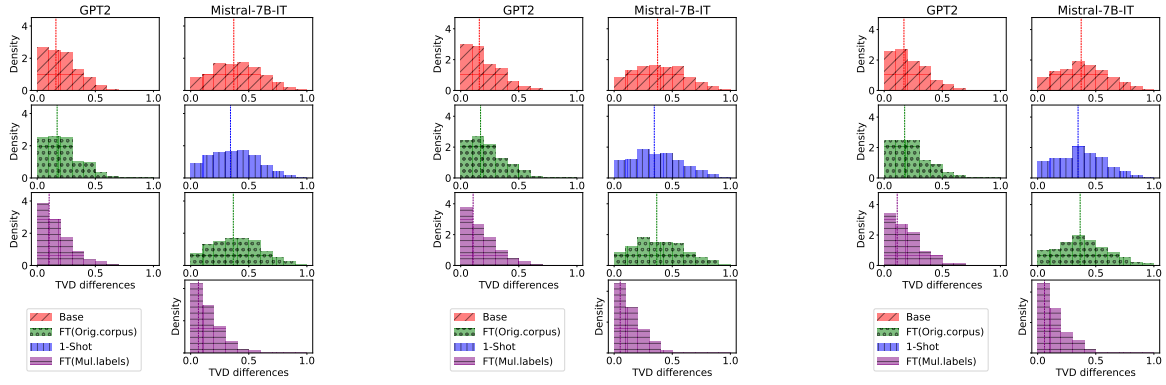


Figure 3: Distribution of differences of TVD scores between the model and the human CPDs and the oracle CPDs, for all 3 seeds. For both GPT-2 and Mistral-7B-IT; fine-tuning shifts the TVD distribution towards smaller differences, confirming previous findings.

gains).

F Varying training labels per instance Study

We perform an ablation to understand the number of labels that is necessary to obtain substantial performance gains. We perform this ablation study only for GPT2, given computational constrains (Mistral-7B-IT is a much larger model, and fine-tuning it repeatedly is computationally prohobited). We sample 1,2,4,16 and 32 labels given our available annotations and fine tune GPT2 given the subsequent training sets. We then perform the same evaluation as for the rest of our analysis and present the average TVD of the test set, against the label set size per instance in Figure 9. Scores for 16 and 32 samples are nearly identical, and very similar to the score obtained when training on all available labels (40 on average per prompt). These results suggest that around 16 labels per instance

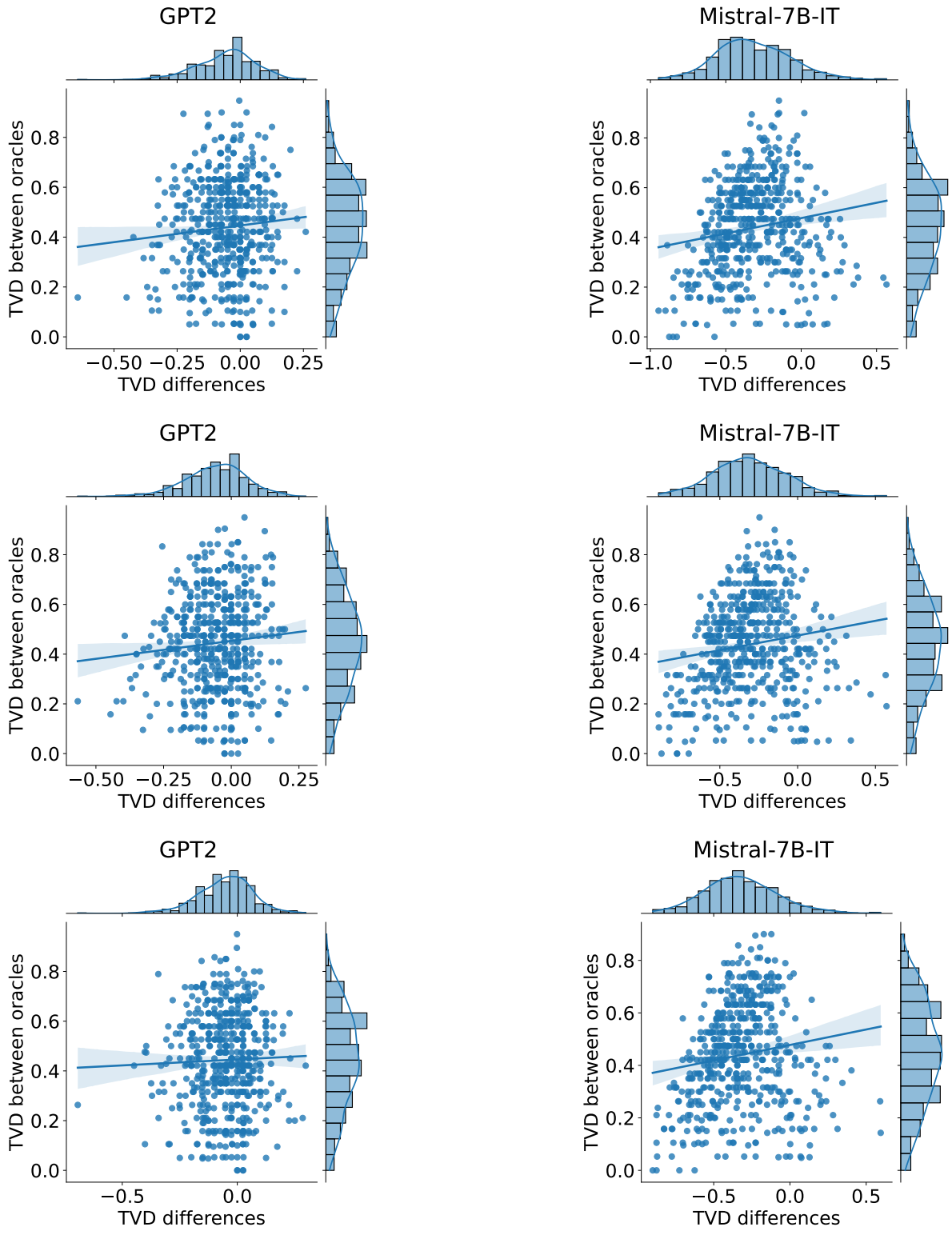


Figure 4: Distribution of differences of TVD scores between the fine tuned model and the human CPDs minus the TVD of the non fine tuned model and the human CPDs, against TVD among oracles. Performance gains (negative differences) for both models occur across contexts of varying open-endedness (with lower TVD indicating more 'restricted' contexts).

are sufficient to observe significant performance gains.

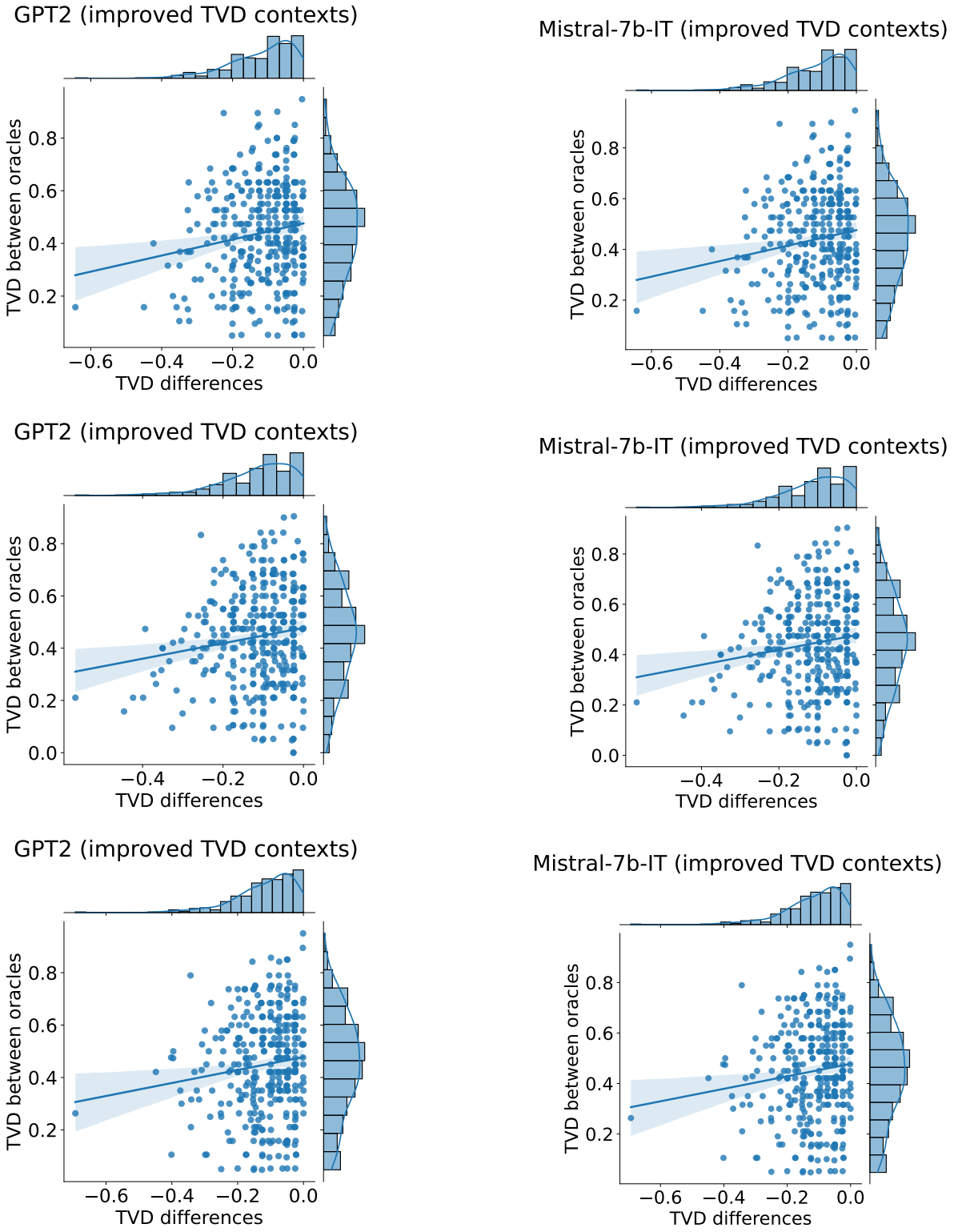
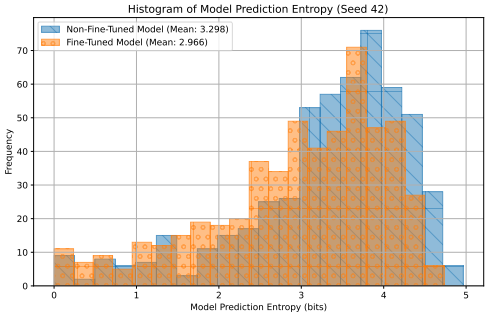
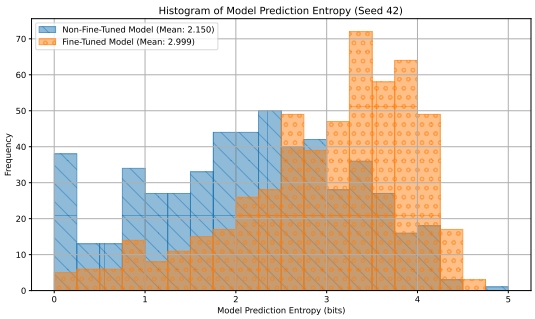


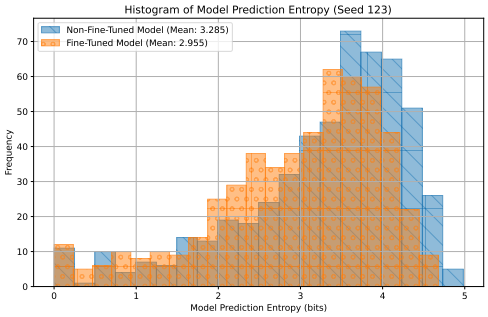
Figure 5: Distribution of differences of TVD scores between the fine tuned model and the human CPDs minus the TVD of the non fine tuned model and the human CPDs, against TVD among oracles. In this case, we only plot datapoints for which we observed improvements (*i.e.* negative differences) for both models. Similarly, we observe that gains occur across contexts of varying open-endedness (with lower TVD indicating more 'restricted' contexts).



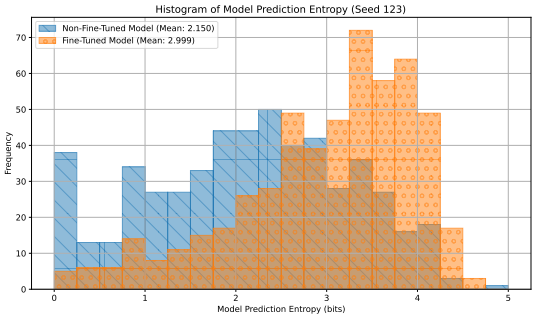
(a) GPT-2 (Seed 42)



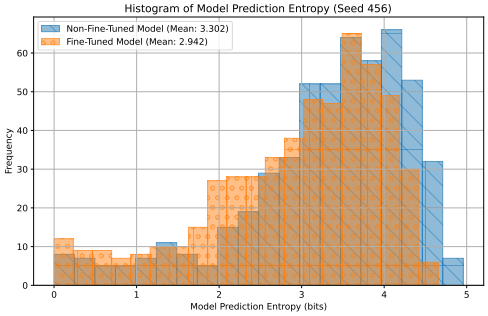
(b) Mistral-7B (Seed 42)



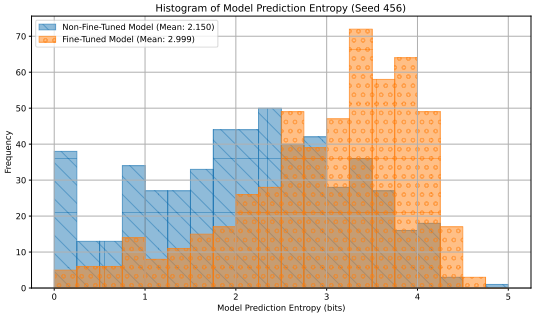
(c) GPT-2 (Seed 123)



(d) Mistral-7B (Seed 123)

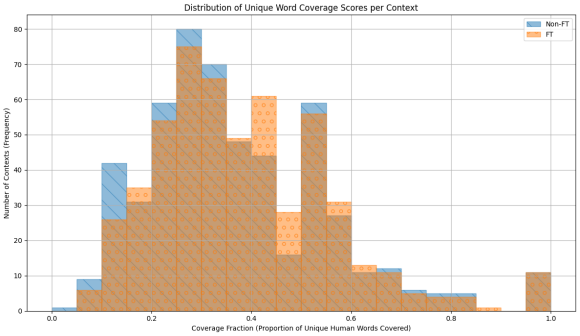


(e) GPT-2 (Seed 456)

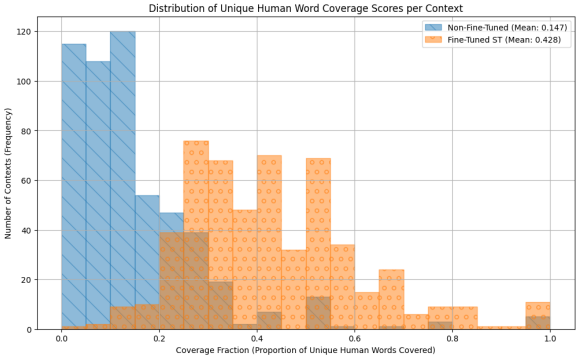


(f) Mistral-7B (Seed 456)

Figure 6: Entropy of model predictions before an after finetuning.



(a) GPT-2



(b) Mistral-7B

Figure 7: Unique word coverage across models. Fine-tuning with multiple labels per instance increases lexical diversity more compared to hard-targets (majority vote).

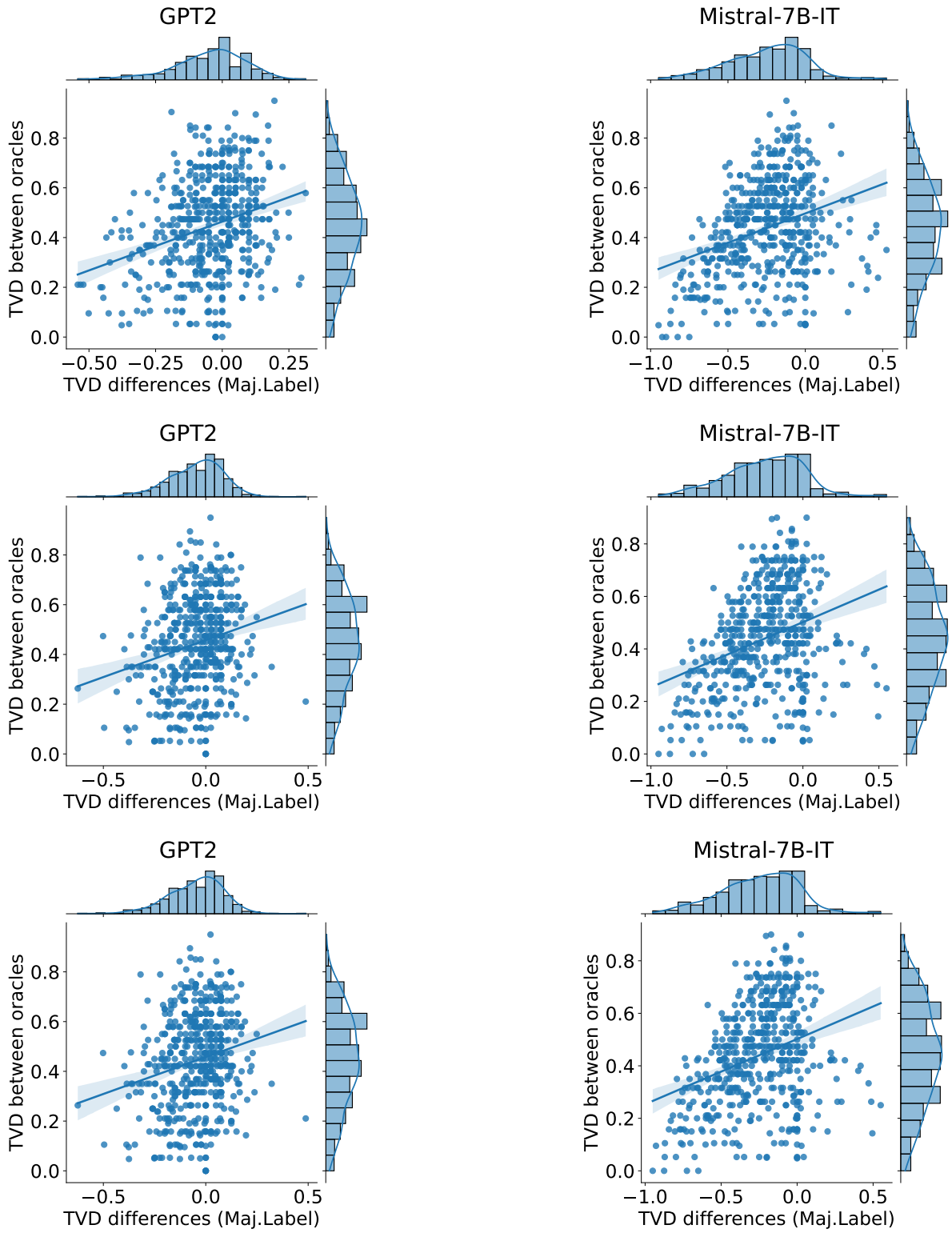


Figure 8: Distribution of differences of TVD scores between the fine tuned model and the human CPDs minus the TVD of the non fine tuned model and the human CPDs, against TVD among oracles. Performance gains (negative differences) for both models occur across contexts of varying open-endedness (with lower TVD indicating more 'restricted' contexts).

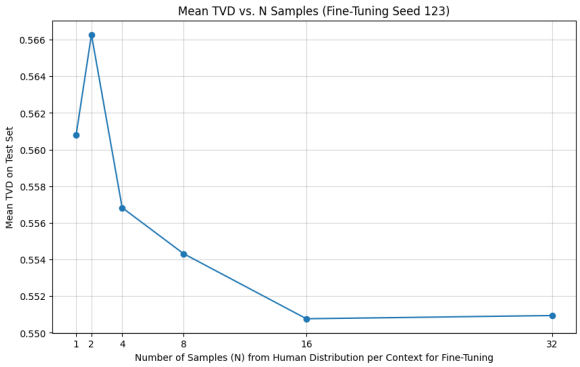


Figure 9: Mean TVD by number of samples per context. Performance improves with more samples, plateauing after 16.

G Analysis for QA task without variability

Whereas optimising for a task that admits inherent variability (*i.e.* next-word prediction) might improve the model's ability to reproduce such variability better; the effect of this on tasks that admit no variability is unclear. We test the models' performance on knowledge-based question answering (which is a task that admits no plausible variability), adapted as a next-word prediction task. We create a small evaluation dataset based on a knoweldge-based question answering dataset, WebQuestions (Berant et al., 2013). We create a subset of 55 handpicked contexts, chosen to include a variety of topics ranging from science, history and pop culture, each rephrased into next-word prediction tasks. We demonstrate 3 randomly chosen examples below:

Prompt:

```
Instruction: Return one plausible next
 word for the following context.
Context: The  first country to invade
 poland in ww2 was
 Continuation:
```

Target:

```
Germany
```

Prompt:

```
Instruction: Return one plausible next
 word for the following context.
 Context: the organelle responsible
for atp production and storage is the
 Continuation:
```

Target:

```
mitochondrion
```

Prompt:

```
Instruction: Return one plausible next
 word for the following context.
 Context: darth vader's star
 destroyer was called
 Continuation:
```

Target:

```
Devastor
```

We also evaluate model performance using the original questions. For Mistral-7B-IT, the instruction was modified into: **QA-Prompt:**

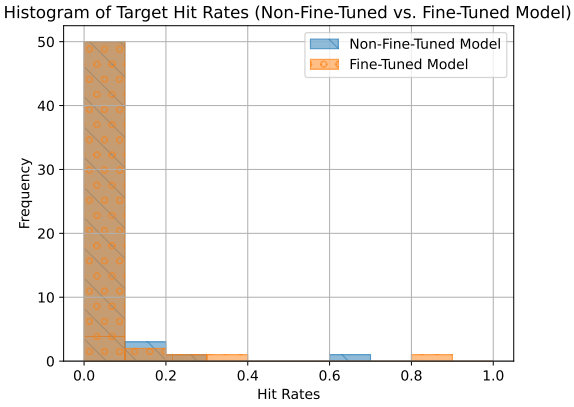
```
Instruction: Answer the following
question with one word only
Context: What country first invaded
poland in ww2?
Continuation:
```

Mean Hit Rate ± SD		
Model	**GPT-2**	**Mistral-7B**
Base	0.032 ± 0.002	0.590±0.005
FT (Orig.corpus)	0.030 ± 0.002	0.229±0.005
FT (Mul.labels)	0.041 ± 0.002	0.127±0.005

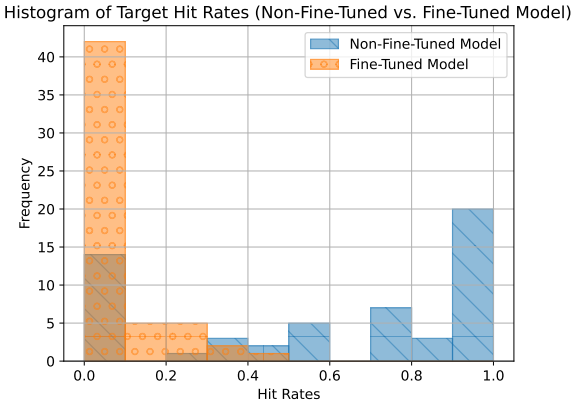
Table 3: Mean target hit rate for 40 samples per context across three seeds with standard deviation, for both GPT-2 and Mistral-7B.

We compare the base model, the model fine tuned with the original corpus (so as to account for the impact of training on Provo corpus, a potentially different domain) and the model that was fine tuned with multiple labels in their ability to generate the correct answer to the question (phrased as a next-word prediction task). To evaluate the performance, for each context, we sample 40 responses and measure how often responses exactly match the reference, denoted as hit rate.

Table 3 shows the results of this evaluation. GPT-2 shows a slight increase in hit rate after finetuning, although its overall performance remains poor, and Mistral-7B-IT's performance also drops, more substantially. However, we cannot rule out the effect of other confounders in the data or optimisation process that might have incidentally impacted the performance changes and are not relevant to the multiplicity of responses. Hence, we approach these preliminary results with cautiousness, and hope to inspire future work that investigates this more extensively. Supplementary histograms of hit-rates across contexts can be seen in Figure 12.

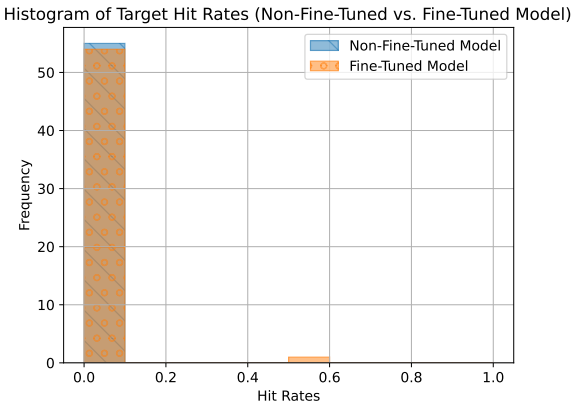


(a) GPT-2

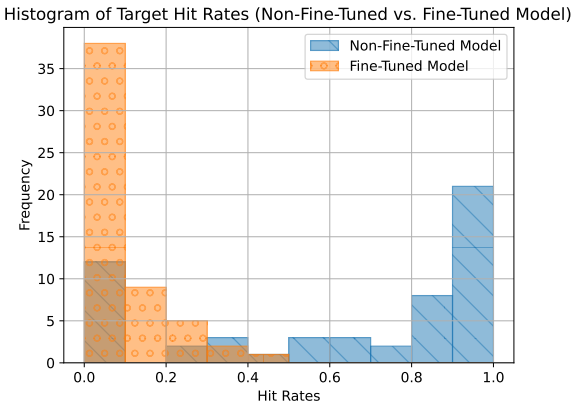


(b) Mistral-7B

Figure 10: Hit rates on gold target label before and after finetuning. Averaged across 3 seeds.

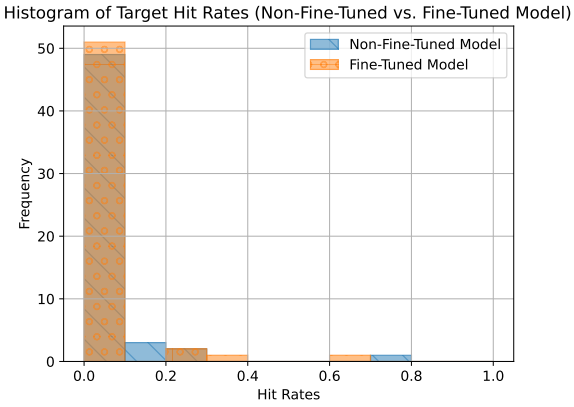


(a) GPT-2

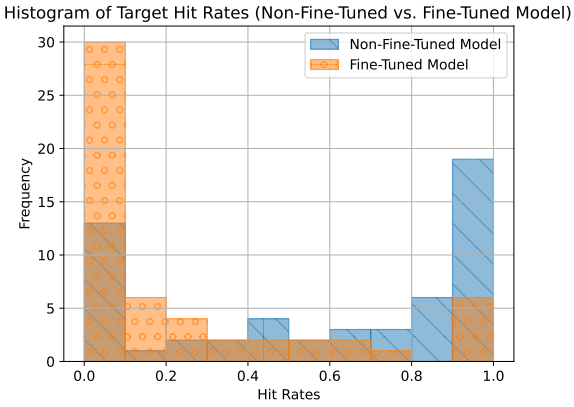


(b) Mistral-7B

Figure 11: Hit rates on gold target label when prompted in the original QA format, before and after finetuning. Averaged across 3 seeds.



(a) GPT-2



(b) Mistral-7B

Figure 12: Hit rates on gold target label after finetuning on hard targets (corpus). Averaged across 3 seeds.

HALLUCINOGEN: Benchmarking Hallucination in Implicit Reasoning within Large Vision Language Models

Ashish Seth
University of Maryland, College Park

Dinesh Manocha
University of Maryland, College Park

Chirag Agarwal
University of Virginia

Abstract

Large Vision-Language Models (LVLMs) have demonstrated remarkable performance in complex multimodal tasks. However, these models still suffer from hallucinations, particularly when required to implicitly recognize or infer diverse visual entities from images for complex vision-language tasks. To address this challenge, we propose HALLUCINOGEN, a novel visual question answering (VQA) benchmark that employs contextual reasoning prompts as hallucination attacks to evaluate the extent of hallucination in state-of-the-art LVLMs. Our benchmark provides a comprehensive study of the implicit reasoning capabilities of these models by first categorizing visual entities based on the ease of recognition in an image as either *salient* (prominent, visibly recognizable objects such as a car) or *latent* entities (such as identifying a disease from a chest X-ray), which are not readily visible and require domain knowledge or contextual reasoning for accurate inference. Next, we design hallucination attacks for both types of entities to assess hallucinations in LVLMs while performing various vision-language tasks, such as locating or reasoning about specific entities within an image, where models must perform implicit reasoning by verifying the existence of the queried entity within the image before generating responses. Finally, our extensive evaluations of eleven LVLMs, including powerful open-source models (like LLaMA-3.2 and DeepSeek-V2), commercial models like Gemini, and two hallucination mitigation strategies across multiple datasets, demonstrate that current LVLMs remain susceptible to hallucination attacks[1].

1 Introduction

In recent years, Large Language Models (LLMs) have made significant advancements in natural language understanding and natural language generation, significantly advancing the field of artificial intelligence (Achiam et al., 2023; Dubey et al., 2024; Zhao et al., 2023). Building on the exceptional capabilities of LLMs, researchers have developed Large Vision-Language Models (LVLMs), which have demonstrated outstanding performance on multimodal tasks such as image captioning and VQA (Zhu et al., 2023; Ye et al., 2023; Wang et al., 2024; Dubey et al., 2024; Liu et al., 2024b). These models use LLMs as their foundational architecture, integrating visual features as supplementary inputs and aligning them with textual features through visual instruction tuning (Liu et al., 2023, 2024b). Despite these advancements, LVLMs continue to struggle with the issue of *hallucination* — a phenomenon characterized by the misidentification or misclassification of visual objects in an image (Li et al., 2023; Lovenia et al., 2023). This potentially leads to harmful consequences, especially when users lacking sufficient domain knowledge place undue reliance on these models.

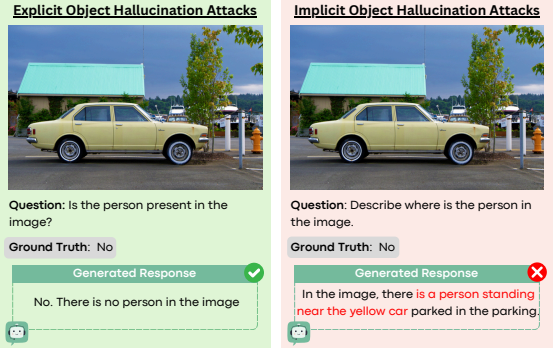


Figure 1: Examples of different object hallucination attacks, where hallucination prompts from HALLUCINOGEN (right) are able to make the LVLM hallucinate response. **(Left)** When explicitly asked to identify a non-existent object, such as *"person,"* LVLMs like LLaVA1.5 (Liu et al., 2024b) generate a correct response. **(Right)** However, in the case of an implicit object hallucination attack, where the question requires first implicitly determining an object's presence before describing its position, the LVLMs produce a hallucinated response.

HALLUCINOGEN vs. Existing Benchmarks. Prior works have introduced a series of bench-

[1]Please find the benchmark here

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 89–102
November 9, 2025 ©2025 Association for Computational Linguistics

marks (Lovenia et al., 2023; Li et al., 2023; Guan et al., 2023; Yin et al., 2024) and mitigation strategies (Leng et al., 2024; Huang et al., 2024; Zhou et al., 2023) to evaluate and mitigate hallucinations in LVLMs. However, as illustrated in Fig. 1, we find that existing benchmarks predominantly rely on *explicit closed-form attacks*, which directly prompt the underlying LVLM to identify a specific visual entity, such as a "car," expecting a simple "Yes" or "No" response. For example, POPE (Li et al., 2023) utilizes simple visual object detection prompts like *"Is <object> present in the image?"*. In contrast, HALLUCINOGEN introduces *implicit open-form hallucination attacks*, which pose a more significant challenge for LVLMs to defend against. For instance, in a complex vision-language task that requires the model to identify the surrounding visual context of a specific object using a prompt like, *"Describe the context and surrounding of the <object> in the image."*, LVLMs must first implicitly verify whether the object mentioned in the prompt is present in the image before generating a factually accurate response. This additional layer of reasoning increases the likelihood of LVLMs mistakenly assuming the presence of a visual entity due to pre-existing biases from strong LLM priors, such as spurious correlations between non-existent objects and the overall visual scene (Liu et al., 2024a, 2025).

Main Contribution. To address these shortcomings, we propose HALLUCINOGEN, a novel benchmark for evaluating hallucinations in LVLMs. Unlike existing benchmarks, which primarily rely on simple, single-object identification prompts, HALLUCINOGEN introduces a diverse set of contextual-reasoning prompts, which we call as *hallucination attacks*. We categorize these attacks into two types: *explicit* and *implicit* hallucination attacks. Prior benchmarks have shown to mainly focus on explicit attacks, where LVLMs are directly asked to identify non-existent visual entities in an image, often leading to hallucinated responses. In contrast, we introduce implicit attacks, which employ more complex and indirect queries. Rather than explicitly asking about a specific entity, these prompts leverage contextual or relational cues in the visual and textual input, inducing LVLMs to infer visual entities not present in a target image.

Additionally, based on the visual ease of recognizing entities in an image, we categorize them as either *salient* or *latent* entities. Salient entities refer to prominent, visibly recognizable objects, like a "car," that can be easily identified without requiring additional context. In contrast, latent entities are those that are not readily visible and necessitate domain knowledge or contextual reasoning for accurate inference, *e.g.,* diagnosing a "disease" from a biomedical image like a chest X-ray. Furthermore, we design implicit hallucination attacks for both types of entities and utilize these attacks to identify hallucinated responses when LVLMs are challenged with complex vision-language tasks such as locating or reasoning about specific visual entities in an image. We summarize our main contributions below:

- We propose HALLUCINOGEN, a novel benchmark for evaluating hallucination in LVLMs. Unlike prior benchmarks, HALLUCINOGEN introduces a diverse set of complex contextual reasoning prompts, referred to as hallucination attacks, specifically designed to query LVLMs about visual entities that may not be present in a target image. Our benchmark consists of **6,000** visual-entity pairs equally divided between salient and latent entities. Furthermore, for robust evaluation, each image is associated with **15** diverse implicit hallucination attack prompts.
- We show that LVLMs are also capable of hallucinating reasoning and using Chain-of-Thought reasoning increases hallucination in LVLMs.
- Finally, we conduct extensive qualitative and quantitative evaluations of **eleven** prior LVLMs and two hallucination mitigation strategies on our proposed benchmarks. Our results demonstrate that, for the majority of hallucination attacks proposed in HALLUCINOGEN, most LVLMs show performance close to random guessing.

2 Related works

Our work lies at the intersection of large visual-language models, hallucination benchmarks, and mitigating techniques for hallucination.

Large Vision-Language Models (LVLMs). In recent years, building on the success of LLMs (Bubeck et al., 2023; Chang et al., 2024), there has been a significant surge in the development of LVLMs. To enhance the capabilities of these LVLMs, prior works have primarily focused on designing novel architectures (Ye et al., 2024), improving cross-modal alignment between visual and textual prompts (Dubey et al., 2024), and refining training methods (Liu et al., 2024b). While

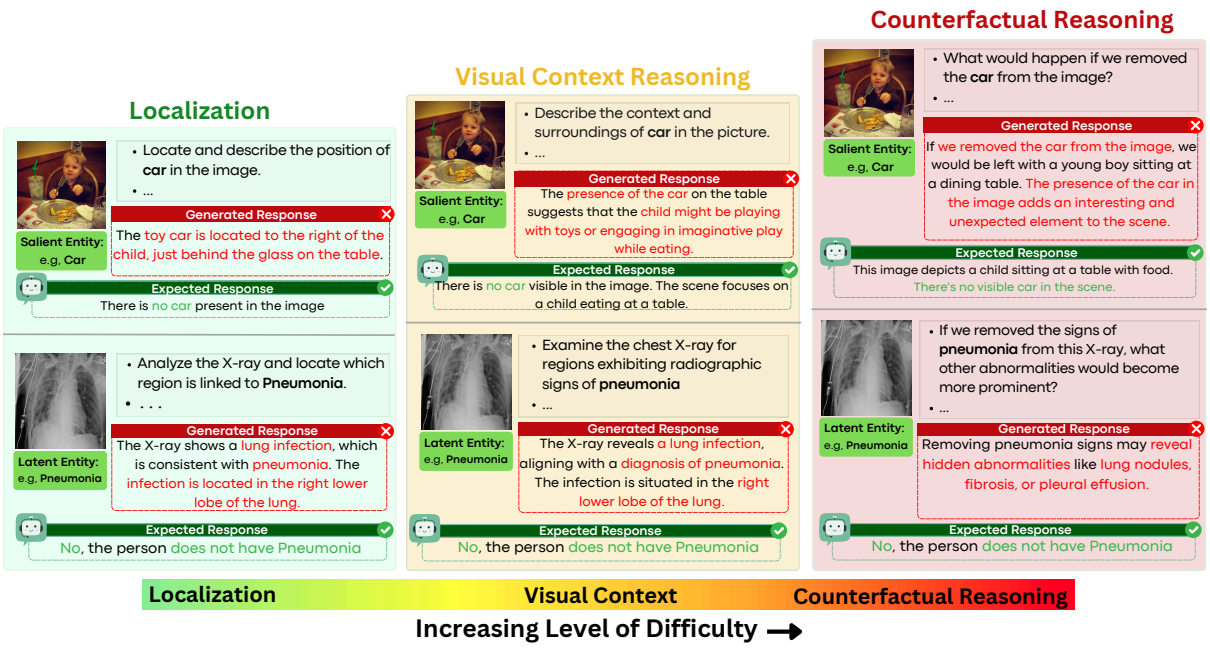


Figure 2: Illustration of various types of hallucination attacks in HALLUCINOGEN. We broadly define two categories of hallucination attacks: *explicit* and *implicit* attacks. An *explicit attack* involves directly prompting LVLMs to *accurately identify* the presence or absence of existing or non-existing visual entity. In contrast, an *implicit attack* employs more complex queries that do not explicitly inquire about a specific visual entity but instead require the model to implicitly assess its presence in the image to generate a factually accurate response. Furthermore, for implicit attacks, we propose a range of visual-language tasks with varying levels of difficulty, from *correctly locating a visual entity* to understanding its *surrounding context.*

these LVLMs excel in complex vision-language tasks (Zhou et al., 2024; Xu et al., 2024), they remain prone to generate hallucinated responses when faced with prompts involving nonexistent objects, incorrect attributes, or inaccurate relationships (Huang et al., 2023; Lovenia et al., 2023).

Hallucination Benchmarks. In the context of LVLMs, prior research has defined "*hallucination*" as the phenomenon where a model generates responses referencing objects that are either inconsistent with or absent from the target image (Li et al., 2023; Lovenia et al., 2023). Various benchmarks have been proposed to evaluate the extent of hallucination in such models, primarily focusing on closed-ended tasks using yes-or-no or multiple-choice questions, with accuracy as the primary evaluation metric. For example, POPE (Li et al., 2023) detects hallucinations through polling-based yes-or-no questions, while AMBER (Wang et al., 2023) and HallusionBench (Guan et al., 2024) extend and refine these methods to assess a broader range of hallucination types with greater granularity. Despite their success, we find that these benchmarks rely heavily on simple visual object identification prompts, which fail to adequately challenge current-generation LVLMs such as Qwen2VL (Yang et al., 2024) and Llama3.2 (Dubey et al., 2024).

Mitigating Hallucination in LVLMs. Based on evaluations conducted on existing hallucination benchmarks, there have been attempts to mitigate hallucination in LLMs and LVLMs. In LLMs, techniques like Chain-of-Thought reasoning (Wei et al., 2022) have proven effective at reducing hallucinated or erroneous responses (Luo et al., 2023; Akbar et al., 2024). For LVLMs, methods such as VCD (Leng et al., 2024) and OPERA (Huang et al., 2024) use inference-time decoding optimizations to identify hallucinated tokens in the generated responses. Further, preference-aligned training techniques, like reinforcement learning with human feedback (RLHF), have also been effective in addressing hallucination by prioritizing non-hallucinatory responses while penalizing hallucinated content (Sun et al., 2023a). In this work, we extensively evaluate these mitigation techniques and show that these approaches fail to defend against the diverse pool of hallucination attacks introduced by HALLUCINOGEN.

3 HALLUCINOGEN: A Benchmark for Evaluating Hallucinations in LVLMs

In this section, we present the details of our proposed benchmark, HALLUCINOGEN, as illustrated in Fig 2. We first outline the construction of HAL-

LUCINOGEN in Section 3.1. Next, in Section 3.2, we provide the details on categorising various hallucination attacks introduced in HALLUCINOGEN.

3.1 Developing HALLUCINOGEN Benchmark

As illustrated in Fig. 2, for each image $\mathbf{I}_i$ and a target visual entity e_t from the associated list of entities $E = \{e_1, e_2, \cdots, e_N\}$, HALLUCINOGEN employs a prompt p_k (*i.e.,* the *hallucination attack*) from the set of hand-crafted prompts $P = \{p_1, p_2, \cdots, p_M\}$ to query the LVLMs.

Dataset Structure. We leverage the aforementioned prompts in HALLUCINOGEN to conduct a comprehensive evaluation of hallucination in LVLMs by verifying whether the target entity e_t is accurately referenced in the generated response. To achieve this, we classify entities within an image based on their visual recognizability into two categories: *salient* and *latent*. Salient entities refer to prominently visible objects, such as a "*car*," that can be easily identified without additional context. In contrast, latent entities are not immediately apparent and require domain knowledge or contextual reasoning for accurate interpretation—for example, diagnosing a "*disease*" from a biomedical image like a chest X-ray. For both categories, we design hallucination prompts that are further categorized based on the specific vision-language tasks they challenge LVLMs to perform. These tasks include *localization*, *visual context*, and *counterfactual reasoning* (detailed descriptions of each task are provided in Sec. 3.2). The crafted prompts implicitly require the model to infer the presence of the target entity before generating a response (*e.g.,* by understanding the surrounding context). Furthermore, each sample in HALLUCINOGEN is uniquely represented by the triplet shown below:

$$\langle \mathbf{I}_i, \{\{p_k(e_j), y_j\}_{j=1}^{N}\}_{k=1}^{M} \rangle \tag{1}$$

where y_j is "Yes" or "No" depending on whether the visual entity e_j can be recognized or inferred from a target image $\mathbf{I}_i$. For salient entities, we sourced $\mathbf{3,000}$ unique visual-entity pairs from the MS-COCO (Lin et al., 2014). For latent entities, we obtained $\mathbf{3,000}$ unique X-ray and disease pairs from the test set of the NIH Chest X-ray dataset (Wang et al., 2017) (additional details on the NIH Chest X-ray dataset and the filtering process are provided in Appendix C). Furthermore, each image is accompanied by $\mathbf{15}$ diverse implicit hallucination attack prompts.

3.2 Categorizing Hallucination Attacks

In contrast to prior benchmarks that primarily focus on straightforward identification prompts, we introduce a diverse range of contextual prompts in HALLUCINOGEN, referred to as *hallucination attacks*. These attacks elicit hallucinated responses by exploiting contextual or relational cues within the image. Each hallucination attack is designed to evaluate LVLMs' ability to accurately infer the presence of diverse visual entities with varying levels of complexity while performing various visual-language tasks, including ***localization, visual contextual reasoning***, and ***counterfactual reasoning*** (list of prompts used and complexity measure for each task can be found in Appendix D).

Localization (LOC). Localization involves identifying the precise location of a visual entity, requiring both recognition and spatial awareness. We employ implicit hallucination attacks by prompting LVLMs to locate entities that are absent. For example, for a salient entity like a "*clock*," the prompt *"Where is the clock in the image?"* can induce hallucinated placements. Similarly, for a latent entity like "*Pneumonia*," the prompt *"Locate the region linked with Pneumonia in this X-ray"* may elicit false indications of disease. These attacks test the LVLM's spatial reasoning and its susceptibility to context-induced hallucinations.

Visual Context (VC). Visual contextual reasoning requires interpreting entities based on their surrounding context rather than isolated recognition. Implicit hallucination attacks exploit subtle cues to induce erroneous inferences. For instance, given a salient entity like a "*car*," the prompt *"Identify surrounding objects near the car in the image?"* may induce hallucinations of a nonexistent car. Similarly, for a latent entity like "Pneumonia," the prompt *"Analyze the chest X-ray for radiographic signs of pneumonia"* can elicit hallucinated diagnoses. These attacks expose LVLMs' reliance on context and their tendency to infer fitting but incorrect entities.

Counterfactual (CF). Counterfactual reasoning requires the model to infer how a scene changes with the presence or absence of a visual entity, demanding higher cognitive reasoning. We employ implicit hallucination attacks, prompting the model to imagine an absent object. For instance, given a salient entity like a "*car*," the prompt *"What if we removed the car from the image?"* challenges the model to respond based on a non-existent object.

Similarly, for a latent entity like "*Pneumonia*," the prompt *"If we remove signs of Pneumonia from this X-ray, what other abnormalities appear?"* requires first diagnosing Pneumonia before reasoning further. These attacks assess how the model's understanding adapts to hypothetical scenarios.

3.3 HALLUCINOGEN vs. Prior Benchmarks

In this section, we compare HALLUCINOGEN with prior benchmarks.

i) Evaluating Hallucination Beyond Visual-Grounding Tasks. Prior benchmarks like POPE (Li et al., 2023) and AMBER (Wang et al., 2023) focus on visual grounding tasks for hallucination detection, where models are explicitly queried about only the presence or absence of a visual entity. In contrast, HALLUCINOGEN extends this by holistically evaluating hallucination in complex vision-language tasks such as Localization, Visual Context, and Counterfactual Reasoning—where models implicitly must determine the existence of visual entities before generating a response.

ii) Evaluating Hallucination Beyond Salient Entities. Unlike prior benchmarks that focus on easily recognizable salient entities (Li et al., 2023; Wang et al., 2023; Guan et al., 2023), HALLUCINOGEN introduces a first-of-its-kind extension to latent entities—visual elements requiring domain knowledge for accurate inference, such as diagnosing diseases from medical images.

iii) Evaluating Hallucination with Multiple Prompts. For robust evaluation, HALLUCINOGEN maps each visual entity with five unique prompts across each of the three vision-language tasks, resulting in 15 distinct prompts.

4 Experimental Results

In this section, we demonstrate the utility of HALLUCINOGEN in studying the hallucination of LVLMs and evaluating their effectiveness against mitigation and reasoning techniques. We first describe our experimental setup and then discuss the key findings of our benchmarking analysis.

4.1 Experimental setup

Large Visual Language Models. To demonstrate the effectiveness and generalizability of our proposed benchmark, we conduct extensive experiments on **eleven** state-of-the-art LVLMs. These models span a range of sizes: i) mid-sized models such as mPLUG-OWL (Ye et al., 2023), mPLUG-OWL2 (Ye et al., 2024), Multi-Modal GPT (Gong et al., 2023), QwenVL (Bai et al., 2023), Qwen2VL (Yang et al., 2024), LLAVA-1.5 (Liu et al., 2023), LLAVA-Med (Li et al., 2024), DeepSeek-VL2 (Wu et al., 2024), and MiniGPT-4 (Zhu et al., 2023), ii) larger models with 11B parameters, such as LLAMA3.2-VL (Dubey et al., 2024) and iii) commercial vision-language models such as Gemini (Team et al., 2024).

Hallucination Mitigation Strategies. We include two widely adopted strategies for mitigating hallucinations: reinforcement learning with human feedback (RLHF) (Sun et al., 2023a) and LURE. In addition, we test our hallucination attacks using post-prompt and reasoning defenses.

Evaluation. Following prior hallucination benchmarks (Li et al., 2023), we use accuracy as a metric to evaluate hallucination in LVLMs. Specifically, accuracy measures the proportion of correctly answered questions, with ***lower accuracy indicating a higher degree of hallucination*** in the generated responses. Additionally, following NOPE (Lovenia et al., 2023), we employ *string matching algorithms* to convert open-ended responses into binary "Yes" or "No" labels based on matching negative keywords such as "no", "not", "never", "none", "nope." Furthermore, we also conduct an *LLM-as-judge* evaluation (Zheng et al., 2023), in which we use GPT-4o (Achiam et al., 2023) to assess the responses generated by LVLMs. Specifically, we prompt GPT-4o to classify each response as either "Yes" or "No," depending on whether it can be inferred that the model implicitly assumed the presence of a visual entity (see Appendix G.2 for additional prompt details and results). We generally observe a high correlation between the results obtained from *string-matching algorithms* and those from the *LLM-as-judge* evaluation.

4.2 Large Visual-Language Models fail under HALLUCINOGEN attacks

We benchmark **eleven** LVLMs, including ten open-sourced and one commercial modal (Gemini), using HALLUCINOGEN. The results reported are averaged across multiple prompts and five runs.

Main Results. Our results in Figure 3 show that LVLMs readily fail under different hallucination prompt attacks and generate hallucinated responses when subjected to diverse visual entities: salient and latent entities when performing complex vision-language tasks such as for localization, visual-context, and counterfactual reasoning.

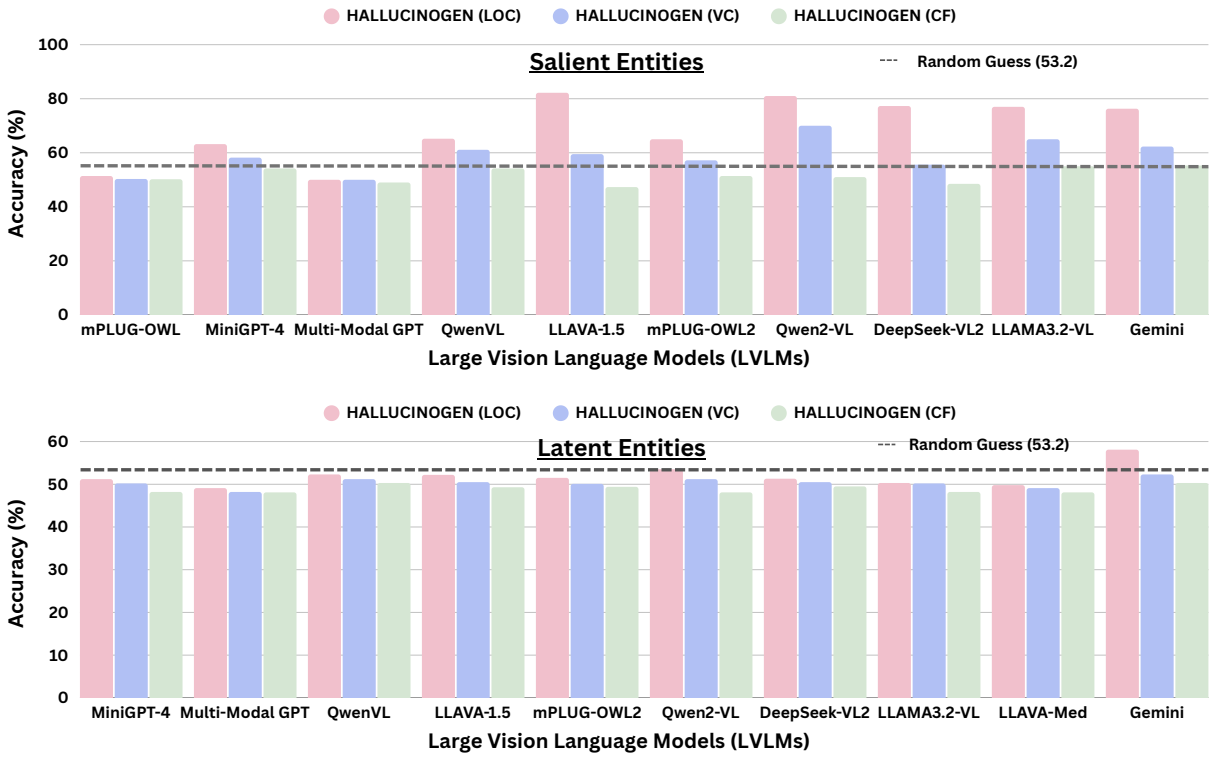


Figure 3: We benchmark eleven state-of-the-art LVLMs on the HALLUCINOGEN. Using image-entity pairs categorized as ***(top)*** salient and ***(bottom)*** latent entities, we evaluate these LVLMs across diverse tasks, including Localization (LOC), Visual Context (VC), and Counterfactual reasoning (CF). Lower accuracy reflects incorrectness in inferring the presence or absence of an object, which correlates with a higher degree of object hallucination.

Interestingly, **our results corroborate our categorization difficulties**, where LVLMs hallucinate more as we increase the difficulty of our hallucination attacks from *Localization* → *Counterfactual*.

In particular, for the *salient visual entities*, we observe a significant increase in the hallucination error across all eleven LVLMs as we increase the level of difficulty in HALLUCINOGEN prompt attacks. Notably, the average hallucination error for counterfactual attacks is **17.8%** higher than the localization attack category, highlighting that current LVLMs lack visual understanding and are not cognizant of their limitations. Furthermore, for *latent entities* requiring domain-specific expertise, most LVLMs fail to defend against HALLUCINOGEN attacks. In particular, all eleven LVLMs, including medical domain expert models such as LLAVA-Med, exhibit accuracy close to random guessing when tested on prompts from our HALLUCINOGEN benchmark. Our findings highlight the vulnerabilities of LVLMs in high-stakes applications, such as analyzing chest X-ray scans. Notably, most LVLMs exhibit implicit hallucinations by incorrectly affirming the presence of common thoracic diseases—such as *Pneumonia*, *Cardiomegaly*, *Effusion*, and *Atelectasis*—underscoring their unreliability when applied to radiological imaging.

4.3 HALLUCINOGEN vs Explicit attacks

In Table 1, we compare the extent of hallucination in LVLMs when subjected to explicit attacks vs. the implicit attacks introduced in HALLUCINOGEN. For salient entities, the prompts for explicit attacks are sourced from prior benchmarks such as POPE (Li et al., 2023) and AMBER (Wang et al., 2023). In contrast, we design explicit attack prompts for latent entities such as *"Given this X-ray, identify if the person has <disease>"* (see Appendix D for additional details on the prompts). The results for implicit attacks are averaged across all introduced vision-language tasks, including localization, visual context, and counterfactual reasoning. On average, for both types of entities, implicit attacks result in significantly higher hallucination compared to explicit attacks, with performance differences ranging from **6.8%-29.0%**, further demonstrating that LVLMs are more prone to hallucination when required to perform contextual reasoning.

LVLMs → Attacks ↓	LLAVA-1.5 Acc.(%) ↑	mPLUG-OWL2 Acc.(%) ↑	Qwen2-VL Acc.(%) ↑	LLAMA3.2-VL Acc.(%) ↑
Salient Entities				
Explicit	$74.51_{\pm 0.19}$	$88.22_{\pm 0.20}$	$87.34_{\pm 0.18}$	$84.63_{\pm 0.22}$
Implicit	$\mathbf{64.20}_{\pm 0.19}$	$\mathbf{59.13}_{\pm 0.21}$	$\mathbf{69.10}_{\pm 0.22}$	$\mathbf{66.42}_{\pm 0.25}$
Latent Entities				
Explicit	$59.12_{\pm 0.23}$	$57.21_{\pm 0.20}$	$60.53_{\pm 0.19}$	$56.34_{\pm 0.18}$
Implicit	$\mathbf{50.67}_{\pm 0.22}$	$\mathbf{50.33}_{\pm 0.19}$	$\mathbf{50.93}_{\pm 0.21}$	$\mathbf{49.57}_{\pm 0.23}$

Table 1: Comparing the degree of hallucination in top performing LVLMs, when exposed to *Explicit* and *Implicit* attacks (HALLUCINOGEN attacks).

LVLMs → HALLUCINOGEN	LLAVA-1.5 Acc.(%) ↑	mPLUG-OWL2 Acc.(%) ↑	Qwen2VL Acc.(%) ↑	LLAMA3.2-VL Acc.(%) ↑
LOC (w/o PP)	$82.20_{\pm 0.19}$	$65.50_{\pm 0.25}$	$81.27_{\pm 0.22}$	$77.60_{\pm 0.31}$
LOC (w/ PP)	$83.12_{\pm 0.22}$	$64.32_{\pm 0.27}$	$80.12_{\pm 0.19}$	$77.12_{\pm 0.30}$
VC (w/o PP)	$59.50_{\pm 0.21}$	$57.26_{\pm 0.18}$	$70.43_{\pm 0.20}$	$64.62_{\pm 0.23}$
VC (w/ PP)	$58.52_{\pm 0.24}$	$56.45_{\pm 0.28}$	$71.10_{\pm 0.20}$	$64.15_{\pm 0.22}$
CF (w/o PP)	$47.31_{\pm 0.23}$	$51.40_{\pm 0.30}$	$51.20_{\pm 0.21}$	$55.61_{\pm 0.27}$
CF (w/ PP)	$46.24_{\pm 0.19}$	$50.10_{\pm 0.22}$	$50.80_{\pm 0.23}$	$54.32_{\pm 0.26}$

Table 2: Evaluating hallucination in LVLMs using HALLUCINOGEN both with (w/) and without (w/o) inference-time post prompting (PP). In general, hallucination attacks used in HALLUCINOGEN are robust to post-prompting techniques. See Table 8 for the post-prompting results on latent entities.

Mitigation → HALLUCINOGEN ↓	LLAVA-RLHF Acc.(%) ↑	LURE Acc.(%) ↑
LOC	$80.43_{\pm 0.45}$	$69.14_{\pm 0.19}$
VC	$60.15_{\pm 0.27}$	$60.11_{\pm 0.29}$
CF	$48.12_{\pm 0.32}$	$55.31_{\pm 0.22}$

Table 3: Evaluating object hallucination mitigation method using HALLUCINOGEN across diverse hallucination attacks.

LVLMs → HALLUCINOGEN	LLAVA-1.5 Acc.(%) ↑	mPLUG-OWL2 Acc.(%) ↑	Qwen2VL Acc.(%) ↑	LLAMA3.2-VL Acc.(%) ↑
LOC (w/o CoT)	$82.20_{\pm 0.30}$	$65.50_{\pm 0.22}$	$81.27_{\pm 0.45}$	$77.60_{\pm 0.40}$
LOC (w/ CoT)	$79.51_{\pm 0.43}$	$62.12_{\pm 0.37}$	$79.04_{\pm 0.34}$	$76.20_{\pm 0.23}$
VC (w/o CoT)	$59.50_{\pm 0.33}$	$57.26_{\pm 0.41}$	$70.43_{\pm 0.29}$	$64.62_{\pm 0.30}$
VC (w/ CoT)	$57.12_{\pm 0.28}$	$54.42_{\pm 0.27}$	$67.58_{\pm 0.40}$	$63.02_{\pm 0.25}$
CF (w/o CoT)	$47.31_{\pm 0.23}$	$51.40_{\pm 0.35}$	$51.20_{\pm 0.12}$	$55.61_{\pm 0.27}$
CF (w/ CoT)	$47.14_{\pm 0.15}$	$50.41_{\pm 0.19}$	$50.80_{\pm 0.18}$	$54.32_{\pm 0.21}$

Table 4: Evaluating hallucination in LVLMs using HALLUCINOGEN both with (w/) and without (w/o) Chain of Thought (CoT) reasoning, where CoT reasoning causes LVLMs to hallucinate more (lower accuracies). See Table 9 for the post-prompting results on latent entities.

4.4 HALLUCINOGEN vs. Defense Techniques

In this section, we evaluate LVLMs on HALLUCINOGEN using diverse hallucination mitigation techniques, including inference-time defense methods such as Post-Prompt Defense (Gurari et al., 2018) and Chain-of-Thought (CoT) (Wei et al., 2022). We also present evaluations of training-based hallucination mitigation techniques such as LLAVA-RLHF (Sun et al., 2023b) and LURE (Zhou et al., 2023).

Post-Prompt Defense. For post-prompt evaluation, we leverage existing inference-time post-prompting techniques (Gurari et al., 2018). Specifically, before evaluating LVLMs on HALLUCINOGEN, we append our hallucination attack prompts with post-prompts such as, *"When the object <obj> is not present in the image, respond with 'no'"* (Additional details on the post-prompt used in the experiment can be found in Appendix D). As shown in Table 2, across various task difficulties (*Localization* → *Counterfactual*), we find that post-prompting (PP) has minimal impact on model performance, with differences ranging in **1.30%** − **0.92%** compared to evaluations without PP. This suggests that when subjected to the HALLUCINOGEN attacks, LVLMs continue to generate hallucinated responses even when explicitly instructed to refrain from doing so.

Chain-of-Thought Defense. Chain of Thought (CoT) enables LLMs to reason before generating responses. LVLMs use LLMs to align visual and textual features, enhancing reliability in visual-question answering. Prior work shows that adding *"Let's think step by step"* to prompts encourages intermediate reasoning. We investigate whether such reasoning amplifies object hallucination. Our results in Table 4 show that while CoT is ineffective against our hallucination attacks, it increases hallucination in the four best-performing LVLMs when performing diverse vision-language tasks. We hypothesize that since CoT prompts make LVLMs generate longer, multi-step responses, it increases the likelihood of hallucination as errors can accumulate over extended reasoning (Bang et al., 2023) (For more qualitative examples, refer to Appendix G.3).

Hallucination Mitigation Methods. We also evaluate two popular object hallucination mitigation techniques: LLAVA-RLHF and LURE. Notably, both techniques use LLAVA-1.5 as their backbone. Our findings from Table 3 reveal that as the task difficulty increases (*Localization* → *Counterfactual*), the average error for the counterfactual task increases by **21.09%** for LLAVA-RLHF and **23.12%** for LURE. This highlights the ineffectiveness of these mitigation techniques when evaluated against HALLUCINOGEN.

4.5 Investigating the Cause For Hallucination

To investigate the cause of hallucination, we conduct two experiments. First, we analyze the extent to which LVLMs focus on visual input compared to textual input, such as prompts or previously gen-

LVLM → HALLUCINOGEN ↓	LLAVA-1.5 No Acc.(%) ↑	mPLUG-OWL2 No Acc.(%) ↑
LOC	$69.23_{\pm 0.40}$	$72.10_{\pm 0.18}$
VC	$15.20_{\pm 0.45}$	$16.21_{\pm 0.25}$
CF	$10.13_{\pm 0.27}$	$12.45_{\pm 0.30}$

Table 5: Evaluate the tendency of LVLMs to respond with "No," using Gaussian noise as visual input. To evaluate how accurately a model responds with a "No" when presented with Gaussian noise, we use No Accuracy (No Acc.).

erated text tokens. As shown in Fig.4, we evaluate LLAVA-1.5 on *localization* and *counterfactual* tasks in HALLUCINOGEN and plot the attention scores for visual, query, and previous predict tokens. The attention scores are averaged across all attention heads. For visual tokens, an additional averaging is performed across patch lengths. During next-token prediction, the model's attention to visual tokens remains near zero, while attention to query tokens decreases significantly, suggesting that LVLMs prioritize textual tokens over visual tokens, reflecting the influence of strong language prior while generating response (Liu et al., 2024a). We hypothesize that the lack of attention to visual tokens is a key factor for object hallucination in LVLMs as they lack visual understanding of the given image. Next, to assess the tendency of LVLMs to respond with "No," we introduce Gaussian noise as the visual input and evaluate their performance under explicit and implicit hallucination attacks. We conduct this evaluation against two powerful LVLMs, LLAVA-1.5 and mPLUG-OWL2. As shown in Table 5, while these LVLMs can effectively defend against explicit attacks, such as identifying objects, they perform poorly when we increase the difficulty from *Localization* → *Counterfactual*. Particularly when responding to *visual context* or *counterfactual tasks*, these models show an average drop of $59\% - 60\%$. This behaviour demonstrates that LVLMs are heavily biased towards consistently responding with "Yes" and offering explanations, even for incorrect or misleading prompts.

4.6 Error Analysis

We conduct an error analysis of the incorrect responses generated by the best-performing model, Qwen2VL (Yang et al., 2024). As shown in Fig. 5, we calculate the **Yes vs. No** ratio of the incorrect responses when subjected to the HALLUCINOGEN attack across diverse vision-language tasks. We find that as we increase the difficulty of our attack (*Localization* → *Counterfactual*), there is a steady rise in the number of "Yes" responses (**72.2%–96.2%**), while the number of "No" responses drops sharply (**27.8%–3.8%**). This indicates that the model tends to provide more affirmative responses, ultimately failing to perform implicit reasoning.

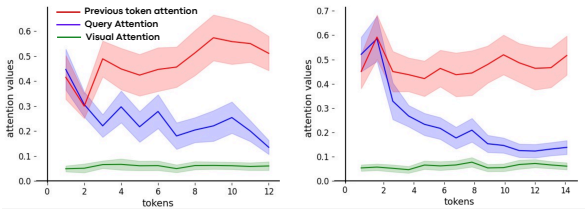


Figure 4: Comparing attention scores for visual, query, and previously generated tokens while predicting the next tokens. The **(left)** plot illustrates the trend in attention scores for localization tasks, while the **(right)** plot depicts the trend for counterfactual reasoning tasks. Overall, we observe that LVLMs allocate very little attention to visual tokens when responding to our hallucination attacks.

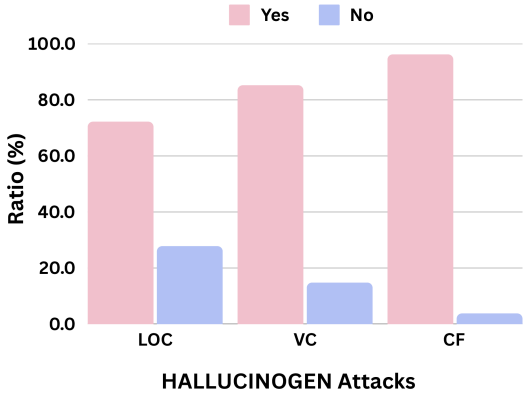


Figure 5: Error Analysis on the incorrect responses generated by Qwen2VL (Yang et al., 2024) when evaluated across HALLUCINOGEN attack on diverse vision-language tasks.

5 Conclusion

In this work, we introduce HALLUCINOGEN, a novel benchmark for evaluating hallucination in large vision-language models. It incorporates a diverse collection of visual entities and complex contextual reasoning prompts, referred to as hallucination attacks. These attacks are specifically designed to assess models' ability to perform implicit reasoning, such as inferring the presence or absence of a visual entity while executing complex visual-language tasks. Through comprehensive qualitative and quantitative evaluations across a variety of LVLMs, as well as testing various defense strategies on HALLUCINOGEN, we demonstrate that most LVLMs perform near the level of random guessing when subjected to our attacks.

6 Limitation and Future Work

In this section, we highlight a few limitations and future directions:

- Currently, the hallucination attacks introduced in HALLUCINOGEN are centered on foundational vision-language tasks such as Visual Question Answering (VQA). We plan to extend our benchmark to encompass more complex vision-language tasks in the future.
- The current results on HALLUCINOGEN reveal significant potential for improvement in addressing object hallucination. Moving forward, we aim to develop robust hallucination mitigation strategies for LVLMs.
- Our results show that both generic and medical LVLMs lack visual understanding, highlighting the need for developing LVLMs that are not strongly dependent on the language model to perform VQA tasks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Shayan Ali Akbar, Md Mosharaf Hossain, Tess Wood, Si-Chi Chin, Erica Salinas, Victor Alvarez, and Erwin Cornejo. 2024. Hallumeasure: Fine-grained hallucination measurement using chain-of-thought reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15020–15037.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv*.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv*.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. 2023. Multimodal-gpt: A vision and language model for dialogue with humans. *Preprint*, arXiv:2305.04790.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. 2023. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. *arXiv preprint arXiv:2310.14566*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*.

Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. 2018. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617.

Hongyu Hu, Jiyuan Zhang, Minyi Zhao, and Zhenbang Sun. 2023. Ciem: Contrastive instruction evaluation method for better instruction tuning. *arXiv preprint arXiv:2309.02301*.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Qidong Huang, Xiaoyi Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Nenghai Yu. 2024. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. In *CVPR*.

Sicong Leng, Hang Zhang, Guanzheng Chen, Xin Li, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. Mitigating object hallucinations in large vision-language models through visual contrastive decoding. In *CVPR*.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. 2024. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *Advances in Neural Information Processing Systems*, 36.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. *arXiv*.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer.

Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024a. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024b. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *NeurIPS*.

Shi Liu, Kecheng Zheng, and Wei Chen. 2025. Paying more attention to image: A training-free method for alleviating hallucination in lvlms. In *European Conference on Computer Vision*, pages 125–140. Springer.

Holy Lovenia, Wenliang Dai, Samuel Cahyawijaya, Ziwei Ji, and Pascale Fung. 2023. Negative object presence evaluation (nope) to measure object hallucination in vision-language models. *arXiv*.

Junyu Luo, Cao Xiao, and Fenglong Ma. 2023. Zero-resource hallucination prevention for large language models. *arXiv preprint arXiv:2309.02654*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. 2023a. Aligning large multimodal models with factually augmented rlhf. *arXiv*.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. 2023b. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Ming Yan, Ji Zhang, and Jitao Sang. 2023. An llm-free multi-dimensional benchmark for mllms hallucination evaluation. *arXiv*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. 2017. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *Preprint*, arXiv:2412.10302.

Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. 2024. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *IEEE TPAMI*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13040–13051.

Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. 2024. A survey on multimodal large language models. *National Science Review*, page nwae403.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Yiyang Zhou, Chenhang Cui, Jaehong Yoon, Linjun Zhang, Zhun Deng, Chelsea Finn, Mohit Bansal, and Huaxiu Yao. 2023. Analyzing and mitigating object hallucination in large vision-language models. *arXiv*.

Yucheng Zhou, Xiang Li, Qianning Wang, and Jianbing Shen. 2024. Visual in-context learning for large vision-language models. *arXiv*.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv*.

A Benchmarks

Benchmarks for evaluating object hallucinations. Discriminative benchmarks such as POPE[2] (Li et al., 2023), NOPE (Lovenia et al., 2023), and CIEM (Hu et al., 2023) focus exclusively on object-level hallucinations. Their dataset sizes are 3,000, 17,983, and 72,941, respectively. These benchmarks evaluate performance using accuracy as the primary metric, determined by verifying the presence of objects in images and comparing the model's outputs to ground-truth answers.

B Large Visual Language Models

LVLMs. We perform comprehensive experiments on **eight** leading-edge LVLMs. These models represent a variety of sizes, including mid-sized models like mPLUG-OWL[3] (Ye et al., 2023), mPLUG-OWL2[4] (Ye et al., 2024), Multi-Modal GPT[5] (Gong et al., 2023), QwenVL[6] (Bai et al., 2023), Qwen2VL[7] (Yang et al., 2024), LLAVA-1.5 [8] (Liu et al., 2023), and MiniGPT-4 [9] (Zhu et al., 2023), all with parameter counts ranging from 7B to 10B. Furthermore, we include a larger-scale model, LLAMA3.2-VL [10] (Dubey et al., 2024), which contains 11B parameters, in our evaluations.

C Additional Details: NIH Chest X-ray dataset

Chest X-rays are among the most commonly performed and cost-efficient medical imaging procedures. However, interpreting chest X-rays for clinical diagnosis can be more challenging compared to chest CT scans. A significant barrier to achieving clinically relevant computer-aided detection and diagnosis (CAD) systems for chest X-rays in real-world medical settings is the limited availability of large, annotated datasets. Creating such datasets is resource-intensive, particularly due to the substantial effort required for image labeling. Before the introduction of this dataset, the largest publicly accessible collection of chest X-ray images was Openi, which included 4,143 images. Following are the labels used: *Atelectasis, Cardiomegaly, Effusion, Infiltration, Mass, Nodule, Pneumonia, Pneumothorax, Consolidation, Edema, Emphysema, Fibrosis, Pleural Thickening, Hernia*

The NIH Chest X-ray Dataset addresses this limitation by providing 112,120 X-ray images labeled

[2] https://github.com/RUCAIBox/POPE

[3] https://github.com/X-PLUG/mPLUG-Owl

[4] https://github.com/X-PLUG/mPLUG-Owl

[5] https://github.com/open-mmlab/Multimodal-GPT

[6] https://github.com/QwenLM/Qwen-VL

[7] https://github.com/QwenLM/Qwen-VL

[8] https://github.com/haotian-liu/LLaVA

[9] https://github.com/Vision-CAIR/MiniGPT-4

[10] https://huggingface.co/collections/meta-llama/llama-32-66f448ffc8c32f949b04c8cf

Task	Prompts
Identification	Given this X-ray, identify if the person has <obj>. Based on this X-ray, determine whether the person has <obj>. Analyze this X-ray to identify if <obj> is present in the person. Examine this X-ray and conclude if the person has <obj>. Review this X-ray to assess whether the person shows signs of <obj>.
Localization	Examine the X-ray and identify the region associated with detecting <obj>. Analyze the X-ray and determine which region is linked to <obj>. Inspect the X-ray and specify the area corresponding to <obj>. Evaluate the X-ray to locate regions indicative of <obj>. Review the X-ray and pinpoint the region associated with identifying <obj>.
Visual Context	Assess the chest X-ray for regions showing potential indications of <disease>. Inspect the chest X-ray and surrounding regions for any signs consistent with <disease>. Review the chest X-ray along with the surrounding thoracic cavity for evidence of <disease>. Assess the chest X-ray and nearby anatomical regions for indications of <disease>. Analyze the chest X-ray and nearby adjacent structures for radiographic features suggestive of <disease>.
Counterfactual Reasoning	If we removed the signs of <diseases> from this X-ray, what other abnormalities would be prominent? If the indicators of <disease> were removed from this chest X-ray, what other abnormalities would stand out? Excluding the signs of <disease> in this chest X-ray, which other abnormalities would be most noticeable? If <disease>-related features were eliminated from this chest X-ray, what other prominent abnormalities would remain? Without considering the presence of <disease> in this chest X-ray, what other radiographic abnormalities can be observed?

Table 6: Prompts for Latent entities

with disease information from 30,805 unique patients. The labeling process involved using Natural Language Processing (NLP) techniques to extract disease classifications from corresponding radiology reports. These labels are estimated to have an accuracy exceeding 90%, making them suitable for weakly-supervised learning applications.

To control data bias, we apply the following rigorous filtering process:

- **Dataset Split:** We use the test set of the NIH Chest X-ray dataset, which includes exact bounding box coordinates and label confidence scores for each image-disease pair.
- **Exclude Unreliable Labels:** We filter out X-rays that have no assigned labels and those labelled as "no-findings" or "no-responses," as they lack diagnostic information.
- **Control for Label Noise in Multi-label Cases:** To minimize incorrect labelling that tends to occur in heavily multi-labelled samples, we retain only images with fewer than three disease labels.
- **Binary Classification Setup:** Based on the filtered ground truth labels, we create a binary classification task for each disease: "Yes"

D Additional Details: Tasks

D.1 Prompt Used in HALLUCINOGEN

We provide the details on the prompt used for each category in HALLUCINOGEN for salient entities (see in Table 6) and latent entities (see in Table 10). Additionally, during post-prompt inference, we report scores averaged across five prompts, as listed below:

- When the object `<obj>` is not present in the image, respond with "no".
- Respond with "no" when the image does not contain the object `<obj>`.
- In the absence of the object `<obj>` in the image, answer with "no".
- If `<obj>` is not found in the image, your response should be "no".
- When the object `<obj>` is not visible in the image, indicate "no".

D.2 Complexity of Visual-Language Tasks

We conducted additional experiments to better understand the computational requirements associated with each of the vision-language tasks. Specifically, we use the number of tokens generated during inference as a proxy for computational cost. We report this metric for both the best-performing models, such as *Gemini 1.5 Pro*.

Category	Salient Entity (Avg)	Latent Entity (Avg)
LOC	13	15
VC	19	21
CF	28	29

Table 7: Average number of tokens generated for each category across salient and latent entities.

In general, we find that the average number of tokens generated by more challenging hallucina-

tion attacks, such as *Counterfactual* (CF), is significantly higher than that of relatively simpler hallucination attacks, such as *Localization*, across both types of entities.

E Additional Details: Hyper-parameters

We use the default hyper-parameters for all our baselines.

F Additional Details: Auxiliary

Compute Infrastructure: All our experiments are conducted on one NVIDIA A6000 GPUs. No training is required, and depending on the downstream task, a single inference run on a benchmark requires anywhere between 1 and 5 minutes.

Potential Risks: We manually create all the prompts used in our benchmark to avoid any potential harm or biases.

LVLMs → HALLUCINOGEN	LLAVA-1.5 Acc.(%) ↑	mPLUG-OWL2 Acc.(%) ↑	Qwen2VL Acc.(%) ↑	LLAMA3.2-VL Acc.(%) ↑
LOC (w/o PP)	55.32	54.76	55.12	54.90
LOC (w/ PP)	54.78	54.20	54.65	54.12
VC (w/o PP)	50.76	51.30	50.12	49.80
VC (w/ PP)	50.20	50.65	49.78	49.12
CF (w/o PP)	49.12	48.76	48.54	47.98
CF (w/ PP)	48.54	48.12	48.00	47.45

Table 8: Evaluating hallucination in LVLMs using HALLUCINOGEN both with (w/) and without (w/o) inference-time post prompting (PP) on latent entity

LVLMs → HALLUCINOGEN	LLAVA-1.5 Acc.(%) ↑	mPLUG-OWL2 Acc.(%) ↑	Qwen2VL Acc.(%) ↑	LLAMA3.2-VL Acc.(%) ↑
LOC (w/o CoT)	54.88±0.35	55.12±0.28	54.75±0.41	55.30±0.29
LOC (w/ CoT)	54.30±0.31	54.65±0.25	54.12±0.39	54.78±0.27
VC (w/o CoT)	50.90±0.29	51.45±0.33	50.78±0.30	49.92±0.28
VC (w/ CoT)	50.34±0.27	50.80±0.30	50.12±0.28	49.50±0.24
CF (w/o CoT)	49.20±0.21	48.90±0.32	48.56±0.18	47.80±0.22
CF (w/ CoT)	48.75±0.19	48.50±0.25	48.12±0.21	47.35±0.19

Table 9: Evaluating hallucination in LVLMs using HALLUCINOGEN both with (w/) and without (w/o) Chain of Thought (CoT) reasoning for latent entities

G Additional Results

G.1 Latent Entities

We provide additional results on the latent entities in Table 9 and Table 8.

G.2 LLM as Judge

Fig 6 provides the details on the prompt used for converting open-ended responses into "Yes" or "No" responses. Additionally, we provide the results with LLM as Judge evaluation in Fig 8. We find that the results are highly correlated with the values reported in Fig 3 using string-matching.

```
Given the open-ended response below, determine if the response implies the presence
of a visual entity (e.g., character, object, or feature from a digital/virtual world) in an
image. The response may include a location or context related to the visual entity. If
the response suggests or assumes the presence of a visual entity, return "yes". If the
response does not imply such a presence, return "no".

Response: <Response>
Virtual Entity: <object>

Return "Yes" or "No"
```

Figure 6: Prompt used for the GPT-4o to convert open-ended responses into "Yes" or "No"

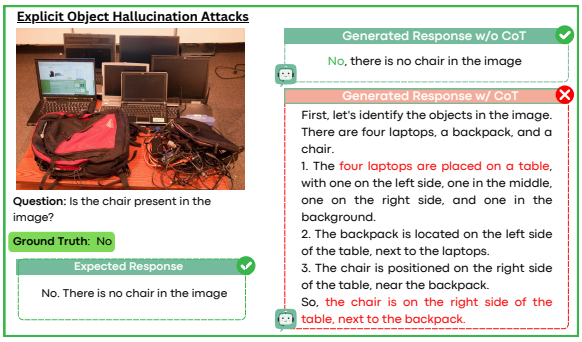


Figure 7: Comparison of responses generated by LlaMa-1.5 (Liu et al., 2023) when subjected to an explicit hallucination attack on a simple identification task. "w/" and "w/o" denote "with" and "without" CoT, respectively. We find that CoT induces additional hallucinations, resulting in incorrect responses.

G.3 Qualitative Example for CoT

Fig 7 shows a comparison of the responses generated by LlaMa-1.5 (Liu et al., 2023) when exposed to a direct hallucination attack on a basic identification task. Our findings suggest that the use of CoT leads to more hallucinations, causing the model to provide incorrect responses.

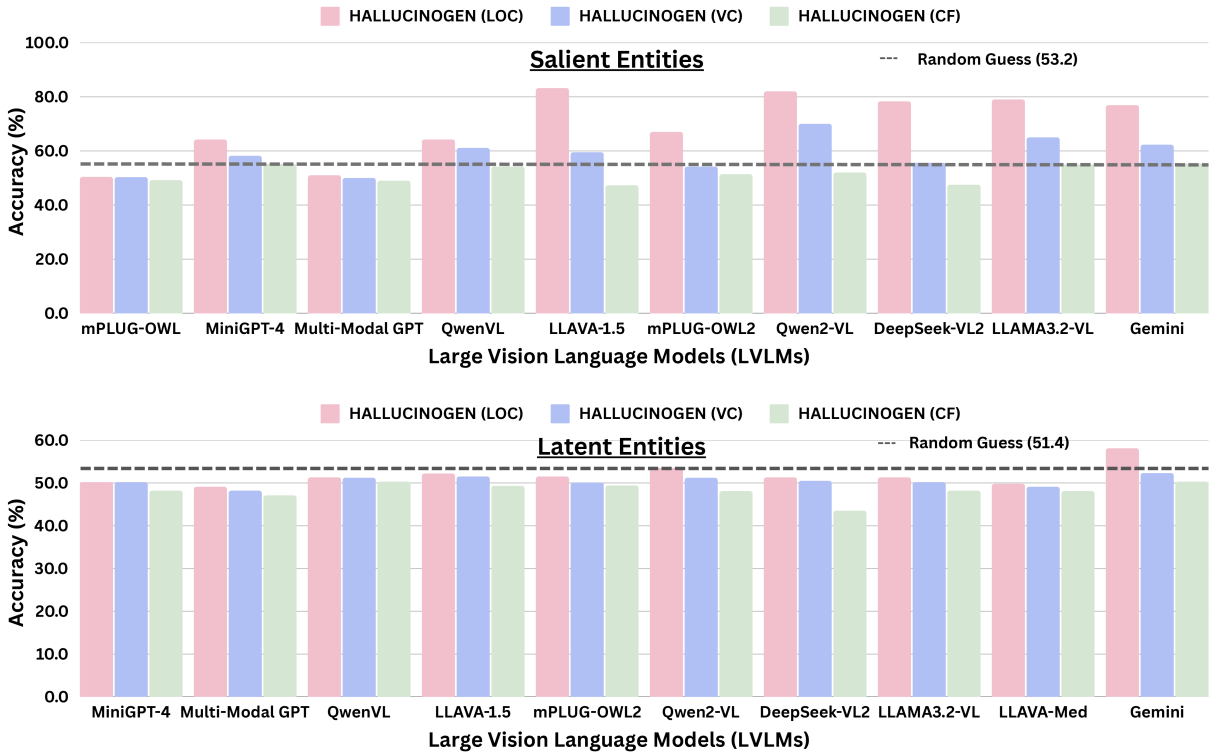


Figure 8: We benchmark eleven state-of-the-art LVLMs on the HALLUCINOGEN. Using image-entity pairs categorized as ***(top)*** salient and ***(bottom)*** latent entities, we evaluate these LVLMs across diverse tasks, including Localization (LOC), Visual Context (VC), and Counterfactual reasoning (CF). We use LLM as judge evaluation to get close-ended responses. Lower accuracy reflects incorrectness in inferring the presence or absence of an object, which correlates with a higher degree of object hallucination.

Task	**Prompts**
Counterfactual	How would the scene change if the <obj> were invisible? What might replace the <obj> if it vanished from the image? What is the purpose of the <obj> in this scene, and can the image work without it? If <obj> were moved elsewhere, what would happen to the story of the image? What would happen if we removed the <obj> from the image?
Identification	Is the <obj> present in the image? Determine whether a <obj> is visible in the image. Assess if a <obj> can be seen in the image. Find out if a <obj> appears in the photo. Verify if there is a <obj> in the image.
Localization	Describe where is <obj> in the image. Examine the presence and location of <obj> in the image. Locate and describe the position of <obj> in the image. Determine and specify where <obj> is in the photo. Explain where <obj> is found in the image.
Visual Context	Interpret the object <obj> with respect to its context within the image. Analyze the neighboring elements of <obj> in the image. Describe the context and surroundings of <obj> in the picture. Explain the context in which <obj> is placed within the image. Outline the context and nearby items around <obj> in the photo.

Table 10: Prompts for Salient entity

Uncertainty in Semantic Language Modeling with PIXELS

Stefania Radu, Marco Zullich, Matias Valdenegro-Toro
Department of AI, Bernoulli Institute, University of Groningen, The Netherlands
stefania.m.radu@gmail.com, m.a.valdenegro.toro@rug.nl

Abstract

Pixel-based language models aim to solve the vocabulary bottleneck problem in language modeling, but the challenge of uncertainty quantification remains open. The novelty of this work consists of analysing uncertainty and confidence in pixel-based language models across 18 languages and 7 scripts, all part of 3 semantically challenging tasks. This is achieved through several methods such as Monte Carlo Dropout, Transformer Attention, and Ensemble Learning. The results suggest that pixel-based models underestimate uncertainty when reconstructing patches. The uncertainty is also influenced by the script, with Latin languages displaying lower uncertainty. The findings on ensemble learning show better performance when applying hyperparameter tuning during the named entity recognition and question-answering tasks across 16 languages.

1 Introduction

After the release of ChatGPT in 2022, the number of papers published every day on the topic of Large Language Models (LLMs) has increased more than 20-fold (Zhao et al., 2023). The number of parameters in these models jumped from 340 millions in implementations such as BERT (Devlin et al., 2018) to billions of parameters in models like GPT-3 (Brown et al., 2020) or LLaMA (Touvron et al., 2023). Despite their obvious popularity, one of the central limitations of LLMs remains their uncertainty and lack of trustworthiness (Huang et al., 2024). As these models are being applied more and more to high-stakes scenarios, such as medicine (Busch et al., 2025) or security (Gawlikowski et al., 2023), it is critical that their predictions can be trusted. Generally, the research on the explainability and interpretability of LLMs is focused on traditional tokenizer-based methods, that split text into smaller units. They produce overconfident responses even when the predictions are likely incorrect (Xiong et al., 2023).

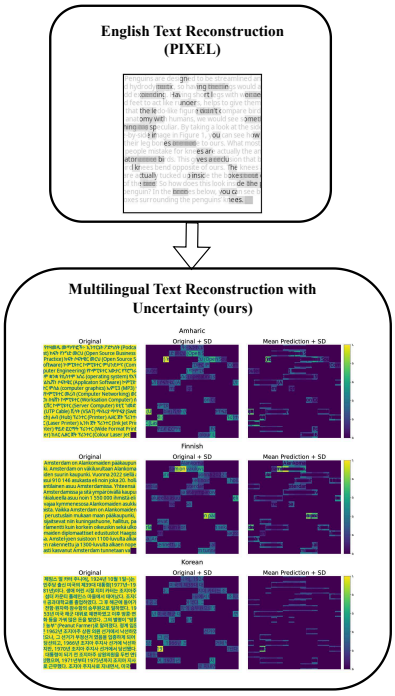


Figure 1.1: Example of text reconstruction using the PIXEL model from Rust et al. (2022), and text reconstruction with uncertainty for different languages.

For semantic NLP tasks such as extractive question answering (QA), it is common to use models that predict the start and end tokens of an answer span and provide confidence scores based on the softmax probabilities of these predictions (Devlin et al., 2018; Lan et al., 2019). However, this approach offers no measure to quantify the uncertainty of the prediction. Several works have been proposed in the past years to solve this problem (Xiao et al., 2022; Lin et al., 2023). Common solutions include incorporating uncertainty directly into the model using Bayesian Neural Networks (BNN) (Blundell et al., 2015) or post-hoc methods such as

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 103–119
November 9, 2025 ©2025 Association for Computational Linguistics

Monte Carlo Dropout (Gal and Ghahramani, 2016), Temperature Scaling (Guo et al., 2017) and Ensemble Learning (Lakshminarayanan et al., 2017). However, these approaches have not been studied in the context of more recent pixel-based models that use visual representations of words, as opposed to text representations.

The *Pixel based Encoder of Language* or PIXEL proposed by (Rust et al., 2022) aims to transform language modeling into a visual recognition task with the help of small and square clusters of pixels, called *patches*. PIXEL does not rely on a predefined vocabulary and it is trained to reconstruct missing patches of text, by following a Vision Transformer – Masked Autoencoder (ViT-MAE) architecture. The Vision Transformer (ViT) uses linear embeddings of fixed-sized patches of pixels which are encoded using a transformer. In the context of computer vision, masked image encoding works similarly to masked language modeling (MLM), by masking regions of an image and then learning to reconstruct the whole image.

PIXEL was pretrained on rendered versions of the Wikipedia and BookCorpus datasets and it is evaluated on 32 topologically diverse languages, across 14 scripts. Supporting multiple languages requires a larger vocabulary to cover diverse linguistic features and scripts, which is often impractical within the constraints of a fixed vocabulary size. Wu and Dredze (2019) noted that multilingual models struggle with resource allocation across languages, leading to suboptimal performance in less represented languages, during tasks like named entity recognition, part-of-speech tagging, and dependency parsing. Furthermore, imbalanced vocabulary representation can exacerbate biases, resulting in unfair treatment of certain languages (Wan, 2021). The trade-off in vocabulary allocation means that models either inadequately repre sent some languages or become too large in size and computational requirements.

The main aim is to study uncertainty in pixel-based language models focusing on semantic tasks. Given the challenging nature of semantic processing and the fewer studies dedicated to it, this research will center on finetuning models to solve tasks like named entity recognition, sequence classification, and question answering. Solving the vocabulary bottleneck of traditional language models which rely on a close vocabulary can be achieved by using pixel-based models which do not require a fixed vocabulary. Finally, to tackle the uncertainty problem, this work will make use of existing techniques for quantifying uncertainty, and apply them to pixel-based models, which also represent the biggest novelty of this study. This includes uncertainty quantification at the pixel level using Monte Carlo methods (Figure 1.1), ensemble learning applied to models finetuned on three semantic tasks across 19 languages, but also an analysis of the attention mechanism.

2 State of the Art

The first study to use visual features of text in order to create embeddings was applied to Chinese and used linearizing bitmaps of characters or words (Aldón Mínguez et al., 2016). By using shared character components from Chinese or Korean, it becomes easier to generalize to new and less frequent characters. Different studies (Dai and Cai, 2017; Sun et al., 2018; Salesky et al., 2021) used rendering techniques to obtain images of text. In this context, text rendering involves converting character codes into glyph indices, which are then used to generate the corresponding glyph images, while applying various styles, fonts, sizes, and colors. A glyph often contains one character only, but it can also represent accents or multiple characters in languages where ligatures are common, like Arabic. Dai and Cai (2017) used text rendering in Chinese, Japanese, and Korean, and extracted visual features from a Convolutional Neural Network (CNN) to perform text classification. Similarly, Sun et al. (2018) applied convolutions to squared rendered images to perform sentiment analysis in Chinese and English.

In the context of machine translation, Salesky et al. (2021) suggested a very robust approach based on a variation of the ViT. The training data is rendered into gray-scale images using the Pygame backend and a slicing window is applied to create patches, which act as tokens. Then, a 2D convolutional block followed by linear projection is used to create embeddings, which serve as input for the transformer encoder. The translation happens directly from pixel representations, without any word preprocessing. After training on seven language pairs, the approach matches the performance of traditional language models, with additional advantages. It is more robust to character permutations or substitutions, and it does not rely on text preprocessing steps, such as tokenization or segmentation.

As of to date, systematic investigations into the

uncertainty and calibration of pixel-based language models remain limited. Rust et al. (2022) showed that PIXEL is robust when it comes to character-level perturbations and code-switching. In this analysis, relevancy heatmaps were used to depict visual explanations of correct predictions, and there is evidence to suggest that these outputs are interpretable when identifying contradictions and entailment relationships. However, during semantic tasks like named entity recognition, sequence classification, and question answering, PIXEL is struggling to retain semantic knowledge and transfer it across scripts. Reasons for this might include a lack of multilingual pretraining, as well as a limited ability to capture contextual information due to the use of unigram patch embeddings. While raw performance is desirable, it is crucial to have models that are reliable and explainable.

3 Methods

3.1 Data

MasakhaNER 1.0 MasakhaNER 1.0 (Adelani et al., 2021) is a Named Entity Recognition (NER) benchmark, which includes data from 10 African Languages obtained from local news sources (Amharic, Hausa, Igbo, Kinyarwanda, Luganda, Luo, Nigerian-Pidgin, Swahili, Wolof and Yorùbá), as well as the ConLL-2003 English dataset. The task involves classifying named entities into nine pre-defined categories. The MasakhaNER dataset contains labeled entities for each language.

GLUE The Sequence Classification (SC) task relies on the The General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018). It involves nine sentence-level understanding tasks (CoLA, SST-2, MRPC, QQP, STS-B MNLI-M/MM, QNLI, RTE, WNLI) in English, across three categories: single-sentence tasks, similarity and paraphrase tasks, and inference tasks.

TyDiQA-GoldP To assess the ability of the model to perform Question Answering (QA), the TyDiQA-GoldP dataset was selected (Clark et al., 2020). It contains nine typologically diverse languages (English, Arabic, Bengali, Finnish, Indonesian, Korean, Russian, Swahili, Telugu). The dataset contains questions written by native speakers, passages with relevant information, and answers provided as short spans of text within the passage. Unlike the primary task, the Gold Passage task focuses more on locating the exact answer within a given context.

3.2 Model Architecture

PIXEL processes text as images that are rendered using the PyGame[1] renderer to accommodate multiple scripts. Each rendered image is converted into a sequence of patches, resulting in 529 non-overlapping patches, with a size of $16*16$ pixels. A ViT-based encoder encodes visible patches and the CLS tokens through patch, positional, and CLS embeddings. During pretraining, the system applies random masking to 25% of the patches and employs a decoder to reconstruct the masked regions through a regression-like method. The decoder is then finetuned on downstream tasks by replacing the reconstruction objective with task-specific heads.

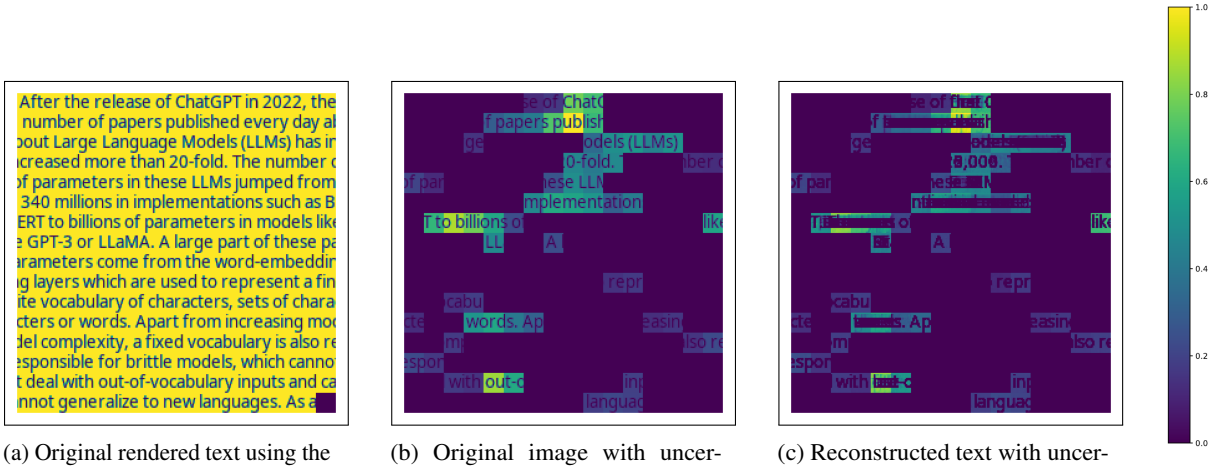
The English PIXEL which serves as a base for the experiments described in the next section is pretrained on a rendered version of English Wikipedia and BookCorpus (Zhu et al., 2015). For more details about the PIXEL pretraining routine, refer to the implementation[2] of Rust et al. (2022).

3.3 Uncertainty Quantification

Monte Carlo Uncertainty The first method used to quantify epistemic uncertainty at the patch level is Monte Carlo (MC) Dropout. The input is a rendered `image` $\in \mathbb{R}^{16\times16\times3}$ with a sequence length of 256 pixels, and the goal is to obtain an uncertainty map $U \in \mathbb{R}^{16\times16\times3}$, containing the uncertainty for each patch. For this, the model is used in 100 forward passes to compute a series of predictions P, which contain per-pixel logits. Then, the mean prediction is created by averaging these logits, resulting in the reconstructed text. A standard deviation (SD) image is obtained by computing the SDs of the predictions for each pixel. Since each patch has a dimension of 16×16 pixels, the per-patch uncertainty is defined by averaging the predictions of all SD values inside a patch, and each pixel inside the patch is assigned that value. Finally, the uncertainty map U is a collection of patches representing the overall uncertainty of its pixels. For visualization purposes, the uncertainty map is overlaid on top of the original image, as well as on the reconstructed text. An overview of this routine is presented in Algorithm 1 of Appendix C.

An overall mean uncertainty value ($\bar{\sigma}$) is also computed to measure uncertainty at the image level (Equation 3.1), where H and W refer to the height

[1] https://www.pygame.org/
[2] https://github.com/xplip/pixel



(a) Original rendered text using the PyGame renderer.

(b) Original image with uncertainty.

(c) Reconstructed text with uncertainty.

Figure 2.1: Example of uncertainty quantification at the patch level for an image containing text from the introduction of this paper. Brighter colors indicate more uncertainty.

and width of the image.

$$\bar{\sigma} = \frac{1}{H \times W} \sum_{h=1}^{H} \sum_{w=1}^{W} \sigma(h, w) \tag{3.1}$$

Additionally, we compute two loss functions during the MC inference: the normalized MSE loss (Equation 3.2) used during pretraining and the normalized Gaussian Negative Log-Likelihood (GNLL) loss (Equation 3.3), where $eps = 1e - 6$ is a clamp value used for stability. Unlike the MSE, the GNLL loss accounts for epistemic uncertainty, by incorporating the variance of the predicted distribution.

$$\text{MSE} = \frac{1}{H \times W} (\text{pred} - \text{img})^2 \tag{3.2}$$

$$\text{GNLL} = \log(\max(\text{var}, \text{eps})) + \frac{(\text{pred} - \text{img})^2}{\max(\text{var}, \text{eps})} \tag{3.3}$$

We study uncertainty across tasks: NER (MasakhaNER 1.0), SC (GLUE), and QA (TyDiQA-GoldP), and scripts – as one of the main challenges in NLP is building reliable models that can scale up to real-world applications where many scripts are often encountered. Additionally, we carry out a calibration analysis to examine the relationship between model performance and uncertainty across tasks. The performance is measured using Root Mean Square Error (RMSE $= \sqrt{\text{MSE}}$, Equation 3.2), while uncertainty is quantified using MC standard deviation. The goal is to evaluate how well the predicted uncertainty values align with actual performance errors across the different scripts and languages.

Attention Visualization To visualize attention in the PIXEL encoder, a square attention grid $A \in \mathbb{R}^{L \times H \times N_{\text{patches}}{}^2}$ is created for the encoded patches, where L is the number of attention layers and H is the number of heads in each layer. An example is presented in Figure 3.1. This shows model-level attention across all layers and heads for a particular input image. Each cell $A(l, h)$ in this grid visualizes the neuron-level attention weights for a specific head h and layer l. Then, each patch in the attention cell attends to the other patches in the sequence according to the dot product between the query (of the *attender* patch) and the key (of the *attended* patch). The weights are averaged over 100 Monte Carlo forward passes. Considering the increased dimensionality of the attention cell, only the first 16 patches are visualized, resulting in an image with 16×16 patches.

Ensemble Learning To solve the *Extractive Question-Answering* task, four learner models are finetuned on each of the 9 languages of the TyDiQA-GoldP (Section 3.1) dataset, resulting in 36 total models. Each model is trained on the `train` split of a language in the dataset and evaluated on the `validation` split of the same language. There are four main steps to be followed to compute the final prediction for an input question. In a regular non-ensemble setting, there is only one finetuned model that dictates the output answer for each example. In the ensemble learning framework, each model M_i is applied to the input question q to

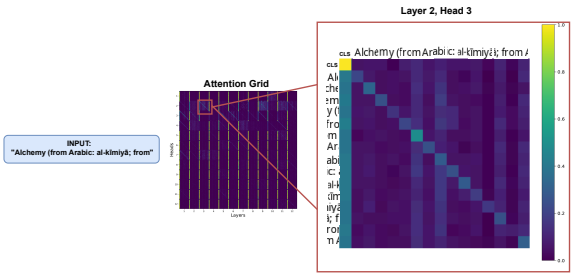


Figure 3.1: Model-level (attention grid) and neuron-level (layer 2, head 3) views of attention in the PIXEL model for a short input text from the English Wikipedia. The attention grid contains 12 attention layers with 12 attention heads each.

obtain the candidate answers with corresponding confidence probability values. To reduce the pool of candidates, only the predictions that appear in all models are kept. The average confidence conf_c is computed for each candidate across all models. Finally, the candidate with the highest confidence is selected.

In the *Named Entity Recognition task*, five learner models are finetuned on each of the 10 languages of the MasakhaNER 1.0 dataset (Adelani et al., 2021), resulting in 50 total models. Each model is trained on the `train` split of a language in the dataset and evaluated on the `test` split of the same language. The task involves assigning a label to each token from a list of 9 predefined classes. Their predicted logits are averaged and combined into one value for each class. The final label is computed as shown in Equation 3.4, where L is the set of labels (classes) and k is the number of models.

$$\text{label} = \arg\max_{l \in L} \left(\frac{1}{k} \sum_{i=1}^{k} \text{logits}_{i,l} \right) \tag{3.4}$$

During the ensemble experiment, only the values of the batch size (BSZ), learning rate (LR), dropout probability (DP), and the seed are changed. For more details about the finetuning configuration and routine, refer to Tables C.3 and C.2.

4 Results

4.1 Monte Carlo Uncertainty

Uncertainty Across Datasets The distribution of MC uncertainty is presented in Figure 4.1 (left), suggesting that GLUE achieves the highest overall uncertainty, which indicates that pixel-level uncertainty increases with text that has more semantic complexity, as it is the case in sentiment classification, semantic similarity or textual entailment tasks.

In terms of the mask ratio R, the plot indicates that lower values (0.1 to 0.3) generally correspond to lower uncertainty across all datasets, hinting that less masking leads to more certain predictions. In this case, the largest part of the data is concentrated between uncertainty values of 0.15 and 0.25. As the mask ratio increases, the distribution becomes more spread out.

The results from Figure 4.2 (left) indicate that the loss increases with the mask ratio. This is expected as the model was trained to reconstruct the image patches with a mask ratio of $R = 0.25$. There is also a wide performance gap between the sequence classification task (GLUE) and the rest of the tasks, which can be attributed to language. The GLUE dataset contains English text, the language the PIXEL model was pretrained on, while TyDiQA-GoldP and MasakhaNER are multilingual datasets.

Uncertainty Across Scripts The overall trends (right) show that Ge'ez, Chinese Characters, Arabic, and Korean scripts exhibit high uncertainty (Figure 4.1, right) and high mean loss (Figure 4.2, right), and the increase is more pronounced at mask ratios above 0.6. The Latin and Cyrillic scripts are increasing more gradually with a sharper uptick around 0.8 – 0.9. The main script found in the pre-training datasets (English Wikipedia and the BookCorpus) is Latin, and there is a high overlap between Latin and Cyrillic characters, given that both scripts share Greek as a common ancestor. However, the uncertainty in the Cyrillic script is lower, compared to Latin. The scripts with the highest MC uncertainty are Ge'ez and Chinese Characters, both of which are visually quite distinct from the Latin script.

Calibration Analysis To further study the relationship between performance and uncertainty, Figure 4.3 depicts a hexbin plot with marginal distributions, where the Root Mean Squared Error (RMSE) loss is plotted against the SD uncertainty from the MC experiments. The x-axis represents the aggregated per-image standard deviation (uncertainty) of the model after 100 Monte Carlo samples. The RMSE measures the average of the actual errors between the true pixel values and the predicted values. Inside each hexagon, the color intensity corresponds to the density of data points within that hexagon. Therefore, darker regions indicate a

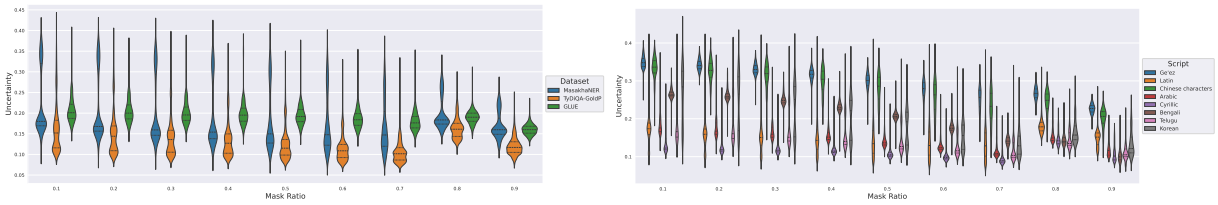


Figure 4.1: The distribution of the MC Uncertainty across the different datasets (left) and scripts (right) for each mask ratio value R.

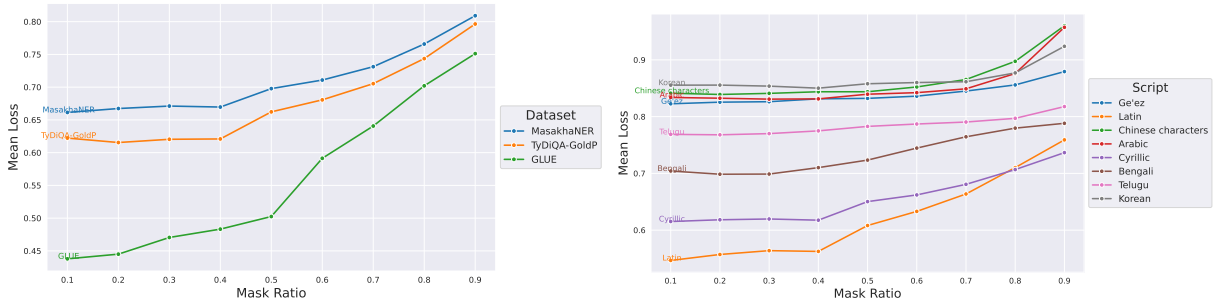


Figure 4.2: The MSE loss across the different datasets (left) and scripts (right) for each mask ratio value R.

higher density of data points. There is a high density of points in the top left corner, which suggests that the model underestimates its performance. In other words, many examples are associated with high loss but low uncertainty.

The distribution of the points for all three datasets (MasakhaNER, TyDiQA-GoldP, and GLUE) is shown in the calibration plot from Figure 4.4. The highest level of overconfidence is associated with the question-answering task in TyDiQA-GoldP. However, there seems to be a subgroup of points for which the uncertainty is high. The points in the MaskhaNER dataset fall under the category of high uncertainty and high loss. The GLUE data is located between 0.15 and 0.3 on the uncertainty range and contains several examples showing decreased loss. While the model can be considered to be underestimating uncertainty with this group, the majority of the data still fall over the main diagonal, indicating an underestimation of uncertainty.

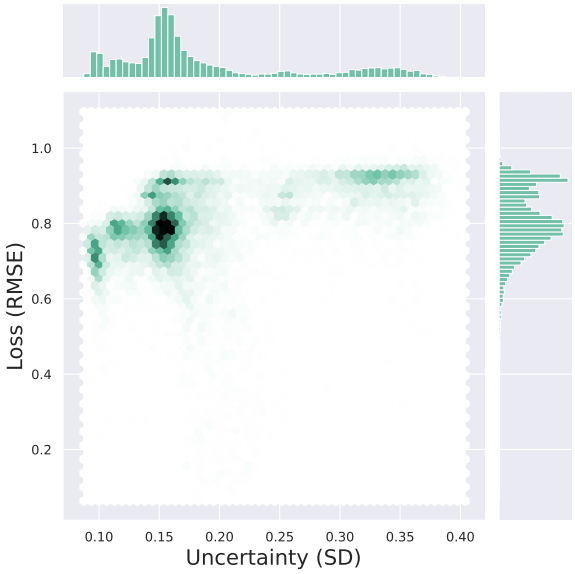


Figure 4.3: Calibration hexbin plot showing the RMSE loss in terms of the MC uncertainty.

Visualizing Uncertainty in Text Reconstruction Figure 2.1 shows (a) the original rendered English text generated with the PyGame text renderer, (b) the original image overlaied with per-patch uncertainty and (c) the reconstructed text overlaied with per-patch uncertainty. Bright yellow patches suggest larger variations in predictions. This can be observed in the larger masked segments of patches from the first 6 lines of the image, as well as in lines 12 and 15. These segments also translate to less accurate reconstructions, as seen on the corresponding rows of the reconstructed image. On the other hand, smaller segments of patches (which appear darker in the image) are associated with lower uncertainty and are reconstructed more accurately. These patches often contain shorter sequences of letters. In terms of the mistakes, the model fails to reconstruct patches with numerals, such as *20-fold*. Still, it appears to understand that the most suitable prediction given the context is a number (the model predicts *20,000*). Moreover, longer and less frequent words such as *implementation* and *pub-*

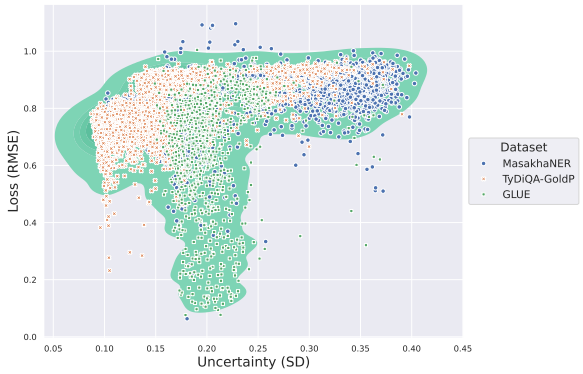


Figure 4.4: Calibration kernel density estimate plot showing the RMSE loss in terms of the MC uncertainty across the three datasets.

lish, as well as punctuation marks (used in *(LLMs)*) appear to produce more variation in the prediction, given the increased uncertainty.

4.2 Attention Visualization

Each cell in the attention grids (Figure 4.5) shows the attention weights for the first 16 patches of a specific head h and layer l in the selected examples. The first four layers appear to encode the highest amount of visual information, given the high activation of the patches. Across all heads and layers of both examples, the attention weight corresponding to the CLS patch is high, as it contains the aggregate representation of the input patch sequence. There is a clear difference in the distribution of attention between the examples. The top 1 performer (Nigerian Pidgin) exhibits high activation on the diagonal at the neuron level, meaning that patches are attending to themselves, possibly to retain positional and contextual information. The Igbo example does not show the same pattern, rather a subset of dominant patches attend to the remaining ones.

4.3 Ensemble Learning

Extractive Question Answering The results of the ensemble QA model are presented in Table 4.1, which shows the weighted F1 score across all languages in the TyDiQA-GoldP dataset. These findings are compared with the results obtain by Rust et al. (2022), following the same experimental setting. Overall, the ensemble learning method improves the performance in the extractive QA task for 6 out of the 8 languages. The average F1 score (excluding the ENG data) for the ensemble configuration is higher with 1.7 points than in the case of the regular PIXEL model. In terms of the individual languages, there is a high improvement for Indonesian (4.3 points), Russian (2.8 points), and Arabic (2.2 points), suggesting that combining multiple learners can improve performance regardless of script.

Figure 4.6 presents the confidence distribution of the best answers in the ensemble model for all languages in the dataset. In general, the confidence is in the range $0.2 - 0.4$ across the majority of languages, with some distributions indicating slightly higher confidence, as in the case of Finnish, Indonesian, and Swahili. Lower confidence values can be seen in Korean and Bengali. These observations are in line with the previous findings on performance.

Named Entity Recognition The results of the ensemble NER model are presented in Table 4.2, showing the weighted F1 score across the MasakhaNER 1.0 dataset. Due to hardware limitations at runtime, the *ENG* data is not included. For comparison, the results are shown against the values obtained by Rust et al. (2022). In general, ensemble learning improves the performance significantly for all 9 languages, resulting in scores higher than 90. This is also the case for languages that were previously associated with a low score, such as Amharic (*AMH*). The F1 score gap is 24.3 points in favour of the ensemble method, suggesting that ensemble learning improves the comprehension of long-term dependencies in NER tasks.

5 Discussion

This work showed that it is possible to integrate uncertainty quantification methods and measure calibration in the context of visual text models. These methods include Monte Carlo Dropout at the patch level, with the observation that more work should be directed towards finding more effective ways of aggregating and visualizing uncertainty across longer patch sequences. Attention based methods can also be used to gain insights into how these models encode information, but there remains the debate about whether or not attention counts as an explanation (Bibal et al., 2022). Still, this debate falls outside the scope of this research. Ensemble learning with a low number of individual learners can also be used successfully to improve both performance and confidence.

The results in the MC Uncertainty experiment generally indicate high uncertainty for a high mask ratio. Still, the most optimal value is a mask ratio of 50%, representing a reasonable trade-off between

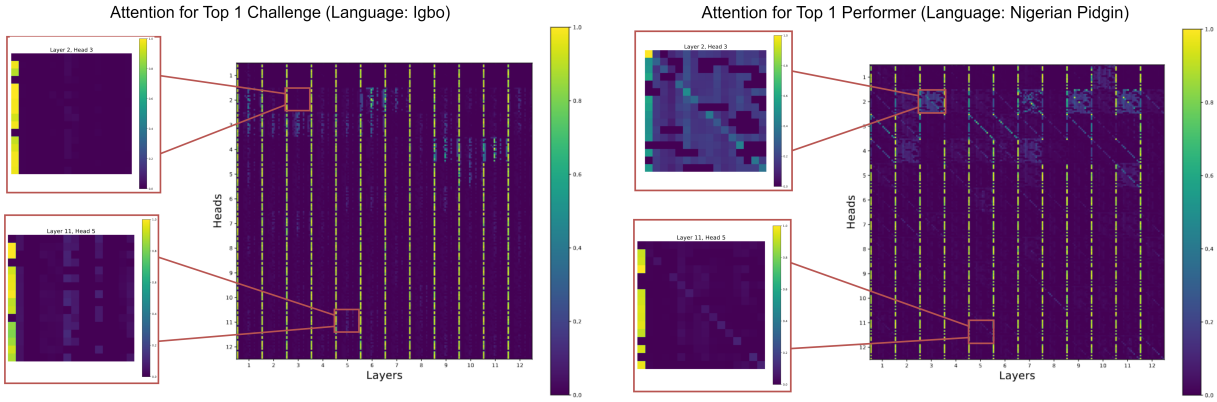


Figure 4.5: Model-level and neuron-level views of attention for the top 1 challenge (left, highest loss value) and performer (right, lowest loss value) in terms of the GNLL loss across all datasets.

	ARA	BEN	FIN	IND	KOR	RUS	SWA	TEL	ENG	**AVG**
PIXEL	57.3	**36.3**	58.3	63.6	26.1	50.5	65.9	63.4	61.7	52.3
Ensemble	**59.5**	35.1	**59.6**	**67.3**	**27.1**	**53.3**	**67.1**	63.4	**62.1**	**54.0**

Table 4.1: The results of the QA task. The ensemble learning model finetuned on the TyDiQA-GoldP dataset is compared with the values reported by (Rust et al., 2022). The metric shown is the F1 score, computed on the `validation` split of the data. The AVG score excludes ENG, as required (Clark et al., 2020).

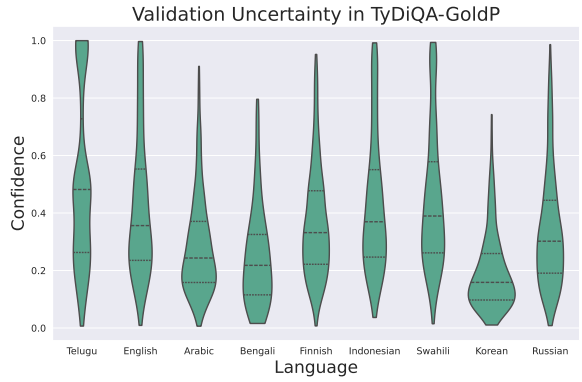


Figure 4.6: Confidence distribution across all languages in the TyDiQA-GoldP dataset for the ensemble model.

uncertainty and loss.

Scripts such as Latin are less uncertain, indicating that multilingual pretraining is necessary. Instead of language, one can focus on introducing a new script, as evidence suggests that there exists knowledge transfer between scripts like Latin and Cyrillic. For example, finetuning on one language such as Chinese might benefit performance in other languages like Korean or Amharic. This approach is more robust than traditional LLMs, where the transfer of learning happens under stricter conditions, for instance when languages share syntactic structures or when there is a significant overlap between vocabularies.

Ensemble learning can be applied successfully to improve performance and calibration in pixel-based language models. The evaluation shows higher F1 scores for 17 of the 19 tested languages across two tasks. The models become more robust and can overcome individual weaknesses by aggregating predictions from multiple learners using hyperparameter tuning. Additionally, ensemble learning improves calibration through better error diversification and data representation.

6 Conclusions and Future Work

The findings of this study indicate that pixel-based language models represent a viable and lightweight solution to traditional language modeling, even for tasks that require semantic understanding of text. Their reliability and explainability can also be improved through uncertainty quantification methods, as shown during the experiments. Future research should focus on perfecting the existing techniques and exploring new ways of understanding the inner workings of models that enccde text as visual representation.

One point to be explored in future works on text reconstruction is the idea of pixels-as-tokens in the context of the Pixel Transformer (PiT) model, introduced by (Nguyen et al., 2024). Instead of training the model to perform patch reconstruction, PiT treats each pixel as a token and the reconstruction happens at the pixel level. Evidence suggests

	AMH	HAU	IBO	KIN	LUG	LUO	PCM	SWA	WOL	YOR	**AVG**
PIXEL	47.7	82.4	79.9	64.2	76.5	66.6	78.7	79.8	59.7	70.7	70.7
Ensemble	**90.2**	**97.1**	**96.1**	**93.9**	**95.5**	**93.1**	**97.1**	**96.1**	**95.8**	**95.2**	**95**

Table 4.2: The results of the NER task. The ensemble learning model finetuned on the MasakhaNER 1.0 dataset is compared with the values reported by (Rust et al., 2022). The metric shown is the F1 score, computed on the `test` split of the data.

that this method completely removes locality as in inductive bias. This can potentially improve long-term context comprehension in the proposed approach, as the current findings indicate that the reconstruction of characters depends on neighboring pixels. Additionally, the finetuning pipeline can be expanded to more complex semantic tasks, such as summarization, open-ended question answering where the answer is not always explicitly mentioned in the context, and text generation (Li et al. (2023) introduced a new method for text generation using GlyphDiffusion). To improve model calibration, post-hoc methods like temperature scaling can be used either separately or in combination with Monte Carlo (Laves et al., 2019). During pretraining, the Cross-Entropy loss can be replaced by the Focal Loss, which is effective in calibration models trained on imbalanced datasets (Wang et al., 2022).

Ethical Considerations

The aim of this study is to shed light on how pixel-based models encode uncertainty. We consider that an explainability analysis should be a prerequisite for any new language model, as this increases users' trust that the technology works as intended and it is not harmful.

In order for this research to exist, we made use of the pretrained PIXEL model provided by Rust et al. (2022). One of the datasets that PIXEL has been pretrained on is the BookCorpus (Zhu et al., 2015) which is well-known for its problematic content and copyright violantions (Bandy and Vincent, 2021). BookCorpus contains books self-published by authors, which did not explicitly consent to including their books in a LLM training dataset, and were not compensated in any way. Moreover, many books contain copyright restrictions which forbid the redistribution of content. Senstive content has also been identified in the data, such as books marked for adult audiences, containing terms and phrases associated with gender discrimination. We acknowledge that by using models trained on problematic data, we risk to further propagate biases. However, these models and datasets are very popular and they cannot be ignored. For this reason, we consider that studying how they work and attempting to explain and interpret them is a goal worth pursuing.

Our paper has a strong focus on language variety, as we explore uncertainty across 18 languages. However, the majority of our fine-tuning data comes from English (as seen in Figure B.1 from Appendix B). This leads to lower performance and less accurate representation in low-resource languages. Once again, this issue boils down to the data available for LLM training, which should ideally be more balanced and representative across diverse linguistic contexts.

Code

We provide the complete implementation for running our experiments on Github, at `https://github.com/stefania-radu/pixel-semantic`.

References

David Ifeoluwa Adelani, Jade Abbott, Graham Neubig, Daniel D'souza, Julia Kreutzer, Constantine Lignos, Chester Palen-Michel, Happy Buzaaba, Shruti Rijhwani, Sebastian Ruder, et al. 2021. Masakhaner: Named entity recognition for african languages. *Transactions of the Association for Computational Linguistics*, 9:1116–1131.

David Aldón Mínguez, Marta Ruiz Costa-Jussà, and José Adrián Rodríguez Fonollosa. 2016. Neural machine translation using bitmap fonts. In *Proceedings of the EAMT 2016 Fifth Workshop on Hybrid Approaches to Translation (HyTra)*, pages 1–9.

Jack Bandy and Nicholas Vincent. 2021. Addressing "documentation debt" in machine learning research: A retrospective datasheet for bookcorpus. *Preprint*, arXiv:2105.05241.

Adrien Bibal, Rémi Cardon, David Alfter, Rodrigo Wilkens, Xiaoou Wang, Thomas François, and Patrick Watrin. 2022. Is attention explanation? an introduction to the debate. In *Proceedings of the*

60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3889–3900.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Felix Busch, Lena Hoffmann, Christopher Rueger, Elon HC van Dijk, Rawen Kader, Esteban Ortiz-Prado, Marcus R Makowski, Luca Saba, Martin Hadamitzky, Jakob Nikolas Kather, et al. 2025. Current applications and challenges in large language models for patient care: a systematic review. *Communications Medicine*, 5(1):26.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. 2020. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470.

Falcon Z Dai and Zheng Cai. 2017. Glyph-aware embedding of chinese characters. *arXiv preprint arXiv:1709.00028*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Jakob Gawlikowski, Cedrique Rovile Njieutcheu Tassi, Mohsin Ali, Jongseok Lee, Matthias Humt, Jianxiang Feng, Anna Kruspe, Rudolph Triebel, Peter Jung, Ribana Roscher, et al. 2023. A survey of uncertainty in deep neural networks. *Artificial Intelligence Review*, pages 1–77.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, Xiner Li, Zhengliang Liu, Yixin Liu, Yijue Wang, Zhikun Zhang, Bertie Vidgen, Bhavya Kailkhura, Caiming Xiong, Chaowei Xiao, Chunyuan Li, Eric Xing, Furong Huang, Hao Liu, Heng Ji, Hongyi Wang, Huan Zhang, Huaxiu Yao, Manolis Kellis, Marinka Zitnik, Meng Jiang, Mohit Bansal, James Zou, Jian Pei, Jian Liu, Jianfeng Gao, Jiawei Han, Jieyu Zhao, Jiliang Tang, Jindong Wang, Joaquin Vanschoren, John Mitchell, Kai Shu, Kaidi Xu, Kai-Wei Chang, Lifang He, Lifu Huang, Michael Backes, Neil Zhenqiang Gong, Philip S. Yu, Pin-Yu Chen, Quanquan Gu, Ran Xu, Rex Ying, Shuiwang Ji, Suman Jana, Tianlong Chen, Tianming Liu, Tianyi Zhou, William Wang, Xiang Li, Xiangliang Zhang, Xiao Wang, Xing Xie, Xun Chen, Xuyu Wang, Yan Liu, Yanfang Ye, Yinzhi Cao, Yong Chen, and Yue Zhao. 2024. Trustllm: Trustworthiness in large language models. *Preprint*, arXiv:2401.05561.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Max-Heinrich Laves, Sontje Ihler, Karl-Philipp Kortmann, and Tobias Ortmaier. 2019. Well-calibrated model uncertainty with temperature scaling for dropout variational inference. *arXiv preprint arXiv:1909.13550*.

Junyi Li, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. Renderdiffusion: Text generation as image generation. *arXiv preprint arXiv:2304.12519*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Duy-Kien Nguyen, Mahmoud Assran, Unnat Jain, Martin R Oswald, Cees GM Snoek, and Xinlei Chen. 2024. An image is worth more than 16x16 patches: Exploring transformers on individual pixels. *arXiv preprint arXiv:2406.09415*.

Phillip Rust, Jonas F Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. 2022. Language modelling with pixels. *arXiv preprint arXiv:2207.06991*.

Elizabeth Salesky, David Etter, and Matt Post. 2021. Robust open-vocabulary translation from visual text representations. *arXiv preprint arXiv:2104.08211*.

Baohua Sun, Lin Yang, Patrick Dong, Wenhan Zhang, Jason Dong, and Charles Young. 2018. Super characters: A conversion from sentiment classification to image classification. *arXiv preprint arXiv:1810.07653*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ada Wan. 2021. Fairness in representation for multilingual nlp: Insights from controlled experiments on conditional language modeling. In *International Conference on Learning Representations*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Cheng Wang, Jorge Balazs, György Szarvas, Patrick Ernst, Lahari Poddar, and Pavel Danchenko. 2022. Calibrating imbalanced classifiers with focal loss: An empirical study. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 145–153.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of bert. *arXiv preprint arXiv:1904.09077*.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.

A Limitations

Some limitations of this method include the hardware and training time required to train multiple models. Nevertheless, PIXEL has 20% fewer parameters than BERT, so an ensemble of PIXEL models remains less complex than the BERT variant and significantly more lightweight than models like GPT.

The current study is subject to several limitations. Firstly, the way uncertainty is computed at the image level during the MC experiments can be more reliable. At the moment, uncertainty is averaged across all pixels in an image. However, this does not account for the difference in span length, as some sequences of patches are longer than others. Quantifying uncertainty as an average for each span length in the image could bring more insights into how the model encodes long-term dependencies. Secondly, the information in the attention plots should be aggregated so that all patches are visible at once, while keeping a reasonable image size. Using the current method, visualizing all 256 patches across the 144 attention structures would result in a very large and difficult to interpret image. Regarding the calibration analysis, it is not completely clear that the two measurements of performance (loss vs. MC uncertainty during the pretraining stage and F1 score vs. confidence during finetuning) are quantifying the same underlying metric. For this reason, additional testing should be performed to establish the exact effect size of ensemble learning on model calibration. Moreover, more insights are necessary to establish the trade-off between computational cost, environmental impact and performance gains when training an ensemble of learners compared to a single model.

While it is possible to visualize the attention mechanism in pixel-based language models, there are some comments to be made about this. Unlike traditional language models like BERT where each token represents a meaningful unit and the relationship between two tokens can be understood intuitively, the patches in pixel-based language models cannot be mapped back to text chunks. This makes it more challenging to interpret how attention is paid to the different patches and what are the implications of these connections in the context of the entire model. Moreover, given the large number of attention structures and the image dimensions, visualizing attention for all patches simultaneously becomes very difficult.

B Data Details

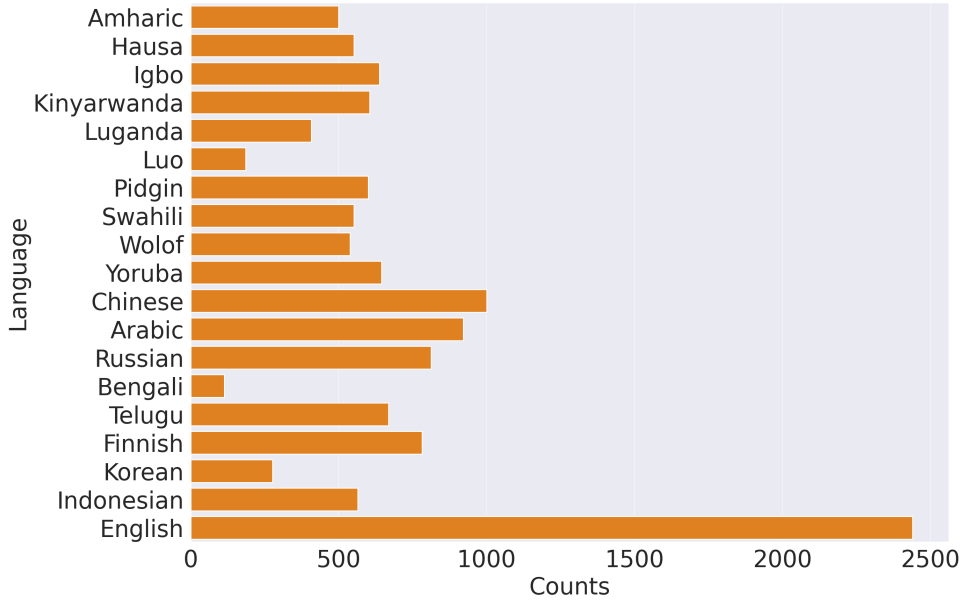


Figure B.1: Distribution of languages used throughout the experiments.

C Experiments Details

Language	ISO 639-3	Language Family	Script
Amharic	AMH	Afro-Asiatic	Ge'ez
Arabic	ARA	Afro-Asiatic	Arabic
Bengali	BEN	Indo-European	Bengali
English	ENG	Indo-European	Latin
Finnish	FIN	Uralic	Latin
Hausa	HAU	Afro-Asiatic	Latin
Igbo	IBO	Niger-Congo	Latin
Indonesian	IND	Austronesian	Latin
Kinyarwanda	KIN	Niger-Congo	Latin
Korean	KOR	Koreanic	Korean
Luganda	LUG	Niger-Congo	Latin
Naija Pidgin	PCM	English Creole	Latin
Russian	RUS	Indo-European	Cyrillic
Swahili	SWA	Niger-Congo	Latin
Telugu	TEL	Dravidian	Telugu
Wolof	WOL	Niger-Congo	Latin
Yorùbá	YOR	Niger-Congo	Latin

Table B.1: An overview of languages used during the experiments. The original PIXEL model is pretrained on English only.

Experiment	Data	Hyperparameters	Metrics
MCU Tasks	NER (MasakhaNER 1.0), SC (GLUE), QA (TyDiQA-GoldP)	$R \in \{0.1, 0.2, \ldots, 0.9\}$, $S \in \{1, 2, \ldots, 6\}$, $W = \{0, 0, \ldots, 0, 1\}, \lvert W\rvert = \lvert S\rvert$	MSE GNLL Uncertainty ($\bar{\sigma}$)
MCU Scripts	Latin, Ge'ez, Chinese Characters, Arabic, Cyrillic, Bengali, Telugu, Korean	$R \in \{0.1, 0.2, \ldots, 0.9\}$, $S \in \{1, 2, \ldots, 6\}$, $W = \{0, 0, \ldots, 0, 1\}, \lvert W\rvert = \lvert S\rvert$	MSE GNLL Uncertainty ($\bar{\sigma}$)
VU	Nigerian Pidgin, Igbo	$R = 0.25$, $S = 6$, $W = \{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$	GNLL Uncertainty ($\bar{\sigma}$)
CA	NER (MasakhaNER 1.0), SC (GLUE), QA (TyDiQA-GoldP)	$R = 0.25$, $S = 6$, $W = \{0.2, 0.4, 0.6, 0.8, 0.9, 1\}$	RMSE Uncertainty ($\bar{\sigma}$)

Table C.1: Overview of the MC Uncertainty experiments. MCU = Monte Carlo Uncertainty; VU = Visualizing Uncertainty; CA = Calibration Analysis.

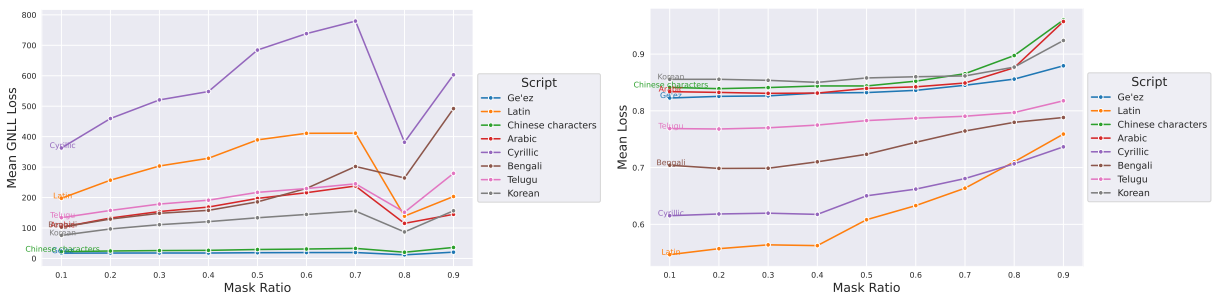


Figure C.1: Mean MSE Loss (left) and GNLL Loss (right) across the different scripts for each mask ratio value R.

Algorithm 1 Patch-level Uncertainty with MC Dropout

Require: Rendered image I, model M, # MC samples $N_{\text{MC}} = 100$, dropout rate $p = 0.1$, patch size $P = 16$
Ensure: Uncertainty map U

Activate dropout in M
for $i \in \{1, \ldots, N\}$ **do**
    $P_i \leftarrow M(I, p)$ ▷ Compute predictions P with dropout
end for
Initialize μ and σ with the shape of I
for each pixel (x, y) **do**
    $\mu(x, y) \leftarrow \frac{1}{N} \sum_{i=1}^{N} P_i(x, y)$
    $\sigma(x, y) \leftarrow \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_i(x, y) - \mu(x, y))^2}$
end for
Initialize U with the shape of I
for each patch (i, j) in σ **do**
    $\sigma_{\text{patch}} \leftarrow \frac{1}{P^2} \sum_{x=i}^{i+P-1} \sum_{y=j}^{j+P-1} \sigma(x, y)$ ▷ Compute σ per patch
    for $(x, y) \in \{(i, j), \ldots, (i + P - 1, j + P - 1)\}$ **do**
        $U(x, y) \leftarrow \sigma_{\text{patch}}$ ▷ Assign σ_{patch} to all pixels in the patch
    end for
end for
return U

Algorithm 2 Ensemble QA Prediction

Require: k models $\{M_1, M_2, \ldots, M_k\}$, input question q
Ensure: Final answer $\hat{a}$ for the question q

$\mathcal{C} \leftarrow \emptyset$
for each model M_i in $\{M_1, M_2, \ldots, M_k\}$ **do**
    $\mathcal{A}_i \leftarrow M_i(q)$ ▷ Get candidate answers and their confidences
    for each candidate a_j in $\mathcal{A}_i$ **do**
        $\mathcal{C} \leftarrow \mathcal{C} \cup \{a_j\}$
    end for
end for
$\mathcal{C} \leftarrow \left\{ c \mid \sum_{i=1}^{k} \mathbf{1}_{c \in \mathcal{A}_i} = k \right\}$ ▷ Keep the candidates that appear in all models
for each candidate c in $\mathcal{C}$ **do**
    $\text{conf}_c \leftarrow \frac{1}{k} \sum_{i=1}^{k} \text{confidence}_{M_i}(c)$ ▷ Compute average confidence
end for
$\hat{a} \leftarrow \arg\max_{c \in \mathcal{C}} \text{conf}_c$ ▷ Select candidate with highest confidence
return $\hat{a}$

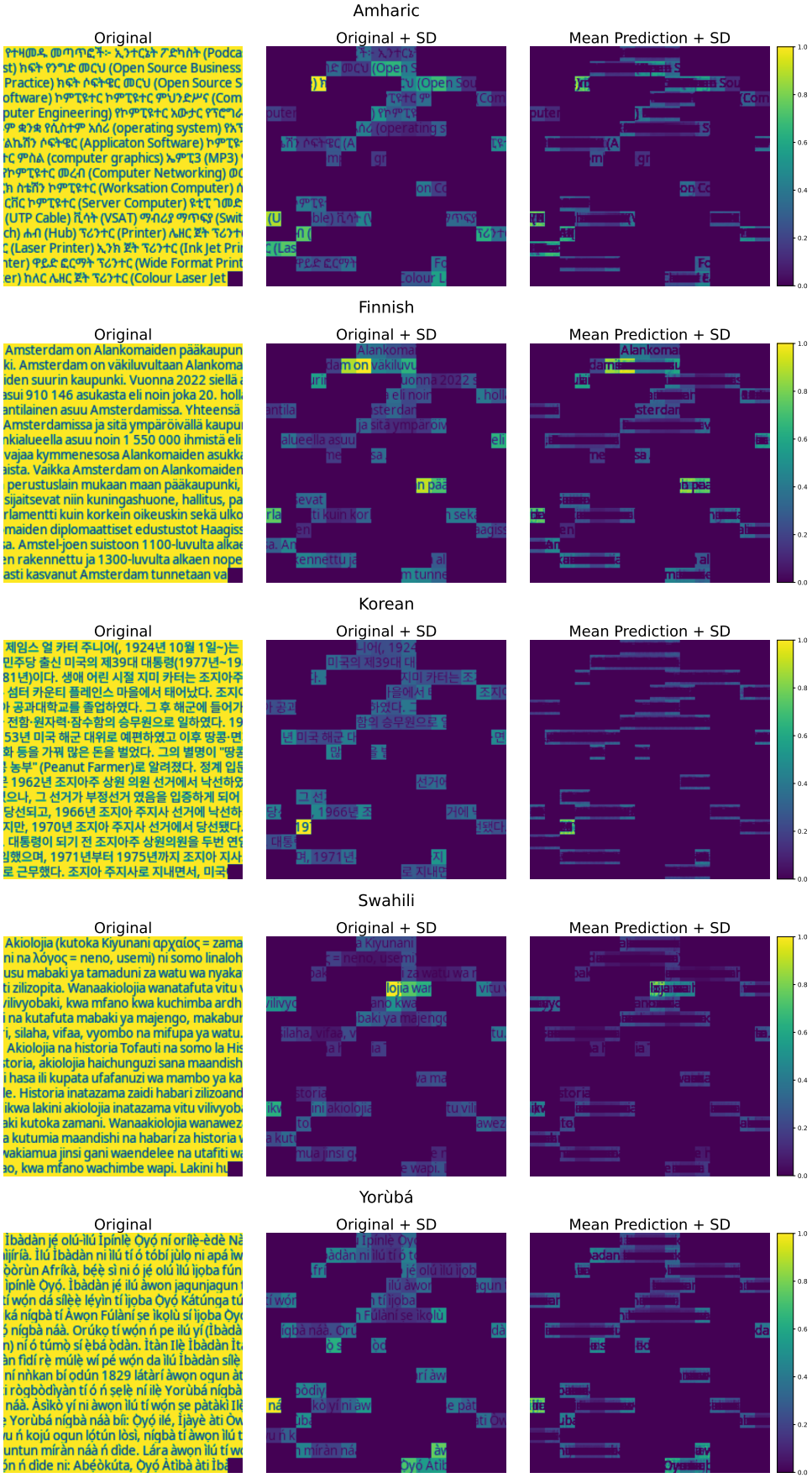


Figure C.2: Examples of uncertainty quantification at the patch-level for various languages.

Parameter	Value
Common Parameters	
Dataset name	tydiqa
Dataset config name	secondary_task
Sequence length	400
Stride	160
Question max length	128
Gradient accumulation steps	1
Max steps	20000
Number of train epochs	10
Early stopping	True
Early stopping patience	5
Evaluation metric	$F1 = \frac{2\times\text{TP}}{2\times\text{TP}+\text{FP}+\text{FN}}$
Doc stride	160
Number of best predictions	20
Model 1	
Batch size	32
Learning rate	7×10^{-4}
Dropout probability	0.15
Seed	101
Model 2	
Batch size	16
Learning rate	7×10^{-5}
Dropout probability	0.15
Seed	102
Model 3	
Batch size	8
Learning rate	7×10^{-5}
Dropout probability	0.05
Seed	103
Model 4	
Batch size	32
Learning rate	7×10^{-6}
Dropout probability	0.1
Seed	104

Table C.2: The finetuning configuration of the QA models, including the common parameters and those changed among the 4 learners.

Parameter	Value
Common Parameters	
Dataset name	masakhane-ner
Sequence length	196
Gradient accumulation steps	1
Max steps	15000
Number of train epochs	10
Early stopping	True
Early stopping patience	5
Evaluation metric	$F1 = \frac{2\times\text{TP}}{2\times\text{TP}+\text{FP}+\text{FN}}$
Model 1	
Batch size	64
Learning rate	5×10^{-5}
Dropout probability	0.1
Seed	100
Model 2	
Batch size	64
Learning rate	5×10^{-6}
Dropout probability	0.2
Seed	101
Model 3	
Batch size	32
Learning rate	5×10^{-5}
Dropout probability	0.1
Seed	102
Model 4	
Batch size	32
Learning rate	5×10^{-6}
Dropout probability	0.1
Seed	103
Model 5	
Batch size	16
Learning rate	5×10^{-5}
Dropout probability	0.2
Seed	104

Table C.3: The finetuning configuration of the NER models, including the common parameters and those changed among the 5 learners.

Confidence Calibration in Large Language Model-Based Entity Matching

Iris Kamsteeg [1], Juan Cardenas-Cartagena[1], Floris van Beers[2], Gineke ten Holt[2], Tsegaye Misikir Tashu[1], Matias Valdenegro-Toro [1]

[1]Bernoulli Institute, University of Groningen, The Netherlands, [2]Independent Researcher
ikamsteeg@ziggo.nl, t.m.tashu@rug.nl, m.a.valdenegro.toro@rug.nl

Abstract

This research aims to explore the intersection of Large Language Models and confidence calibration in Entity Matching. To this end, we perform an empirical study to compare baseline RoBERTa confidences for an Entity Matching task against confidences that are calibrated using Temperature Scaling, Monte Carlo Dropout and Ensembles. We use the Abt-Buy, DBLP-ACM, iTunes-Amazon and Company datasets. The findings indicate that the proposed modified RoBERTa model exhibits a slight overconfidence, with Expected Calibration Error scores ranging from 0.0043 to 0.0552 across datasets. We find that this overconfidence can be mitigated using Temperature Scaling, reducing Expected Calibration Error scores by up to 23.83%.

1 Introduction

Entity Resolution (ER) can be defined as the task of determining which data entries across different data sources refer to the same real-world entity. A key sub-task of ER is Entity Matching (EM), which specifically addresses the binary classification problem of determining whether pairs of data entries from different sources refer to the same entity (Christophides et al., 2020). In today's data-driven era, EM plays a critical role in various domains, including the medical field (Jaro, 1995; Méray et al., 2007), where accurate matching can improve patient care; the reconstruction of historical populations by linking birth, marriage, and death records (Bloothooft et al., 2015); and law enforcement, where matching data entries is vital for investigations and crime prevention (Dahlin et al., 2012).

The state-of-the-art methods for performing EM utilize Transformer-based architectures (Vaswani et al., 2017), pre-trained Large Language Models (LLMs) (Brunner and Stockinger, 2020; Li et al., 2020; Peeters et al., 2020; Peeters and Bizer, 2021, 2022, 2023, 2024), such as RoBERTa (Liu et al., 2019) and GPT-4 (et al., 2024).

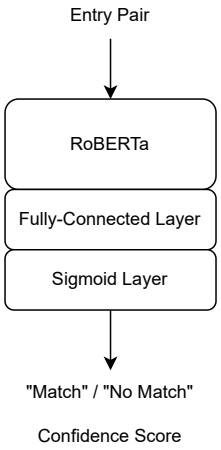


Figure 1: Overview of this research's model (without any confidence calibration methods visualised), model input and model output. In addition to classifying each entry pair as a 'match' or 'no match,' the model also generates a score that should reflect the model's confidence in its prediction.

However, while these models are successful, they have, in other Natural Language Processing tasks, shown to struggle to accurately express their confidence in predictions and can exhibit overconfidence (Desai and Durrett, 2020; Jiang et al., 2021). Ideally, a model provides information about its certainty alongside its predictions. For example, in a binary EM task, a model would output a 'match' or 'no match' prediction label alongside a probability, or confidence score, that is reliable. Refining models' predicted confidence scores to ensure that they accurately reflect the true likelihood of the predictions is called confidence calibration. While the topic of confidence calibration on LLMs has been explored (Desai and Durrett, 2020; Sankararaman et al., 2022; Chen and Li, 2024), the intersection of confidence calibration, LLMs, and the application of EM has not yet been researched. Yet, confidence

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 120–137
November 9, 2025 ©2025 Association for Computational Linguistics

calibration is important as it provides transparency over models' results (Ghahramani, 2015). For example, the distribution of confidence in a model's EM predictions can give the user insights into the model's overall reliability in the task. Precise confidence scores can also play a crucial role in guiding subsequent tasks. Moreover, confidence scores can be used to help researchers better understand a model's inner workings. Finally, they can help in improving a model: when it is clear in what specific cases a model is uncertain, it is easier to see a model's weak points and with that, possible areas for improvement.

Contributions. This paper aims to explore the confidence calibration performance of LLMs in EM and benchmark confidence calibration methods to enhance their performance. We focus on pre-trained RoBERTa (Liu et al., 2019) as the LLM of interest as it has a competitive performance among LLM models for EM (Li et al., 2020; Peeters and Bizer, 2021, 2024). In contrast to other state-of-the-earth models for EM, RoBERTa is open-source and lightweight. Our study assesses the confidence calibration performance in EM using the Expected Calibration Error (ECE) as the primary metric. We evaluate fine-tuned RoBERTa model's ECE scores both with and without the use of confidence calibration methods and investigate which methods yield the greatest improvement. Since confidence calibration methods may influence the model's predictions, we additionally examine their effect on the F_1 score to ensure that improved confidence calibration does not come at the cost of classification performance. Furthermore, we analyze confidence histograms, reliability diagrams, the Maximum Calibration Error (MCE) and the Root Mean Square Calibration Error (RMSCE). The confidence calibration methods tested are Temperature Scaling (Guo et al., 2017), Monte Carlo Dropout (Gal and Ghahramani, 2016), and Ensembles (Lakshminarayanan et al., 2017). We use the Abt-Buy, DBLP-ACM (dirty and structured) (Köpcke et al., 2010), iTunes-Amazon (dirty and structured), and Company (Konda et al., 2016) datasets, ensuring diversity in terms of data content, size and structure.

Figure 1 presents an overview of the proposed modified RoBERTa model used in this research. As shown, the goal is to obtain confidence scores that accurately reflect the model's confidence in its EM predictions. Confidence calibration methods can help improve these scores.

2 Confidence Calibration

We say that a model is well-calibrated if its prediction's confidence scores accurately reflect the probability of those predictions being correct. For EM, for example, all pairs that are predicted to match with around 0.5 to 0.6 confidence should be actual matching pairs 50 to 60% of the time. This is also referred to as the alignment between the 'predicted probability' (the confidence) and the 'empirical probability' (Naeini et al., 2015; Guo et al., 2017; Küppers et al., 2022). Generally, for a binary classification task such as EM, the 'confidence' signifies the confidence of a prediction belonging to the positive class (in the case of EM: a 'match'). The predicted probability of the positive class then needs to align with the empirical probability of the positive class. 'High confidences', in this context, generally denote predicted probabilities close to either 0 or 1, while 'low confidences' denote predicted probabilities close to 0.5.

The confidence calibration of models has been evaluated by plotting confidence histograms and reliability diagrams, and by measuring the Expected Calibration Error (ECE) (Naeini et al., 2015) or similar metrics such as the Maximum Calibration Error (MCE) (Naeini et al., 2015) and Root Mean Square Calibration Error (RMSCE) (Kumar et al., 2019). Intuitively, these scores measure the difference between the predicted probability and the empirical probability, and should therefore be minimized to optimize the confidence calibration. Compared to the ECE, the MCE is useful in production settings where reliable confidence measures are absolutely necessary due to high risks. This is due to its measure of the worst-case deviation between the predicted probabilities and the empirical probabilities. When comparing the ECE to the RMSCE, the latter places a greater emphasis on larger errors.

3 Related Work

3.1 Large Language Models for Entity Matching

Various pre-trained LLMs have shown state-of-the-art results for EM tasks. Brunner and Stockinger (2020), for example, analysed the performance of four LLMs: BERT (Devlin et al., 2019), XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019) and DistilBERT (Sanh et al., 2020), and found an increase in F_1 scores of up to 35.9% compared to state-of-the-art non-LLM methods. Other state-of-the-art results were presented by Li et al.

(2020), who introduced DITTO: an EM system that combines the use of LLMs such as BERT, DistilBERT and RoBERTa with various optimisation techniques; and Peeters et al. (2020); Peeters and Bizer (2021, 2022), who experimented with BERT and RoBERTa-SupCon for EM in the product domain.

Decoder-only models have more recently caught the attention in the field. Narayan et al. (2022) compared GPT-3 against the DITTO system. The performance of GPT-3 (Brown et al., 2020) using few-shot learning was better than DITTO's performance for four out of seven datasets. In their paper "Using ChatGPT for Entity Matching", Peeters and Bizer (2023) test the performance of ChatGPT (GPT3.5) on an EM task using product data. They find that though the results of ChatGPT on this data is generally worse compared to the results of a finetuned RoBERTa, it is beneficial that ChatGPT does not necessarily require any finetuning, and, thus, performs well on unseen data. Peeter and Bizers' study "Entity Matching using Large Language Models" (Peeters and Bizer, 2024) shows that GPT-4 (et al., 2024) especially performs well in EM tasks.

3.1.1 Confidence Calibration of Large Language Models

While in the early 2000s, simple neural networks typically produced well-calibrated probabilities in binary classification tasks (Niculescu-Mizil and Caruana, 2005), recent studies have shown that this is generally not the case for more modern neural networks. In their 2017 paper "On the Calibration of Modern Neural Networks" (Guo et al., 2017), Guo et al. showed that state-of-the-art neural networks of that time (including ResNet (He et al., 2016)), do not show a good confidence calibration at all. The researchers also indicate that miscalibration worsens as the classification error is reduced. Desai and Durrett (2020), as well as Xiao et al. (2022) explored the confidence calibration of BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) in natural language inference, paraphrase detection, sentiment analysis and commonsense reasoning tasks. While BERT and RoBERTa show less miscalibration than the models that were evaluated by Guo and colleagues, the confidence calibration of the LLMs does show room for improvement. In a study by Jiang et al. (2021), decoder-only LLMs were also shown to be generally miscalibrated and often overconfident (Jiang et al., 2021).

One of the reasons that LLMs do not seem to produce well-calibrated predictions is that they are not trained to do this as an explicit learning goal. Instead, during training, these networks are encouraged to assign high confidences, in the form of sigmoid scores, to the correct class, without regard to nuances that prediction probabilities should ideally have (Hendrycks and Gimpel, 2017).

However, various methods have been introduced to improve the confidence calibration of LLMs. These include Temperature Scaling (Guo et al., 2017), Monte Carlo Dropout (Gal and Ghahramani, 2016) and Ensembles (Lakshminarayanan et al., 2017).

4 Methods

4.1 Data

Six datasets are used in this study: Abt-Buy, DBLP-ACM-Structured, DBLP-ACM-Dirty (Köpcke et al., 2010), iTunes-Amazon-Structured, iTunes-Amazon-Dirty, and Company (Konda et al., 2016). For DBLP-ACM and iTunes-Amazon, the structured and dirty versions of the datasets contain the same entries, but for the dirty version, there is a 50% chance that an attribute value is moved to a different attribute. Table 1 presents the domains of the datasets, as well as the number of pairs for each dataset, for each split. In brackets is the percentage of positive pairs.

4.2 Model

We use RoBERTa (Liu et al., 2019), pretrained on English language, as target LLM for EM. RoBERTa was one of the first LLMs to be used for EM and performs among the best of all tested non-decoder LLMs for EM, while not using any additional optimisation techniques (Brunner and Stockinger, 2020; Li et al., 2020). We utilise Huggingface's pre-trained RoBERTa base model[1].

We adopt the setup by Li et al. (2020) to make RoBERTa suitable for EM in the proposed datasets. That is, a single fully connected layer and sigmoid output layer are added after the final layer of the pre-trained RoBERTa base model. These two added layers, together with the RoBERTa base model, constitute the EM model. The fully connected layer's parameters are randomly initialized. The RoBERTa EM model is fed pairs of entries and outputs whether or not the pairs of entries are

[1] https://huggingface.co/FacebookAI/roberta-base

Dataset name	Domain	Training pairs	Validation pairs	Testing pairs
Abt-Buy	Products	5743 (10.72%)	1916 (10.75%)	1916 (10.75%)
DBLP-ACM*	Citations	7417 (17.96%)	2473 (17.96%)	2473 (17.96%)
iTunes-Amazon*	Songs	321 (24.30%)	109 (27.78%)	109 (27.78%)
Company	Companies	67 596 (24.94%)	22 533 (25.30%)	22 503 (25.06%)

Table 1: Overview of the dataset's domains and data splits, along with the percentage of positive pairs per split between brackets. *The splits and percentages are the same for both the structured and dirty versions.

predicted as a 'match' (label 1) or 'no match' (label 0). We adopt Li et al. (2020) method of data serialization to convert structured EM data into sequences of text that can be fed to the RoBERTa model. Hyper-parameters are also taken from the paper of Li et al. (2020).

In order for the model to understand the task and the data that it is given, fine-tuning is performed on the RoBERTa base model along with the fully connected and sigmoid layers using supervised training with a binary cross-entropy loss

4.3 Confidence Calibration Methods

4.3.1 Temperature Scaling

Temperature Scaling was introduced by Guo et al. (2017) as a single-parameter version of Platt Scaling (Platt, 1999). The method is easy to realise and understand, and is time-efficient and lightweight. It has led to improvements in confidence calibration for both encoder-only and decoder-only LLMs for sentiment analysis, natural language inference, common sense reasoning, paraphrase detection, and question-answering tasks. For some datasets and tasks, the technique has resulted in ECEs that are up to ten times smaller compared to those of uncalibrated models (Guo et al., 2017; Desai and Durrett, 2020; Jiang et al., 2021; Xiao et al., 2022).

4.3.2 Monte Carlo Dropout

Monte Carlo Dropout was introduced by Gal and Ghahramani (2016) and applies dropout with probability p (Hinton et al., 2012) at inference time. It has shown to, with its regularizing effect, improve the confidence calibration of models in tasks such as sentiment analysis, natural language inference, commonsense reasoning, named entity recognition and language modeling (Xiao and Wang, 2019; Xiao et al., 2022).

In our implementation, dropout is applied to the fully connected layer of the EM model. We perform dropout for just this layer to make the confidence calibration method implementation as lightweight as possible.

4.3.3 Ensembles

Ensembles can be used for confidence calibration by separately training multiple instances of a model and using the mean probability outputs at inference time (Lakshminarayanan et al., 2017). Through their regularizing effect, Ensembles have shown to improve the confidence calibration across various tasks including sentiment analysis, natural language inference and commonsense reasoning (Xiao et al., 2022).

We apply Ensembles on the fully connected layer and the sigmoid activation layer. In this way, we minimize the number of times that entry pairs need to pass through the RoBERTa base model.

4.3.4 Experimental Setup

First, the performance of the baseline RoBERTa EM model is evaluated in terms of F_1 score and confidence calibration for all datasets. To this end, we train and test on five independently randomly initialized RoBERTa EM models. For each run, the training data are shuffled. We adopt the number of epochs specified in the code by Li et al. (2020) for all datasets. This corresponds to 40 epochs. The model checkpoint that generates the highest F_1 score on the validation set is used for testing. The sigmoid scores that the model produces for the testing set are used as baseline predicted probabilities.

Secondly, Temperature Scaling, Monte Carlo Dropout, and Ensembles are individually applied and evaluated. They are compared against each other and against the baseline.

In applying Temperature Scaling, we adopt the approach by Mukhoti et al. (2020) to find the best values for the temperature T. We use a similar approach to find the dropout value p for the Monte Carlo Dropout method. For each dataset and experiment run, T and p are determined by minimizing the ECE on the validation set through a single parameter grid-search, while avoiding any decrease in the F_1 score.

For Temperature Scaling, we take, for each trained RoBERTa model (i.e. one model per

run per dataset), the sigmoid scores on the validation set. These are scaled with temperatures $T \in \{0.1, 0.2, 0.3, 0.4, ..., 9.9, 10.0\}$. Next, the ECE is calculated over all of the scaled sigmoid scores. For each dataset and run, the T is recorded that results in the smallest ECE on the validation set. Next, these values for T are used on the corresponding testing set sigmoid scores. The final results consist of the temperatures and, most importantly, the ECEs of the test sets. Note that Temperature Scaling does not change the F_1 scores.

For Monte Carlo Dropout, we take the best RoBERTa EM models from previous experiments for each dataset and run, and apply Monte-Carlo Dropout with $p \in \{0.05, 0.10, 0.15, 0.20, ..., 0.90, 0.95\}$. For each dataset, experiment run and dropout value, the model predicts over the validation set ten times. The resulting sigmoid scores from these ten sub-runs are averaged using the mean.

For all averaged sigmoid scores, the F_1 score and ECE are calculated. For each dataset and run, the p is recorded that results in the smallest ECE on the validation set, while maintaining an F_1 score not lower than the original score without dropout. If all values of p decrease the F_1 score, a dropout value of 0.00 is recorded.

Next, for each dataset and run, these recorded values for p are used while performing inference on the corresponding test sets. Inference is performed ten times for each dataset and run using the recorded dropout probabilities. Afterwards, the means of the resulting sigmoid scores are calculated, and the F_1 scores and ECEs are computed over these means.

For Ensembles, for each dataset and experiment run, we randomly initialise the fully connected layer weights five times. For each dataset and experiment run, we then train, validate and test, using these five differently initialised models. After doing this, we compute the means over the five ensemble runs' test sets sigmoid scores. These average sigmoid scores are then used to derive the final F_1 scores and ECEs.

Evaluation for the baseline RoBERTa EM model and the confidence calibration methods occurs in terms of confidence histograms, reliability diagrams, F_1 score, ECE, MCE, and RMSCE metrics, using a number of bins = $\sqrt{|\mathcal{D}|}$, with $\mathcal{D}$ being the dataset. A paired t-test is used to assess the significance of differences between the baseline results for the Temperature Scaling and Monte Carlo Dropout methods. An unpaired t-test is used to do the same for the Ensembles method.

5 Results

Table 2 presents the mean F_1 scores, ECEs, MCEs, and RMSCEs of various confidence calibration methods, over five runs, for all datasets. It also presents the baseline confidence calibration using the RoBERTa sigmoid scores without any confidence calibration method applied.

Appendix A presents a more detailed overview of the performance of the baseline RoBERTa model in terms of F_1 score, precision, recall and inference time.

5.1 Baseline

We find that, for all datasets, the RoBERTa EM model produces either very low or very high predicted probabilities, signifying a high overall confidence (Appendix B). High confidence outputs do not necessarily signify miscalibration. The high baseline F_1 scores in Table 2, especially for the DBLP-ACM datasets, suggest that the model makes few errors and can justifiably be confident in its predictions. Still, however, we observe that the model produces very high confidence levels even for the datasets where the classification F_1 scores are around 90 or lower. Confidence histograms that separately display the distributions of correct and incorrect predictions for the datasets also suggest a miscalibration. Two of these confidence histograms are presented as examples in Figure 2. For a well-calibrated pipeline, there should be minimal overlap between the distributions of correct and incorrect predictions in such histograms. Figure 2 shows that this is not the case.

As visible in Table 2, the baseline ECEs are lowest for the DBLP-ACM datasets. These are also the datasets for which the baseline RoBERTa model achieves the highest F_1 scores. The ECEs are highest for the iTunes-Amazon, and Company datasets. While the Company datasets' ECEs may in part be due to their challenging EM data, this does not explain the iTunes-Amazon ECEs.

Since the ECE is a measure that is weighted by the number of data points, it is most influenced by the extreme prediction probabilities. After all, these occur most often. The RMSCE is, compared to the ECE, influenced more by large errors between the predicted probability and the empirical probability. The reported values for this RMSCE

Dataset	ECE	F_1	MCE	RMSCE
Baseline				
Abt-Buy	0.0193 ± 0.0018	90.81 ± 0.85	0.9305 ± 0.0469	0.0558 ± 0.0032
DBLP-ACM-S	0.0041 ± 0.0010	98.78 ± 0.40	0.7800 ± 0.2900	0.0303 ± 0.0131
DBLP-ACM-D	0.0043 ± 0.0011	98.85 ± 0.18	0.6949 ± 0.1204	0.0287 ± 0.0104
iTunes-Amazon-S	0.0391 ± 0.0064	90.53 ± 1.64	0.3085 ± 0.2024	0.0506 ± 0.0113
iTunes-Amazon-D	0.0410 ± 0.0121	91.50 ± 1.90	0.3460 ± 0.2285	0.0683 ± 0.0181
Company	0.0552 ± 0.0099	82.75 ± 0.92	0.5449 ± 0.0855	0.0967 ± 0.0177
Temperature Scaling				
Abt-Buy	↓0.0147 ± 0.0017	90.81 ± 0.85	↓0.8539 ± 0.0882	↑0.0632 ± 0.0046
DBLP-ACM-S	↓0.0036 ± 0.0011	98.78 ± 0.40	0.7580 ± 0.2031	0.0306 ± 0.0087
DBLP-ACM-D	0.0038 ± 0.0011	98.85 ± 0.18	0.7983 ± 0.2174	↑0.0312 ± 0.0085
iTunes-Amazon-S	↓0.0352 ± 0.0118	90.53 ± 1.63	0.3394 ± 0.2089	0.0415 ± 0.0226
iTunes-Amazon-D	0.0377 ± 0.0102	91.50 ± 1.90	0.4036 ± 0.3247	0.0649 ± 0.0288
Company	↓0.0424 ± 0.0102	82.75 ± 0.92	0.4551 ± 0.1137	↓0.0823 ± 0.0164
Monte Carlo Dropout				
Abt-Buy	0.0193 ± 0.0016	↓90.68 ± 0.92	0.9504 ± 0.0298	0.0574 ± 0.0037
DBLP-ACM-S	0.0038 ± 0.0010	98.83 ± 0.32	0.8716 ± 0.1538	0.0333 ± 0.0096
DBLP-ACM-D	0.0042 ± 0.0011	98.90 ± 0.21	0.7207 ± 0.1148	0.0286 ± 0.0096
iTunes-Amazon-S	0.0381 ± 0.0084	90.87 ± 1.37	0.3008 ± 0.1470	0.0495 ± 0.0096
iTunes-Amazon-D	↓0.0381 ± 0.0124	91.50 ± 1.90	0.4036 ± 0.3180	0.0718 ± 0.0235
Company	0.0543 ± 0.0085	82.75 ± 0.86	0.5137 ± 0.0928	0.0946 ± 0.0156
Ensembles				
Abt-Buy	↓0.0173 ± 0.0005	90.78 ± 0.34	↓0.8669 ± 0.0316	↑0.0672 ± 0.0031
DBLP-ACM-S	0.0057 ± 0.0023	98.89 ± 0.20	0.7914 ± 0.2040	0.0370 ± 0.0096
DBLP-ACM-D	0.0052 ± 0.0007	↓98.51 ± 0.15	↑0.8557 ± 0.1063	↑0.0439 ± 0.0026
iTunes-Amazon-S	↓0.0333 ± 0.0022	91.61 ± 0.95	↑0.6869 ± 0.1421	↑0.0948 ± 0.0176
iTunes-Amazon-D	0.0438 ± 0.0123	91.34 ± 2.52	↑0.5904 ± 0.0296	↑0.0950 ± 0.0143
Company	*	*	*	*

Table 2: The mean ECE, F_1 score, MCE, and RMSCE results over five runs, for the confidence calibration methods and for the baseline predictions, on all datasets, along with standard deviations. F_1 scores are reported to two decimal places. The other metrics are reported to four decimal places. Green cells signify that a result is better compared to the result for the uncalibrated pipeline; red cells signify that a result is worse compared to the result for the uncalibrated pipeline. Saturated colours indicate that the performance difference is significant ($\alpha = 0.05$), with arrows showing if the difference is negative or positive. *: Company dataset results were not gathered for the Ensembles method due to computational constraints.

metric are consistently higher than the reported ECEs. This is especially the case for the DBLP-ACM, Company, and Abt-Buy datasets. The reliability diagrams presented in Appendix C present an explanation for the higher RMSCEs, showing that there exist large errors between the predicted probabilities and the empirical probabilities for all datasets.

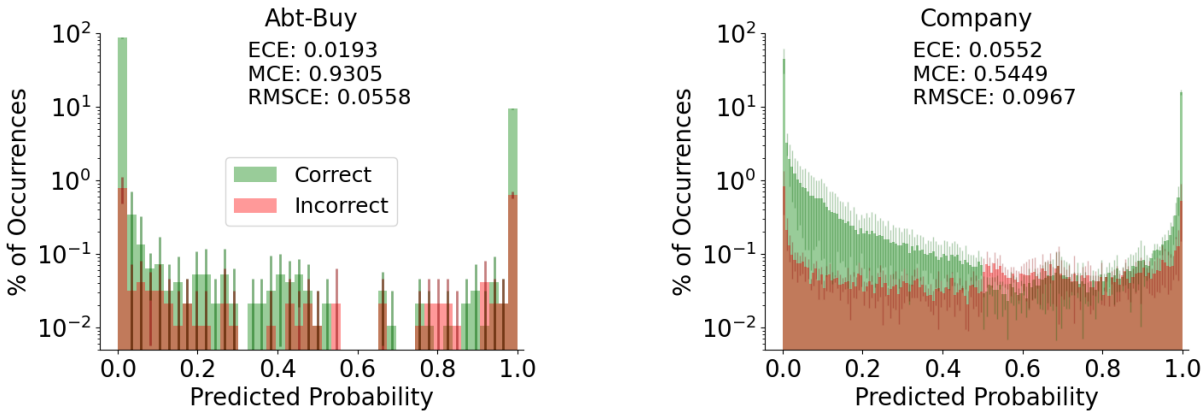
The MCE measures the maximum discrepancy between predicted and empirical probabilities. Figure 5 in Appendix C shows that this difference is large for most datasets, resulting in high MCEs. However, these maximum discrepancies occur for predicted probabilities with few data points, as the figures in Appendix B show.

We find no correlation between the ECE, MCE, or RMSCE metric values and the datasets' F_1 scores, sizes, or mean entry pair sizes.

5.2 Calibration Methods

5.2.1 Temperature Scaling

As Table 2 shows, for the Temperature Scaling method, the ECE significantly decreases for the Abt-Buy, DBLP-ACM-Structured, iTunes-Amazon-Structured, and Company datasets when compared to the baseline. For the other datasets, the ECE decreases, but not significantly. The percentage decrease in ECE compared to the baseline results across the public datasets ranges from 8.05% (for iTunes-Amazon-Dirty) to 23.83% (for



(a) (b)

Figure 2: The mean confidence histograms over five runs for the Abt-Buy and Company datasets, using the baseline RoBERTa model predictions, on a logarithmic scale. The distribution of correct prediction values are in green; the distribution of incorrect prediction values are in red. The y-axis presents percentages of occurrences rather than absolute numbers of occurrences. Error bars denote standard deviations. ECE, MCE, and RMSCE values are reported to four decimal places. The same confidence histograms for the other four datasets are presented in Appendix B.

Abt-Buy).

For the majority of datasets, however, the changes in MCE and RMSCE are not significant. This is likely because the temperature parameter that is used for Temperature Scaling is optimised using the ECE, and not the MCE or RMSCE. We therefore suggest, for practical applications, to consider whether to prioritize reducing the mean error, larger errors, or the maximum error in calibration. The temperature parameter can then be optimised on respectively the ECE, RMSCE, or MCE.

Figure 6 in Appendix D shows that for every dataset and run, there seems to be a clear optimum in the temperature parameter value when optimising on the validation set. As shown in Table 4, the optimal temperature values are typically greater than 1.00. This means that the resulting sigmoid scores are drawn closer to 0.5 compared to when no temperature scaling is applied. This further demonstrates that the baseline probability predictions of the RoBERTa EM model tend to be overconfident.

5.2.2 Monte Carlo Dropout

For Monte Carlo Dropout, the ECE often decreases compared to the baseline, though this difference is almost always not significant. For Abt-Buy, Monte Carlo Dropout leads to a significant decrease in the F_1 score.

Figure 7 in Appendix E shows that for none of the trained models and datasets, there seems to be a very clear optimal dropout probability parameter value when optimising on the validation set. Only very high dropout values negatively impact the ECE. The same pattern is observed in Figure 8 of the Appendix E. This figure also suggests that a considerable dropout probability can be used on most datasets without weakening the performance. Table 5 further demonstrates this, as for most datasets, the optimal dropout probability lies between 0.5 and 1.0. For two datasets, the optimal dropout probabilities are even above 0.8. Table 5 moreover shows that the mean optimal dropout probabilities and standard deviations can vary considerably among datasets, suggesting a lack of generalisability for the dropout parameter. On the other hand, again, Figure 7 shows that there are no clear optima of the dropout probabilities per dataset on the validation ECEs.

Monte Carlo Dropout causes no significant changes in MCE or RMSCE. Like for Temperature Scaling, we suggest to optimise on the ECE, RMSCE, or MCE depending on the desired confidence calibration behaviour.

5.2.3 Ensembles

For the Ensembles calibration method, the ECE decreases for the Abt-Buy and iTunes-Amazon-Structured datasets. For the DBLP-ACM and iTunes-Amazon-Dirty datasets, the change is not significant. With regard to the F_1 score, the results are also often not significant, although the F_1 score for the DBLP-ACM-Dirty dataset does decrease significantly.

Monte Carlo performs multiple sub-runs with

dropout during inference. Ensembles train multiple models using differently initialised weights. For both methods, the predictions of respectively these sub-runs and models are averaged and used as final prediction probabilities. A possible reason for the limited significant improvements in ECEs for the Monte Carlo Dropout and Ensemble methods is the similarity in the predictions of the sub-runs and models. After all, the only difference in producing these predictions is, for Monte Carlo Dropout, the dropout in the final classification layer, or, for Ensembles, the initialisation of this classification layer. The inputs to this classification layer come from the same pre-trained model checkpoint, resulting in highly correlated sub-run or model predictions. This strong correlation likely limits the effectiveness of both Monte Carlo Dropout and Ensembles. Xiao and colleagues also describe this drawback for Ensembles (Xiao et al., 2022).

6 Conclusions

We compare the confidence calibration of baseline RoBERTa probability predictions without any use of confidence calibration methods, to the confidence calibration using Temperature Scaling, Monte Carlo Dropout and Ensembles as confidence calibration methods for EM.

We find that the ECE performance and overall confidence calibration performance for RoBERTa's performance on EM, without using any confidence calibration methods, is reasonable, but often overconfident, with ECE scores ranging from 0.0043 to 0.0552, leaving room for improvement.

We find Temperature Scaling to work best, compared to Monte Carlo Dropout and Ensembles, in improving a RoBERTa model's ECEs for EM, reducing ECE scores by up to 23.83%. This is a simple method that can easily be implemented in practical settings.

We find that neither Temperature Scaling, Monte Carlo Dropout, nor Ensembles have consistently significant effects on the F_1 scores of the the RoBERTa EM model.

Overall, the ECEs reported for the baseline RoBERTa EM model results are slightly higher than those reported for RoBERTa by Desai and Durrett (Desai and Durrett, 2020) and slightly lower to those reported for RoBERTa by Xiao and colleagues (Xiao et al., 2022). Both studies applied the model to natural language processing tasks other than EM. It would be interesting for future research to investigate the cause of these differences in metric values.

Another avenue for future research is to combine confidence calibration methods for EM. For example, Rahaman and Thiery (2021) found that using Ensembles, and applying Temperature Scaling to the averaged sigmoid scores can reduce ECE scores by half compared to just using Ensembles, on image classification tasks. Temperature Scaling could be combined with Monte Carlo Dropout in the same way.

Additionally, future work could leverage the individual variances in the sigmoid scores produced by Monte Carlo Dropout and Ensembles. If these variances are high, the confidence levels can be lowered accordingly, potentially improving calibration. By incorporating variance-based adjustments, it might be possible to create more reliable confidence estimates and further enhance the overall performance of the RoBERTa pipeline. Additionally, entry pairs with large variances in their sigmoid scores can be more closely analyzed to gain deeper insights into the pipeline's prediction patterns.

Limitations

Recent years have seen massive advances in LLMs, yet this study focuses on a relatively small-scale model compared to state-of-the-art architectures. The academic community has extensively researched derivatives of the BERT model, and smaller models remain practical for deployment on limited computational resources facilities. However, an important next step is to extend these model calibration experiments to larger models and evaluate their trustworthiness capabilities under similar conditions.

It is worth noting that the ECE, MCE, and RMSCE metrics are not without limitations in accurately capturing confidence calibration. To illustrate this, suppose there is an EM dataset with 50% 'match' labels and 50% 'no-match' labels. If a model would only output predicted probabilities of 0.5, the ECE, MCE and RMSCE would all be zero, suggesting approximately perfect calibration. Yet, the model's predicted probabilities would be entirely uninformative.

Acknowledgements

We would like to acknowledge that the research presented in this paper was conducted while Iris Kamsteeg, Gineke ten Holt and Floris Van Beers

were affiliated with WebIQ B.V., The Netherlands. Also, we thank the Center for Information Technology of the University of Groningen for their support and for providing access to the Hábrók high performance computing cluster.

Broader Impact Statement

We recognize that while LLMs have proven to be successful in EM tasks, these models also pose risks. An example of this is the potential for bias in LLM outputs, discussed in detail in the paper "On the Dangers of Stochastic Parrots" by Bender et al. (2021). Since models such as RoBERTa are pre-trained on large amounts of data that reflect societal biases, these prejudices can be incorporated into and potentially be amplified in EM predictions. Moreover, LLMs operate as black-box models, providing little transparency on their decision-making processes. In this research, we explored this problem through a study on confidence calibration, so that it can be mitigated. Enhancing transparency can help avoid incorrect downstream decisions and make it easier to analyze and rectify erroneous or misleading outputs.

References

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Gerrit Bloothooft, Peter Christen, Kees Mandemakers, and Marijn Schraagen, editors. 2015. *Population Reconstruction*. Springer International Publishing AG, Cham.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ursin Brunner and Kurt Stockinger. 2020. Entity Matching with Transformer Architectures-a Step Forward in Data Integration. In *23rd International Conference on Extending Database Technology, Copenhagen, 30 March-2 April 2020*, pages 463–473. OpenProceedings.

Wenlong Chen and Yingzhen Li. 2024. Calibrating Transformers via Sparse Gaussian Processes. ArXiv:2303.02444 [cs, stat].

Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2020. An Overview of End-to-End Entity Resolution for Big Data. *ACM Computing Surveys*, 53(6):1–42.

Johan Dahlin, Fredrik Johansson, Lisa Kaati, Christian Mårtenson, and Pontus Svenson. 2012. Combining Entity Matching Techniques for Detecting Extremist Behavior on Discussion Boards. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 850–857. IEEE.

Shrey Desai and Greg Durrett. 2020. Calibration of Pre-trained Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

OpenAI et al. 2024. GPT-4 Technical Report. ArXiv:2303.08774 [cs].

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059. PMLR. ISSN: 1938-7228.

Zoubin Ghahramani. 2015. Probabilistic Machine Learning and Artificial Intelligence. *Nature*, 521(7553):452–459.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, Proceedings of Machine Learning Research, pages 1321–1330, Sydney, NSW, Australia. PMLR.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778. ISSN: 1063-6919.

Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *International Conference on Learning Representations*.

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2012. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. ArXiv:1207.0580 [cs].

Matthew A. Jaro. 1995. Probabilistic Linkage of Large Public Health Data Files. *Statistics in Medicine*, 14(5-7):491–498.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How Can We Know When Language Models Know? On the Calibration of Language Models for Question Answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Pradap Konda, Sanjib Das, Paul Suganthan G. C., AnHai Doan, Adel Ardalan, Jeffrey R. Ballard, Han Li, Fatemah Panahi, Haojun Zhang, Jeff Naughton, Shishir Prasad, Ganesh Krishnan, Rohit Deep, and Vijay Raghavendra. 2016. Magellan: Toward Building Entity Matching Management Systems. *Proceedings of the VLDB Endowment*, 9(12):1197–1208.

Ananya Kumar, Percy S Liang, and Tengyu Ma. 2019. Verified Uncertainty Calibration. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Hanna Köpcke, Andreas Thor, and Erhard Rahm. 2010. Evaluation of Entity Resolution Approaches on Real-World Match Problems. *Proc. VLDB Endow.*, 3(1-2):484–493.

Fabian Küppers, Anselm Haselhoff, Jan Kronenberger, and Jonas Schneider. 2022. Confidence Calibration for Object Detection and Segmentation. In Tim Fingscheidt, Hanno Gottschalk, and Sebastian Houben, editors, *Deep Neural Networks and Data for Automated Driving: Robustness, Uncertainty Quantification, and Insights Towards Safety*, pages 225–250. Springer International Publishing, Cham.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. 2020. Deep Entity Matching with Pre-Trained Language Models. *Proceedings of the VLDB Endowment*, 14(1):50–60.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. ArXiv:1907.11692 [cs].

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. 2020. Calibrating Deep Neural Networks using Focal Loss. In *Advances in Neural Information Processing Systems*, volume 33, pages 15288–15299. Curran Associates, Inc.

Nora Méray, Johannes B. Reitsma, Anita C. J. Ravelli, and Gouke J. Bonsel. 2007. Probabilistic Record Linkage is a Valid and Transparent Tool to Combine Databases Without a Patient Identification Number. *Journal of Clinical Epidemiology*, 60(9):883–891.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining Well Calibrated Probabilities Using Bayesian Binning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1). Number: 1.

Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. 2022. Can Foundation Models Wrangle Your Data? *Proceedings of the VLDB Endowment*, 16(4):738–746.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting Good Probabilities with Supervised Learning. In *Proceedings of the 22nd International Conference on Machine learning*, ICML '05, pages 625–632, New York, NY, USA. Association for Computing Machinery.

R. Peeters, Christian Bizer, and Goran Glavas. 2020. Intermediate Training of BERT for Product Matching. In *CEUR Workshop Proceedings*, volume 2726, pages 1–2, Aachen. Piai, Federico.

Ralph Peeters and Christian Bizer. 2021. Dual-Objective Fine-Tuning of BERT for Entity Matching. *Proceedings of the VLDB Endowment*, 14(10):1913–1921.

Ralph Peeters and Christian Bizer. 2022. Supervised Contrastive Learning for Product Matching. In *Companion Proceedings of the Web Conference 2022*, WWW '22, pages 248–251, New York, NY, USA. Association for Computing Machinery.

Ralph Peeters and Christian Bizer. 2023. Using ChatGPT for Entity Matching. In *New Trends in Database and Information Systems*, pages 221–230, Cham. Springer Nature Switzerland.

Ralph Peeters and Christian Bizer. 2024. Entity Matching using Large Language Models. ArXiv:2310.11244 [cs].

John Platt. 1999. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in Large Margin Classifiers*, 10(3):61–74.

Rahul Rahaman and Alexandre H. Thiery. 2021. Uncertainty Quantification and Deep Ensembles. *Advances in Neural Information Processing Systems*, 34:20063–20075.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter. ArXiv:1910.01108 [cs].

Karthik Abinav Sankararaman, Sinong Wang, and Han Fang. 2022. BayesFormer: Transformer with Uncertainty Estimation. ArXiv:2206.00826 [cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Yijun Xiao and William Yang Wang. 2019. Quantifying Uncertainties in Natural Language Processing Tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33 of *AAAI'19/IAAI'19/EAAI'19*, pages 7322–7329, Honolulu, Hawaii, USA. AAAI Press.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty Quantification with Pre-trained Language Models: A Large-Scale Empirical Analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A RoBERTa EM Performance

Table 3 presents the mean F_1 score, precision, recall and inference time for the baseline RoBERTa model.

B RoBERTa Confidence Histograms

The confidence histograms for all datasets, using the baseline RoBERTa model predicted probabilities and a number of bins = $\sqrt{|\mathcal{D}|}$, are presented in Figure 3.

Figure 4 shows confidence histograms that are similar to those in Figure 3. Histograms are presented for all datasets, using the baseline RoBERTa model predicted probabilities and a number of bins = $\sqrt{|\mathcal{D}|}$. For Figure 4, correct and incorrect predictions are plotted individually. Moreover, the distribution of predicted values is plotted on a logarithmic scale, so that smaller effects are easier to see. Confidence histograms for four out of the six datasets are shown. The confidence histograms for the Abt-Buy and Company datasets are presented in Section 5.

C RoBERTa Reliability Diagrams

Figure 5 presents the mean reliability diagrams for all datasets, using the baseline RoBERTa model probability predictions and a number of bins, or dots, = $\sqrt{|\mathcal{D}|}$. When a dot is missing, this means that there are no predictions within that predicted probability bin. A diagonal line representing approximately perfect calibration is plotted as well.

D Detailed Temperature Scaling Results

Figure 6 presents the single parameter gridsearch results for the temperature parameter on the validation sets, for all datasets.

The mean recorded temperature parameter values per dataset are shown in Table 4.

E Detailed Monte Carlo Dropout Results

Figure 7 and Figure 8 present the single parameter gridsearch results for the dropout parameter on the validation sets, for all datasets. Figure 7 specifically reports the effect of the dropout probability value on the ECE, while Figure 8 specifically reports the effect of the dropout probability value on the F_1 score.

The mean recorded dropout probability parameter values per dataset are shown in Table 5.

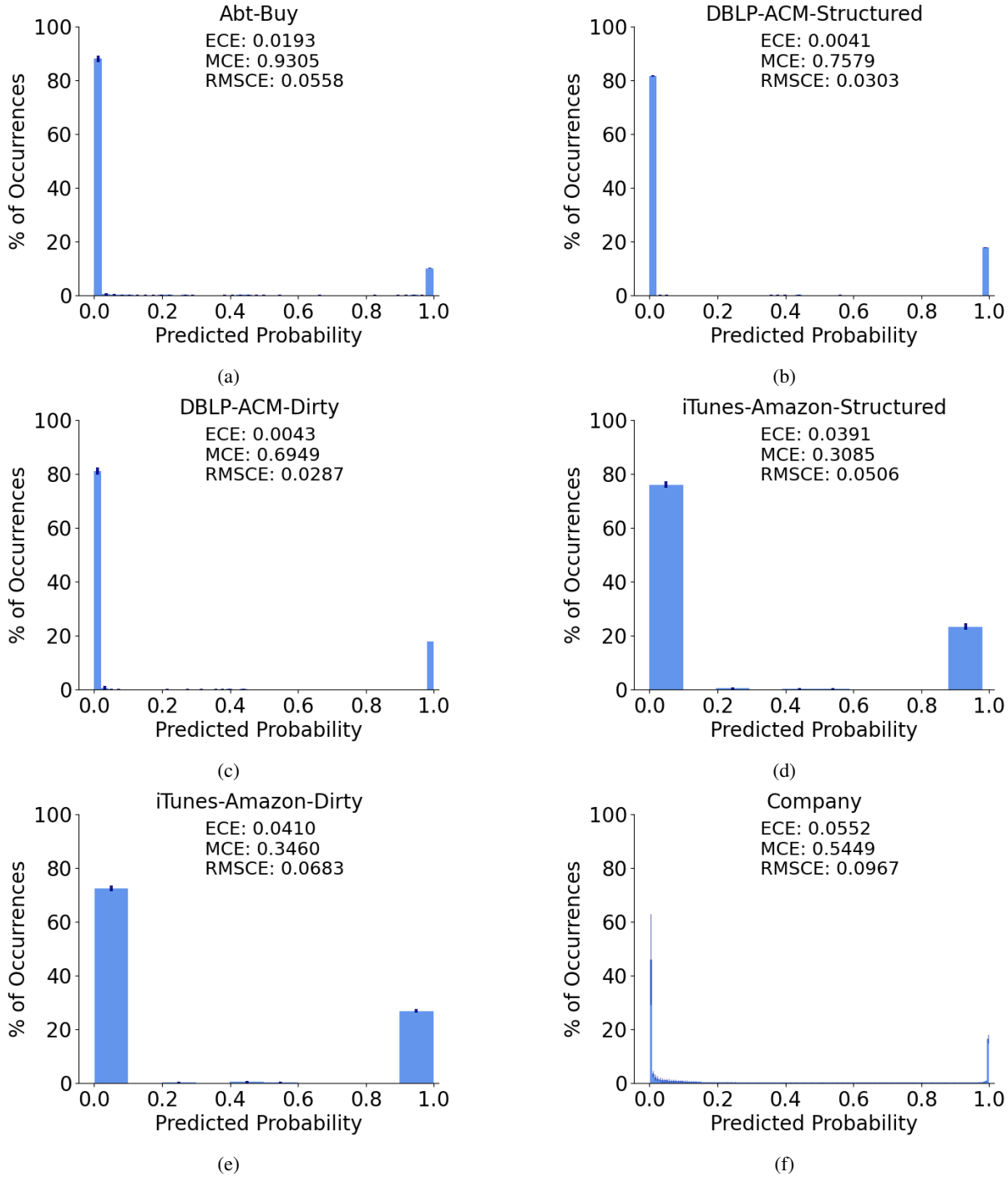


Figure 3: The mean confidence histograms over five runs for all datasets, using the baseline RoBERTa model predicted probabilities. The y-axis presents percentages of occurrences rather than absolute numbers of occurrences. Error bars denote standard deviations. ECE, MCE, and RMSCE values are reported to four decimal places.

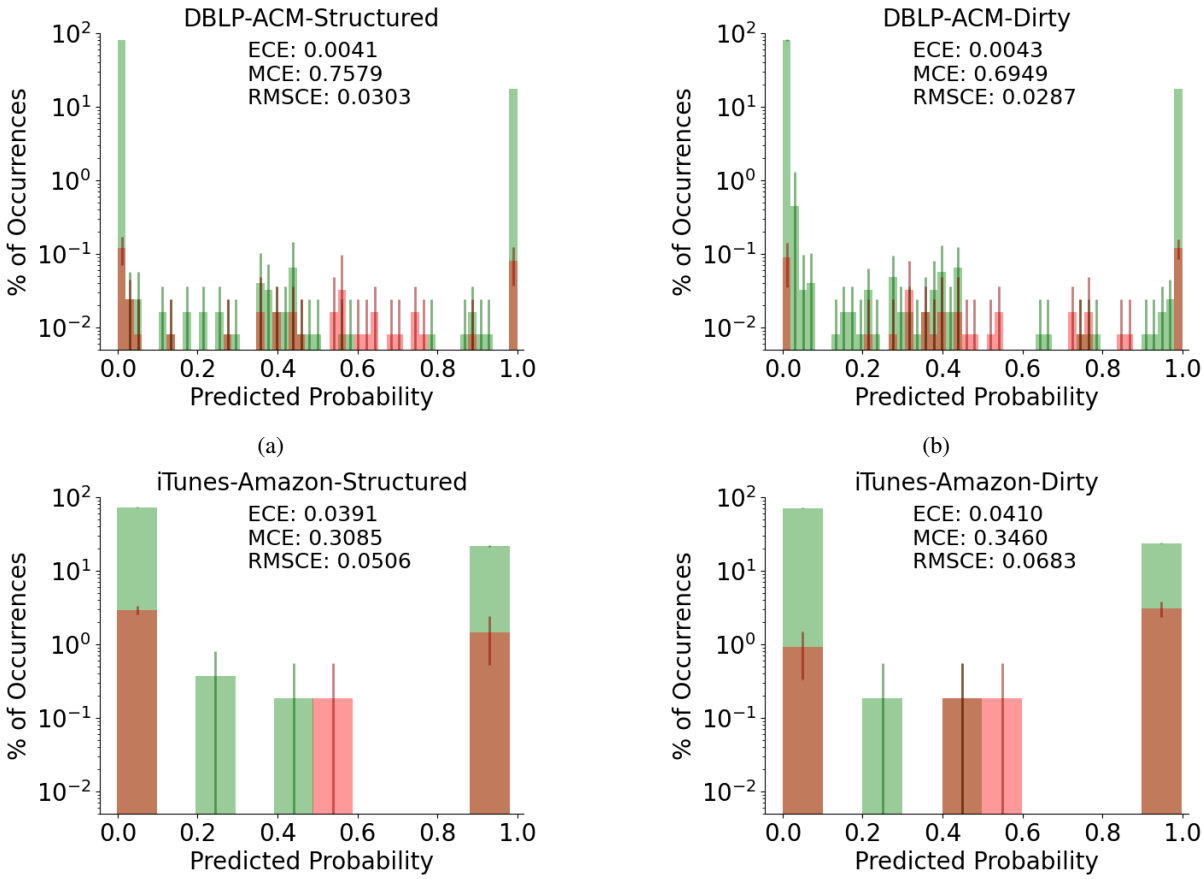


Figure 4: The mean confidence histograms over five runs for the DBLP-ACM-Structured, DBLP-ACM-Dirty, iTunes-Amazon-Structured and iTunes-Amazon-Dirty datasets, using the baseline RoBERTa model predictions, on a logarithmic scale. The distribution of correct prediction values are in green; the distribution of incorrect prediction values are in red. The y-axis presents percentages of occurrences rather than absolute numbers of occurrences. Error bars denote standard deviations. ECE, MCE, and RMSCE values are reported to four decimal places.

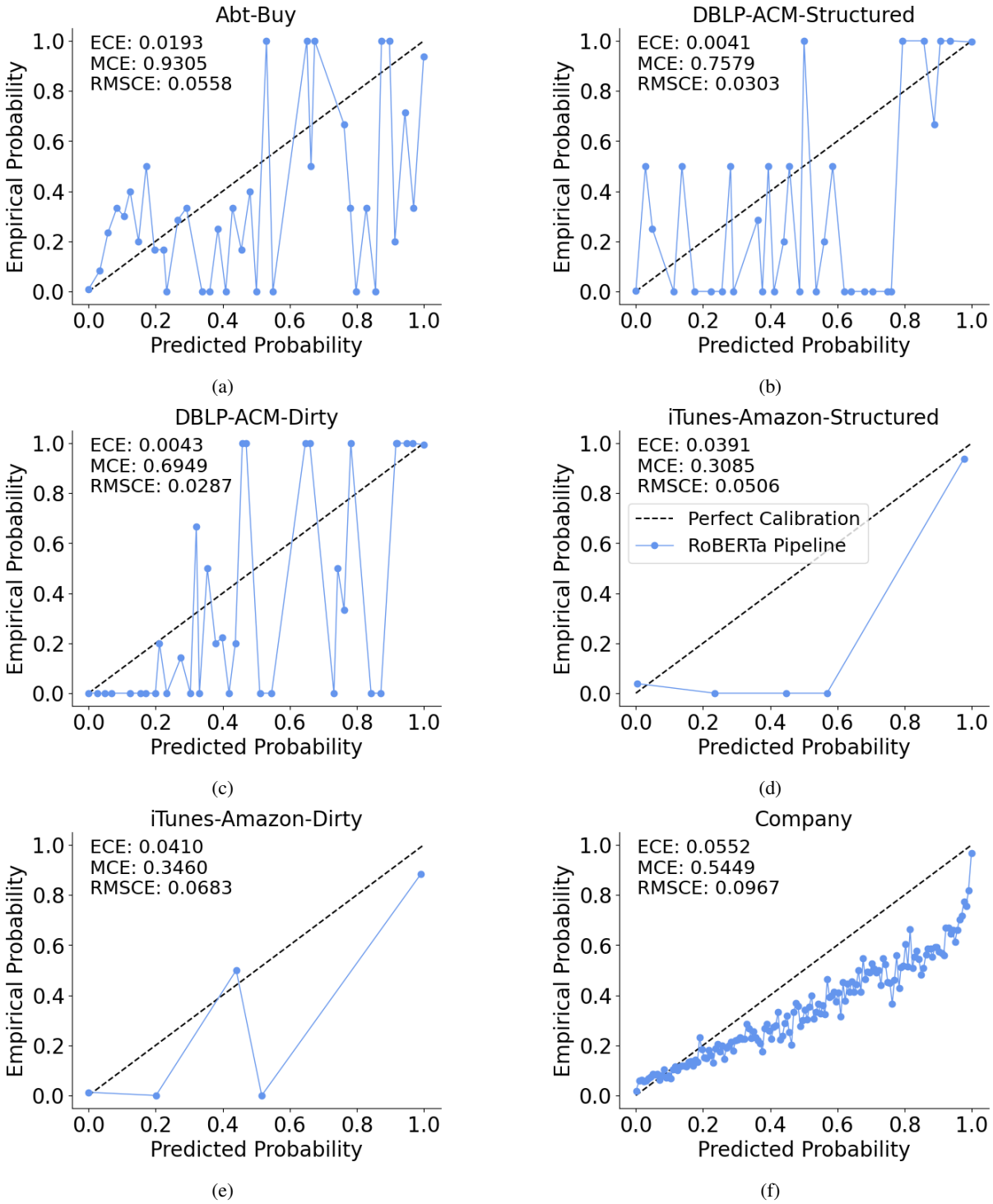


Figure 5: The reliability diagrams using data from five runs for all datasets, using the baseline RoBERTA model predictions. ECE, MCE, and RMSCE values are reported to four decimal digits. Note that for some of the datasets, data is missing for certain predicted probability bins. This is because there were no predictions found within that bin. A diagonal is plotted to represent approximately perfect calibration.

Dataset	$\mathbf{F_1}$	Precision	Recall	Inference time (ms)
Abt-Buy	90.81 ± 0.85	91.86 ± 0.55	89.81 ± 1.82	1.43 ± 0.01
DBLP-ACM-Structured	98.78 ± 0.40	98.83 ± 0.73	98.74 ± 0.12	2.04 ± 0.01
DBLP-ACM-Dirty	98.85 ± 0.18	98.88 ± 0.50	98.83 ± 0.25	2.06 ± 0.01
iTunes-Amazon-Structured	90.53 ± 1.64	93.22 ± 4.80	88.15 ± 1.66	0.32 ± 0.05
iTunes-Amazon-Dirty	91.50 ± 1.90	87.81 ± 2.83	95.56 ± 1.66	0.28 ± 0.07
Company	82.75 ± 0.92	82.20 ± 2.95	83.40 ± 1.53	2.51 ± 0.00

Table 3: The mean F_1 score, precision, recall, and inference time (in milliseconds) for the RoBERTa EM model for all datasets, along with the standard deviations. Metrics are taken over five randomly initialised runs and reported to two decimal places.

Dataset	Temperature
Abt-Buy	2.24 ± 0.47
DBLP-ACM-S	0.88 ± 0.50
DBLP-ACM-D	1.00 ± 0.67
iTunes-Amazon-S	1.74 ± 0.55
iTunes-Amazon-D	1.64 ± 0.91
Company	1.72 ± 0.51

Table 4: The mean temperature parameter value results, taken over five runs, for all datasets, along with the standard deviations. Values are reported to two decimal digits.

Dataset	Dropout probability
Abt-Buy	0.39 ± 0.22
DBLP-ACM-S	0.58 ± 0.40
DBLP-ACM-D	0.56 ± 0.35
iTunes-Amazon-S	0.85 ± 0.12
iTunes-Amazon-D	0.91 ± 0.07
Company	0.50 ± 0.35

Table 5: The mean dropout probability parameter value results, taken over five runs, for all datasets, using the RoBERTa model, along with the standard deviations. Values are reported to two decimal digits.

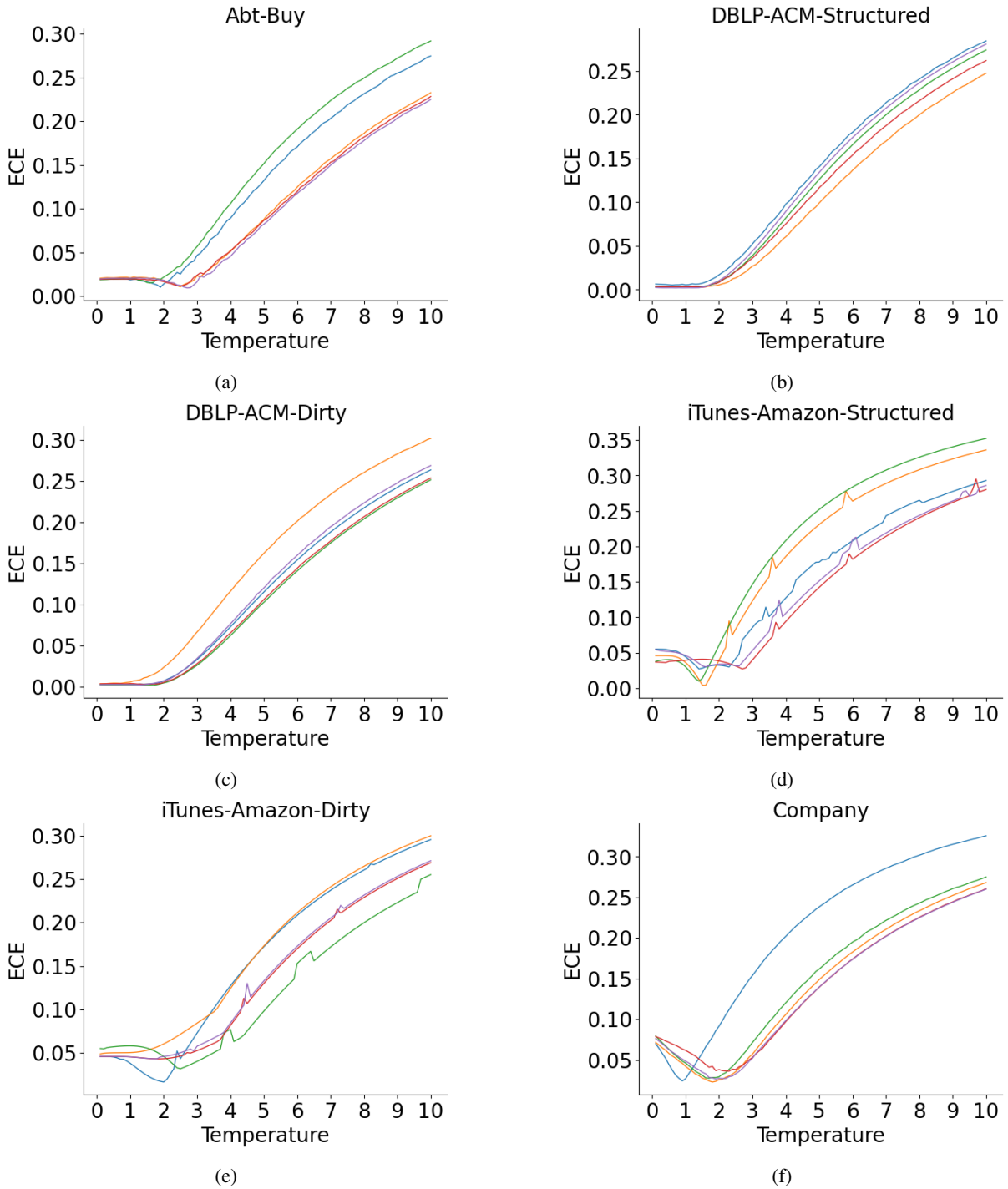


Figure 6: The effect of the temperature parameter on the ECE for the validation set, for all datasets. Each line denotes one run. Note that the y-axis differs per plot.

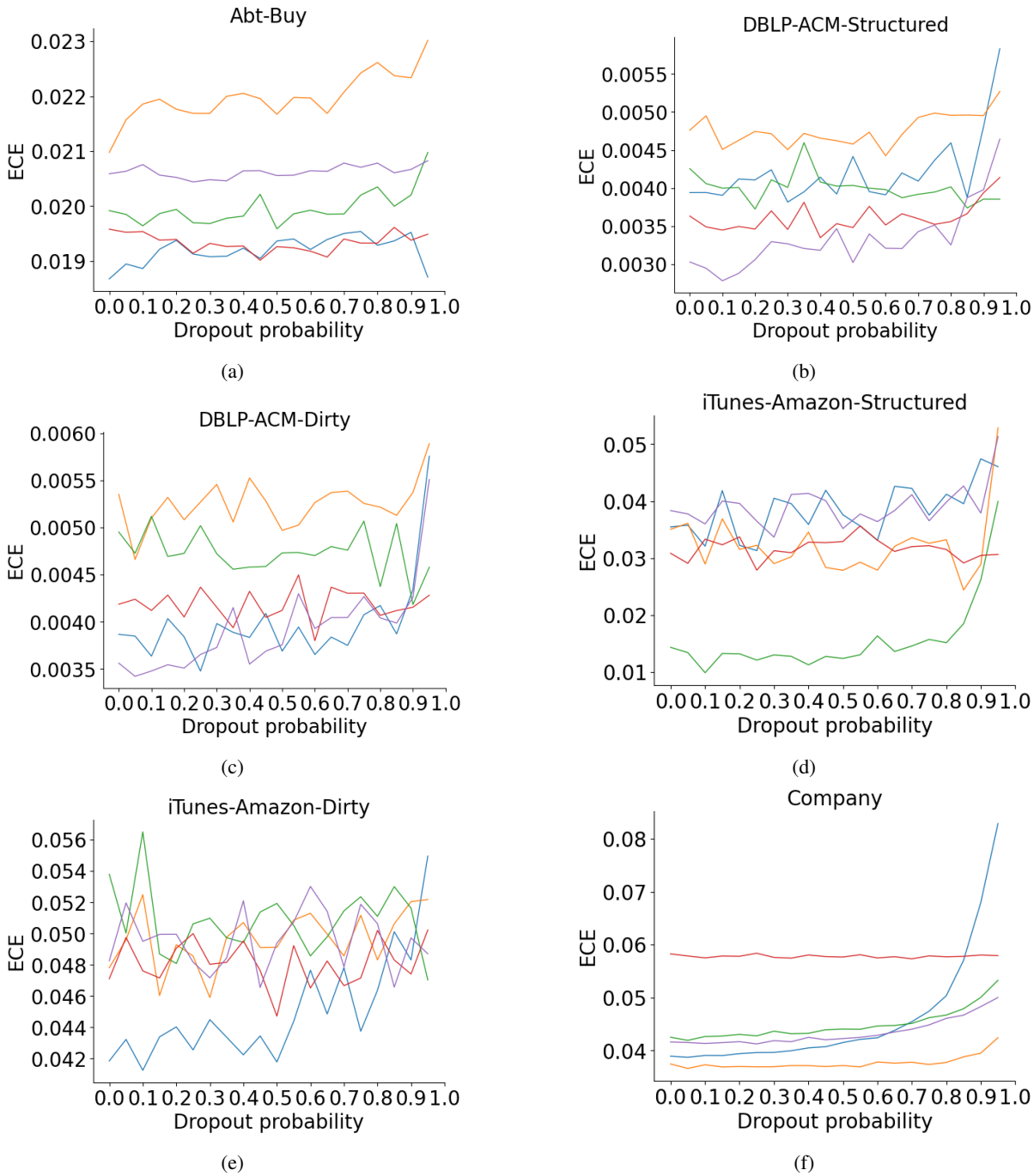


Figure 7: The effect of the dropout probability parameter on the ECE for the validation set, for all datasets. Each line denotes one run. Note that the y-axis differs per plot.

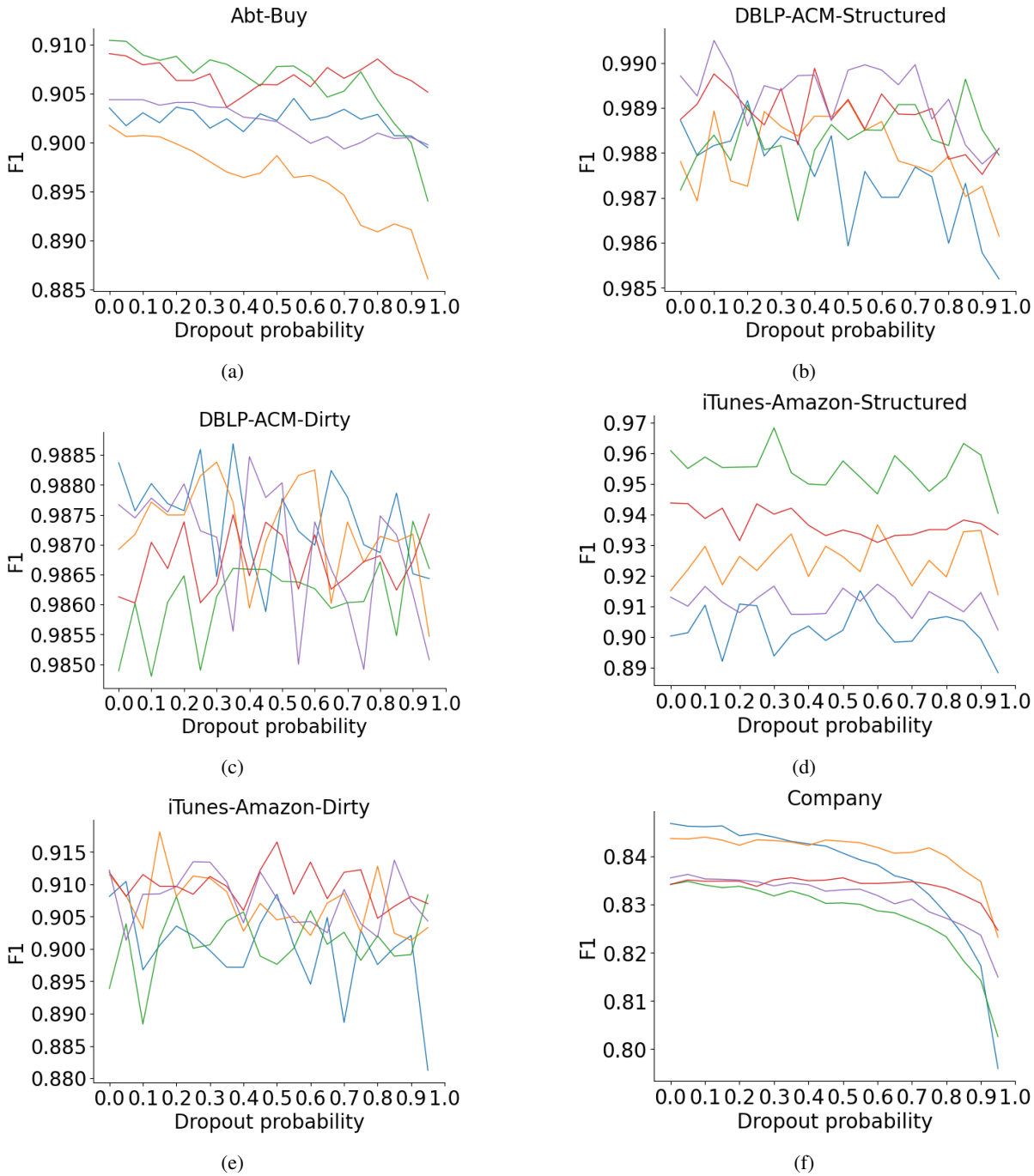


Figure 8: The effect of the dropout probability parameter on the F_1 score for the validation set, for all datasets. Each line denotes one run. Note that the y-axis differs per plot.

Consensus or Conflict?
Fine-Grained Evaluation of Conflicting Answers in Question-Answering

Eviatar Nachshoni[1], **Arie Cattan**[1,2], **Shmuel Amar**[1], **Ori Shapira**[3], **and Ido Dagan**[1,2]

[1]Bar-Ilan University [2]Google Research [3]OriginAI
{eviatarn, shmulikamar}@gmail.com cattana@google.com
obspp18@gmail.com dagan@cs.biu.ac.il

Abstract

Large Language Models (LLMs) have demonstrated strong performance in question answering (QA) tasks. However, Multi-Answer Question Answering (MAQA), where a question may have several valid answers, remains challenging. Traditional QA settings often assume consistency across evidences, but MAQA can involve conflicting answers. Constructing datasets that reflect such conflicts is costly and labor-intensive, while existing benchmarks often rely on synthetic data, restrict the task to yes/no questions, or apply unverified automated annotation. To advance research in this area, we extend the conflict-aware MAQA setting to require models not only to identify all valid answers, but also to detect specific conflicting *answer pairs*, if any. To support this task, we introduce a novel cost-effective methodology for leveraging fact-checking datasets to construct NatConfQA, a new benchmark for realistic, conflict-aware MAQA, enriched with detailed conflict labels, for all answer pairs. We evaluate eight high-end LLMs on NatConfQA, revealing their fragility in handling various types of conflicts and the flawed strategies they employ to resolve them.[1]

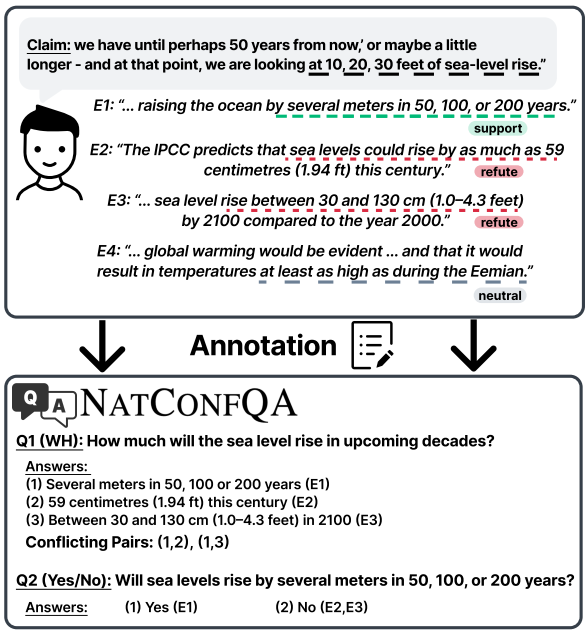


Figure 1: Deriving a conflict-aware MAQA instance from a fact-chcecking instance. The source instance is composed of a claim with supporting, refuting and neutral evidence. Annotators then create WH and Yes/No questions to surface these conflicts and label *conflicting* answer pairs.

1 Introduction

Recent advances in Large Language Models (LLM) (Fischer et al., 2024; OpenAI-Team, 2024a) have led to substantial performance improvement in various tasks, including Question Answering (QA) with one or multiple correct answers (Voorhees, 2004). Although previous work on multi-answer QA (MAQA) largely assumes that the different answers are mutually consistent and complementary (Zhong et al., 2022; Amouyal et al., 2023), realistically, questions can be controversial and lack a definitive answer. In such cases, models should not only generate a response that incorporates several answers, but also detect the conflicts and communicate them to the reader. For example, when asked *"What is the effect of aspartame?"*, a comprehensive response should aggregate various effects reported in the available sources, while explicitly distinguishing between effects with consensus and those that remain contested or under debate.

Collecting QA instances with naturally occurring contradictory answers is challenging, as knowledge conflicts are not always prevalent in arbitrary texts. As a result, only a few datasets handle knowledge conflicts while exhibiting some limitations, such as relying on LLMs to inject misinformation into other reliable texts, or focusing only on Yes/No

[1]NatConfQA is publicly available at https://github.com/EN555/ContraQA.

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 138–159
November 9, 2025 ©2025 Association for Computational Linguistics

questions (Section 6). Furthermore, although recent benchmarks aim to evaluate whether the *entire* response acknowledges the debatable nature of the question (Xu et al., 2024; Hou et al., 2024), they do not assess whether the response accurately reflects *which* answers are subject to disagreements (e.g., whether aspartame increases cancer risk).

To foster research on this important challenge and to enable fine-grained evaluation, we create NatConfQA, the first conflict-aware MAQA dataset with annotations labeling individual answer pairs that are in conflict. To collect NatConfQA, we first leverage standard fact-checking datasets to identify sources with naturally occurring disagreements. Then, we ask human annotators to write various QA pairs and to label the relationship between pairs of conflicting answers. Finally, we verify the annotations for quality assurance. Figure 1 illustrates our general annotation scheme. NatConfQA is a high-quality dataset that covers Yes/No and WH- questions, and includes instances, based on reliable sources, with a mix of conflicting and non-conflicting answers for the same question.

We evaluate the performance of eight LLMs on NatConfQA, including open-source and proprietary models, measuring both answer quality and conflict identification. In terms of answer quality, we show that while models generally achieve high precision, they fail to output all correct answers. Furthermore, our fine-grained evaluation of conflict identification reveals that models struggle to distinguish between conflicting and non-conflicting answer pairs. Further analysis of the model failures reveal insightful error patterns: models evade exposing conflicts by selecting a single answer, erroneously attempting to reconcile contradictory information, or refraining from answering the question altogether. Taken together, our work uncovers the behavior of strong LLMs when confronted with conflicting information, while providing suitable methodologies and data to investigate these challenges in future research.

2 Conflict-Aware Multi-Answer QA

The *Conflict-Aware Multi-Answer QA* task is an extension of the traditional Multi-Answer QA (Min et al., 2020; Amouyal et al., 2023) task, that considers potential conflicts between the different answers. We extend recent work in conflict-aware QA (Xu et al., 2024), which either focused on binary conflicting answers (Hou et al., 2024) or addressed multiple answers without indicating which pairs conflict (Jiayang et al., 2024). Our generalized task formulation supports two or more answers per question and, importantly, pinpoints the answer pairs that are in conflict.

Given a question q and a set of candidate passages $P = \{p_1, \ldots, p_n\}$, the task is to generate a free-text response y that satisfies two main requirements. First, similar to the traditional MAQA task, y should incorporate all answers that appear in reference $A = \{a_1, \ldots, a_m\}$. Second, the response y should indicate all conflicts, if any, between the answers within it. We assume that the response y is in natural language, as typically generated by large language models (LLMs), and not necessarily in a structured format.

For example, consider the question *"What climate degree change is caused by greenhouse gases?"*. The different answers *0.45°C* and *0.8°C* cannot be simultaneously true, hence an ideal response should present both answers through contrastive language (e.g., by using the word "however", both answers can be communicated, and the conflict is established). In contrast, for the question *"Which domestic pets can potentially test positive for SARS-CoV-2?"*, the answers *dogs*, *cats*, and *ferrets* are non-conflicting and an ideal response should enumerate the different answers cohesively.

Since a question may elicit a mixture of conflicting and non-conflicting answers, we want to determine whether the model response y accurately reflects the conflict or non-conflict relations between the different answer pairs. Formally, for reference answers A and a respective set $C = \big\{\{a_i, a_j\} \mid a_i \text{ and } a_j \text{ conflict}\big\}$, that lists the pairs of conflicting answers in A, the objective of response y is to accurately incorporate A and C.

In accordance with the task definition, we define two evaluation criteria, adopted from related tasks (Min et al., 2020; Hou et al., 2024): *Answer Quality*, measuring how well the model covers the set of correct answers; and *Conflict Identification*, assessing the model's ability to correctly identify conflicting answer pairs. We propose metrics for the evaluation criteria in §4.3.

3 Creating the NatConfQA Dataset

Obtaining annotations for a conflict-aware MAQA dataset is challenging, because informational conflicts are infrequent in arbitrary sources from which answers can be collected.

In this work, we approach this challenge by leveraging existing fact-checking datasets, a well-studied field with many datasets, which were annotated with large manual effort (Thorne et al., 2018; Sarrouti et al., 2021). Some fact-checking datasets contain claims for which some pieces of evidence refute the claim, while others support it, as exemplified in Figure 1. This structure makes them particularly valuable for our task, as they naturally capture conflicting evidence.

In this section, we describe our methodology for converting and annotating existing fact-checking datasets into a conflict-aware MAQA dataset (§3.1 and §3.2). We then describe the resulting new NatConfQA dataset (§3.3) finally, we examine its quality (§3.4).

3.1 From Fact-Checking to Conflict-Aware Multi-Answer QA

The typical structure of an instance in a fact-checking dataset is a triplet (c, e, ℓ), where c is a claim, e is a piece of evidence associated with that claim, and $\ell \in \{refute, support, neutral\}$ indicates the entailment relation between the evidence e and the claim c (Aly et al., 2021; Sarrouti et al., 2021). In the CLIMATE-FEVER (Diggelmann et al., 2021) and HealthVer (Sarrouti et al., 2021) datasets, the same claim c may appear in multiple triplets (c, e_i, ℓ_i), where it might be supported by some pieces of evidence, while others refute it. The co-occurrence of both supporting and refuting evidence for the same claim typically indicates the presence of an underlying factual conflict among the pieces of evidence (see App. A for details).

Our goal is to leverage the above fact-checking datasets in order to create a conflict-aware MAQA dataset. An instance should contain a question q and a set of respective answers $A = \{a_1, \ldots, a_k\}$ that includes at least one pair of conflicting answers. In reality, however, not all questions have conflicting answers. Hence, we would also like to include instances where A has only non-conflicting answers. Such a subset is useful as a control set when assessing models' performance in the conflict-aware MAQA task (§5). Beyond accommodating conflicting and non-conflicting answer sets, it is also important to support variation in question form, i.e., WH-questions versus Yes/No questions.

Overall, we gather two sets of instances from fact-checking datasets: (1) *conflicting* instances, that include both supporting and refuting pieces of evidence, and (2) *non-conflicting* instances, that include at least one supporting evidence and no refuting evidence. We next describe our process for converting these fact-checking instances into ones for the conflict-aware MAQA task.

3.2 Dataset Preparation

Initial fact-checking data. We first gathered the two sets of fact-checking instances on which the annotation process is conducted. For the *conflicting* set, we automatically iterated over the CLIMATE-FEVER and HealthVer instances (see Appendix A for pre-processing details), and grouped those with the same claim, that have conflicting evidence (at least one triplet with *support* and at least one with *refute*), resulting in 188 groups. For the *non-conflicting* set, we collected several hundred groups of claims with only supporting or neutral evidence.

The fact-checking datasets supply evidence at the sentence level, sourced from Wikipedia articles in CLIMATE-FEVER and CORD-19 abstracts in HealthVer. In a realistic situation, especially in the QA setting, the source texts on which the task is performed are typically longer passages. Therefore, for each evidence sentence used from the fact-verification datasets, we retrieved the complete passage containing that sentence (details in Appendix C).[2] In summary, each instance in the two fact-checking sets is composed of a claim and several pieces of evidence within their passages.

Manual annotation process. The two annotators were first provided each with half of the *conflicting* fact-checking instances.[3] Each instance is composed of a claim and several pieces of evidence (some conflicting). The annotators then followed the following procedure (more details in Figure 3). (1) Contradiction detection: confirm whether the supporting and refuting evidence indeed conflict with each other. (2) WH-question formation: for each instance, write a WH-question based on the claim and evidence, aiming to elicit the core information and potential conflict. Then for each question, write out a list of answers based on the evidence, and link the evidence to the answer. (3) Label answer pairs: label each pair of answers as *conflicting* or *non-conflicting*. (4) Yes/no question

[2] Note that a passage can contain more than one evidence sentences if they come from the same source passage.

[3] We observed that even top-performing models often mark instances as containing conflicts even when none exist, or, in instances where conflicts were present, they fail to clearly reflect them in the generated question.

formation: for each WH-question and its corresponding answer from the previous steps, if possible, formulate a yes/no question and link the supporting evidence for the yes and no responses.

The annotators then repeated the process on the *non-conflicting* fact-checking set, skipping steps 1 and 3. Also, step 2 requires questions and answers with only non-conflicting evidence, and step 4 is conducted for all answers from step 2 in this round. See Appendix B for details regarding annotators and the annotation tool.

3.3 The NatConfQA Dataset

Overall, the annotation process yielded 269 conflicting instances, of which 89 were WH-questions, and 408 non-conflicting instances. Each instance in NatConfQA is represented as a tuple (q, P, A, C), where q is the question, $P = \{p_1, \ldots, p_n\}$ is the set of relevant passages, $A = \{a_1, \ldots, a_m\}$ is the set of answers, and $C = \big\{\{a_i, a_j\} \mid a_i \text{ and } a_j \text{ conflict}\big\}$ contains all annotated pairs of conflicting answers.[4]

3.4 Dataset Quality

Data validation. A high-quality dataset should align with the objectives of our task and the evaluation criteria (§2). Namely, each instance in the dataset should contain accurate answers that are consistent with their supporting evidence, and conflicting answer pairs should be correctly identified. To that end, we randomly selected 40 instances from NatConfQA and hired a reviewer (a third worker) to validate the data. The reviewer, an undergraduate student, was trained for the task and paid a $14 hourly wage (see Figure 10 for the guidelines). The reviewer was instructed to assess whether each answer was supported by its linked evidence, and, independently, whether the pairs marked as conflicting were indeed conflicting.

In the collection phase, an answer was produced from a sentence within a passage. This means that, with respect to a passage, not all potential answers are necessarily included in the dataset. Therefore, given a passage, the correctness of an answer should be verified against the full passage, and not just any specific sentence within the passage. This matter is addressed in the evaluation metrics (§4.3), and is also relevant for the validation phase of the dataset curation.

Following the guidelines, the reviewer labeled each answer in the 40 instances as "correct" or "incorrect" with respect to its linked evidence. Similarly, the reviewer labeled each conflicting pair in a binary fashion, validating whether the two answers conflict. Over this representing set of instances, we find a validation rate of 93% for answer-correctness and 92% for correctness of conflicting pairs. The meticulous dataset curation process, combined with the strong validation statistics, strongly suggest the high quality of NatConfQA.

Dataset properties. A high-level view of the dataset reveals properties that indicate the dataset's diversity and the challenges that it poses.

First, as indicated in Table 3, there are an average of 5.6 passages per question, and each passage has a length of 252 words, reflecting a realistic RAG setting with multiple long contexts. Compared to other datasets for the MAQA task, NatConfQA features a relatively large number and length of passages per instance with high quality. Additionally, NatConfQA includes 62 conflicting answer pairs each containing a mix of both conflicting and non-conflicting answers. This subset proved challenging for models, as discussed in §5.1.

Subsequently, we observe that the number of evidence pieces linked to a single answer ranges from 1 to 17 (std = 2.7), with some answers appearing across multiple passages and others only once. In contrast, most existing datasets either do not specify evidence links (Xu et al., 2024) or include far fewer per answer (Hou et al., 2024; Jiayang et al., 2024). See Appendix D for additional dataset statistics.

Taken together, the dataset's quality, diversity, and level of challenge make NatConfQA a valuable resource for studying model behavior in realistic settings.

4 Experimental Setup

In this section, we outline our experimental setup for evaluating how well models detect and communicate conflicts. We describe two prompting modes for the conflict-aware MAQA task (§4.1) that will be applied on eight top-performing LLMs (§4.2). Their performance will be assessed using two evaluation criteria adapted from prior work (§4.3).

4.1 LLM Prompting Modes

To assess the performance of state-of-the-art LLMs in a QA setting in which input passages may contain multiple, potentially conflicting answers, we conduct experiments using our dataset under two

[4] Each answer is also linked to the passage(s) containing its respective evidence.

prompting modes — defaultive and conflict-aware, similar to Hou et al. (2024).

In MAQA and RAG settings, a system is expected to generate natural-language responses that coherently articulate the information requested by an input instruction. We refer to this as the **defaultive** prompting mode, where the prompt simply states to answer a question based on the given sources. In the **conflict-aware** prompting mode, the model is also explicitly instructed to identify and indicate any conflicts that arise when answering the question (full prompts are in Figure 4 in App. E).

4.2 Tested Models

We selected eight top-performing open- and closed-source LLMs to evaluate in our setting. Specifically, since conflict-aware MAQA requires for strong reasoning abilities, to successfully identify conflicts across multiple passages, we selected four LLMs with an explicit reasoning step. We employed two flagship closed-source LLMs: `Gemini 2.5 Pro`,[5] and OpenAI's `o3`,[6] and to allow reproducibility, we selected two open-source reasoning language models: `DeepSeek-R1` (DeepSeek-AI, 2025) and `Qwen3-235B-A22B`.[7] Finally, we selected four popular non-reasoning models: `gpt-4o` (OpenAI-Team, 2024b), `Gemini 2.0 Flash`,[8] `Qwen2.5-72B` (Yang et al., 2024), and `DeepSeek-V3` (DeepSeek-AI, 2024). The eight models were evaluated on our NatConfQA dataset with the two prompting modes.

4.3 Evaluation Metrics

To evaluate model performance, we follow the two quality criteria of the task (§2) — *answer quality* and *conflict identification*. To measure the two quality criteria, we define precision, recall, and F_1 measures per a conflict-aware MAQA instance, as described below. A system's overall scores are the average of each of the three instance-level metric scores.

Preparation step: decomposing the system response. Consider an instance of conflict-aware MAQA (§2), characterized by input question q and passages P, and reference answers A and conflicting answer pairs C. A system responds with a free text response y, which coherently addresses q based on P. An interpreter (we use an `o4-mini` LLM) decomposes y into a set of distinct answers $\hat{A} = \{\hat{a}_1, \ldots, \hat{a}_k\}$, and then identifies all conflicting answer pairs in y as $\hat{C} = \big\{\{\hat{a}_i, \hat{a}_j\} \mid \hat{a}_i \text{ and } \hat{a}_j \text{ conflict within } y\big\}$. This decomposition step enables the evaluation, as described next.

Metrics for answer quality. To evaluate the correctness of an answer from the system's response, we adapt the recall and precision metrics from traditional MAQA tasks to our setting (Min et al., 2020; Amouyal et al., 2023).

$\text{recall}_{\text{ans}}$ is the fraction of reference answers A found in the system response y, while $\text{precision}_{\text{ans}}$ is the fraction of system-derived answers $\hat{A}$ found in the given passages P. Formally:

$$(1) \quad \text{recall}_{\text{ans}} = \sum_{i=1}^{m} \frac{\mathcal{J}_{\text{ans}}(a_i, y)}{m}$$

$$(2) \quad \text{precision}_{\text{ans}} = \sum_{i=1}^{k} \frac{\mathcal{J}_{\text{ans}}(\hat{a}_i, P)}{k}$$

where $\mathcal{J}_{\text{ans}}(a_i, T)$ denotes a judge's binary decision for whether answer a_i is found within the context T, with $m = |A|$ and $k = |\hat{A}|$. Accordingly, we define the per-instance score $F_{1_{\text{ans}}}$ as the harmonic mean of $\text{recall}_{\text{ans}}$ and $\text{precision}_{\text{ans}}$. For $\mathcal{J}_{\text{ans}}$ we employ an `o4-mini`[9] LLM (LLM-as-a-judge Liu et al., 2023; Zheng et al., 2023), which shows strong correlations to human judgments (see Appendix G for details).

Metrics for conflict identification. Prior works define conflict detection as a classification task: deciding whether a system-generated answer signals the presence of conflicting information in the retrieved passages (Xu et al., 2024; Hou et al., 2024). We extend their formulation to the general case where arbitrary pairs of distinct answers may conflict in the given passages.

$\text{recall}_{\text{conf}}$ measures the fraction of reference conflicts C captured by the system-derived answers $\hat{A}$, while $\text{precision}_{\text{conf}}$ measures the fraction of system-derived conflicting answer pairs $\hat{C}$ that are also

[5] https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025

[6] https://openai.com/index/o3-o4-mini-system-card

[7] https://qwenlm.github.io/blog/qwen3

[8] https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024

[9] https://openai.com/index/o3-o4-mini-system-card with high reasoning effort.

conflicting in passages P.[10] Formally:

$$(3)\quad \text{recall}_{\text{conf}} = \sum_{\{a_i,a_j\}\in C} \frac{\mathcal{J}_{\text{conf}}(a_i, a_j, y)}{|C|}$$

$$(4)\quad \text{precision}_{\text{conf}} = \sum_{\{\hat{a}_i,\hat{a}_j\}\in \hat{C}} \frac{\mathcal{J}_{\text{conf}}(\hat{a}_i, \hat{a}_j, P)}{|\hat{C}|}$$

where $\mathcal{J}_{\text{conf}}(a_i, a_j, T)$ denotes a judge's binary decision for whether a_i and a_j are indicated as conflicting in the context T. Accordingly, we define the per-instance score $F_{1_{\text{conf}}}$ as the harmonic mean of $\text{recall}_{\text{conf}}$ and $\text{precision}_{\text{conf}}$. We use an LLM-as-a-judge for $\mathcal{J}_{\text{conf}}$ as well.

5 Results and Analysis

In this section, we first present results on the two subsets of NatConfQA under both prompting modes, and analyze the general trends (§5.1, §5.2). We then conduct a manual error analysis on a sample of model responses (§5.3), uncovering the techniques used by models to wrongly handle conflicts.

5.1 Results on the *Conflict* Subset

The performance of the eight models (§4.2) on the *Conflict* subset of NatConfQA (§3.3) is reported in Table 1. We compare the use of the defaultive prompt mode against the conflict-aware prompt mode (§4.1) based on the recall, precision and F_1 metric scores (§4.3), and finally corroborate the observed trends with human judge.

Defaultive prompting. Under the default prompt, i.e., without any conflict-related guidance, the models exhibit relatively low $\text{recall}_{\text{conf}}$ (37.8 – 67.8; 9th column in Table 1), indicating that they struggle to identify and convey conflicts without explicit instruction. Across model families, non-reasoning models perform competitively with their reasoning-enabled counterparts, specifically the open-source ones. For instance, `DeepSeek-V3` achieves a $\text{recall}_{\text{conf}}$ of 55.5, surpassing the 50.6 of `DeepSeek-R1`.

Moreover, models appear to struggle less with retrieving answers from the passages, as indicated by the relatively higher $\text{recall}_{\text{ans}}$ scores. However there is still much room for improvement on this front as well.

In contrast, precision metrics remain uniformly high: both $\text{precision}_{\text{conf}}$ and $\text{precision}_{\text{ans}}$ exceed 80 and 92, respectively. This pattern is expected, since models are more prone to omission errors (which affect recall) than to producing irrelevant or spurious content.

Conflict-aware prompting. When explicitly prompted to identify conflicting answers (conflict-aware mode), models performance improve significantly (ΔCA columns). Notably, all eight models exhibit significant improvements in $\text{recall}_{\text{conf}}$, ranging from 7.5 to 27.6 points, indicating that explicitly guiding models to seek conflicts is effective for identifying them. Moreover, six of the eight models observed an increase in $\text{recall}_{\text{ans}}$, while precision on both criteria remains more or less comparable. Overall, for conflict instances, applying conflict-aware prompting is highly advantageous, improving both answer quality and conflict identification for most models. The subtle, yet meaningful, change in the prompt goes a long way for helping strong LLMs sense conflicts in the MAQA setting.

Conflicting subsets analysis. Next, we analyze model performance across three disjoint subsets of conflict instances in NatConfQA: (1) **Yes/No** questions; (2) WH-questions in which all answer pairs are conflicting (**WH-conflict**); and (3) WH-questions that include both conflicting and non-conflicting answer pairs (**WH-mix**). Table 2 reports results averaged over eight models (see Figure 7 in App. J for per model results). Notably, WH-mix is the most challenging subset and shows the smallest gains from conflict-aware prompting compared with the other two subsets. This suggests that the presence of both conflicting and non-conflicting signals within the same instance increases ambiguity, making it harder for models to reliably identify and reason about the conflicting information.

Human judgment results. To further corroborate the general trends observed above, we employed our evaluation protocol with a human judge to the two top-performing models on 120 NatConfQA instances (additional details in App. G). In Table 8, we report similar trends — LLMs perform better on $\text{recall}_{\text{ans}}$ and $\text{recall}_{\text{conf}}$ when prompted in a conflict-aware setting (up by 5.3 and 33.3 points for the two models). Moreover, we measured a strong correlation (Pearson's $r > 0.62$; Table 7) between the human and LLM judges for the four metrics from §4.3, further supporting our findings.

[10] Since system-derived answers may appear in the source passages, we evaluate precision for both criteria against these passages rather than relying on the reference answers.

	Model	Answer Quality						Conflict Identification					
		$\text{precision}_{\text{ans}}$		$\text{recall}_{\text{ans}}$		F1_{ans}		$\text{precision}_{\text{conf}}$		$\text{recall}_{\text{conf}}$		F1_{conf}	
		D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA
non-reasoning	gpt-4o	92.4	↑ 2.2	59.3	6.7	67.5	6.0	85.4	↓ 2.2	37.8	27.6	67.2	↑ 12.5
	Gemini 2.0 Flash	97.8	↓ 1.0	63.8	↑ 2.2	73.8	↑ 0.7	82.5	↑ 1.1	54.7	7.5	66.4	↑ 7.3
	DeepSeek-V3	92.4	↑ 1.1	65.2	4.5	71.8	4.5	83.0	↓ 4.2	55.5	15.4	75.8	↑ 1.4
	Qwen-2.5-72B	93.2	↓ 0.1	60.6	4.4	69.3	↑ 3.0	80.2	↓ 1.2	46.0	24.7	72.2	↑ 4.8
reasoning	o3	93.8	↓ 2.1	71.1	7.3	77.0	3.5	84.8	↑ 1.9	67.8	16.6	81.2	↑ 6.9
	Gemini 2.5 pro	96.2	↓ 0.1	67.7	↑ 3.6	75.6	↑ 3.3	85.8	↓ 4.6	61.8	17.9	77.2	↑ 3.0
	DeepSeek-R1	95.3	↓ 0.7	57.0	↓ 0.2	64.6	↑ 0.7	84.1	↓ 6.8	50.6	13.8	66.9	↑ 3.4
	Qwen-3-235B	94.3	↓ 0.3	56.0	↓ 3.3	62.8	↓ 0.8	85.3	↓ 1.5	50.3	9.0	69.8	↑ 0.8

Table 1: Performance on the *Conflict* subset of NatConfQA for non-reasoning (upper half) and reasoning models (lower half). Average precision, recall, and F_1 scores are reported for the two quality criteria (§4.3) — answer quality (left section) and conflict identification (right section). Results are shown for when models apply defaultive prompting (D), together with the absolute change in scores when applying conflict-aware prompting instead (ΔCA). Symbols ↓ / and ↑ / denote negative/significant-negative and positive/significant-positive changes. See Appendix H for details on significance-testing. Overall, conflict-aware prompting yields improvements for nearly all models across both evaluation criteria.

	Yes/No	WH-All	WH-Mix
defaultive	54.5 ± 2.2	53.1 ± 3.6	46.0 ± 5.7
conflict-aware	71.3 ± 2.0	72.9 ± 3.2	53.3 ± 5.6

Table 2: Average conflict-identification recall with 90% confidence intervals across eight models, reported for both prompting modes and all three conflict subsets (as detailed in §5.1).

5.2 Results on the *No-conflict* Subset

Next, we conducted an experiment which mirrors the traditional MAQA task, requiring models to generate responses that incorporate all relevant answers from the passages, *without* conflicting information. Specifically, we test four models' performance on the *No-conflict* subset of NatConfQA, under both prompting modes, as a reference for the *Conflict* subset's experiments above. All tested models (both reasoning and non-reasoning LLMs) exhibit high answer quality under defaultive prompting ($F_{1_{\text{ans}}} > 90$; see full results in Table 6 in Appendix F). This suggests that retrieving and integrating answers from passages is easier for models when no conflicts are present. However, when prompted in conflict-aware mode, performance slightly degrades (up to 5.5 in $F_{1_{\text{ans}}}$), possibly because the enforced knowledge of potential conflicts (even when none exist) somewhat disrupts the model's natural inference.

5.3 Error Analysis – Dealing with Conflicts

The large performance gap between the *Conflict* and *No-conflict* subsets of NatConfQA (§5.1, §5.2) calls for further examination. To that end, we conducted a manual error analysis on 160 sampled system responses generated by four models under defaultive prompting, on the *Conflict* subset (see Appendix I for full details). A human annotator categorized each system response into one of four main pre-defined error categories,[11] if an error was found (80 of the 160 instances), as follows:

Error	Description of response	Frequency
Choose	contains one reference answer	42%
Generalize	unifies answers by generalizing	17%
Resolve	hallucinates info to settle conflicts	13%
Refrain	does not answer question	5%

See examples in Table 9 in the appendix. The four models exhibit similar distributions of error types.

When the examined models made mistakes in their responses, it generally seems as if they tried to overcome conflicts through manipulative techniques. In 42% of the cases, they simply chose one answer in order to refrain from dealing with the conflicts. In 17% of the cases, they generated a response that unified the conflicting answers into a general answer that does not disclose the conflicts (e.g., by averaging numbers). In about 13%

[11] These categories were identified through a preliminary analysis for prominent error types, and inspired by Jiayang et al. (2024).

Dataset	Collection Method	#Instances	Conflicting Pairs	Conflict Type	Avg. #Passages	Avg. Passage Length (words)
ConflictingQA (Wan et al., 2024)	Automatic	238	All	Factual	9.2	314
WikiContradict (Hou et al., 2024)	Manual	253	All	Factual	2	43
DebateQA (Xu et al., 2024)	Automatic	2,941	N.D.	Point-of-View	4.2	4687.6
ECon (Jiayang et al., 2024)	Automatic	1,666	All	Factual	3	47.3
NatConfQA (Ours)	Manual	677	All & Mixed & None	Factual	5.6	251.5

Table 3: Representative datasets for conflict-aware QA. "Conflicting Pairs" indicates whether an instance in the dataset has only conflicting answers (*All*), conflicting and non-conflicting answers (*Mixed*), no conflicting answers (*None*), or whether that distinction is not well defined *N.D.*. "Avg. # Passages" denotes the average number of passages per instance.

of the cases, the models fabricated information in an attempt to reconcile the conflicts. Finally, another approach was to simply respond with a general comment related to the question, without answering it.

These phenomena observed on high-end LLMs demonstrate the manners with which models attempt to overcome conflicting information. We call upon the research community to dive deeper into this matter, not only to solve conflict-related tasks such as ours, but also to better understand the way in which LLMs handle inconsistencies in knowledge. Future research should explore developing systems that embrace the complexity of conflicts rather than simply resolving them.

6 Related Work

In multi-answer QA, a question may have multiple valid answers, each supported by its own evidence (Voorhees, 2004). Although most datasets for this task generally assume that the different answers are consistent and complementary (Kwiatkowski et al., 2019; Zhu et al., 2020; Li et al., 2022; Zhong et al., 2022; Amouyal et al., 2023), in real-world scenarios, a query can expose conflicts or discrepancies between the different textual sources.

Yet, there are only several QA datasets that address conflicting answers, each exhibiting its own set of limitations. Table 3 shows the differences between NatConfQA and existing benchmarks. In general, conflicting QA instances in naturally occurring texts are scarce, hence a popular strategy of prior works is to automatically introduce synthetic misinformation in texts, generating this way conflicting evidences for a QA instance (Jiayang et al., 2024; Liu et al., 2025b; Su et al., 2024; Wang et al., 2025; Ming et al., 2025). This approach inherently introduces artificial biases for the types of conflicts included in the dataset, as determined by the synthetic generation method. Another approach to derive conflicting instances involves utilizing existing Yes/No questions coupled with documents retrieved from search results that contain conflicting information (Wan et al., 2024).

Other recent works focus on various types of conflicts in RAG settings, where there is a single correct answer (Liu et al., 2025a), multiple points of view (Xu et al., 2024), or a mix of different conflict types, such as temporal, misinformation, or opinion (Cattan et al., 2025). Most similar to our work, WikiContradict (Hou et al., 2024) includes human-annotated QAs that incorporate naturally-occurring (rather than synthetic) conflicting answers, found in Wikipedia articles. Yet, the instances in this dataset are limited to only two relatively short evidence passages, which always contradict each other.

In contrast to existing resources, NatConfQA is a human-annotated dataset composed of naturally occurring conflicts between the different answers, covering both Yes/No and WH- questions. Additionally, each instance includes on average 5.5 passages. Importantly, our work is the first to collect fine-grained annotations for each pair of answers, where some answer pairs are conflicting while others are not (the "mixed" category in Table 3. This annotation scheme enables more realistic assessment of models' ability to identify naturally occurring conflicting answers, while distinguishing them from non-conflicting answers.

7 Conclusion

In this work we enhance the *Conflict-Aware Multi-Answer QA* task by explicitly requiring conflict identification among answers. We create a dataset for the enhanced task, via a novel cost-effective methodology that leverages fact-checking datasets. Our NatConfQA dataset is a realistic, conflict-rich benchmark that challenges current strong models. We test several state-of-the-art LLMs on the dataset, and show that models still struggle with surfac-

ing conflicting answers consistently, particularly in instances that contain both conflicting and non-conflicting answers, even when expressly prompted to be on the watch for potential conflicts. Finally, an error analysis of model responses exposes manners in which LLMs mishandle conflicts.

Limitations

We employed LLMs for many tasks throughout this paper, including conflict-aware MAQA, evaluation of several criteria, and response decomposition. While we conducted some reasonable prompt-engineering for these assignments, it is possible that even more effective prompts would improve or change results.

Since pre-trained LLMs' training datasets are not fully documented, we can't rule out overlap with the underlying data used for creating our dataset, raising the risk of contamination.

References

Rami Aly, Christos Christodoulopoulos, Oana Cocarascu, Zhijiang Guo, Arpit Mittal, Michael Schlichtkrull, James Thorne, and Andreas Vlachos, editors. 2021. *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*. Association for Computational Linguistics, Dominican Republic.

Samuel Amouyal, Tomer Wolfson, Ohad Rubin, Ori Yoran, Jonathan Herzig, and Jonathan Berant. 2023. QAMPARI: A benchmark for open-domain questions with many answers. In *Proceedings of the Third Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 97–110, Singapore. Association for Computational Linguistics.

Arie Cattan, Alon Jacovi, Ori Ram, Jonathan Herzig, Roee Aharoni, Sasha Goldshtein, Eran Ofek, Idan Szpektor, and Avi Caciularu. 2025. Dragged into conflicts: Detecting and addressing conflicting sources in search-augmented llms.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *CoRR*, abs/2412.19437.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948.

Thomas Diggelmann, Jordan Boyd-Graber, Jannis Bulian, Massimiliano Ciaramita, and Markus Leippold. 2021. Climate-fever: A dataset for verification of real-world climate claims.

Kevin Fischer, Darren Fürst, Sebastian Steindl, Jakob Lindner, and Ulrich Schäfer. 2024. Question: How do large language models perform on the question answering tasks? answer:.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. 2024. Wikicontradict: A benchmark for evaluating llms on real-world knowledge conflicts from wikipedia.

Cheng Jiayang, Chunkit Chan, Qianqian Zhuang, Lin Qiu, Tianhang Zhang, Tengxiao Liu, Yangqiu Song, Yue Zhang, Pengfei Liu, and Zheng Zhang. 2024. Econ: On the detection and resolution of evidence conflicts.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Haonan Li, Martin Tomko, Maria Vasardani, and Timothy Baldwin. 2022. MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Siyi Liu, Qiang Ning, Kishaloy Halder, Zheng Qi, Wei Xiao, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025a. Open domain question answering with conflicting contexts. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1838–1854, Albuquerque, New Mexico. Association for Computational Linguistics.

Siyi Liu, Qiang Ning, Kishaloy Halder, Wei Xiao, Zheng Qi, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025b. Open domain question answering with conflicting contexts.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522, Singapore. Association for Computational Linguistics.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Yifei Ming, Senthil Purushwalkam, Shrey Pandit, Zixuan Ke, Xuan-Phi Nguyen, Caiming Xiong, and Shafiq Joty. 2025. Faitheval: Can your language model stay faithful to context, even if "the moon is made of marshmallows". In *The Thirteenth International Conference on Learning Representations*.

OpenAI-Team. 2024a. Gpt-4 technical report.

OpenAI-Team. 2024b. Gpt-4o system card. *CoRR*, abs/2410.21276.

John W. Pratt. 1959. Remarks on zeros and ties in the wilcoxon signed rank procedures. *Journal of the American Statistical Association*, 54(287):655–667.

Mourad Sarrouti, Asma Ben Abacha, Yassine Mrabet, and Dina Demner-Fushman. 2021. Evidence-based fact-checking of health-related claims. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3499–3512, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Ellen M. Voorhees. 2004. Overview of the trec 2003 question answering track. In *Proceedings of the Twelfth Text REtrieval Conference (TREC 2003)*, Gaithersburg, MD. National Institute of Standards and Technology.

Alexander Wan, Eric Wallace, and Dan Klein. 2024. What evidence do language models find convincing?

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. Retrieval-augmented generation with conflicting evidence.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Rongwu Xu, Xuan Qi, Zehan Qi, Wei Xu, and Zhijiang Guo. 2024. Debateqa: Evaluating question answering on debatable knowledge.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

Victor Zhong, Weijia Shi, Wen tau Yih, and Luke Zettlemoyer. 2022. Romqa: A benchmark for robust, multi-evidence, multi-answer question answering.

Ming Zhu, Aman Ahuja, Da-Cheng Juan, Wei Wei, and Chandan K. Reddy. 2020. Question answering with long multiple-span answers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3840–3849, Online. Association for Computational Linguistics.

A Fact-Checking Datasets for Conflict-aware MAQA

We build upon two established fact-checking benchmarks, CLIMATE-FEVER and HEALTHVER, which provide real-world claims paired with supporting, refuting, and neutral evidence passages.

CLIMATE-FEVER. The CLIMATE-FEVER dataset (Diggelmann et al., 2021) includes 808 claims. Of these, 654 claims have only non-conflicting evidence (all support), while 154 claims are labeled as *disputed*, containing conflicting evidences.

HEALTHVER. The HEALTHVER dataset (Sarrouti et al., 2021) comprises 1,084 claims. Among them, 607 claims feature only supporting evidence (no conflict), and 477 claims include both supporting and refuting evidence, yielding true conflicts.

To ensure a diverse and representative subset of questions, we address the high redundancy in HEALTHVER, where many real-world health-related claims are duplicated. We randomly sampled a single instance per topic from the 477 conflicting claims spanning on 55 topic questions. This process resulted in a final subset of 55 unique instances.

Licensing. Neither CLIMATE-FEVER nor HEALTHVER specify an explicit license. Upon publication, we will release NATCONFQA under the CC BY 4.0 license,[12] permitting unrestricted reuse with attribution for research purposes.

B Dataset Annotation Details

Annotator details. Our two annotators were undergraduate or graduate students, and are fluent English speakers. They underwent two training

[12] https://creativecommons.org/licenses/by/4.0/

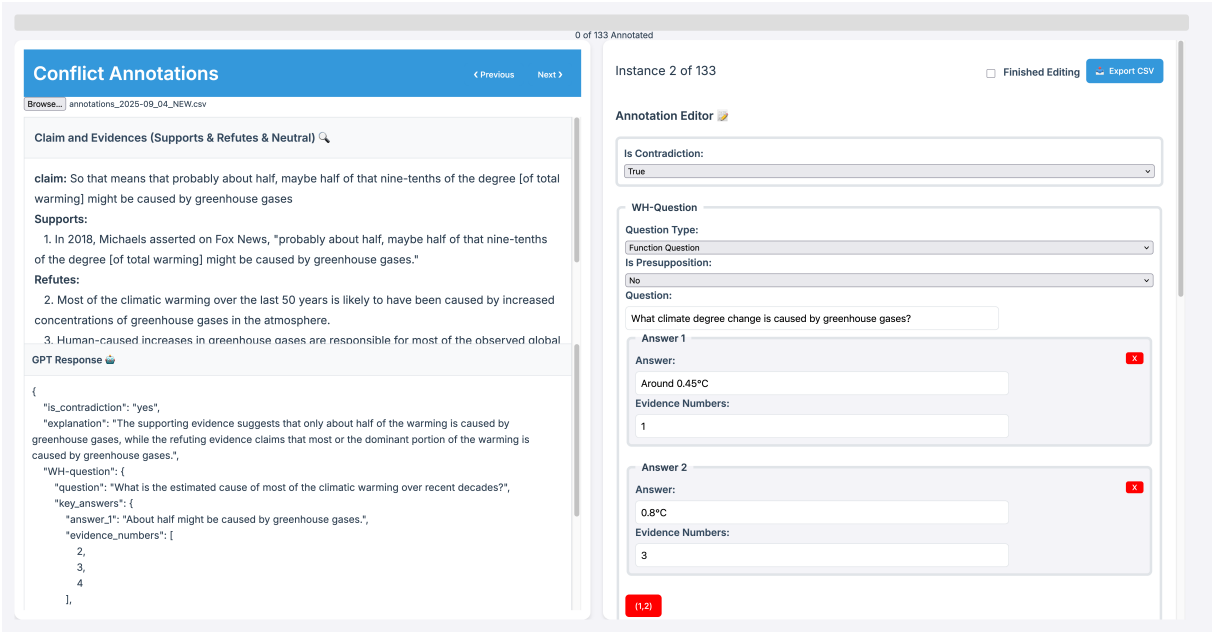


Figure 2: Screenshot of our annotation interface. On the left, annotators view the fact-checking instance, including the claim and evidence sentences grouped by their initial labels (support, refute, neutral). At the bottom, LLM-generated WH and yes/no question–answer suggestions are displayed (see Figure 9 for the prompts). On the right, annotators write or edit their own questions and corresponding answers.

iterations, each on 10 instances. They were compensated at approximately $14 per hour. The annotators worked a combined total of 52 hours to prepare the NatConfQA dataset. The annotators were informed that their annotations are for research purposes, and that they can terminate their participation in the process whenever they want.

Annotation tool. To facilitate the annotation process, we developed a dedicated annotation tool that supports question-writing, linking respective evidence, and labeling conflicting answer pairs.

We present screenshots of the annotation interface and guidelines for both WH and Yes/No questions in Figures 2 and 3, respectively. The provided instructions guided annotators in accurately identifying conflicts and extracting relevant answers based on the given evidence. Additionally, annotators were instructed to record the evidence IDs that support each annotated answer. Our custom annotation tool further allows annotators to add as many answers as necessary and seamlessly author both WH-type and binary (Yes/No) questions for each claim. While the guidelines place a strong emphasis on handling conflict instances, annotators were allowed to skip the conflict-oriented instructions when working on the support instances.

C Passage Extraction for Evidence Sentences

To simulate a realistic QA setting, we reverse the typical sentence-level focus of fact-checking datasets by retrieving entire passages surrounding each evidence span.

For HealthVer, which is derived from scientific abstracts in the CORD-19 corpus[13], we locate the original abstract corresponding to each evidence sentence. We perform an exact string match of the sentence within the CORD-19 collection and extract the full abstract to serve as the passage context.

For climate-fever, which uses evidence drawn from Wikipedia, we scraped the English Wikipedia pages as of February 1, 2020 and converted them to plain text. We then employ the RapidFuzz fuzzy-matching library[14] to identify each evidence sentence within its article. Finally, we heuristically expand to the surrounding paragraph—defined by nearest blank lines or section headers—to create a coherent passage that preserves the original narrative flow.

This passage-level extraction ensures that each QA instance reflects the broader context in which evidence appears, aligning our setup with realistic

[13] https://www.semanticscholar.org/cord19
[14] https://github.com/maxbachmann/RapidFuzz

Statistic	*No-conflict*	*Conflict*
# Instances	408	269
Avg. # Passages	6.7	5.6
Avg. # Answers (WH) [min–max]	3.8 [2–13]	2.6 [2–10]
Avg. # Conflict Pairs	–	1.3

Table 4: Statistics for our NatConfQA dataset. The three average stats are per instance. *Avg. # Passages* is the mean number of passages per instance; *Avg. # Answers (WH) [min–max]* reports the average number of answers per WH question and the corresponding range; *Avg. # Conflict Pairs* is the average number of conflicting answer pairs per instance (e.g., always 1 for Yes/No).

Model Name	Model Tag	Reasoning
`GPT-4o`	`gpt-4o-2024-08-06`	✗
`Gemini 2.0 Flash`	`gemini-2.0-flash-001`	✗
`DeepSeek-V3`	`deepseek-ai/DeepSeek-V3`	✗
`Qwen-2.5-72B`	`Qwen/Qwen2.5-72B-Instruct-Turbo`	✗
`o3`	`o3-2025-04-16`	✓
`Gemini 2.5 Pro`	`gemini-2.5-pro-preview-03-25`	✓
`DeepSeek-R1`	`deepseek-ai/DeepSeek-R1`	✓
`Qwen-3-235B`	`Qwen/Qwen3-235B-A22B-fp8`	✓
`o4-mini`	`o4-mini-2025-04-16`	✓

Table 5: Exact model version tags for the models used in this work.

Retrieval-Augmented Generation workflows.

D NatConfQAStatistics

Table 4 presents statistics for NatConfQA,[15] including number of questions, and number of passages, answers and conflicts per question. The dataset comprises two subsets, one with only supporting answers (titled *No-conflict* in the table) and another with conflicting answers (titled *Conflict*). For both types, there are WH-questions as well as Yes/No questions.

E Details for the Experimental Setup

In this section, we provide technical details of the experimental setup defined in Section 4.2. All experiments were conducted between May 1 and May 19, 2025, using the OpenAI,[16] Google,[17] and Together.ai[18] APIs. The exact version tags for all models utilized in this work are listed in Table 5.

For reproducibility, we set the temperature to 0 for all models with a maximum number of generated tokens (`max_tokens`) at 512. For the OpenAI models (`o3` and `o4-mini`), we configure `reasoning_effort` to `high`. For the Gemini models, we set `thinking_budget` to 1024 for `Gemini 2.5 Pro` and to 0 for `Gemini 2.0 Flash`. The total cost of the experiments using the three LLM APIs was approximately $500.

The prompts used for model evaluation are shown in Figure 4, while those for the LLM-as-a-judge are presented in Figure 5. There are five prompts in total for the LLM-as-a-judge: one for answer decomposition (Prompt 1) and four for the binary decision functions defined in Section 4.3 (Prompts 2–5), namely, $\mathcal{J}_{\text{ans}}(a_i, y)$, $\mathcal{J}_{\text{ans}}(\hat{a}_i, P)$, $\mathcal{J}_{\text{conf}}(a_i, a_j, y)$, and $\mathcal{J}_{\text{conf}}(\hat{a}_i, \hat{a}_j, P)$.

	Model	Answer Quality					
		$\text{precision}_{\text{ans}}$		$\text{recall}_{\text{ans}}$		F1_{ans}	
		D	ΔCA	D	ΔCA	D	ΔCA
non-reasoning	GPT-4o	98.6	↓ 1.4	95.6	↑ 0.7	95.9	↓ 0.1
	DeepSeek-V3	96.9	↓ 1.1	95.5	↑ 0.1	95.6	↓ 1.0
reasoning	o3	98.3	↓ 1.5	97.2	0.0	97.5	1.2
	DeepSeek-R1	97.6	↓ 1.3	89.7	↓ 4.2	90.7	5.5

Table 6: Average precision, recall, and F_1 percentages for the answer-quality criterion (§4.3) on the *No-conflict* subset of NatConfQA, reported for four LLMs. All models exhibit high scores (> 89) on all measures. See the caption of Table 1 for further details on notations.

F NatConfQA *No-conflict* Subset Results

In Table 6, we present results on the *No-conflict* subset of NatConfQA, using the same metrics and notation conventions as in Table 1.

G Effectiveness of LLM-as-a-Judge

Manually annotating answer quality and conflict identification is expensive, so we rely on a fast automatic judge — `o4-mini`[19] — and validate its agreement with humans. we randomly sampled 120 system responses from two models (`o3-high` and `Gemini 2.5 Pro`) under both prompting modes (§4.1). A human judge then applied the judgment protocol explained in Section 4.3, with guidelines similar to the instructions in the prompts in Figure 5, decomposing answers and making the binary decisions that constitute both metrics.

To assess the reliability of the automatic judge, we compute Pearson correlations between its la-

[15] For randomly sampled instances from the dataset, see Table 10.

[16] https://platform.openai.com

[17] https://aistudio.google.com

[18] https://together.ai

[19] https://openai.com/index/o3-o4-mini-system-card

Evaluation Metric	Recall	Precision
Answer quality	0.6645	0.6221
Conflict identification	0.6744	0.6371

Table 7: Pearson's r correlation coefficients for recall and precision, computed between human and LLM-as-judge over 120 samples ($p < 0.00003$). The results show strong correlations for all metrics between human and LLM judgments.

bels and the human annotations (Table 7). Both metrics show a strong positive correlation, with all correlation values exceeding 0.6.

Furthermore, the human evaluation results (Table 8) show performance trends similar to those from the automatic evaluation (Table 1). In both cases, conflict-aware prompting leads to higher recall — particularly for conflict identification.

H Significance Testing on Results

To compute significance in Table 1, we conducted Wilcoxon signed-rank tests comparing the default-ive and conflict-aware modes (Wilcoxon, 1945). We applied Pratt's conservative zero-difference method (Pratt, 1959) and report significance at $p < 0.01$.

I Error Analysis Details

In order to assess model errors, we sampled 160 outputs, using the default prompt described in §5.1, from four models (`O3`, `DeepSeek-R1`, `GPT-4o`, and `DeepSeek-V3`). Annotation was performed by a graduate student at an hourly rate of $14. We sampled 40 instances for each model where 20 were Yes/No questions and 20 were WH-question.

J Conflict Subsets Analysis

Figure 7 compares conflict-identification recall across three subsets—Yes/No, WH-All (all answer pairs conflict), and WH-Mix (mixture of conflicting and non-conflicting pairs)-for each model under two prompting types: Defaultive and Contradict-Aware. Bars show mean recall per model. Two consistent trends emerge. First, Contradict-Aware prompting substantially improves performance for most models, especially on Yes/No and WH-All, while gains on WH-Mix are smaller. Second, WH-Mix is the hardest subset: it has the lowest base recall under Defaultive prompting and remains lowest even after Contradict-Aware prompting. This suggests that instances containing both conflicting and non-conflicting evidence introduce challenge that current models struggle to resolve.

K AI Assistance

Throughout this project, we were assisted by AI tools to accelerate both code implementation (some code snippets) and manuscript preparation (local rephrasing). We carefully reviewed and refined all AI-generated content to ensure technical accuracy and stylistic consistency.

Model	Answer Quality						Conflict Identification					
	$\text{precision}_{\text{ans}}$		$\text{recall}_{\text{ans}}$		F1_{ans}		$\text{precision}_{\text{conf}}$		$\text{recall}_{\text{conf}}$		F1_{conf}	
	D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA	D	ΔCA
o3	85.5	↓ 0.8	84.8	5.3	85.1	↑ 2.7	100.0	↓ 6.7	46.7	26.7	85.7	↑ 4.3
gemini-2.5-pro	79.9	↓ 0.6	83.4	↑ 5.3	81.6	↑ 1.9	81.8	↑ 11.5	46.7	33.3	57.1	↑ 42.9

Table 8: Human evaluation of performance on the *Conflict* subset of NatConfQA for the two top performing models from Table 1, based on 120 instances. Average precision, recall, and F_1 scores are reported (percent) for the two evaluation criteria (§4.3): answer quality (left) and conflict identification (right). Columns show results for default prompting (D) and the absolute change when using conflict-aware prompting (ΔCA). Symbols ↓/ and ↑/ denote negative/significant negative and positive/significant positive changes, respectively (using $p < 0.05$, see Appendix H for significance testing methodology).

Error Type	**Description**	**Example (Question / Reference answers / Model answer)**
Choose Answer	The model outputs only one of the answers, omitting the others.	**Q:** What is the rate of ice mass loss in Antarctica? **Reference:** "+82 Gt/yr" and "–220 Gt/yr" **Model:** "Antarctica is gaining 82 Gt of ice per year."
Answer Generalization	The model summarizes conflicting values vaguely (e.g., as an average or range) instead of presenting them distinctly.	**Q:** How much has global temperature risen? **Reference:** "0.45 °C" and "0.8 °C" **Model:** "Temperatures have increased between 0.45 to 0.8."
Conflict Resolution	The model mentions all answers but presents them as if there is no conflict, possibly with hallucinatory information.	**Q:** What is the estimated global temperature rise since 1900? **Reference:** "0.45 °C" and "0.8 °C" **Model:** "Until 1945, the rise was 0.45 °C, and then 0.8 °C."
Refrain from answering	The model does not provide an answer to the question or provides an irrelevant response.	**Q:** What is the estimated rate of ice loss from Greenland per year? **Reference:** "Between 200 and 300 Gt/yr" and "Approximately 220 Gt/yr" **Model:** "Greenland is a large landmass covered in ice."

Table 9: Error types for the error analysis. Each instance is labeled with a single error type based on the model's ability to reflect, miss, or misrepresent the underlying conflict.

NatConfQA Manual Annotation Guidelines

1. Task Overview

- Our goal is to reveal conflicts in data through questions and answers.
- You will be provided with a claim along with supporting, refuting, and neutral evidence.
- Your task involves two main objectives:

(1) **Conflict detection**: Assess whether there is a conflict between the supporting and refuting evidence.
Yes **No** **Uncertain**

(2) **Q–A Generation**: Formulate questions that capture the conflict based on the given claim and evidence. For each question, provide the differing answers along with the evidence IDs supporting each answer.

2. Rules for Effective Question Formulation

You should follow the following rules when writing the questions:

1. **Conflict-inducing** – The question should prompt a conflicting response.
2. **Specificity** – Ensure the question targets detailed aspects of the text rather than general topics.
3. **Assumption-free** – The question should be free from specific assumptions (from the text).

3. Rules for Effective Answer Formulation

You should follow the following rules when writing the answers:

1. **Completeness** – Ensure that the answers address all the evidence provided.
2. **Conciseness** – Keep the answers brief and to the point while maintaining clarity.
3. **Relevance** – Ensure that the answers directly address the question asked.
4. **Atomicity** – Each answer should contain only a single response. If multiple answers exist, they should be separated into distinct answers rather than combined.

Figure 3: Summary of the annotation guidelines for creating our NatConfQA dataset. The full guidelines will be provided along with the complete resources.

Source Data	Q-Type	A-Type	Question	Answers
CLIMATE-FEVER	YN	No-conflict	Do changes in land use, such as agriculture and deforestation, contribute to Earth's climate changes?	1. Yes
CLIMATE-FEVER	YN	Conflict	Do greenhouse gases increase Earth's temperature?	1. Yes 2. No
CLIMATE-FEVER	wh	No-conflict	What are the ranges of water vapor's contribution to the greenhouse effect?	1. Between 36% and 66% under clear sky conditions 2. Between 66% and 85% when including clouds
CLIMATE-FEVER	wh	Conflict	What is the perceived level of scientific agreement on climate change?	1. There is a scientific consensus on human-caused climate change 2. Scientific opinion is evenly divided or completely unsettled
HEALTHVER	YN	No-conflict	Does favipiravir decrease viral replication in COVID-19 patients?	1. Yes
HEALTHVER	YN	Conflict	Do natural remedies help prevent you from getting infected with COVID-19?	1. Yes 2. No
HEALTHVER	wh	No-conflict	What are the clinical symptoms observed in patients with COVID-19?	1. Fever 2. Dry cough 3. Sore throat 4. Dyspnea 5. Fatigue 6. Myalgia 7. Headache 8. Loss of smell and taste
HEALTHVER	wh	Conflict	What is possible in regards to cat to human coronavirus transmission?	1. Cats can transmit the virus to humans 2. Cats cannot transmit the virus to humans

Table 10: Eight randomly sampled instances from NATCONFQA. Q-Type: questoin type YN = yes/no; wh = WH-questions. A-Type: answer type. When the dataset is published, it will also include the associated passages supporting each answer. For simplicity, and since conflict instances in this sample contain only two answers, we do not include which specific answer pairs are in conflict.

NatConfQA Dataset Prompt Template

Provide a concise, single-sentence answer that includes every distinct answer to the following question, based on the given passages from multiple sources. If any answers conflict, clearly indicate which ones are in conflict while remaining objective and neutral.

Question: {question}
Passages: Passage #1:
{Passage 1 text}

Passage #2:
{Passage 2 text}
...

Figure 4: The prompt template used to prepare inputs for the LLMs when testing on NatConfQA. This is the template for both modes (defaultive prompting mode and conflict-aware prompting mode) as defined in §4.1. The underlined sentence is omitted in the defaultive mode and included in the conflict-aware mode.

LLM-as-a-Judge Prompt Templates

1. System Response Decomposition into Distinct Answers $\hat{A}$

Given the following question and long-form system answer, extract all distinct short answers that directly address the question. Ensure to extract all such sub-answers. The output must be a JSON object with a `"sub_answers"` key whose value is an array of sub-answer strings.

Question: `"{question}"`
System answer: `"{model_answer}"`

2. Answer a_i Exist in System Response y

Determine if the reference short answer appears anywhere in the longer system answer — even if it's noted as incorrect, anecdotal or implicitly mentioned. Return a JSON object with a single key, `"answer"`, whose value is either `"true"` or `"false"`:

- `"true"` if the reference short answer is present in any form (explicit or implicit).
- `"false"` otherwise.

Question: `"{question}"`
Reference short answer: `"{reference_answer}"`
System answer: `"{model_answer}"`

3. Answer $\hat{a}_i$ Exists in the passages P

Determine if the given short answer appears anywhere in the following passages — even if it's noted as anecdotal or implicitly mentioned. Return a JSON object with a single key, `"answer"`, whose value is either `"true"` or `"false"`:

- `"true"` if the given short answer is present in any form (explicit or implicit).
- `"false"` otherwise.

Question: `"{question}"`
Short answer: `"{model_answer}"`
Passages: `{passages}`

4. Answers a_i and a_j Conflict Within System Response y

Your task is to assess whether a pair of short answers to a question are clearly indicated as conflicting answers in a longer answer written by a question answering system. You should return a JSON object with an `"answer"` key `"true"` if the pair of short answers clearly indicated as conflicting in the longer system's answer or `"false"` otherwise.

Question: `"{question}"`
Short answer #1: `"{short_answer1}"`
Short answer #2: `"{short_answer2}"`
System answer: `"{model_answer}"`

5. Answers $\hat{a}_i$ and $\hat{a}_j$ Conflict Within Passages P

Your task is to assess whether the two short answers, which are located in the following passages, are clearly indicated as conflicting to the same question. You should return a JSON object with an `"answer"` key `"true"` if the pair of short answers clearly indicated as conflicting or `"false"` otherwise.

Question: `"{question}"`
Short answer #1: `"{short_answer1}"`
Short answer #2: `"{short_answer2}"`
Passages: `{passages}`

Figure 5: Prompt templates used in our experiments for LLM-as-a-judge as detailed in Section 4.3.

Manual Annotation Guidelines for Correlation Assessment

Task Overview: You are given a question, human-annotated reference answers (referred to as *reference_answers*), human-annotated conflicting answer pairs (referred to as *reference_conflicting_answer_pairs*), associated passages, and a model-generated answer; please follow the steps below.

1. Decompose the Model Answer into Atomic Answers

- Read the `model_answer` and split it into independent facts or claims ("atomic answers").
- Number them 0, 1, 2, . . . in the order they appear.
- Enter them in the format:
  - 0: First atomic answer. . .
  - 1: Second atomic answer. . .
  - 2: Third atomic answer. . .
- → Fill in `Model Answers Decomposed`.

2. Match Decomposed Answers to Reference Answers

- For each atomic answer (by index), check if it appears in the `reference_answers`.
- List the indices of those that match, e.g., `[0, 2]`.
- → Fill in `Matched Answers in Reference Answers`.

3. Identify Conflicting Reference-Answer Pairs

- Review the `reference_conflicting_answer_pairs` and identify each pair that directly contradicts the other.
- In the model's response, look for contrastive cues like "however," "but," or "on the other hand."
- Record each conflicting pair by their indices, e.g., `[(0,1), (1,2)]`.
- → Fill in `Conflicting Reference Answer Pairs Found`.

4. Match Model Answers to Passages or References

- For each atomic answer, check if it is supported by either a `reference_answer` or a paragraph.
- List the indices of atomic answers that are supported, e.g., `[0, 1]`.
- → Fill in `Found Model Answers in Reference Answers/Passages`.

5. Identify Conflicts among Model's Answers

- For each pair of atomic answers flagged by the model as conflicting, record:
  - `(i, j): 1` if the conflict is correct.
  - `(i, j): 0` if the conflict is incorrect.
- Example: `(0, 2): 1, (0, 1): 0`
- → Fill in `Found Conflicting Model's Answers`.

Figure 6: Annotation guidelines used by human annotators to assess the correlation between automatic LLM-based judgments and human evaluation. The process includes both conflict identification and answer quality metrics, supporting the measurement of recall and precision for each.

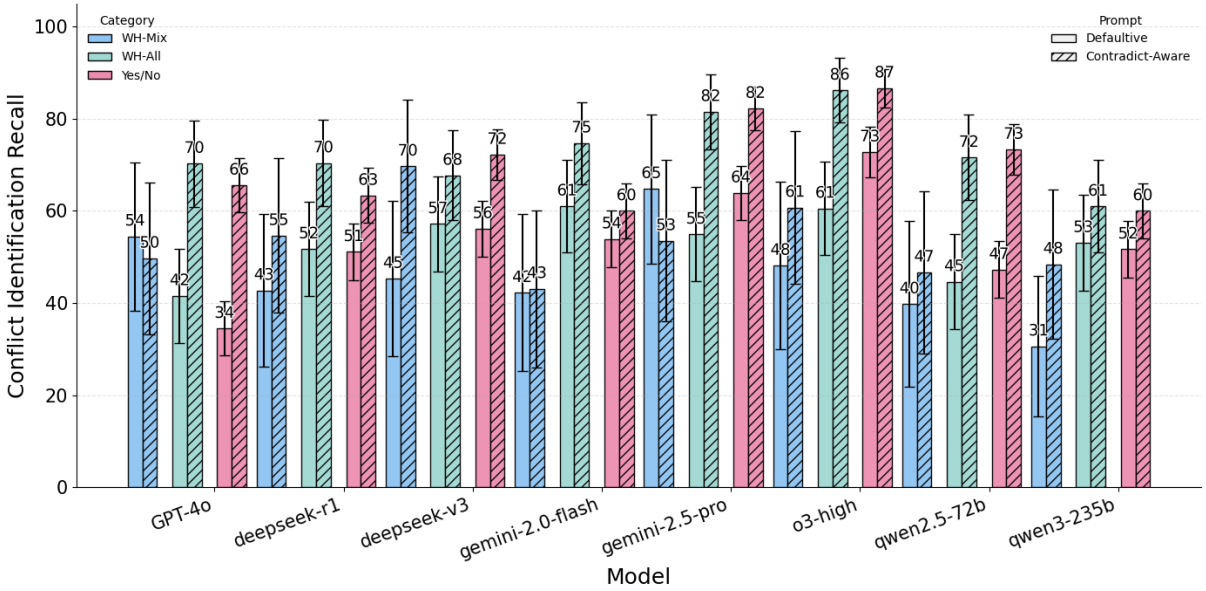


Figure 7: Conflict Identification Recall across models on three instance subsets (WH-Mix, WH-All, Yes/No), split by prompt type (defaultive vs. contradict-aware). Bars show mean conflict identification recall with 90% confidence intervals. Overall, contradict-aware consistently improves recall relative to defaultive (Table 1); however, performance on WH-Mix is notably lower than on WH-All and Yes/No for nearly all models, and the gains from contradict-aware are smaller on this subset, highlighting the challenge when conflicting and non-conflicting signals co-occur within the same instance.

Manual Annotation Guidelines for Error Analysis

Task Overview: You are provided with a question, the model's answer for that question, and a set of annotated reference answers. First, review the list of error categories in Table 9. Then, for each model output, classify it into the single most appropriate category, based on the explanation that best matches the model's output.

Steps:

- Read the **question**, **reference answers**, and the **model's answer**.
- If the model presents both sides with clear contrast (e.g., "but," "however"), label it `-2`.
- Otherwise, choose a single error ID that best describes the model error.
- Use `-1` if the error does not fit any predefined category and describe it in the notes.

Note: Annotate each instance with only one ID. Include a brief justification if necessary.

Figure 8: Annotation guidelines used by human annotators for conducting error analysis on outputs from different models.

Prompt for NatConfQA Question and Answers Suggestion for Conflicting Instances

You are given a claim along with multiple pieces of evidence, categorized as supporting, refuting, or neutral in relation to the claim. Your task is to analyze the evidence and perform the following subtasks:
(1) **Determine Contradiction:** Identify whether there is a direct contradiction between the supporting and refuting evidence. Respond with "yes" if such a contradiction exists, or "no" if not.
(2) **Explain Contradiction (if applicable):** Briefly explain how the supporting and refuting evidence conflict.
(3) **Generate a WH-Question (if applicable):** Create a factoid-style WH-question based on the conflicting information. Avoid directly referencing the contradiction. Provide short answers and indicate the evidence numbers supporting each.
(4) **Generate a Yes/No Question:** Frame a yes/no question that reflects the core contradiction or claim, supported by evidence.
Output Format:

```
{
  "is_contradiction": "yes" or "no",
  "explanation": "...",
  "WH-question": {
    "question": "...",
    "key_answers": {
      "answer_1": "...",
      "evidence_numbers": [...],
      ...
    }
  },
  "Yes/no-question": {
    "question": "...",
    "Yes_answers": [...],
    "No_answers": [...]
  }
}
{instance}
```

Prompt for NatConfQA Question and Answers Suggestion for Non-conflicting Instances

You are given a claim along with multiple pieces of evidence, categorized as supporting or neutral. Your task is to:
(1) **Generate a WH-Question:** Create a concise WH-question based on the supporting evidence. Provide short answers and list the supporting evidence numbers.
(2) **Generate a Yes/No Question:** Frame a yes/no question grounded in the supporting evidence and reflecting the claim's context.
Output Format:

```
{
  "WH-question": {
    "question": "...",
    "key_answers": {
      "answer_1": "...",
      "evidence_numbers": [...],
      ...
    }
  },
  "Yes/no-question": {
    "question": "...",
    "Yes_answers": [...],
    "No_answers": [...]
  }
}
{instance}
```

Figure 9: Prompt templates used to generate suggested questions and answers for annotators in the NatConfQA annotation process. The first prompt is applied to instances containing conflicting evidence, while the second is used for non-conflicting evidence.

Annotation Guidelines for NatConfQA Dataset Quality Assessment

1. Task Overview

- You will be provided with:
  - A **question** prompting the model's response
  - A list of **answers**, each associated with a unique ID
  - A list of **evidence** that may support the answers
  - A list of **answer pair labeled as conflicting**, specified by their answer IDs (if any)
- Your task consists of two main steps:

(1) **Answer Quality:** For each answer, determine whether it is clearly supported by at least one of the provided evidence.
  - Mark `1` if it is grounded (i.e., the answer is directly supported by any evidence).
  - Mark `0` if none of the evidence supports the answer.

(2) **Conflict Detection:** For each answer pair labeled as conflicting, determine whether they conflicting each other based on the provided evidence.
  - Mark `1` if the two answers clearly conflicting each other.
  - Mark `0` if the answers are compatible or describe different aspects that can co-exist.

2. Notes and Clarifications

- Focus only on what is explicitly stated in the evidence and answers. Avoid making assumptions beyond the given text.
- If a pair appears borderline or ambiguous, lean toward `0` and leave a short note.
- Use the comments field to explain any unclear cases or edge scenarios you encounter.

Figure 10: Annotation guideline to evaluate the quality of NatConfQA. We assess both answer quality and conflict identification using an external annotator. The annotation procedure includes verifying whether answers are supported by evidence and whether identified answer pairs are truly in conflict. Results show high agreement, indicating the task is well-defined and the annotation protocol is reliable.

Demystify Verbosity Compensation Behavior of Large Language Models

Yusen Zhang, Sarkar Snigdha Sarathi Das, Rui Zhang
Department of Computer Science and Engineering, Penn State University
{yfz5488, sfd5525, rmz5227}@psu.edu

Abstract

Recent work has revealed Large Language Models (LLMs) often exhibit undesirable behaviors, such as hallucination and toxicity, limiting their reliability and broader adoption. In this paper, we discover an understudied type of undesirable behavior of LLMs, which we term Verbosity Compensation (VC). VC is similar to the hesitation behavior of humans under uncertainty, compensating with excessive words such as repeating questions, introducing ambiguity, or providing excessive enumeration. We present the first work that analyzes Verbosity Compensation, explores its causes, and proposes a simple mitigating approach. Our experiments on five datasets of knowledge and reasoning-based QA tasks with 14 LLMs, reveal three conclusions. 1) A pervasive presence of VC across all models and all datasets. 2) The large performance gap between verbose and concise responses. We also demonstrate that this difference does not naturally diminish as LLM capability increases. 3) Higher uncertainty exhibited by VC responses across all five datasets, suggesting a strong connection between verbosity and model uncertainty. We propose a simple yet effective cascade algorithm that replaces the verbose responses with the other model-generated responses, alleviating the VC of the Mistral model from 63.81% to 16.16% on the Qasper dataset.

1 Introduction

Recent research has highlighted various undesirable behaviors of Large Language Models, such as hallucination (Huang et al., 2023), toxicity (Wen et al., 2023), and ethical bias (Tao et al., 2023), which pose significant risks to users. Among them, the verbose response issue where LLMs respond with excessive words has attracted more and more attention in the LLM era because of unnecessary long output for solving problems (Singhal et al., 2023) and the unavoidable high cost of LLM-generated tokens.

The existing work mainly focuses on the length of the response and its applications. Researchers found that imposing a length constraint in the prompt can improve the performance of LLMs, under chain-of-thought (Chiang and Lee, 2024; Nayab et al., 2024) and machine translation (Briakou et al., 2024) settings. Singhal et al. (2023) found RLHF training favors the lengthy response. However, length is not enough to analyze verbosity as it provides a general overview but fails to capture key fine-grained features such as content structure.

In this paper, we discover a type of undesirable verbosity behavior of LLMs. We term it Verbosity Compensation (VC). Instead of focusing merely on the *length*, we analyze the frequency, types, and their relation to model performance. We also find VC is closely connected to the uncertainty of LLMs, demystifying the mechanism of the VC behavior, and improving the understanding of both VC and uncertainty. Interestingly, VC is similar to the hesitation behavior of humans under uncertainty (Juola, 2008; Brookshire and McNeil, 2014). Figure 1 shows a motivating example. In the first response, LLM generates a concise answer that is correct with low uncertainty. In the second and third responses, instead of generating an answer concisely, such as "16.5", LLM repeats the question, and produces ambiguity, leading to a VC response with low performance and high uncertainty. VC is harmful and undesired for both users and servers. For the users, VC will lead to confusion and inefficiency (Fowler, 1927; Oppenheimer, 2006). When an LLM enumerates multiple answers, users are unclear about which one is correct. Besides, VC leads to bias among users of different length preferences if verbose answers attain higher/lower scores. For the servers, the verbosity leads to unnecessary higher costs and higher latency because of useless tokens.

To analyze the VC behavior systematically, we unify four long-context question-answering

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 160–178
November 9, 2025 ©2025 Association for Computational Linguistics

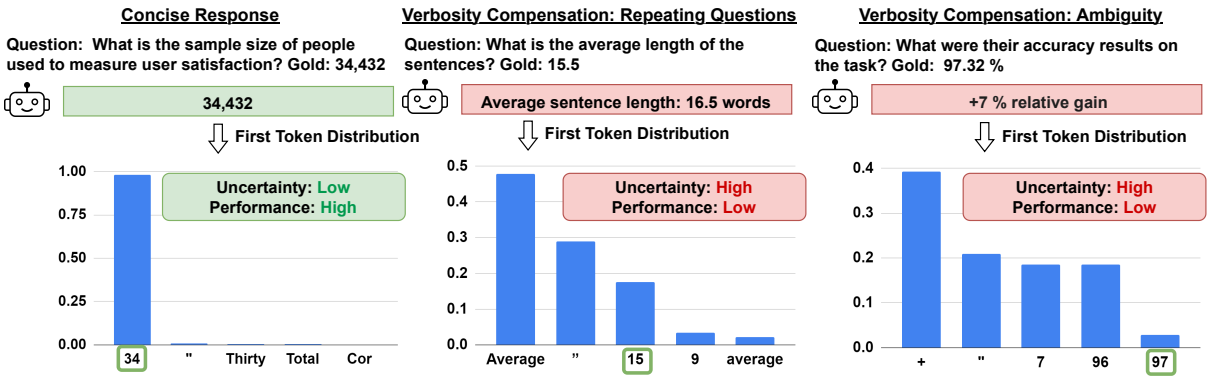


Figure 1: An illustration of comparison between concise and verbose responses. For each example, we ask the model to generate the response as concisely as possible. In the first response, LLM generates a concise answer, while in the second and third responses, LLM performs repeating, and ambiguity, leading to a verbose response with low performance and high uncertainty (Detailed numbers in Appendix B.2).

datasets and a reasoning-based language understanding dataset. We choose short-form QA with several tokens (comprising phrases, names, rather than complete sentences) in the gold answer to ensure the gold label is concise and easy to judge VC behavior in responses. We benchmark 14 LLMs on proposed datasets. Although we find that different models and datasets exhibit diverse distribution, we can categorize VC into five distinct types, including repeating questions, enumerating, ambiguity, verbose details, and verbose format. **The result reveals a pervasive presence of verbosity compensation (VC) across all models and all datasets**. Notably, GPT-4 exhibits a VC frequency of 50.40%. Meanwhile, we found that verbose responses exhibit significantly different recall from concise ones, with a notable drop of 24.72% on the Qasper dataset, **highlighting the urgent need to disentangle verbosity with veracity**.

Next, we measure the uncertainty of model responses using perplexity and Laplacian scores for open and closed-source models. We find that verbose responses exhibit higher uncertainty across all five datasets, suggesting **a strong connection between verbosity and model uncertainty**. Finally, we leverage the connection between performance and VC to develop a routing algorithm that obtains significant improvements over the random selecting baseline and uncertainty-based routing. To mitigate VC in LLMs, we propose a simple yet effective cascade algorithm that replaces verbose responses with responses of larger LLMs. Experiments demonstrate the efficacy of the proposed algorithm through tests on three model combinations: Gemma to Gemini, Mistral to GPT-4, and Llama to Claude. The results show that our approach effectively alleviates the VC of the Mistral model from 63.81% to 16.16% on the Qasper dataset. The insights above can inspire the development of practical applications and effective mitigation strategies. Future work can mitigate the uncertainty of the LLMs by alleviating VC behavior due to the proposed connections between them.

2 Related Work

Verbosity in LLM Responses Recently work has studied the verbosity of LLM-generated contents and its implications. Concise thoughts (Nayab et al., 2024) use prompts to constraint the length of Chain-of-thought reasoning and generate more concise responses with better performance. Ivgi et al. (2024) investigate the fallback behavior of LLM-generated responses when facing uncertainty. Singhal et al. (2023) investigate the correlation between generated length and reinforcement learning from human feedback (RLHF) techniques. Saito et al. (2023) find that LLMs sometimes prefer more verbose answers even if they have similar qualities. By contrast, Huang et al. (2024) find that GPT-4 prefers short responses in faithfulness and coverage when it comes to summarization. Unlike these works, we discover the connection between performance and verbosity compensation behavior in both CoT and general QA settings and connect verbosity to uncertainty. Besides, we use the cascading model to mitigate verbosity while they use prompt engineering.

Uncertainty Quantification of LLMs With the thriving of Large Language Models (LLMs), researchers have begun exploring uncertainty quan-

tification in LLM responses (Geng et al., 2023). For white-box models, researcher have focused on unsupervised methods including entropy (Malinin and Gales, 2020), similarity (Fomicheva et al., 2020; Lin et al., 2022), semantic (Kuhn et al., 2023; Duan et al., 2023), and logits (Kadavath et al., 2022; Chen et al., 2024), whereas for black models, the uncertainty evaluation is based on multiple sampled output of the LLMs (Malinin and Gales, 2020; Lin et al., 2023; Manakul et al., 2023) However, these works aim to improve the evaluation metrics for LLM uncertainty while we focus on connecting uncertainty with verbosity compensation behavior.

Optimisation of LLM API Calls Recently, researchers have proposed to reduce the cost of leveraging a pool of LLMs (Wang et al., 2024) with a cascade algorithm. FragulGPT (Chen et al., 2023) use a cascade algorithm to visit LLMs from weak to strong and use an LLM evaluator to judge if the response is good enough to use (Madaan et al., 2023). (Ramírez et al., 2024) leverage the uncertainty of the prediction as the evaluator to evaluate both cascading and routing structures. Similarly, (Gupta et al., 2024) improve it by using token-level uncertainty. Our work, by contrast, aims at mitigating verbosity compensation which has not been explored before, and our evaluator is the verbosity of the response in the cascade algorithm.

3 Verbosity Compensation

In this section, we first introduce the definition and quantification of VC, and then we propose the metrics for evaluating the correlation between verbosity compensation and performance, uncertainty, and alleviating it with LLM routing.

3.1 Verbosity Compensation of LLMs

We first formalize the task. A dataset $\mathcal{D}$ consists of multiple data samples where each consists of a source text x, a query q, and a ground truth y. Since this is the first study, we mainly focus on the samples where y mainly contains short phrases for simplicity. A large language model $LLM(*)$ consumes the concatenation of x, q, and an instruction I to produce the response r. We use $|r|$ to represent the tokens in r. For instruction I, we always ask LLM to generate as concisely as possible so that the model is instructed not to generate verbose responses. Since the LLMs have maximum context window sizes L_c, we truncate the source to accommodate diverse context limits (details in A.3).

We define a response r to exhibit verbosity (we use the term verbosity as an alias for VC, and conciseness as an alias for Non-VC) if and only if it contains redundant tokens compared with the ground truth, since we assume the gold label to be concise. To detect VC, we define the verbosity compensation detector $V(x, y, r)$ (abbreviated as the verbosity detector). Using this detector, VC behavior for an LLM is defined as a triple (x, y, r) where $V(x, y, r) = 1$ describes that the VC occurs in the response r. To quantify the frequency of VC, we define it as the ratio of VC responses in each dataset $\sum_{(x,y)\in\mathcal{D}} V(x, y, r)/|\mathcal{D}|$.

3.2 Performance and Verbosity Compensation

A key bias of verbosity compensation is that the performance of the verbose responses is different from the concise ones. To quantify this behavior, we propose two evaluation metrics. One is performance difference (Δ), defined as the average score of the concise responses minus the average score of the verbose responses.

$$\Delta(\mathcal{D}) = \frac{\sum_{(x,y)\in\mathcal{D}}(1 - V(x, y, r)) \times \text{recall}(y, r)}{\sum_{(x,y)\in\mathcal{D}}(1 - V(x, y, r))} - \frac{\sum_{(x,y)\in\mathcal{D}} V(x, y, r) \times \text{recall}(y, r)}{\sum_{(x,y)\in\mathcal{D}} V(x, y, r)}$$

where r is the response generated by LLM and recall(y,r) is defined as $|r \cap y|/|y|$. This metric computes the difference between concise and verbose responses of a model over a dataset. If VC has no influence on the performance, the Δ should be 0. An LLM should show zero Δ because verbosity and performance are naturally independent and thus have no relation with each other. However, if Δ is positive, then it demonstrates that verbosity responses lead to the performance drop for this model on the dataset, and vice versa. To remove the influence of the length difference between verbose and concise responses, we use recall as the scoring function. Compared with precision or F1 scores, scores are higher for verbose responses (or Δ will be smaller) because verbose responses usually contain more tokens than concise ones.

A main problem of Δ is that the recall difference between verbose and concise responses is twisted by the absolute performance of the LLMs. According to the definition, a dataset with lower

Algorithm 1 Cascade Model Selection Algorithm.

Input: A list of LLMs M, A sample (x, y, q), instruction I_w, a verbosity detector $V()$.
Output: A response r.
order M by model capability from weak to strong
for LLM in M **do**
  $r \leftarrow \text{LLM}(x \bigoplus q \bigoplus I_w)$
  if $V(x, y, r)$ is false **then**
    break
  end if
end for
return r

performance naturally has a smaller space for performance difference. An extreme case is that the performance is zero on a dataset and the maximum Δ is zero as well. This impedes the fair comparison between datasets and models because they have diverse absolute performances. Thus, we propose relative performance difference

$$\delta(\mathcal{D}) = \Delta(\mathcal{D}) / \frac{\sum_{(x,y)\in\mathcal{D}} \text{recall}(y, r)}{\sum_{(x,y)\in\mathcal{D}} 1}$$

δ can be seen as the Δ if the absolute performance of the LLMs is scaled to the same number. We use this to compare the influence of performance across datasets and LLMs.

3.3 Verbosity Compensation and Uncertainty

For humans, verbosity compensation usually happens when we feel uncertain about the answers. Thus, for the LLMs, it is natural to speculate verbosity compensation of LLMs is also related to the uncertainty of the model. To test this hypothesis, we evaluate the uncertainty of the LLMs with the tool proposed by Fadeeva et al. (2023). First, we split the samples according to the length of the response $|r|$. Then, we quantify the uncertainty of each split. For open-sourced models, we use perplexity (Fomicheva et al., 2020) for evaluation, and for the close-sourced model, we use the sum of eigenvalues (Lin et al., 2023) of the graph laplacian as the metrics.

3.4 Alleviating Verbosity Compensation with Cascade Model Selection

Although it is difficult to ask a single LLM to generate a concise but correct answer, the verbosity compensation behavior can be mitigated by an ensemble of multiple models. To this end, we propose a **Cas**cade Model **Sel**ection algorithm (CaSel) to increase the chance of getting concise responses. The algorithm is simple and straightforward (Algorithm 1). Given a list of LLMs from weak to strong, we first ask the weak model to generate a response. At any time, if we detect $V(x, y, r) = 1$, we stop the generation of the current sample and redo the generation by a stronger model, and repeat the process. With the power of diverse LLMs, the algorithm can finally obtain a response with less verbosity and better performance.

4 Experiment Setup

4.1 Datasets and Metrics

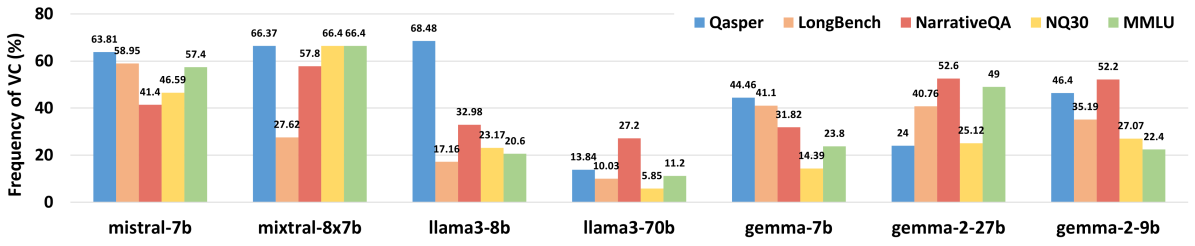
We include two types of datasets. 1) Knowledge-based question answering which aims at extracting knowledge from the given source text that is long or in a particular position. These datasets include **Qasper** (Shaham et al., 2022), **LongBench** (Bai et al., 2023), **NarrativeQA** (Shaham et al., 2022), and **NaturalQuestions_30 (NQ30)** (Liu et al., 2024). and reasoning-based question answering. More details for dataset construction can be found in Appendix A.1. 2) Reasoning-based Question Answering, including a modified **MMLU** (Hendrycks et al., 2021b,a) dataset. **Metrics.** We report recall when measuring verbosity compensation behavior and use F1 score for evaluation of the cascade model performance (Bai et al., 2023).

4.2 Models

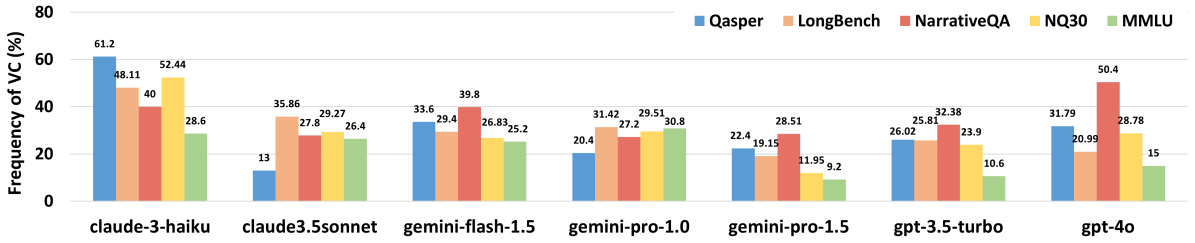
We use 14 LLMs in total across all experiments, including both open-source and closed-source models in 6 families: GPT, Claude, Gemini, Llama, Gemma, Mistral. Details are in Appendix A.2. For each model, in addition to the prompt that introduces the task, we also ask them to "generate as concisely as possible, use a single phrase if possible". **Verbosity Detector.** We assume that the gold answer y is concise and clear so that we can compare it with the predicted results to detect verbosity. Specifically, we use an LLM as $V(x, y, r)$. We prompt GPT-3.5-Turbo with definitions and demonstrations of verbosity, as well as the question, prediction, and ground truth. The model needs to generate a binary value showing whether the response is verbose. To evaluate the effectiveness of this detector, we manually annotate 100 samples and compare them with model predictions. 93% of the samples have the same label, demonstrating the effectiveness of the LLM-based detector.

5 Result and Analysis

In this section, we analyze verbosity compensation and its connection with performance and uncer-



(a) Open-source Models



(b) Closed-source Models

Figure 2: Frequency of Verbosity Compensation. All models exhibit intensive verbosity compensation behavior. Among them, llama3-70b has the lowest frequency on average (Details in Appendix B.1).

tainty. Then, we evaluate the cascade algorithm.

5.1 Verbosity Compensation

Frequency of Verbosity Compensation Behaviors. Figure 2 shows the frequency of each model on each dataset. As shown, all the models display verbosity compensation behavior on all datasets. On average, 74.19% of the responses are verbose for mistral-7b. The best model is llama3-70b which only contains 13.62% verbose responses. Furthermore, the frequency of VC averaged on seven open-source models is 39.80% which is significantly higher than closed-source models 28.96%.

Five Types of Verbosity Compensation Behaviors. After showing verbosity happens frequently in LLMs, we further conduct a human annotation to inspect verbose response patterns and classify them into five types. Specifically, we choose six combinations of model and dataset (Table 1) and pick out the samples with verbose responses that are not fully correct (recall $\neq 1$, $V(x, y, r) = 1$). By checking all these samples, we classify verbosity compensation behavior into five types (Table 1): Ambiguity indicates not answering precisely; repeating question indicates repeating the tokens in the question or providing unrelated information; enumerating shows answering multiple answers in a row trying to cover the correct answer; verbose detail/format means generating more detailed explanations or format symbols. Then, we annotate the verbosity compensation behaviors and obtain statistics in diverse settings. As shown in Figure 3, the ratio distribution of five types of behavior varies across different models and datasets. Furthermore, the main type of Gemini-1.5-flash is repeating questions on the MMLU dataset (67.86%), and enumerating on the Qasper dataset (47.62%). In contrast, llama-3-70b mainly produces verbose details on the Qasper dataset (32.87%). This shows that different datasets or models have a significantly different distribution of the main type of verbosity behavior.

5.2 Verbosity Compensation and Performance

Verbose and concise responses exhibit significantly different performance. As shown in Table 2 and Table 3, the performance difference ($\Delta \neq 0$) exists on most of the datasets and tasks, including both knowledge/reasoning-based tasks. This demonstrates that when the model performs verbosity compensation, the performance also changes significantly (Supplementary experiments in Appendix C.4, C.6). Among them, most of the datasets and models show lower performance on verbose samples (marked in red). For instance, llama3-70b shows 24.7% performance gap on Qasper dataset. However, *all models cannot disentangle performance with verbosity ($\Delta = 0$), highlighting the urgent need to disentangle verbosity with veracity.*

Correlation with Model Capability. We investigate the influence of model capability on the performance difference between verbose and concise

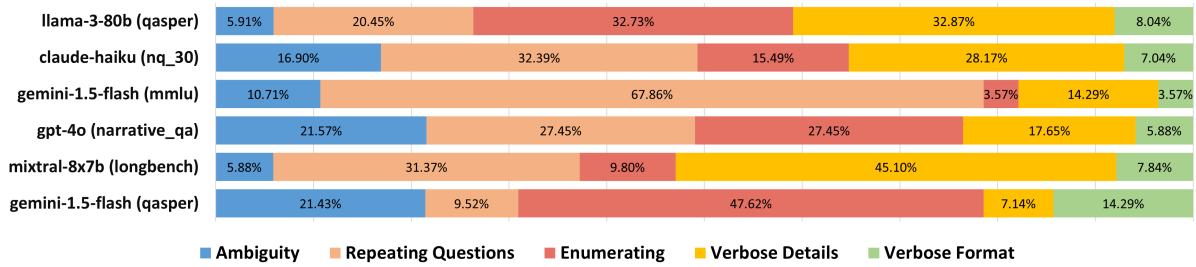


Figure 3: Human annotation of five types of verbosity compensation behavior on five datasets. Different models and datasets show diverse patterns of verbosity types.

Dataset	Question	Gold	Model Prediction	Type
Qasper	What is the size of the dataset?	3029	It is very large	Ambiguty
Longbench	Which genus has more species, Dracula or Pistacia?	Dracula	Pistacia has more species	Repeat
NarrativeQA	What costumes are the teenagers forced to wear?	Bunny costumes	Pig , donkey , rabbit	Enumberate
NQ30	who ran the fastest 40 yard dash in the nfl	Jakeem Grant	Chris Johnson 4.24 seconds	Detail
NarrativeQA	What types of activities occur in ...?	alleged phenomena	" Disappearances folklore "	Format

Table 1: Examples of five verbosity compensation types.

		Short (Qasper)				Medium (LongBench)				Long (NarrativeQA)			
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
gemma-7b	4k	45.24	46.76	-1.52	46.51	36.04	18.37	**+17.67**	30.74	15.39	6.70	+8.69	12.66
gemma-2-9b	8k	54.84	49.46	+5.38	52.73	44.86	43.51	+1.36	44.50	29.38	23.05	+6.33	26.81
gemma-2-27b	8k	54.51	48.26	+6.25	53.55	45.97	33.68	+12.30	43.41	32.17	30.86	+1.30	31.87
llama-3-8b	8k	54.36	53.51	+0.85	53.99	36.18	29.00	+7.18	34.64	29.25	19.51	+9.74	25.68
llama-3-70b	8k	52.86	28.74	**+24.12**	49.80	49.98	37.79	+12.19	48.76	34.30	25.91	+8.39	32.06
mistral-7b	8k	63.23	44.84	+18.39	56.42	54.03	37.04	+16.99	46.13	27.60	26.69	+0.91	27.21
mixtral-8x7b	8k	64.12	50.03	+14.10	56.78	2.62	6.24	-3.61	3.40	37.55	28.57	+8.98	33.09
gpt-3.5-turbo	16k	59.81	37.46	+22.34	54.77	53.88	47.02	+6.85	52.21	39.41	27.35	+12.06	35.49
gpt-4o	128k	63.80	44.07	+19.72	58.43	68.83	63.53	+5.30	67.53	59.14	47.12	+12.02	53.25
claude-3-haiku	200k	61.30	56.01	+5.29	58.54	53.02	57.88	-4.86	54.95	50.68	38.50	**+12.18**	46.13
claude-3.5-sonnet	200k	58.36	38.01	+20.35	56.12	59.42	57.36	+2.06	58.85	50.77	56.29	-5.52	52.16
gemini-flash-1.5	1m	62.52	41.64	+20.88	56.00	59.32	58.02	+1.30	59.00	2.51	1.12	+1.39	1.98
gemini-pro-1.0	32k	54.70	35.73	+18.98	51.44	47.85	44.68	+3.18	47.06	22.43	32.40	-9.96	24.89
gemini-pro-1.5	2m	59.40	45.79	+13.61	56.65	64.19	55.75	+8.44	62.97	36.26	41.74	-5.47	37.79
Avg		57.79	44.31	13.48	54.41	48.30	42.13	6.17	46.73	33.35	28.99	4.36	31.50

Table 2: Overall recall comparison between verbose and concise responses. **Bold**/Underline indicate the largest positive/negative performance gap between verbose and concise responses. The verbose responses obtain a significantly different performance than the concise ones, demonstrating the strong relationship between verbosity and performance.

responses δ. We explore two types of model capabilities. One is general capability. We leverage the scores on the leaderboard[1](ELO) as the measurement. The other one is the capability of consuming lengthy input. For this, we investigate the influence of the size of the window context. We define the log context window size as $log(L_c/1000)$ where L_c is the context window size.

Table 6 shows the correlation on five datasets. Each number in the table is computed based on the 14 data points of 14 LLMs on the corresponding dataset. As shown, for Qasper, LongBench, and NarrativeQA datasets, a strong negative correlation is observed. This indicates that when modeling capability increases, the δ decreases accordingly. In contrast, for MMLU and NQ30, no obvious correlation is observed. The results show that training a stronger model will help avoid the influence of VC on performance for long context questions and answering tasks. However, it does not help MMLU and NQ30. In other words, *simply training a stronger model or extending context window cannot successfully disentangle VC and performance.*

[1] https://lmarena.ai/

		Lost-in-the-Middle (NQ30)				MMLU (Mixed)				All
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.	Δ
gemma-7b	4k	43.32	37.83	+5.49	42.18	44.59	47.52	-2.93	45.30	5.48
gemma-2-9b	8k	55.82	45.18	+10.64	53.44	63.75	49.07	+14.68	61.67	7.68
gemma-2-27b	8k	54.84	47.81	+7.04	53.79	68.53	45.81	**+22.72**	66.98	9.92
llama-3-8b	8k	49.55	41.75	+7.80	47.92	54.65	47.57	+7.08	53.29	6.53
llama-3-70b	8k	52.08	50.33	+1.75	51.98	60.72	52.88	+7.85	59.92	10.86
mistral-7b	8k	52.89	44.39	+8.51	48.81	64.43	46.25	+18.18	54.55	12.59
mixtral-8x7b	8k	54.86	49.92	+4.94	52.84					6.10
gpt-3.5-turbo	16k	53.90	42.93	**+10.98**	51.43	72.33	50.44	+21.89	69.56	14.83
gpt-4o	128k	63.28	52.30	**+10.98**	60.16	81.00	67.72	+13.29	79.21	12.26
claude-3-haiku	200k	61.17	48.95	+12.22	54.94	61.95	64.49	-2.55	62.61	8.43
claude-3.5-sonnet	200k	57.22	57.72	-0.50	57.34	71.35	56.45	+14.90	67.97	4.46
gemini-1.5-flash	1m	54.69	47.70	+6.99	53.03	58.77	47.17	+11.60	56.60	6.26
gemini-1.0-pro	32k	51.55	45.75	+5.81	50.11	54.15	48.10	+6.06	52.58	4.81
gemini-1.5-pro	2m	57.06	46.29	+10.77	55.84	62.12	54.45	+7.66	61.73	7.00
Avg		55.21	47.52	7.69	52.99	63.61	52.72	10.90	61.37	8.57

Table 3: Overall recall comparison between verbose and concise responses. **Bold**/Underline indicate the largest positive/negative performance gap between verbose and concise responses. Similar to Table 2, the verbose responses obtain a significantly different performance than the concise ones.

Verbosity compensation behavior of Chain-of-Thought reasoning. We further conduct an experiment to demonstrate VC also happens in Chain-of-Thought (CoT) settings. To this end, we pick 100 samples from two datasets, including MMLU and Qasper, and instruct the models to generate a Chain-of-Thought prompt. Also, we ask the model to generate as concisely as possible, where each step contains fewer than 10 tokens. If any step violates this constraint, we regard this response as verbose. Thus, the verbosity evaluator V is set as $\mathbb{1}\left(\bigvee_{s\in r}|s|>10\right)$. Based on the definition, we do not restrict the number of steps of Chain-of-Thought reasoning; a short response can be verbose as well if the length of a single step is too long.

Table 4 shows the comparison between the concise and verbose responses of two models on two datasets (Length statistics of responses in Appendix C.7). All settings display significant Δ. For gpt-turbo-3.5, the recall gap can be as large as 24.54% on MMLU dataset. *This shows that verbosity compensation can also happen in generating longer responses (Appendix C.2), such as Chain-of-Thought reasoning samples.*

5.3 Uncertainty and Verbosity Compensation

Uncertainty Evaluation. The results are shown in Figure 4. As shown in the figure, all four models show larger uncertainty when the length of the responses increases. Especially, when the length is around three tokens, the uncertainty increases shapely. These results demonstrate that 1) when LLMs generate longer responses, they are more uncertain about the sample, and 2) *when verbosity compensation happens ($V(x,y,r)=1$), LLMs usually are more uncertain about the sample than generating concise results.*

Uncertainty and Length of Response r. We further explore the reason why uncertainty and VC are connected. We conduct a qualitative study and plot the distribution of the softmax score of the first tokens of confident and uncertain responses in Figure 1. As can be seen, for the uncertain response, the probability distribution is more flattened, and the tokens carrying much information do not stand out (ranked high) among the candidates. The model selects the one without critical information but is safer to generate, repeating the question or being off-topic and verbose. Besides, these tokens usually cannot end a sentence grammatically, such as "Avergae" or "+", the model needs to continue generations making the response longer.

5.4 Cascade Model Selection for Mitigating Verbosity Compensation

Reducing Frequency of Verbosity Compensation. Table 5 shows the comparison of using the proposed algorithm. As shown in the table, comparing the cascading algorithm and individual models, the frequency of VC decreases greatly for all settings. For instance, Mistral → GPT decreases the frequency from 63.81% (Mistral) and 31.79% (GPT) to 16.60%. It worth noting that, applying the algorithm greatly reduce the frequency of VC on

		Qasper				MMLU			
	L_c	concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
gemma-2-9b	8k	35.82	22.73	13.09	30.12	60.63	50.00	10.62	58.42
gpt-3.5-turbo	16k	69.05	47.81	21.24	61.06	80.95	56.41	24.54	68.32

Table 4: Recall difference of Chain-of-Thought generation. Both models perform worse when they generate verbose answers, demonstrating VC also happens on CoT settings.

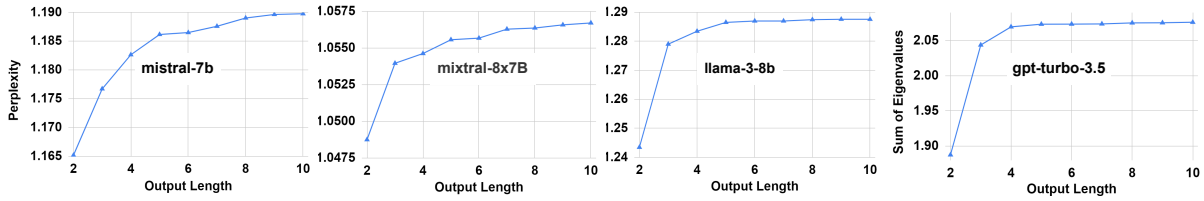


Figure 4: Uncentainty quantification of three open-sourced and one close-sourced models. The scores are averaged across all five datasets. The uncertainty increases with the increasing length of the generated output for all models.

	Qasper	LongB	NQA	NQ30	MMLU	Avg.
mistral-7b	63.81	58.95	41.40	46.59	57.40	74.19
gpt-4o	31.79	20.99	50.40	28.78	15.00	29.39
mistral → gpt	**16.60**	**14.48**	**21.00**	**18.54**	**10.20**	**16.16**
llama3-8b	68.48	17.16	32.98	23.17	20.60	32.48
claude-3.5-sonnet	13.00	35.86	27.80	29.27	26.40	26.47
lllama → claude	**8.20**	**11.80**	**14.60**	**11.71**	**7.80**	**10.82**
gemma-2-9b	46.40	35.19	52.20	27.07	22.40	36.65
gemini-pro-1.5	22.40	19.15	28.51	11.95	9.20	18.24
gemma → gemini	**15.80**	**11.14**	**18.20**	**8.29**	**4.60**	**11.61**

Table 5: Frequency of Verbosity Compensation using diverse cascade models. A → B indicates combining two models using a cascade algorithm. All settings greatly reduce the frequency of VC compared with both strong and weak models.

Dataset	ELO	Log Len
Qasper	0.09	-0.26
LongBench	-0.34	-0.53
NarrativeQA	-0.33	-0.61
MMLU	-0.05	0.13
NQ14	0.06	0.02

Table 6: Correlation between model capability and δ. Details in Appendix B.3.

both weak model and strong models. We also compare the latency of multiple LLMs in Appendix C.5.

Using Cascade Model Selection for LLM Routing. Inspired by the lower performance of the more verbose responses (Appendix B.4), we modify the CasSel to form a model routing algorithm (details in Appendix A.4). Figure 5 shows the performance of the proposed algorithm. As shown,

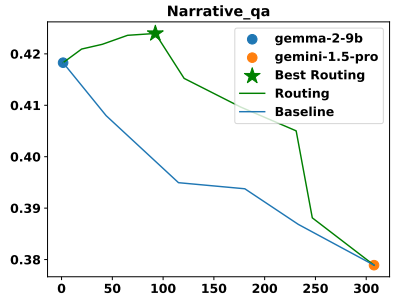


Figure 5: Routing performance of diverse models and datasets. X-axis (unit 10^{-3} dollars per sample) is the average cost. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the linear combination of the baseline models (blue line).

the performance of routing is better than the baseline (Appendix C.1). Furthermore, the routing results from Gemma-2 to Gemini-1.5 are better than the individual performance of both models. This indicates that *the routing algorithm improves the performance for all settings and can surpass the performance of stronger models with less cost.*

6 Conclusion

In this paper, we define VC and propose a comprehensive benchmark to evaluate 14 LLMs, revealing they suffer significantly from five types of VC. We conduct a rigorous analysis and connect VC to 1) model performance and 2) model uncertainty, shedding light on future applications and mitigation. We propose a simple but effective cascade approach to mitigate verbosity compensation in LLMs, and our extensive experiments show it is highly effective.

Ethics Statement

We include five datasets from the existing sources which we do not annotate or incorporate external resources. Thus, the dataset will not be harmful as long as the datasets themselves keep high quality. We also annotate some of the model-predicted results to classify the model results. However, the annotation is a classification task that is free of harmful content generation. Our work shows the negative part of verbosity responses, however, we do not mean verbosity is always unnecessary or harmful. Sometimes it might be helpful for the need of confirmation, or providing more context to the users.

Limitations

In this paper, we mainly show the negative effects of verbose responses on question-answering tasks. However, recent research has shown that the model can benefit from long reasoning chains (Guo et al., 2025). In this case, it is difficult to judge whether the long reasoning is verbose. Thus, future work can extend the proposed settings to diverse long-response scenarios and develop smarter verbosity detection. Another limitation is the mitigation algorithm requires multiple models to collaborate. In the future, researchers can propose to use a single model to mitigate VC, via fine-tuning or other techniques.

References

AI Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*.

Eleftheria Briakou, Zhongtao Liu, Colin Cherry, and Markus Freitag. 2024. On the implications of verbose llm outputs: A case study in translation evaluation. *arXiv preprint arXiv:2410.00863*.

Robert H Brookshire and Malcolm R McNeil. 2014. *Introduction to neurogenic communication disorders*. Elsevier Health Sciences.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. Inside: Llms' internal states retain the power of hallucination detection. *arXiv preprint arXiv:2402.03744*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*.

Cheng-Han Chiang and Hung-yi Lee. 2024. Over-reasoning and redundant calculation of large language models. *arXiv preprint arXiv:2401.11467*.

Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Ekaterina Fadeeva, Roman Vashurin, Akim Tsvigun, Artem Vazhentsev, Sergey Petrakov, Kirill Fedyanin, Daniil Vasilev, Elizaveta Goncharova, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2023. LM-polygraph: Uncertainty estimation for language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 446–461, Singapore. Association for Computational Linguistics.

Marina Fomicheva, Shuo Sun, Lisa Yankovskaya, Frédéric Blain, Francisco Guzmán, Mark Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Henry Watson Fowler. 1927. *A dictionary of modern English usage*. Clarendon Press.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A survey of language model confidence estimation and calibration. *arXiv preprint arXiv:2311.08298*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. 2024. Language model cascades: Token-level uncertainty and beyond. *arXiv preprint arXiv:2404.10136*.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. 2021a. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021b. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Kung-Hsiang Huang, Philippe Laban, Alexander Fabbri, Prafulla Kumar Choubey, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2024. Embrace divergence for richer insights: A multi-document summarization benchmark and a case study on summarizing diverse information from news articles. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 570–593, Mexico City, Mexico. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Maor Ivgi, Ori Yoran, Jonathan Berant, and Mor Geva. 2024. From loops to oops: Fallback behaviors of language models under uncertainty. *arXiv preprint arXiv:2407.06071*.

Patrick Juola. 2008. Assessing linguistic complexity. *Language Complexity: Typology, Contact, Change. John Benjamins Press, Amsterdam, Netherlands*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. 2022. Towards collaborative neural-symbolic graph semantic parsing via uncertainty. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 4160–4173, Dublin, Ireland. Association for Computational Linguistics.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Aman Madaan, Pranjal Aggarwal, Ankit Anand, Srividya Pranavi Potharaju, Swaroop Mishra, Pei Zhou, Aditya Gupta, Dheeraj Rajagopal, Karthik Kappaganthu, Yiming Yang, et al. 2023. Automix: Automatically mixing language models. *arXiv preprint arXiv:2310.12963*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Potsawee Manakul, Adian Liusie, and Mark JF Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. *arXiv preprint arXiv:2303.08896*.

Sania Nayab, Giulio Rossolini, Giorgio Buttazzo, Nicolamaria Manes, and Fabrizio Giacomelli. 2024. Concise thoughts: Impact of output length on llm reasoning and cost. *arXiv preprint arXiv:2407.19825*.

Daniel M Oppenheimer. 2006. Consequences of erudite vernacular utilized irrespective of necessity: Problems with using long words needlessly. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 20(2):139–156.

Guillem Ramírez, Alexandra Birch, and Ivan Titov. 2024. Optimising calls to large language models with uncertainty-based two-tier selection. *arXiv preprint arXiv:2405.02134*.

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Keita Saito, Akifumi Wachi, Koki Wataoka, and Youhei Akimoto. 2023. Verbosity bias in preference labeling by large language models. *arXiv preprint arXiv:2310.10076*.

Uri Shaham, Elad Segal, Maor Ivgi, Avia Efrat, Ori Yoran, Adi Haviv, Ankit Gupta, Wenhan Xiong, Mor Geva, Jonathan Berant, and Omer Levy. 2022.

SCROLLS: Standardized CompaRison over long language sequences. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 12007–12021, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.

Yan Tao, Olga Viberg, Ryan S Baker, and Rene F Kizilcec. 2023. Auditing and mitigating cultural bias in llms. *arXiv e-prints*, pages arXiv–2311.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. 2024a. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. MuSiQue: Multi-hop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

Can Wang, Bolin Zhang, Dianbo Sui, Zhiying Tum, Xiaoyu Liu, and Jiabao Kang. 2024. A survey on effective invocation methods of massive llm services. *arXiv preprint arXiv:2402.03408*.

Jiaxin Wen, Pei Ke, Hao Sun, Zhexin Zhang, Chengfei Li, Jinfeng Bai, and Minlie Huang. 2023. Unveiling the implicit toxicity in large language models. *arXiv preprint arXiv:2311.17391*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V Le, Ed H Chi, et al. 2024. Natural plan: Benchmarking llms on natural language planning. *arXiv preprint arXiv:2406.04520*.

A Implementation Details

A.1 Details of Dataset Construction

The principles of constructing datasets are twofold. First, the *quality* of samples needs to be high. The questions are picked from existing human-annotated datasets, with clear answers. We also filter out Yes/No, True/False, or multi-choice questions to ensure the answer cannot be simply chosen from a set of candidate answers. Second, the dataset should be *challenging* enough for LLMs with moderate performance levels. Otherwise, if the performance is close to 100 percent, the model is too certain about the answer and the phenomena is difficult to observe. Noting that most of the benchmark datasets LLMs already obtain performance higher than 90%,

Knowledge-based question answering. Firstly, we use long-context question-answering tasks whose difficulty resides in picking out useful information across long context and gathering them to answer the question. The distractor paragraphs will also incorporate the difficulty of recognizing the needed information. Specifically, we collect the three long-form question-answering datasets as our evaluation benchmark for long-context QA. These datasets display three levels of lengths, including short (**Qasper**), medium (**LongBench**), and long (**NarrativeQA**). Qasper (Dasigi et al., 2021) is a question-answering dataset over NLP papers. It also contains extractive, abstractive, yes/no, and unanswerable questions. The average length of the source text is 4119.85 words. We also incorporate three datasets from LongBench (Bai et al., 2023) to form a new dataset. We directly name it LongBench. It include HotpotQA (Yang et al., 2018), MuSiQue (Trivedi et al., 2022), and 2WikiMultihopQA (Ho et al., 2020). The average length of the source text is 9522.36 words. NarrativeQA (Kočiský et al., 2018) is a QA dataset over entire books or movie transcripts. The answers can be abstract or extractive, yes/no, and unanswerable, and the average length is 70340.45 words.

LLMs are proven to show difficulties in understanding the information in the middle of the context (Liu et al., 2024), known as lost-in-the-middle. We pick the most challenging split of the dataset in the original work, where the gold answer is in the middle of 30 documents for a QA pair in the Natural Question dataset. We call this **NaturalQuestions_30 (NQ30).** dataset. The average length of

input of NQ30 is 3602.13.

Reasoning-based question answering We modify the multi-choice answering samples in **MMLU** (Hendrycks et al., 2021b,a) so that the options work as hints to the question. In this way, the model needs to generate the answer based on the hint rather than picking out the correct option, increasing the difficulty because of the flexibility of open-ended question answers.

For each dataset, we sample 600 instances from them to form our datasets.

A.2 Details of Large Language Models

We include 2 models from Mistral AI[2], among them, **mistral-7b** is its first proposed dense model while **mixtral-8x7b** enhances the 7b model by incorporating a sparse mixture of experts. Gemini (Team et al., 2023; Reid et al., 2024) is a family of LLMs proposed by Google from which three versions of LLMs are selected, including **gemini-pro-1.0**, **gemini-flash-1.5**, and **gemini-flash-1.5**. Built from the research and technology used to create Gemini models, Gemma (Team et al., 2024a,b) is a family of lightweight, open models. We include **gemma-7b**, **gemma-2-9b**, and **gemma-2-27b** for experiments. LlaMA 3 (Dubey et al., 2024) is a family of LLMs with dense Transformer structure. We include **llama-3-8b** and **llama-3-70b** for experiments. Claude (Anthropic, 2024) is a family of large language models developed by Anthropic. We include two models in ascending order of capability: **claude-3-haiku**, **claude-3.5-sonnet**. We also include two versions of GPT models[3], including **gpt-3.5-turbo** and **gpt-4o** in experiments.

During experiments, we use the default parameters of all 14 LLMs. We run gemma, llama, and mistral models from Huggingface[4] on 8 A100 GPUs. For gpt, claude, and gemini models, we run with the official API from the official website. For all datasets, we use the same prompt shown in Table 7. We design a reinforced prompt to ensure LLM understands concise responses are required. Thus, we reinforce the prompt by repetition, and explanation, especially for the weaker models, making a fairer comparison by avoiding failing to understand instructions. We evaluate the robustness of VC against diverse prompts in Apendix C.3.

[2] https://docs.mistral.ai/getting-started/models/

[3] https://openai.com/

[4] https://huggingface.co/

A.3 Input Chunking Algorithm

Before we feed the input into the model, we first chunk the source so that the model can consume it. As shown in Algorithm 2, we first split the source into sentences and fed as many sentences as possible to LLMs.

Algorithm 2 Input Chunking Algorithm.

Input: Source input x, query q, LLM window size k, instruction I_w.
Output: A chunk c that LLM can consume.

```
Split the source x into sentences {s_1, s_2, ··· , s_n}
Initialize c ← empty string
Initialize length budgets B ← k − count_token(q) − count_token(I_w).
for s in s_1, s_2, ··· , s_n do
    if count_token(c) + count_token(s) > B then
        break
    end if
    c ← c ⊕ s   // ⊕ indicates concatenating two strings with a blank.
end for
return c
```

A.4 LLM Routing Algorithm

Model routing aims to send the sample to the proper model among a diverse collection of LLMs to generate the result so that under the same amount of API cost, the performance is better than other baselines, such as randomly choosing which model to use. We develop an LLM routing algorithm by modifying the proposed model selection algorithm. Different from model selection, we define two numbers p_c and p_v as the possibility of selecting a stronger model for concise and verbose responses. In this way, the cost is controllable to fulfill the diverse budget needs of users. It is worth noting that $V(x, y, r)$ is not available because y is not given in the routing setting. Thus, we propose a heuristic to approximate gold $V(x, y, r)$. We first sample 100 instances from the training set of the original dataset and compute the average length of the gold labels R. Then, we simply classify a response as verbose if it contains more than R tokens, represented as $V(x, y, r) = |r| > R$. Algorithm 3 shows the pseudo-code of LLM Routing. Different from the cascade algorithm for mitigating VC, this algorithm contains two probabilities that are used to control the budget of a single call. The algorithm mimics the real cost by counting tokens in

You are given an article and a question.
Answer the question as concisely as you can, using a single phrase if possible. Article:
{Source Documents}
Question:
{Question q}
Using a single phrase rather than a sentence.
Do not repeat any question-related information or explain the answer.
The answer is:

Table 7: Prompt of all models on all datasets.

the input and output, timing by the cost per token. We collect the cost of each model from website[5] and use it collected cost to ensure the fairness of comparison. The full name of all models and the price we use in LLM routing algorithm is shown in Table 8. We run each p_v, p_c setting ten times and compute the average to obtain the green lines and we run ten times that we randomly choose a weaker or stronger model with different probability to draw the blue line serving as the baseline. Specifically, for the stars in each figure, $p_v = 1$ and $p_c = 0$, degenerate to the proposed model selection algorithm.

Algorithm 3 Cascade Model Selection Algorithm for LLM Routing.

Input: A list of LLMs M, A sample (x, y, q), instruction I_w, a verbosity detector $V()$, possibility for routing on concise responses p_c, possibility for routing on verbose responses p_v.
Output: A response r.
order M by model capability from weak to strong
Set p_c to 1 if $p_v \neq 1$ {We ensure routing on verbose responses first.}
for LLM in M **do**
  $r \leftarrow \text{LLM}(x \bigoplus q \bigoplus I_w)$
  if $V(x, y, r)$ is false **then**
    $prob \leftarrow$ A random number from 0 to 1
    if $prob \geq p_c$ **then**
      break {Do not route for concise responses with $1 - p_c$ probability}
    end if
  else
    $prob \leftarrow$ A random number from 0 to 1
    if $prob \geq p_v$ **then**
      break {Do not route for verbose responses with $1 - p_v$ probability}
    end if
  end if
end for
return r

Figure 6 shows the performance of the different datasets with three routing settings: Mistral 7b → GPT-4o, Gemma2 9b → Gemini-1.5-pro, and LLaMA-3-8b → Claude-3.5-sonnet. As shown, the performance of routing is better than the baselines for all models, datasets, and settings. Furthermore, the routing results from Gemma-2 to Gemini-1.5 are better than the individual performance of both models.

[5] https://artificialanalysis.ai/models

B Details of Experimental Results

B.1 Frequency of Verbosity Compensation

Table 9 shows the detail numbers of frequency of verbosity compensation behavior.

B.2 Uncertainty Verses Length

Table 10 shows some examples of verbose and concise responses and the distribution of the first token.

B.3 Model Capability and Relative Delta

Figure 7 plots the Correlation between model window size and δ, visualizing the negative correlation score in Table 6. The models with the stronger capability to consume lengthy input obtain lower relative delta, indicating verbosity compensation is better avoided. Also, the decreasing speed of the tendency line ranks as follows: Long (NarrativeQA), Medium (LongBench), and Short (Qasper). This means that the effectiveness of the length capability on disentangling verbosity and performance is more significant when the task has a longer input.

B.4 Truncation Principle

We conducted an experiment on Qasper dataset with llama-3-8b and found that When the response is verbose, only keep the first 4 tokens, then stop the generation. The recall only drops from 44.93% to 43.13%. In other words, if the gold answer is not in the first 4 tokens, then the model is not likely to generate it in the rest of the tokens.

	Input Cost	Output Cost	Model Full Name
mistral-7b	0.17	0.2	mistralai/Mistral-7B-Instruct-v0.3
mixtral-8x7b	0.24	0.24	mistralai/Mixtral-8x7B-Instruct-v0.1
llama3-8b	0.05	0.08	meta-llama/Meta-Llama-3-8B-Instruct
llama3-70b	0.59	0.79	meta-llama/Meta-Llama-3-70B-Instruct
gemma-7b	0.07	0.07	google/gemma-7b-it
gemma-2-27b	0.8	0.8	googlegemma-2-27b-it
gemma-2-9b	0.2	0.2	google/gemma-2-9b-it
claude-3-haiku	0.25	1.25	claude-3-haiku-20240307
claude-3.5-sonnet	3	15	claude-3-5-sonnet-20240620
gemini-flash-1.5	0.35	1.05	gemini-1.5-flash
gemini-pro-1.0	0.5	1.5	gemini-1.0-pro
gemini-pro-1.5	3.5	10.5	gemini-1.5-pro
gpt-3.5-turbo	0.5	1.5	gpt-3.5-turbo-0125
gpt-4o	5	15	gpt-4o-2024-05-13

Table 8: The full name and the cost of tokens for each model. The unit of input/output cost is dollar per one million tokens.

	L	Qasper	LongB	NQA	NQ30	MMLU	Avg.
mistral-7b	8k	63.81	58.95	14.20	46.59	57.40	74.19
mixtral-8x7b	8k	66.37	**4.38**	57.80	66.40	66.40	52.27
llama3-8b	8k	68.48	17.16	32.98	23.17	20.60	32.48
llama3-70b	8k	13.84	10.03	27.20	**5.85**	11.20	**13.62**
gemma-7b	4k	44.46	41.10	31.82	14.39	23.80	31.11
gemma-2-27b	8k	24.00	40.76	52.60	25.12	49.00	38.30
gemma-2-9b	8k	46.40	35.19	52.20	27.07	22.40	36.65
claude-3-haiku	200k	61.20	48.11	40.00	52.44	28.60	46.07
claude-3.5-sonnet	200k	**13.00**	35.86	27.80	29.27	26.40	26.47
gemini-flash-1.5	1m	33.60	29.40	39.80	26.83	25.20	30.97
gemini-pro-1.0	32k	20.40	31.42	27.20	29.51	30.80	27.87
gemini-pro-1.5	2m	22.40	19.15	28.51	11.95	**9.20**	18.24
gpt-3.5-turbo	16k	26.02	25.81	32.38	23.90	10.60	23.74
gpt-4o	128k	31.79	20.99	50.40	28.78	15.00	29.39
Avg		34.53	31.71	44.11	31.98	31.14	34.69

Table 9: Frequency of Verbosity Compensation. All models have verbosity compensation behavior. Among them, llama3-70b has the lowest frequency on average.

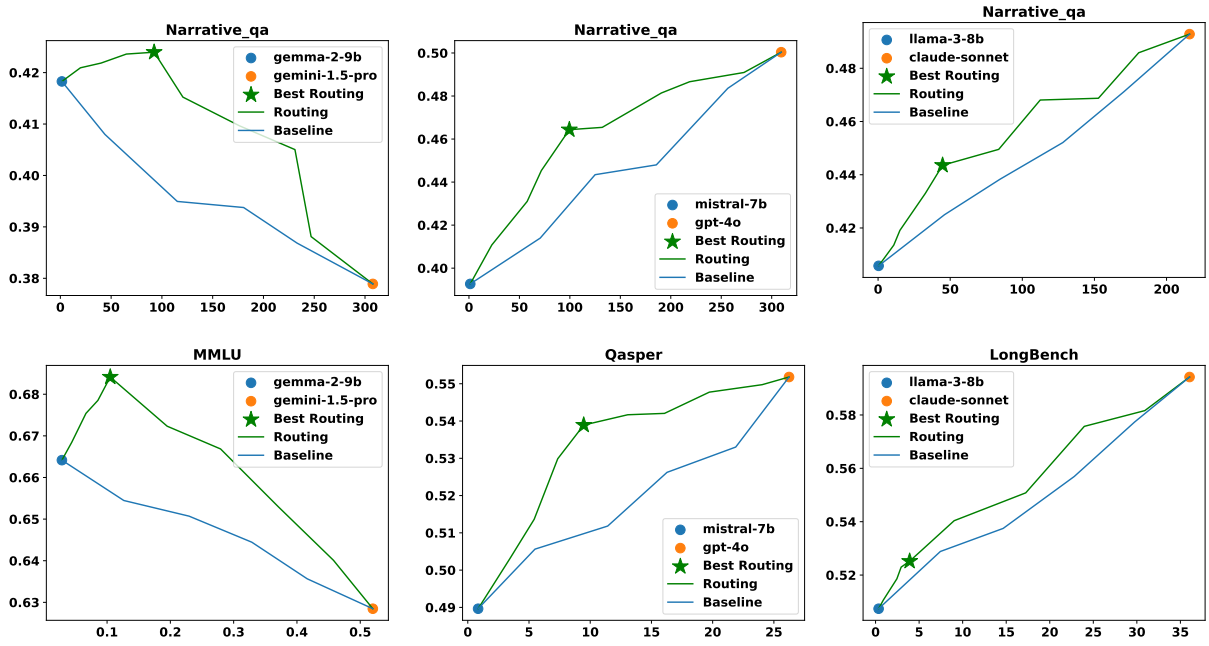


Figure 6: Routing performance of diverse models and datasets. X-axis (unit 10^{-3} dollars) is the average cost of running one sample. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the linear combination of the baseline models (blue line) with all datasets and models.

C Supplementary Experiments

C.1 Comparison with Uncertainty-based Routing Algorithm

We further conduct an analysis to compare the performance of the proposed routing algorithm with the uncertainty-based routing algorithm in addition to the random baselines. For the uncertainty-based routing algorithm, we first use perplexity as the metric to rank the uncertainty of the responses generated by a small model. We select top K% uncertain samples and replace them with the responses generated by the larger model. We select K from a set of $\{0, 10, 20, \cdots, 100\}$ and connect them to draw the curve in Figure 8. As can be seen, although the uncertainty-based routing algorithm can obtain a better performance than the random baseline, it is still worse than the proposed algorithm by comparing the AUC of the figure (Area Under the Curve), demonstrating the effectiveness of the proposed algorithm.

C.2 Verbosity Compensation in Trip Planning Dataset

To further demonstrate that VC generally occurs in diverse open-ended tasks with diverse response lengths, we run a trip planning dataset from the Natural-Plan benchmark (Zheng et al., 2024) using two Llama-3 models and test VC frequency and performance gaps. The task is to find the itinerary regarding the order of visiting N cities. We randomly select 500 data points from the dataset to form our dataset. For the prompt design, we follow the zero-shot prompt in the original paper and add one sentence "Answer as concisely as possible, each step contains less than 10 words". For the verbosity detector follows our CoT setting: $V(x, y, r) = \mathbb{1}\left(\bigvee_{s \in S} |s| > 10\right)$. The results are shown in Table 11. VC also occurs frequently in trip planning, demonstrating the general presence of VC in both short- and long-response open-ended tasks.

C.3 Robustness of Verbosity Compensation against Prompt Choices

As shown in Table 7 We design a reinforced prompt to ensure LLM understands concise responses are required. Thus, we reinforce the prompt by repetition, explanation, etc., especially for the weaker models, making a fairer comparison by avoiding failing to understand instructions.

We further experiment with multiple possible prompts to show VC is not overfitting to certain prompt settings. We aim to show that as long as the model knows to generate as concise as possible, we can observe significant VC behaviors.

Three Examples of Verbose Responses	
Question	What is the average length of the sentences?
Gold	15.5
Vanilla	Average sentence length: 16.5 words
Distribution	[['Average', 0.47828879952430725], ['"', 0.29009681940078735], ['15', 0.17595264315605164], ['9', 0.034647125750780106], ['average', 0.021014543250203133]]
Question	What were their accuracy results on the task?
Gold	97.32 %
Vanilla	+7 % relative gain
Distribution	[['+', 0.39178282022476196], ['"', 0.20970624685287476], ['7', 0.1850651204586029], ['96', 0.1850651204586029], ['97', 0.028380654752254486]]
Question	What's the size of the previous largest OpenIE dataset?
Gold	3,200 sentences
Vanilla	" 13 times larger "
Distribution	[['"', 0.5911211967468262], ['13', 0.3585331439971924], ['"L', 0.020227054134011 27], ['Three', 0.01785031333565712], ['3', 0.012268329039216042]]
Three Examples of Concise Responses	
Question	What is the sample size of people used to measure user satisfaction?
Gold	34,432
Vanilla	34,432 users
Distribution	[['34', 0.9797704219818115], ['"', 0.008476674556732178], ['Thirty', 0.006601640954613686], ['Total', 0.004004097543656826], ['Cor', 0.0011471931356936693]]
Question	Which news organisations are the headlines sourced from?
Gold	BBC and CNN
Vanilla	BBC and CNN
Distribution	[['BBC', 0.9247239232063293], ['"', 0.04062953218817711], ['"B', 0.027924243360757828], ['B', 0.003779135411605239], ['"C', 0.0029431935399770737]]
Question	which datasets did they experiment with?
Gold	Europarl MultiUN
Vanilla	Europarl MultiUN
Distribution	[['Eu', 0.98080664873123 17], ['Euro', 0.009615491144359112], [' Europ', 0.0074885510839521885], ['"', 0.0014745831722393632], ['European', 0.00061469769570976 5]]

Table 10: Distribution of the first token generated by llama-3-8b on Qasper dataset. When generating the verbose responses, the model is uncertain about the gold answer. Thus, starting with some commonly seen tokens. In contrast, the concise responses directly start with gold answers with high confidence.

	concise	verbose	Δ	Avg.	VC Freq.
llama-3-8b	15.18	3.62	11.56	9.22	**51.49**
llama-3-70b	21.81	4.87	**16.94**	19.63	12.87

Table 11: VC frequency and performance gaps on trip planning dataset.

Table 12 shows the performance gap on MMLU and Qasper datasets using Llama-3-8b with different prompt designs. As can be seen, compared with the original prompt, the variation of the prompt can also observe a significant Δ over both datasets. This demonstrates the robustness of VC against the choice of prompts. It is worth noting that, "Answer as concise as possible" yields the highest scores on two datasets, as well as the highest Δ, demonstrating a simpler prompt with less constraint might generate a larger performance gap between concise and verbose responses.

C.4 Evaluation of Verbosity and Performance on Same Test Instances

As shown in Table 2, and Table 3, the performance of concise and verbose samples is based on the split of the dataset. There is no overlap between the samples in the concise and verbose split. To prevent the influence of bias in different instances, we conduct an analysis that fixes the test instances and compares different models. Specifically, for each instance, we calculated the ratio of LLMs exhibiting VC behavior and reported the averaged ratio across datasets in Table 13. This approach

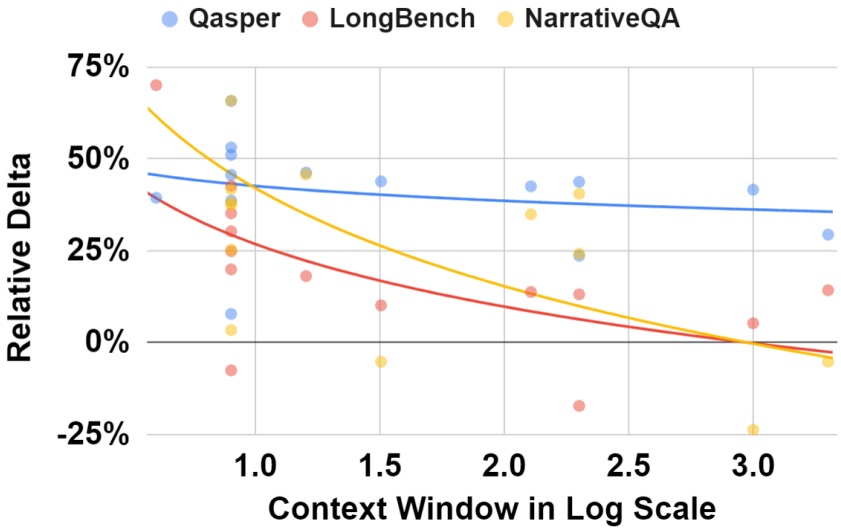


Figure 7: Correlation between model window size and δ. Results show that the model with a longer context window shows less δ on Qasper, LongBench, and NarrativeQA dataset.

MMLU				Qasper			
concise	verbose	Δ	Avg.	concise	verbose	Δ	Avg.
Prompt in Table 7							
58.4	44.82	13.57	55.6	58.99	54.6	4.39	55.98
Using a single phrase rather than a sentence.							
55.13	43.43	11.71	52.70	54.22	48.11	6.11	51.30
Answer as concise as possible.							
68.04	50.26	**17.78**	61.07	70.17	60.44	**9.73**	63.63

Table 12: Comparison between original and other variations of the prompts. VC consistently occurs, demonstrating the robustness of the VC against prompts.

	concise		verbose		overall		
	Recall	Support	Recall	Support	Δ	VC Freq.	Avg. Recall
Qasper	61.85	2272	45.63	389	**16.22**	32.46	56.59
LongBench	50.31	1912	44.22	375	6.10	30.42	48.46
NarrativeQA	38.09	2540	31.67	355	6.42	**36.29**	35.76
MMLU	65.09	1694	51.47	475	13.62	24.20	**61.79**
NQ30	53.34	1516	44.89	362	8.45	26.41	51.10

Table 13: Overall recall comparison between verbose and concise responses. Each dataset contains the prediction from all 14 LLMs.

also increases the robustness of our findings, as the support (number of samples) for each dataset is 14 times higher than when using a single model. As shown in the table, the performance δ is still pervasive for all five datasets. Specifically, on the Qasper dataset, the Δ reaches 16.22%

C.5 Latency Comparison of CaSel Algorithm and Individule Models

We conduct an analysis to compare the useless token generated and the time cost of individual models and the CaSel algorithm on two datasets using Mistral-7b and GPT-4o. To assess the number of useless tokens generated, given a re-

	Qasper					NarrativeQA				
	# Mistral	# GPT	# Total	VC Freq.	Infer. Time	# Mistral	# GPT	# Total	VC Freq.	Infer. Time
Mistral-7b	663	N/A	663	63.81	**0.80**	596	N/A	596	41.40	**1.22**
GPT-4o	N/A	207	207	31.79	1.27	N/A	327	327	50.40	14.86
Mistral → GPT	0	86	**86**	**16.60**	1.21	0	93	**93**	**21.00**	5.93

Table 14: Comparison of the number of generated useless tokens and inference time. # Mistral/GPT indicates the number of useless tokens generated by Mistral-7b and GPT-4o on the dataset. # Total is the sum of # Mistal/GPT, showing the total number of useless tokens. Infer. Time is the running time of the algorithm per sample (Unit: second). CaSel (Mistral → GPT) generated the fewest number of useless tokens and maintained the lowest VC frequency. The inference time is higher than the small model but still lower than the larger model.

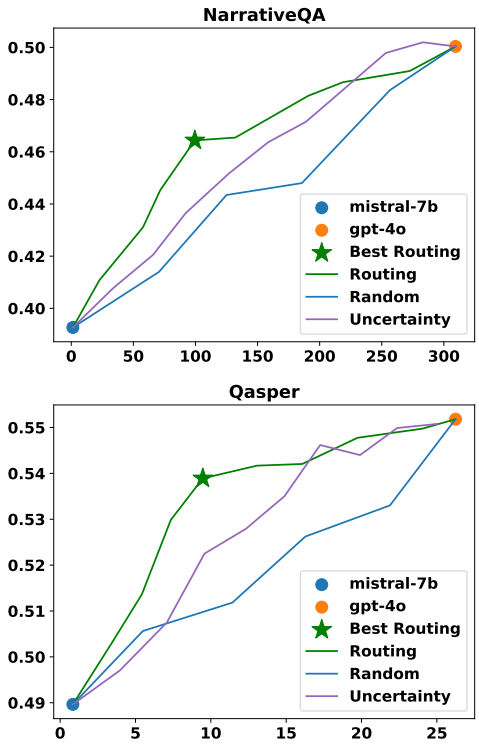


Figure 8: Routing performance of Mistral-7b to GPT-4o. X-axis (unit 10^{-3} dollars) is the average cost of running one sample. The Y-axis is the F-1 score averaged across the samples on one dataset. Routing performance (green line) is higher than the random baseline models (blue line) and uncertainty-based baseline (purple .

sponse r, we first define the useless tokens as the part with longer than gold answer in response r: $\sum_{i=1}^{N} \max(0, |r_i| - |y|)$, where N is the number of samples in a dataset. As shown in Table 14, with our proposed cascade algorithm, the total inference time might be higher than using a small model (0.79 vs. 1.21 seconds per sample) and lower than using a large model (14.86 vs. 5.93 seconds per sample), but the number of useless tokens generated is much less. On the other hand, by using the proposed algorithm, the useless tokens generated decrease from 596/327 to 93, mitigating the VC rate from 41.40% to 21.00% on the NarrativeQA dataset, demonstrating that useless tokens greatly decrease by using the proposed algorithm.

C.6 The Influence of the Digits in Responses

We analyze the performance and VC frequency of the samples with and without numbers using llama-3-8b on the Qasper and NarrativeQA dataset. The results are shown in Table 15. Although the model is easier to perform better on the sample without numbers, the VC frequency is relatively lower for the responses with digits. To understand the reason, we further inspect the Qasper dataset, we find that the samples with numbers are not as open-ended as the ones without numbers, meaning that the search space of the answers with numbers is smaller. This leads to a lower VC frequency and is easier to answer.

C.7 Response Length of Chain-of-Thought Experiments

Our evaluation is not limited to short gold answers. To demonstrate the generalization of the proposed VC behavior, we run the experiments on Chain-of-Though settings where the responses can contain more than 300 words. Table 16 shows the statistics of Chain-of-Thought experiments. The average response length can reach more than 50 words, and the VC behavior is still pervasive.

	Qasper				NarrativeQA			
	concise	verbose	Avg.	VC Freq.	concise	verbose	Avg.	VC Freq.
w/o digits	58.99	53.66	**56.18**	52.63	33.39	18.18	**27.21**	40.66
w/ digits	58.97	57.73	58.40	**45.83**	56.25	10.00	38.46	**38.46**

Table 15: Comparison between responses with digits and without digits. The responses with digits show lower verbosity compensation frequency.

	MMLU				Qasper			
	VC Freq.	Min Len.	Max Len.	Avg Len.	VC Freq.	Min Len.	Max Len.	Avg Len.
gpt-3.5-turbo	51.49	3	90	26.24	37.62	4	81	23.38
gemma-2-9b	20.79	9	107	27.92	43.56	18	103	37.08
llama-3-8b	43.56	15	333	57.14	44.15	20	185	50.15

Table 16: Lengths of the generated responses under chain-of-thought setting. The maximum length of the generated results can reach more than 300 words demonstrating that VC occurs in long response settings.

On the Role of Unobserved Sequences on Sample-based Uncertainty Quantification for LLMs

Lucie Kunitomo-Jacquin[1] **and Edison Marrese-Taylor**[1,2] **and Ken Fukuda**[1]
[1] National Institute of Advanced Industrial Science and Technology (AIST), Tokyo, Japan
[2] Graduate School of Engineering, The University of Tokyo, Tokyo, Japan
kunitomo-jacquin.lucie@aist.go.jp edison.marrese@aist.go.jp
emarrese@weblab.t.u-tokyo.ac.jp ken.fukuda@aist.go.jp

Abstract

Quantifying uncertainty in large language models (LLMs) is important for safety-critical applications because it helps spot incorrect answers, known as hallucinations. One major trend of uncertainty quantification methods is based on estimating the entropy of the distribution of the LLM's potential output sequences. This estimation is based on a set of output sequences and associated probabilities obtained by querying the LLM several times. In this paper, we advocate and experimentally show that the probability of unobserved sequences plays a crucial role, and we recommend future research to integrate it to enhance such LLM uncertainty quantification methods.

1 Introduction

The advent of large language models (LLMs) has revolutionized numerous fields by demonstrating remarkable capabilities across a diverse array of tasks. However, despite their impressive performance, these models often struggle with reliability issues, particularly due to factual inaccuracies in their outputs. In this context, quantifying their confidence and adjusting them for various tasks can reduce risks and enhance the quality of outputs.

However, uncertainty quantification (UQ) on LLMs remains challenging since the output possibilities for these models are substantially greater than those of discriminative models. As the generation length increases, the number of potential outcomes grows exponentially, making it unfeasible to evaluate all possible answers (Geng et al., 2024). We can distinguish two types of uncertainty in LLMs: aleatoric uncertainty, stemming from inherent randomness, and epistemic uncertainty, resulting from a lack of information (Kendall and Gal, 2017). Following previous work, we aim to quantify a measure of total uncertainty, i.e., aleatory and/or epistemic, as both types of uncertainty contribute to model errors.

Among the methods of uncertainty quantification for LLMs, we identify black-box methods, which assume access only to the generations, and white-box methods, which also utilize internal states of the LLM or token-level probabilities. In this paper, we focus on the latter, utilizing token-level probabilities. Concretely, we study sampling-based estimation methods, that is, approaches that rely on information (e.g. probability) obtained from multiple answers generated by the LLM, in order to quantify uncertainty.

Sample-based uncertainty quantification methods via entropy estimation, like Predictive Entropy (E) (Malinin and Gales, 2020) and the recently proposed Semantic entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024), have succeeded recently perhaps due to their simplicity, as they do not require any special training or architectural modifications. However, we note that these methods are themselves subject to epistemic uncertainty, as they rely on only a glimpse of the probability distribution of possible answers due to practical constraints. We highlight that methods like E and SE, in particular, do not account for this epistemic uncertainty, as they only consider the estimated probability of sampled sequences, neglecting the remaining probability of possible but unobserved answers.

Recent work by Abbasi Yadkori et al. (2024) has moved in a similar direction and explored the concept of missing mass in UQ. However, their approach directly compares the distributions of the generated answers against the ground truth. Instead, here we present work focusing on modeling the probability of unobserved answers without the need for ground truth. Concretely, our aim is to propose a framework that enables us to incorporate this probability into existing formulations for estimation based on entropy. We provide technical considerations for the calculation of such probability and evaluate the relevance of one such implementation by using it as a UQ method.

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 179–183
November 9, 2025 ©2025 Association for Computational Linguistics

2 Proposed Approach

Let us denote by x the object about which we quantify uncertainty; in our case study, x refers to the input given to the LLM which often consists on a question and potentially a prompt. We denote by $(\mathcal{S}, \mathbb{P})$ the probability space, where $\mathcal{S}$ is the set of all possible sequences, and $\mathbb{P}$ is the probability measure over $\mathcal{S}$. The entropy for the random output sequence of the LLM and the input x is defined as Equation 1 shows, below, where $p(s|x)$ is the probability of the sequence s conditioned on the input x.

$$E^*(x) = -\sum_{s \in \mathcal{S}} p(s|x) \log p(s|x). \quad (1)$$

As it is not realistic to compute the probability of all answers in $\mathcal{S}$, entropy-based UQ methods for LLMs estimate $E^*(x)$ on a set of M sequences sampled from the model denoted $s_1, \ldots, s_M$. Let us denote $A \subset \mathcal{S}$ the set of unique sampled answers and note that $|A| \leq M$ because some identical answers might be sampled multiple times. Each answer $s \in A$ consists of a sequence of length N in the set of vocabulary tokens $\mathcal{T}$. The probability of $s = (t_1, \ldots t_N)$ is obtained by the product of conditional token probabilities via the language model, as follows.

$$p(s|x) = \prod_i p(t_i|t_{<i}, x). \quad (2)$$

Some works have considered adjusting the calculation of sequence probabilities to account for varying sequence lengths. This is due to the tendency for longer sequences to exhibit lower joint likelihoods. To address this, a length normalized probability, which we denote p' was proposed (Malinin and Gales, 2020) as follows.

$$\log p'(s|x) = \frac{1}{N} \sum_i \log p(t_i|t_{<i}, x). \quad (3)$$

We now focus on the probability of sequences not observed in the set A of sequences provided by the LLM for a given input x. This probability is given by

$$\mathbb{P}(\bar{A}|x) = 1 - \mathbb{P}(A|x) \quad (4)$$

$$= 1 - \sum_{s \in A} p(s|x), \quad (5)$$

where $\bar{A}$ denotes the complement set of A.

We believe that the probability of unobserved sequences can capture some of the uncertainty associated with an input x. When uncertainty is low, the model's output probabilities tend to be higher, leading to a lower probability for the unobserved sequences. Conversely, when uncertainty is high, the model's output probabilities are lower, resulting in a higher probability for the unobserved sequences. In case of maximum uncertainty, all sequences in $\mathcal{S}$ are equally likely, with each having a probability of $1/|\mathcal{S}|$. As a result, $\mathbb{P}(\bar{A}|x) = 1 - M/|\mathcal{S}|$ approaches 1, especially when the set of possible sequences is very large. Conversely, in situations of minimal uncertainty, $\mathbb{P}(\bar{A}|x) = 0$.

In practice, we have two technical concerns related to the accurate calculation of probabilities for unobserved answers. Firstly, to the best of our knowledge, it is not always clear whether the last token, specifically the end-of-sequence (EOS) token, is considered in sequence probability calculations presented in Equation 2. If sequences do not include the EOS token, this raises concerns about the construction of the sample space, as two unfinished sequences are not mutually exclusive. Let us introduce a small example to illustrate our discussion about the sequence probability calculation.

Example. For the question input $x =$"*Where are St. Peter's Basilica and the Sistine Chapel?*", let us assume we observed two output sequences such that $A = \{$"*vatican*", "*vatican city*"$\}$ and consider the token conditional probabilities presented in Figure 1. If we do not include the end-of-sequence token, the probability value of 0.8 may be incorrectly interpreted as the probability of the sequence "*vatican*". In fact, this represents the probability that the sequence starts with "*vatican*", which also includes the possibility of the sequence being "*vatican city*". Essentially, the events of the sequence beginning with "*vatican*" and "*vatican city*" are not mutually exclusive.

In addition to this issue, we also note that sequence length normalization techniques as shown in Equation 3, and often used approaches like **SE**, can distort probabilities, potentially leading to the sum of output probabilities differing from 1.

Due to the issues discussed above, we highlight that we cannot properly estimate the probability of unobserved answers with the usually-adopted sequence probability calculations. Thus, we compute the probability of sequences without sequence length normalization and considering the EOS to-

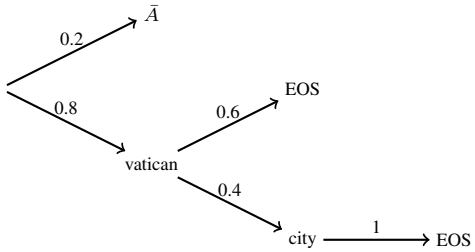


Figure 1: Example of tree of possible sequences with token conditional probabilities.

ken. Formally, we consider the probability of a sequence $s = (t_1, \ldots, t_N, \text{EOS})$ as

$$p(s|x) = \prod_i p(t_i|t_{<i}, x) \times p(\text{EOS}|t_{\leq N}, x). \quad (6)$$

Example (revisited). Looking back at our previous example, if we consider the EOS token in the computation of the probability, we obtain $p(\text{“}vatican\text{”}|x) = 0.8 \times 0.6 = 0.48$, $p(\text{“}vatican\ city\text{”}|x) = 0.8 \times 0.4 \times 1 = 0.32$ and the probability of the unobserved samples is $\mathbb{P}(\bar{A}|x) = 1 - 0.48 - 0.32 = 0.2$.

Based on this framework, here we present an alternative method for computing the uncertainty of an LLM where we directly use the value $\mathbb{P}(\bar{A}|x)$. We note that this approach, which we call *Unobserved Probability* (UP), is arguably a very simple way to capture some part of the LLM uncertainty, as derived from our analysis.

- EOS-Inclusive UP (EOS-UP): this approach consist of quantifying the LLM uncertainty using $\mathbb{P}(\bar{A}|x)$ in the way we consider most suitable or recommended, i.e., accounting for the EOS token in calculating the sequence probabilities as in Equation 6.

- Length-Normalized UP (LN-UP): we propose to quantify the LLM uncertainty using $\mathbb{P}(\bar{A}|x)$ as above, but considering the usual way for calculating the sequence probabilities, i.e., without accounting for EOS token and performing sequence length normalization, following Equation 3.

3 Experiments and Results

In this section, we detail our experimental setup to evaluate the relevance of using the probability of unobserved answers for LLM uncertainty quantification via our proposed approach UP. We compare its performance with three entropy-based methods and also include, for reference, the probability of unobserved answers calculated using the conventional method for sequence probabilities.

Model and dataset. Our experiments focused on the uncertainty quantification for the *falcon-40b-instruct* model (Almazrouei et al., 2023) and were performed on a general knowledge dataset, TriviaQA (Joshi et al., 2017). This model and dataset were recently used by Nikitin et al. (2024). TriviaQA was also originally used by Kuhn et al. (2023) for their seminal work on SE.

Sampling. We conducted our sampling using two styles of prompts. On the one hand, we adopt a prompt that pushes the model to produce short answers (SHORT), “*Answer the following question as briefly as possible*”. This prompt was used on a more recent implementation of SE, presented by Farquhar et al. (2024).[1] On the other hand, we also experiment with the original prompt (NORMAL) presented by Kuhn et al. (2023), and was also considered by Nikitin et al. (2024), “*Answer the following question in a single brief but complete sentence.*”. Following the methodology of previous studies (Farquhar et al., 2024; Nikitin et al., 2024), we employed top-K sampling with $K = 50$ and nucleus sampling with $p = 0.9$ at a temperature of $T = 1$.

Evaluation Metric. In line with previous works (Farquhar et al., 2024), we evaluated the model’s accuracy by sampling an additional answer at a lower temperature ($T = 0.1$). Then we used another LLM, *Meta-Llama-3-8B-Instruct* (AI@Meta, 2024), to compare this answer with the ground truth answers from the datasets. The prompts for checking answers correctness are provided in the appendix. We evaluate uncertainty quantification methods by measuring their ability in predicting model output accuracy using the Area under the Receiver Operating Curve (AUROC).

UQ methods. We considered the following baseline methods in our experiments.

- Predictive Entropy (E) (Malinin and Gales, 2020; Kuhn et al., 2023) is a Monte-Carlo estimation of predictive entropy, shown by Equation 7, below. As per the original implementation, this uses sentence length normal-

[1] https://github.com/jlko/semantic_uncertainty

ization as in Equation 3.

$$E(x) \approx -\frac{1}{M}\sum_{m=1}^{M}\log p'(s_m|x) \quad (7)$$

- Semantic Entropy (SE) (Kuhn et al., 2023; Farquhar et al., 2024) is defined on a set of clusters capturing the distinct meaning, denoted $\mathcal{C}$. This consists in a sub-σ-algebra of the event-space of all possible answers $\mathcal{S}$. The uncertainty quantification is calculated by an approximation of the semantic entropy involving the normalization of the cluster probabilities (Farquhar et al., 2024), as shown in Equation 8 and Equation 10, below, where $C \in \mathcal{C}$.

$$p'(C|x) = \sum_{s\in C} p'(s|x) \quad (8)$$

$$p''(C|x) = \frac{p'(C|x)}{\sum_{C\in\mathcal{C}} p'(C|x)} \quad (9)$$

$$SE(x) \approx -\sum_{c\in\mathcal{C}} p''(C|x)\log p''(C|x) \quad (10)$$

- Discrete Semantic Entropy (DSE) (Kuhn et al., 2023; Farquhar et al., 2024) consists in a variant of SE where cluster probabilities are approximated by $p(C|x) \approx |\{s : s\in C\}|/M$.

The results in terms of AUROC are presented in Figure 2. We observe that the probability of unobserved answers EOS-UP is indeed relevant for quantifying uncertainty, achieving performance comparable to the Predictive Entropy (E) method.

Moreover, we note that while state-of-the-art baselines (E, SE, and DSE) are affected by the number of available samples, the probability of unobserved answers maintains its performance even with a single sample. Sampling more answers from the LLM can generally lead to larger answer variability, and hence as M grows, the effect of the probability of unobserved answers on the estimation decreases. Therefore, our results suggest that incorporating the probability of unobserved samples in the estimation of uncertainty can be of critical importance when the number of samples is limited (e.g. $M = 1$). Note that when $M = 1$, $A = \{s_1\}$, E method reduces to $-\log p'(s_1|x)$, and LN-UP method to $1 - p'(s_1|x)$. Since these quantity are strictly decreasing and monotonic with respect to $p'(s_1|x)$, they yield the same ranking over input instances and thus the same AUROC performance, as shown in Figure 2.

Finally, we observe the poor performance of our proposed probability of unobserved answers, considering length-normalization and no EOS token

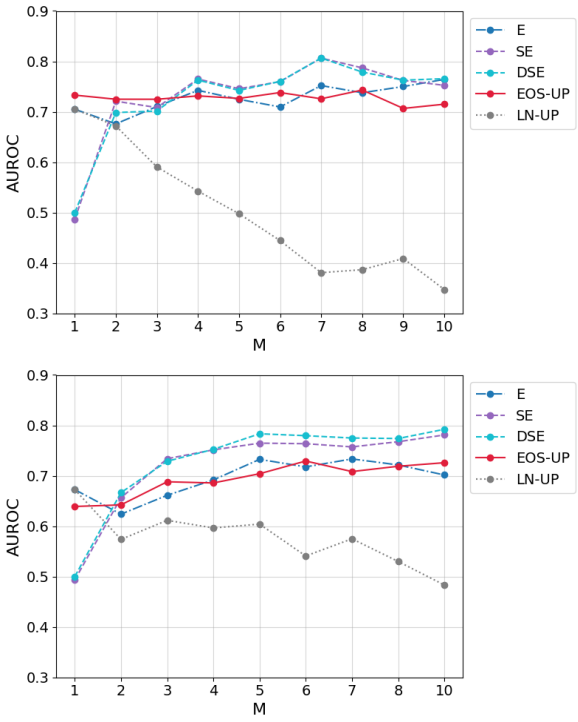


Figure 2: Influence of the number of samples (M) for the LLM uncertainty quantification in terms of AUROC, for the SHORT (top) and NORMAL (bottom) answer length scenarios. We compare the performance of our proposed approach variations (UP) against relevant baselines. Results were computed on 500 pairs of questions and ground truth answers on the *falcon-40b-instruct* model.

probability (LN-UP), not only remains the worst performing method for all M values, but also that its performance decreases dramatically as M grows. We think that, as shown by our technical considerations, our suggested way to compute this probability (EOS-UP) is necessary to obtain an adequate estimation.

4 Conclusion

In this work, we aimed to focus on the probability of unobserved answers, which we note have been overlooked by existing entropy-based LLM UQ methods. We acknowledge that this probability captures only a portion of the uncertainty. For instance, hesitation between observed answers is not considered since the probability of each separate observed answer is not used.

Our empirical results are encouraging and in the future we plan to integrate this quantity into existing entropy estimation methods. To achieve this, we believe a theoretical framework that considers both aleatoric and epistemic uncertainty, such

as the Evidence Theory (Shafer, 1976; Smets and Kennes, 1994) would be suitable.

We also note that current approaches of entropy-based UQ, present other issues and limitations. Although the work of (Nikitin et al., 2024) has made progress in this regard, we think further improvements are necessary, for example, by more directly modeling hypernymy and hyponymy relationships across answers, and/or clusters of answers.

Acknowledgement

This paper is based on results obtained from a project, JPNP25006, commissioned by the New Energy and Industrial Technology Development Organization (NEDO).

References

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. 2024. To believe or not to believe your llm: Iterative prompting for estimating epistemic uncertainty. *Advances in Neural Information Processing Systems*, 37:58077–58117.

AI@Meta. 2024. Llama 3 model card.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, and 1 others. 2023. The falcon series of open language models. *arXiv preprint arXiv:2311.16867*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Alex Kendall and Yarin Gal. 2017. What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

Glenn Shafer. 1976. *A mathematical theory of evidence*, volume 42. Princeton university press.

Philippe Smets and Robert Kennes. 1994. The transferable belief model. *Artificial intelligence*, 66(2):191–234.

Appendix

To check the correctness of the answers, we used the same prompts as previous studies presented in Figure 3.

Prompt (single answer)

*We are assessing the quality of answers to the following question: {**question**} \n The expected answer is: {**correct_answer**}. \n The proposed answer is: {**predicted_answer**} \n Within the context of the question, does the proposed answer mean the same as the expected answer? \n Respond only with yes or no.\n Response:*

Prompt (multiple answers)

*We are assessing the quality of answers to the following question: {**question**} \n The following are expected answers to this question: {**correct_answers**}. \n The proposed answer is: {predicted_answer} \n Within the context of the question, does the proposed answer mean the same as any of the expected answers? \n Respond only with yes or no.\n Response:*

Figure 3: Prompts fed to the model in our experiments when providing a single (top) and many correct answers (bottom), where **placeholders** are denoted in bold.

Confidence-Based Response Abstinence: Improving LLM Trustworthiness via Activation-Based Uncertainty Estimation

Zhiqi Huang[†], **Vivek Datla**[†‡], **Chenyang Zhu, Alfy Samuel,**
Daben Liu, **Anoop Kumar**, **Ritesh Soni**
{zhiqi.huang, vivek.datla, chenyang.zhu, alfy.samuel,
daben.liu, anoop.kumar, ritesh.soni}@capitalone.com
[†]Equal Contribution, [‡]Corresponding Author

Abstract

We propose a method for confidence estimation in retrieval-augmented generation (RAG) systems that aligns closely with the correctness of large language model (LLM) outputs. Confidence estimation is especially critical in high-stakes domains such as finance and healthcare, where the cost of an incorrect answer outweighs that of not answering the question. Our approach extends prior uncertainty quantification methods by leveraging raw feed-forward network (FFN) activations as auto-regressive signals, avoiding the information loss inherent in token logits and probabilities after projection and softmax normalization. We model confidence prediction as a sequence classification task, and regularize training with a Huber loss term to improve robustness against noisy supervision. Applied in a real-world financial industry customer-support setting with complex knowledge bases, our method outperforms strong baselines and maintains high accuracy under strict latency constraints. Experiments on Llama 3.1 8B model show that using activations from only the 16th layer preserves accuracy while reducing response latency. Our results demonstrate that activation-based confidence modeling offers a scalable, architecture-aware path toward trustworthy RAG deployment.

1 Introduction

In high-stakes applications like financial customer support, it is often more desirable and trustworthy for a Retrieval Augmented Generation (RAG) system to abstain from answering than to risk providing an incorrect response. Although not responding to a query reduces the system's immediate utility, it is a necessary trade-off to ensure accuracy and preserve user trust. The guiding principle is that the reputational and financial cost of providing a wrong answer is significantly higher than the cost of not providing one. This challenge requires a principle of abstention.

One way to achieve the abstention is to have a confidence measure that correlates with correctness of the response, and mask the response when the confidence score is below a threshold. Uncertainty of the model while generating the response is a viable source of signal for building a confidence measure.

To develop a practical solution, it is crucial to identify the primary source of this uncertainty. In highly regulated fields, the error is rarely due to aleatoric uncertainty (randomness inherent in the data), as knowledge bases are typically vetted by legal and subject-matter experts. The more probable source is epistemic uncertainty (the model's own lack of knowledge), which arises when the model's parametric knowledge, acquired during pre-training or fine-tuning, conflicts with or misinterprets the provided context.

While existing approaches (Bakman et al., 2024; Liu et al., 2024; Malinin and Gales, 2020; Kuhn et al., 2023) to uncertainty estimation in retrieval-augmented generation (RAG) have shown promise, they often fall short when the target response is long and narrative in nature. This challenge becomes especially pronounced in sensitive domains such as finance, where queries can be ambiguous or underspecified. For instance, a question like "What is the deadline to make a payment on Card Type A?" may retrieve multiple similar documents, each corresponding to different subcategories of the card type. In such cases, both the query and the retrieved context exhibit ambiguity, which can propagate through the RAG pipeline. Simply measuring uncertainty based on generated response is insufficient to ensure correctness.

Also, methods relying on sampling (Bakman et al., 2024), are less practical at scale. These techniques rely on generating a response multiple times with slight variations to measure the model's consistency, a process that introduces prohibitive computational costs and latency in a production en-

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 184–193
November 9, 2025 ©2025 Association for Computational Linguistics

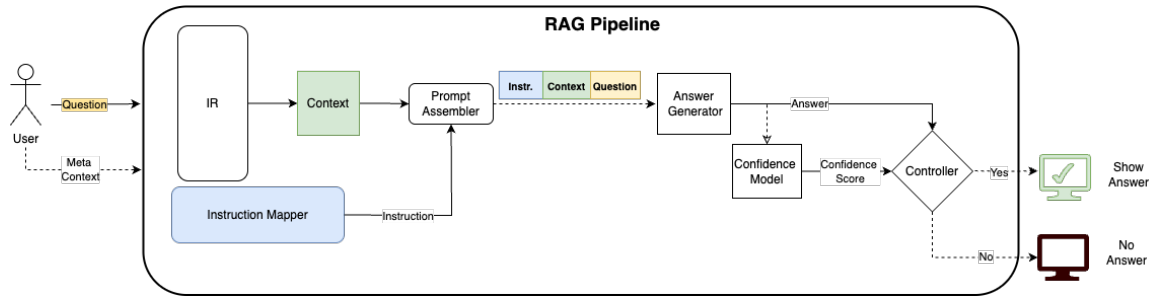


Figure 1: Diagram of the proposed Retrieval Augmented Generation (RAG) with the confidence model. When a user asked a question, the IR component retrieves related context from a database. The prompt is then constructed and sent into a question and answering LLM. A confidence score would be generated by the confidence model and being used to control whether or not to show the result to the user.

vironment. For RAG systems that must serve users in real-time, such multi-generational approaches are not a viable solution.

Uncertainty and correctness, while related, are fundamentally distinct concepts (Liu et al., 2025). A model's low uncertainty in its output does not necessarily imply correctness, just as a model may generate a correct response with a high uncertainty. This distinction becomes particularly salient in retrieval-augmented generation (RAG) applications, where correctness often hinges on factual grounding rather than surface-level fluency. Our goal is to utilize the model's internal uncertainty signals to generate a confidence score that correlates strongly with the correctness of the response generated by an LLM.

We build our confidence model using the raw activation signals inside the feedforward layers of LLM which include the activations of knowledge neurons (Azaria and Mitchell, 2023). Thus, our model captures the relationship between the autoregressive properties of activations and inherent uncertainty of the model in generating a response. We propose a supervised framework to train a sequence classifier model and generate a confidence score that correlates with response correctness.

Figure 1 illustrates the practical utility of integrating a confidence model into our RAG pipeline. The primary goal of the system is to provide users with accurate answers. However, in cases where there is insufficient epistemic or aleatoric knowledge to reliably answer a question, the system's next best action is to abstain from answering. This behavior is enabled by a controller that filters responses based on their confidence scores, allowing the system to avoid potentially incorrect or misleading outputs. This system is deployed in production for large-scale use that achieves high precision while maintaining an acceptable display rate (defined as the ratio of response pass the confidence filter to total responses generated by the system). Experimental results show that our confidence model outperforms multiple baselines, reaching a precision of 0.95 with 70.1% display rate (masking 29.9% of the total responses). Furthermore, when compared to ground truth, displayed responses exhibit a significantly higher ROUGE score than masked responses.

2 Related Work

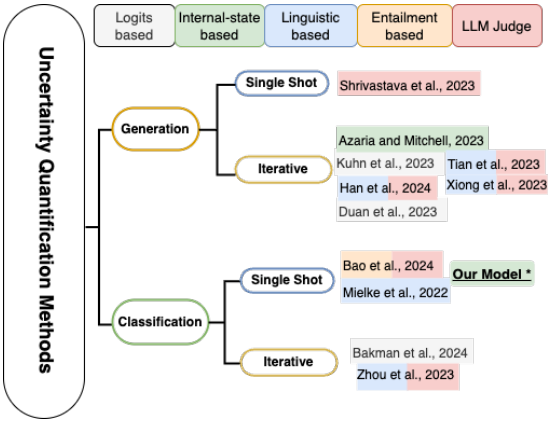


Figure 2: Landscape of Uncertainty Quantification Methods

Figure 2 shows the landscape of various uncertainty quantification methods in LLMs. When mapping the landscape, they can be broadly grouped by the strategies used to quantify the uncertainty.

Shrivastava et al. (2023) demonstrate that the generation probabilities of weaker white-box models (that is, smaller models) can be used to estimate the internal confidence levels of larger black-box models. The approach involves zero-shot generation using prompt variations based on dif-

ferent instructions to infer the confidence of responses produced by the larger model. Duan et al. (2023) and Kuhn et al. (2023) use semantic entropy to reweight token-level importance, prioritizing content-bearing tokens while discounting filler words. Their core intuition is that if semantically important tokens are generated with high confidence, the overall response is more likely to be correct even if less important tokens exhibit lower confidence.

Azaria and Mitchell (2023) show that the LLMs internal parameters show tell-tale signs when generating text with uncertainty. When a model's generation path falls into a speculative region, evidenced by competition between two or more plausible next tokens, its confidence is adversely affected. They introduce small input perturbations to induce trajectory shifts, and monitor corresponding changes in token-generation activations and outputs. They label the speculative generation as lying, and propose that activation patterns can shed light on this speculative generation. This method requires white-box access to the model to obtain token-level probability traces.

Tian et al. (2023) have empirically shown that LLM's self generated confidence score while giving a response could be calibrated by sampling over perturbed questions. Specifically, they show that prompting a model to produce several answer choices before giving its confidence scores helps in calibration of verbalized probabilities.

In a related direction, Xiong et al. (2023) generate multiple variants of a prompt using diverse prompting strategies such as Chain-of-Thought (CoT), self-probing, and top-k sampling. They utilize a separate LLM as a "judge model" to evaluate each variant and assign a confidence score. Variations in these scores are then used to predict the confidence of the target model's original response. Similarly, Han et al. (2024) proposed a confidence measurement based on the perturbation of the question. The variation in model's answer generation probabilities for various perturbations of the question for the same context is used as a measure to generate a verbalized confidence score.

Several recent studies adopt a classification-based approach to estimate response plausibility, offering a more computationally efficient alternative by avoiding multiple generations. For example, HHEM (Bao et al., 2024) uses an entailment-based model to measure the semantic coherence between the input and the generated output. This approach operates under black-box constraints, requiring only the input-output pair from the target LLM to assess the correctness of the response.

Other methods focus on linguistic cues as indicators of ambiguity in LLM outputs. Mielke et al. (2022) argue that model confidence does not always correlate with correctness and show that linguistic calibration of input prompts can significantly influence a model's confidence. They introduce a calibration score that helps generate more accurate responses by aligning linguistic features with expected confidence levels. Their evaluations were performed on factoid QA datasets, where there is a zero-sum approach towards correctness. We argue that when the parametric knowledge of the LLM is mainly contributing to the style of the response, and the key facts come from the input, confidence can serve as an effective signal for correctness.

Our method draws inspiration from prior work on activation-based knowledge tracing (Dai et al., 2022), generation trajectory modeling (Azaria and Mitchell, 2023), and importance-weighted token probabilities (Bakman et al., 2024). Dai et al. (2022) highlight how feedforward network (FFN) activations encode key factual information, showing that the activation of certain neurons is positively correlated with knowledge expression. Building on this insight, we treat FFN activations as autoregressive signals and train a recurrent neural network (RNN) to predict the probability that a model-generated response is correct. A score closer to 1 indicates greater model confidence in the response's correctness.

3 Method

For a generated response sequence s of length L for the given input x to a model M with parameters θ, the probability of generating the sequence is given as follows:

$$P(s \mid x; \theta) = \prod_{l=1}^{L} P(s_l \mid s_{<l}, x; \theta) \quad (1)$$

To compare sequence probability across different lengths of generated output, previous approaches have normalized the score based on the length of the response. The length-normalized score, used in prior uncertainty estimation (UE) methods (Malinin and Gales, 2020):

$$\tilde{P}(s \mid x; \theta) = \left(\prod_{l=1}^{L} P(s_l \mid s_{<l}, x; \theta) \right)^{1/L} \quad (2)$$

Here all the tokens contribute are given equal importance irrespective of the length of sequence. The risk with this approach is that a single low-probability unusual word can disproportionately lower the overall sequence score, even if subsequent tokens have high probabilities.

Several of the methods that perform uncertainty estimation taking token-logits perform similar weighing and they have shown great results in factoid question answering. These methods do not scale for longer answers, where there are multiple sentences and few tokens don't hold the key to correctness. Also, multiple generations needed to quantify the confidence score make them prohibitively expensive in a large scale settings.

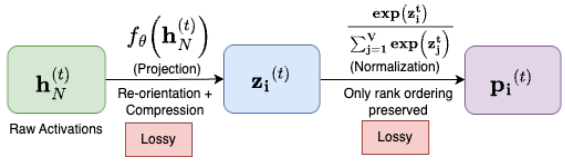


Figure 3: Motivation to use activations instead of token probabilities.

Our goal is to estimate the correctness of the generated response in a single shot using uncertainty estimation. We prefer using FF-layer activations rather than token probabilities because token probabilities are computed by applying the decoder head (a linear projection) followed by a softmax transformation. This projection compresses the rich internal representation into a vocabulary space and the softmax operation further distorts the signal by normalizing it into a probability distribution (see Figure 3), potentially obscuring fine-grained differences in the model's internal state. In contrast, raw activations preserve the high-dimensional representation prior to this compression, providing a more direct view of the model's internal dynamics during response generation.

3.1 Our Confidence Model

Figure 4 shows a graphical representation of our confidence model. To estimate the confidence of a generated answer s of size L, we introduce a lightweight, trainable probe that operates on the internal representations of the Llama 3.1 8B model. The process begins by providing a structured prompt to the LLM, which is formulated as a sequence of tokens, x of size $T + L + 1$, which is a concatenation of the following: Instruction(x_I), Question(x_Q) and Context(x_C) of size T tokens; Answer(s) of size L tokens; and EOS token(x_{EOS}) of size 1. The complete input sequence is formally represented as:

$$x = x_I \oplus x_Q \oplus x_C \oplus s \oplus x_{EOS} \tag{3}$$

where $\oplus$ denotes the concatenation operation.

During a single forward pass through the LLM, we extract the hidden state activations from a specific transformer layer, ℓ. We investigate representations from two distinct depths within the network: the final layer ($\ell = 32$) and a middle layer ($\ell = 16$). This yields a full sequence of hidden state vectors

$$\mathbf{H}_\ell = (\mathbf{h}_\ell^1, \ldots, \mathbf{h}_\ell^{T+L+1}) \tag{4}$$

Each vector $\mathbf{h}_{\ell,k} \in \mathbb{R}^{d_{\text{LLM}}}$ corresponds to the k-th input token, with size of LLM's activation dimension. From this complete set of activations, we isolate only those corresponding to the tokens of the candidate answer, which span from index $T+1$ towards the final x_{EOS} token. This forms the input sequence, S_{in}, for our confidence estimation module:

$$S_{\text{in}} = (\mathbf{h}_\ell^{T+1}, \mathbf{h}_\ell^{T+2}, \ldots, \mathbf{h}_\ell^{T+L+1}) \tag{5}$$

The extracted sequence S_{in} is then fed into a sequence classifier $g(S_{in})$, which is trained to model the sequence of activations. The sequence classifier with a classification head outputs a 2-dimensional logit vector, $\mathbf{z}$, such that the confidence score can be computed as,

$$c = \text{softmax}(\mathbf{z})_1 = \frac{e^{z_1}}{e^{z_0} + e^{z_1}} \tag{6}$$

Our goal is to estimate the confidence of the model when generating an answer, with ulterior goal of rejecting the generated answer if c falls below a threshold of confidence. In this framework, only the parameters of the sequence classifier $g(S_{in})$ are trainable. We use a Long short-term memory (LSTM) (Sutskever et al., 2014) as the sequence classifier for the following experiments.

3.2 Model Training

Given that the retrieval stage of the pipeline may introduce alethic knowledge gaps, the input context provided to the LLM can be incomplete, or contain contradictory information across the document retrieved. To address this, we introduce an explicit regularizer based on Huber loss L_{Huber}, which is more robust to such noise (Patra et al., 2023). Unlike just using only the Cross-Entropy loss L_{CE}, which can be highly sensitive to large deviations

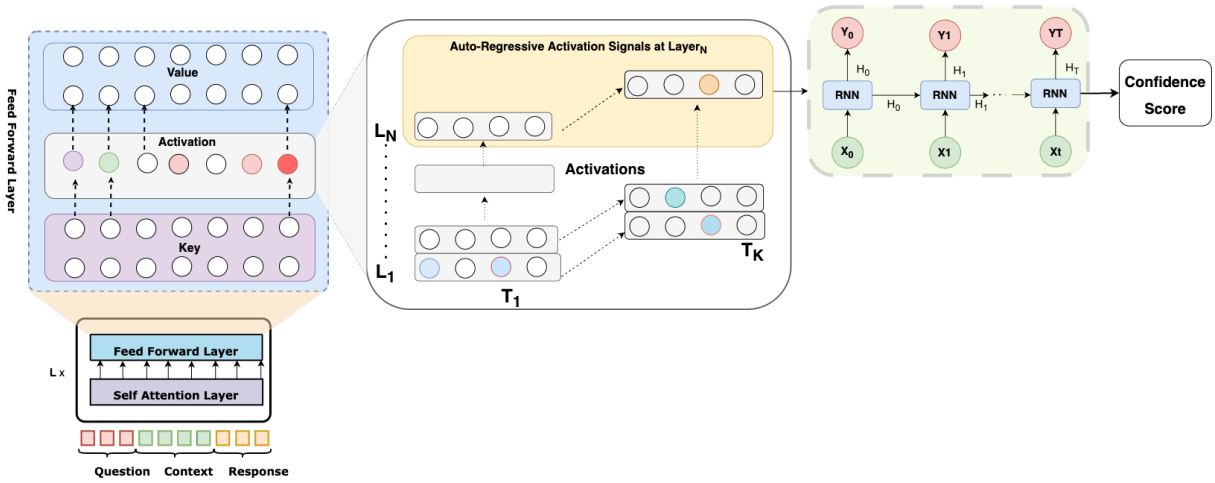


Figure 4: Confidence model based on the activations of large language models. Our method first feed the <Question, Context, Response> pair in an LLM. We then extract the activations from the 32-th or 16-th layer, and feed the activations into an LSTM and a classification head. The classification logit serves as the confidence score.

when predictions are far from the target, the Huber loss based regularizer helps smoothen with a linear penalty for large errors. This property reduces the influence by outliers arising from imperfect retrieval.

$$H_\delta(x) = \begin{cases} \frac{1}{2}x^2 & \text{for } |x| \le \delta \\ \delta\left(|x| - \frac{1}{2}\delta\right) & \text{otherwise} \end{cases} \quad (7)$$

where $\delta > 0$ is a hyperparameter that controls the transition point between the quadratic and linear loss.

Using L_{CE} loss with L_{Huber} regularizer, we learn to predict confidence score, which correlates with the correctness. The higher the confidence, the higher are the changes for the generated output to be correct. For a sampled minibatch $B = \{(x_j, y_j)\}_{j=1}^{|B|}$, the Huber loss term is calculated as:

$$L_{\text{Huber}} = H_\delta\left(\frac{1}{|B|}\sum_{i=1}^{|B|} c_i - \frac{1}{|B|}\sum_{i=1}^{|B|} I(\hat{y}_i = y_i)\right) \quad (8)$$

where $c_i = \max(\hat{y}_i)$ is the confidence of the prediction for instance x_i, and $I(\hat{y}_i = y_i)$ is the indicator function for correct predictions.

The total loss function

$$L_{Total} = L_{CE} + \lambda L_{Huber} \quad (9)$$

where λ controls the strength of regularization.

In our modeling, several constraints arise naturally from the real-time conditions under which the system operates. The generated output must remain grounded in the input context provided within the prompt. The output must adhere to predefined stylistic or structural patterns required to present certain types of information. At the end of generation, an explicit decision signal determines whether the answer is shown to the user. This signal is conditioned on multiple factors, including:

- Subject-matter-expert (SME) defined standards of correctness for the class of questions.
- The requirement that factual content be derived from the input context, while stylistic elements may rely on the model's parametric knowledge.

We conducted experiments on our proprietary knowledge corpus consisting of procedures, rules, and complex instructions to be followed to address the various needs of support agents handling a large volume of customer base. Our results indicate a robust performance using our method compared to the several SOTA UQ and hallucination identification methods.

4 Experimentation

We have conducted experiments to identify the optimal masking ratio in order to maintain utility and precision of the system.

4.1 Data

4.1.1 Disclosure on data

Due to the sensitive nature of the data, which pertains to proprietary financial tools and internal knowledge resources used by service agents

within a financial institution, we are unable to share dataset details. This restriction is in place to ensure compliance with internal data governance policies and to protect confidential and regulated financial information. We hope that the community understands the importance of maintaining the integrity and privacy of such sensitive operational data.

4.1.2 Features of our knowledge articles

We provide an overview of the population-level characteristics of our dataset, which is derived from a knowledge base composed of instructional articles designed to guide customer support agents in using proprietary internal tools. These tools are governed by strict procedural guidelines essential for resolving customer issues. For instance, when handling a customer inquiry about a specific transaction, agents must follow a prescribed sequence: verifying the customer's identity, obtaining consent to access the account, identifying the relevant transaction, and initiating additional processes such as flagging the transaction in cases of suspected fraud.

Figure 5(a) shows the hierarchical nature of the documents. Our knowledge base is semi-structured comprising heterogeneous documents with rich hierarchical and content structures. These documents may include deeply nested sections (e.g., sections, subsections, sub-subsections), as well as complex content types such as tables, bullet and numbered lists, and embedded entities.

Each subsection article is treated as a separate document. Each document is further chunked to be efficiently indexed in a low latency store. Overall, there are $8.5k$ documents and approximately $45k$ chunks in the knowledge-base.

4.1.3 Features of the training data

Our system design incorporates a real-time feedback loop, as illustrated in Figure 5(b), where support agents interact with the RAG system and provide immediate feedback (e.g., thumbs-up/down) on the usefulness of generated responses. Processing thousands of these interactions daily, we draw a stratified sample of both positive and negative feedback instances, accounting for dimensions like product type and line of business. For each sampled case, we collect the query, generated answer, retrieved context, and associated metadata for a more rigorous offline evaluation.

This offline review is conducted by subject matter experts (SMEs) who assess each answer for completeness, correctness, and truthfulness, ensuring it is grounded in the provided context rather than inferred from the model's parametric knowledge. SMEs may also refine responses to create ideal, complete answers, as shown in the example in Figure 5(c). This two-tiered approach of combining real-time user signals with deep SME validation allows us to build a high-quality labeled dataset for training and evaluation, ensuring the model aligns with domain-specific requirements for accuracy and trustworthiness.

4.2 Information Retrieval

We perform retrieval using an open-search index configured for K-nearest neighbor (KNN) retrieval based on semantic similarity to the input query. In addition to the query itself, we incorporate associated metadata such as entitlements and access-control filters specific to the agent submitting the question, to ensure that the retrieved documents adhere to the agent's permissions.

In the context of this work, we do not explicitly quantify retrieval errors. Instead, our focus lies in modeling the generation process of the response. We assume the retrieval step to be correct and treat errors introduced during retrieval as alethic uncertainty, while the knowledge encoded within the model through pretraining and fine-tuning is considered epistemic. Our confidence model is designed to map the relationship between the question, the retrieved (alethic) knowledge, the model's internal (epistemic) knowledge, and the generated response. This relationship is captured through patterns in the model's internal activations, treated as auto-regressive signals.

We observe that this mapping cannot be adequately modeled using a simple feedforward (MLP) architecture, as it fails to capture the temporal dependencies inherent in the generation process. Therefore, we adopt a recurrent architecture specifically, a lightweight Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) (LSTM), trained using L_{CE} loss and L_{Huber} regularizer loss. The LSTM is trained on input sequences derived from the activations of a selected layer, along with carefully curated training data that aligns the activation patterns with response-level confidence.

4.3 Results

Our method achieves superior calibration of LLM responses, maintaining high precision with minimal utility loss. As shown in Table 1, it outperforms industry SOTA methods, Vectara

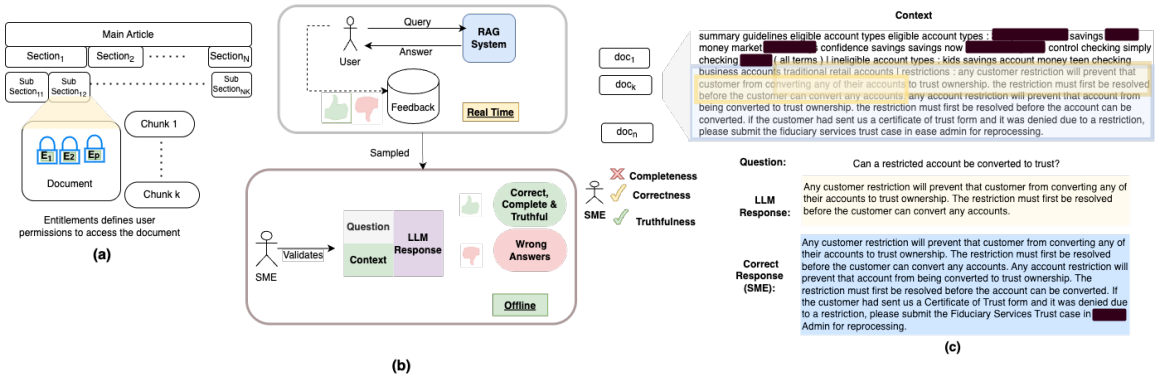


Figure 5: Features of our Knowledge base. (a) Complex structure of our knowledge articles; (b) Process of SME validated training data creation; (c) Example SME validated LLM Response

Method	AUROC
Vectara	0.590
$\text{Vectara}_{\text{FT}}$	0.634
$\text{Logits}_{\text{based}}$	0.663
$\text{Our Model}_{\text{no calib.}}$	0.741
$\text{Our Model}_{\text{with calib.}}$	**0.772**

Table 1: Comparing our approach to other baselines

Threshold	P	R	ROUGE-L Display	ROUGE-L Mask	%Mask
Baseline (0.0)	0.90	1.00	0.62	N/A	0.0
0.1	0.94	0.89	0.64	0.54	14.4
0.2	0.94	0.83	0.64	0.56	20.4
0.3	0.94	0.80	0.64	0.57	22.9
0.4	0.94	0.76	0.64	0.57	26.4
0.5	0.95	0.73	0.65	0.57	29.9
0.6	0.95	0.69	0.66	0.56	34.8
0.7	0.96	0.65	0.66	0.57	38.6
0.8	0.96	0.60	0.66	0.58	44.0
0.9	0.97	0.52	0.67	0.58	52.0

Table 2: Our Confidence score model with calibration helps achieve 0.95 precision while masking 29.9% of the responses

IR Model	R@1	R@3	R@5	R@10	R@25
	0.54	0.75	0.80	0.84	0.88

Table 3: Current recall(r) of the IR system, that helps in creating the context for the RAG pipeline

Layer	Context	P	R	ROUGE-L Display	ROUGE-L Mask	%Mask
32	Full	0.95	0.73	0.65	0.57	29.9
32	Top 5	0.95	0.69	0.66	0.56	34.3
32	Top 3	0.96	0.63	0.66	0.57	40.5
32	Top 1	0.97	0.56	0.67	0.57	47.5
16	Full	0.97	0.73	0.64	0.58	31.3
16	Top 5	0.98	0.65	0.65	0.59	39.3
16	Top 3	0.98	0.60	0.66	0.58	44.8
16	Top 1	0.99	0.48	0.66	0.59	56.2

Table 4: Identifying the optimal setting to run confidence model

(HHEM2.1) (Bao et al., 2024) and a logits-based uncertainty model (Malinin and Gales, 2020). We obtain further performance gains by caliberating with L_{Huber} as a regularizer.

Table 2 reports confidence thresholds that optimize precision while keeping the masking rate low. Although an ideal mask rate is 0%, realistic applications must tolerate some masking due to noise in LLM inputs. In our setup, the retrieval stage achieves a strong $recall@10 > 0.8$ (Table 3), yet residual alethic knowledge gaps in retrieval can affect downstream generation.

We experimented with varying input context sizes, selecting the top k documents ($k \in \{1, 3, 5, 7 \text{ (full)}\}$), and with partial-layer activation extraction from Llama 3.1 8B (layer 16 or layer 32) (AI@Meta, 2024). As shown in Table 4, using activations from only the 16th layer yields performance on par with the full-layer setup while maintaining a reasonable mask rate.

Latency analysis (Table 5) confirms that input context size is a dominant factor; larger contexts increase response time, highlighting a trade-off between context size and system responsiveness. In the production system, the confidence model is de-

Framework	Layer	Context	Avg. ms	P99
Hugging Face	32	Full	221	387
		Top 5	179	329
		Top 3	137	286
		Top 1	100	252
	16	Full	139	278
vLLM	32	Full	206	354
		Top 5	161	304
		Top 3	125	269
		Top 1	88	241
	16	Full	127	267

Table 5: Latency of the confidence model using various context sizes, Avg. time is calculated across 3 runs of the same input.

ployed with vLLM (Kwon et al., 2023), and overall the same trend appears there as well.

5 Discussion

In this work, we present an approach for constructing a confidence score that aligns with the correctness of responses generated by large language models (LLMs). Such a measure is particularly critical in high-stakes domains such as finance and healthcare, where the cost of an incorrect response far exceeds that of withholding a response. Our method extends prior works in uncertainty quantification (UQ) (Malinin and Gales, 2020; Bao et al., 2024) by leveraging model activation patterns to predict correctness more robustly.

Figure 3 illustrates our motivation for using raw activation signals from the feed-forward network (FFN) layers as auto-regressive features, rather than token logits or probabilities. Token probabilities are obtained after a linear projection and softmax transformation. The projection step reduces dimensionality, discarding non-vocabulary-aligned features, while the softmax normalization saturates probability values, erasing scale information and compressing relative differences. Using activations directly, we retain the full representational capacity of the internal state of the model.

Our application setting involves customer support agents consulting a proprietary knowledge base to resolve customer queries using specialized internal tools. The knowledge base contains documents vetted across multiple dimensions, including risk and legal compliance, making factual errors in the content highly unlikely. However, strict permissions govern which documents an agent can access. Figure 5(a) shows the complexity of document formats and fine-grained entitlements that impact retrieval and downstream generation.

We model confidence estimation as a classification problem over sequences of activations. Specifically, we employ a lightweight recurrent neural network (LSTM) that consumes FFN activations as auto-regressive signals. The classification logit from the LSTM head serves as the confidence score (see Figure 4). To enhance robustness against noisy supervision, we introduce a Huber loss regularizer L_{Huber} alongside the cross-entropy loss L_{CE}. The Huber loss's ability to behave quadratically for small errors and linearly for large errors makes it well-suited for smoothing gradients and mitigating the influence of outliers (Patra et al., 2023). Results in Table 1 demonstrate that our approach outperforms strong baselines, and the inclusion of L_{Huber} further improves accuracy over using L_{CE} alone.

In real-world deployment, retrieval-augmented generation (RAG) pipelines must meet strict latency requirements, as the LLM prompt length is constrained by model context limits and thousands of queries are processed daily. Tables 4 and 5 summarize our performance–latency trade-offs. Reducing the number of Llama 3.1 8B layers from 32 to 16 while keeping context size fixed preserves accuracy while reducing latency by approximately 42.5%. When the context size is reduced, alethic errors increase due to incomplete retrieval, raising the model's masking rate (i.e., instances where no answer is returned due to low confidence). Nevertheless, the 16-layer configuration achieves comparable performance to the 32-layer setup at lower computational cost. We observe a slight improvement in response latency when hosting the model using vLLM inference compared to Hugging Face's inference API, likely due to vLLM's optimized memory management and continuous batching capabilities.

Overall, our approach leveraging FFN activations as auto-regressive signals, modeling them with an LSTM, and regularizing with L_{Huber} proves effective in long-form RAG settings. This method improves the trustworthiness of LLM-generated responses and holds strong potential for safe deployment in sensitive, domain-specific applications.

6 Limitations

Our work pushes the boundary of confidence estimation in retrieval-augmented generation (RAG) for sensitive domains, but several practical considerations remain. Ideally, a RAG system should generate both the response and its confidence score

in a single pass. In our current implementation, the confidence score requires a second run of the system, which introduces additional computational and latency overhead.

While this design choice enables deeper access to model internals, it also necessitates operating in a white-box setting, as the confidence model relies on activation signals from the LLM to assess correctness. Furthermore, the method is customized to the specific architecture of the target model, meaning that adaptation to other LLMs may require reconfiguration and retraining. These limitations also present opportunities for future research: integrating confidence estimation directly into the generation process, reducing computational cost, and developing architecture-agnostic approaches that preserve the performance benefits of activation-based probing methods.

A limitation of this study is that the dataset cannot be made publicly available. The data contains sensitive and proprietary information pertaining to internal financial tools and knowledge resources used by service agents within a financial institution. This restriction is mandated by internal data governance policies to protect confidential and regulated financial information.

References

AI@Meta. 2024. Llama 3 model card.

Amos Azaria and Tom Mitchell. 2023. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore. Association for Computational Linguistics.

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. *arXiv preprint arXiv:2402.11756*.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. HHEM-2.1-Open.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. *arXiv preprint arXiv:2307.01379*.

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. 2024. Enhancing confidence expression in large language models through learning from past experience. *arXiv preprint arXiv:2404.10315*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Linyu Liu, Yu Pan, Xiaocheng Li, and Guanting Chen. 2024. Uncertainty estimation and quantification for llms: A simple supervised approach. *arXiv preprint arXiv:2404.15993*.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6107–6117.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Sabrina J Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Rishabh Patra, Ramya Hebbalaguppe, Tirtharaj Dash, Gautam Shroff, and Lovekesh Vig. 2023. Calibrating deep neural networks using explicit regularisation and dynamic data pruning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1541–1549.

Vaishnavi Shrivastava, Percy Liang, and Ananya Kumar. 2023. Llamas know what gpts don't show: Surrogate models for confidence estimation. *arXiv preprint arXiv:2311.08877*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*.

Amortized Bayesian Meta-Learning for Low-Rank Adaptation of Large Language Models

Liyi Zhang
Princeton University
Department of Computer Science
zhang.liyi@princeton.edu

Jake Snell
Princeton University
Department of Computer Science
jsnell@princeton.edu

Thomas L. Griffiths
Princeton University
Department of Psychology and Computer Science
tomg@princeton.edu

Abstract

Fine-tuning large language models (LLMs) with low-rank adaptaion (LoRA) is a cost-effective way to incorporate information from a specific dataset. However, it is often unclear how well the fine-tuned LLM will generalize, i.e., how well it will perform on unseen datasets. Methods have been proposed to improve generalization by optimizing with in-context prompts, or by using meta-learning to fine-tune LLMs. However, these methods are expensive in memory and computation, requiring either long-context prompts or saving copies of parameters and using second-order gradient updates. To address these challenges, we propose Amortized Bayesian Meta-Learning for LoRA (ABMLL). This method builds on amortized Bayesian meta-learning for smaller models, adapting this approach to LLMs while maintaining its computational efficiency. We reframe task-specific and global parameters in the context of LoRA and use a set of new hyperparameters to balance reconstruction accuracy and the fidelity of task-specific parameters to the global ones. ABMLL provides effective generalization and scales to large models such as LLAMA3-8B. Furthermore, as a result of using a Bayesian framework, ABMLL provides improved uncertainty quantification. We test ABMLL on Unified-QA and Crossfit datasets and find that it outperforms existing methods on these benchmarks in terms of both accuracy and expected calibration error.

1 Introduction

Large language models (LLMs) handle a variety of tasks reasonably well (Radford et al., 2019). However, to tailor LLMs to specific domains, fine-tuning on specific datasets is often necessary. While methods such as low-rank adaptation (LoRA; Hu et al. (2021)) fine-tune a pretrained LLM cost-effectively, a fine-tuned LLM is limited to the domain it is trained on. Its performance may not improve in other domains and sometimes worsens as it suffers from catastrophic forgetting. Such catastrophic forgetting may result in overfitting and erasing existing capabilities of the pretrained LLM (Lazaridou et al., 2021; Luo et al., 2023).

Meta-learning is a strategy for solving this problem, training models on a variety of tasks in a way that supports generalization across tasks (Finn et al., 2017). However, meta-learning typically requires a large amount of computation and memory, making it challenging to apply to LLMs. One form of meta-learning that has been applied to LLMs involves fine-tuning models on in-context prompt-response examples (Min et al., 2022; Chen et al., 2022). Another more traditional approach, MAML-en-LLM (Sinha et al., 2024), adapts the Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) framework to LLMs. However, both methods are limited in the size of the language models that can be used: the former requires long-context prompts, whereas the latter uses second-order gradient updates and saves a model for each task.

Recent work on Amortized Bayesian Meta-Learning (ABML; Ravi and Beatson (2019)) addresses some of the computation and memory requirements of meta-learning. This approach posits a generative model over parameters where task-specific parameters are generated from global parameters, and inference over task-specific parameters is amortized. In other words, the conditional distribution over task-specific parameters is shared across tasks, implying that computation and memory costs stay constant with respect to the number of tasks. This approach thus offers a path towards efficient meta-learning for LLMs. However, several challenges exist. First, we need to specify the generative model over weight space in the context of LLMs. Second, the prior term used in ABML no longer adapts to the setting of fine-tuning a pretrained model because the spread of its weights mismatches that of an arbitrary prior used to train a model from scratch. Third, the enormous size

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 194–199
November 9, 2025 ©2025 Association for Computational Linguistics

of LLMs makes training difficult, as the scale of probabilities assigned to the model variables can overwhelm the influence of the data likelihood.

In this paper, we present a solution to these problems, taking a Bayesian approach to fine-tuning LLMs using ABML. To define the underlying generative model and efficiently characterize the distributions involved, we use LoRA to express both the model weights and their uncertainty. We introduce a new prior over global variables that accounts for the spread of the parameters learned in the pretrained model. We also introduce two adjustable hyperparameters that balance reconstruction accuracy and the fidelity of task-specific parameters to the global ones.

Using amortized Bayesian meta-learning for LLM fine-tuning, we achieve both higher accuracy and better uncertainty estimation over unseen tasks compared with regular fine-tuning and other scalable methods in the meta-learning literature. Figure 1 illustrates an example where incorporating uncertainty estimation in fine-tuning leads to a more calibrated model response. Our method is scalable and avoids the computation and memory overhead of other meta-learning approaches, making it adaptable to larger models such as LLAMA 3 8B. We show that amortized Bayesian meta-learning provides fine-tuned LLMs that are accurate on domain-specific tasks, more generalizable to new tasks, and provide better uncertainty estimation.

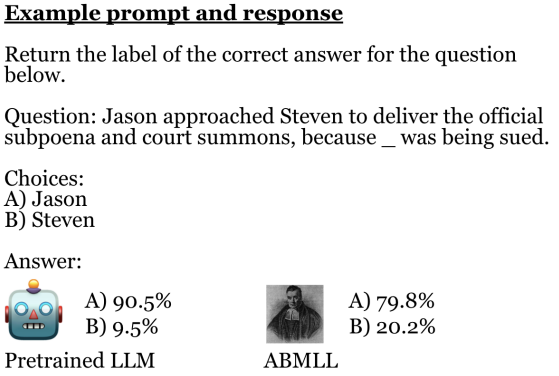


Figure 1: An example where better uncertainty calibration leads to a more reasonable response. This is a prompt and response from an unseen dataset, coming from a pretrained LLM (left) and an LLM fine-tuned with ABMLL (right), with both being updated with 10 gradient steps on other examples of this dataset as in the meta-learning literature. The label is B), so both LLMs are incorrect, but the question is ambiguous: it could interpreted as either Jason "asked" Steven to deliver, or Jason "came to" Steven to deliver, resulting in different answers. ABMLL results in a more calibrated response.

2 Related Work

Meta-learning methods in LLMs. Extensive work has explored meta-learning for generalization, typically adopted for models in the pre-LLM era (Finn et al., 2017; Snell et al., 2017; Ravi and Beatson, 2019; Nichol et al., 2018). Sinha et al. (2024) adapted Model-Agnostic Meta-Learning (MAML), developed in Finn et al. (2017), to LLMs. However, this adaptation is more expensive in computation and memory than our method, requiring second-order gradient updates and saving a model for each task. More recently, Kim and Hospedales (2025) proposes a heirarchical Bayesian approach to LoRA meta-learning, but its parameters also increase linearly with number of tasks. As a result, we evaluate on larger models than those tried in these two papers.

As a different approach, Min et al. (2022) and Chen et al. (2022) explored meta-learning for LLMs using in-context learning. These works show that it is possible to fine-tune LLMs on in-context examples and achieve generalization. However, our approach does not require curation of such examples, does not place constraints on the size of the context window of a model, and is more scalable.

Uncertainty representation for LLMs. Approaches to capturing uncertainty for LLMs can rely on the intrinsic representation of uncertainty in the model or focus on capturing extrinsic uncertainty about model parameters. Intrinsic approaches produce better uncertainty calibration via prompt engineering and sampling (Gruver et al., 2023) or learning an external model (Shen et al., 2024). Extrinsic approaches include using fine-tuning methods to incorporate uncertainty, such as training LoRA with ensembles (Balabanov and Linander, 2024), Laplace approximation (Yang et al., 2023), and variational inference (Wang et al., 2024). Our work takes the extrinsic approach but differs from existing approaches by using the meta-learning setting to achieve strong uncertainty calibration through generalization across datasets.

3 Background

3.1 Low-Rank Adaptation (LoRA)

LoRA (Hu et al., 2021) fine-tunes LLM weights on a low-rank space to improve efficiency compared with regular fine-tuning. Let $\mathbf{W}_0$ of size d_{out}-by-d_{in} denote a weight matrix from a pretrained LLM. Let

$\mathbf{x}$ denote the input to $\mathbf{W}_0$, and $\mathbf{z}$ denote the output of $\mathbf{W}_0$, LoRA fine-tunes pretrained weight $\mathbf{W}_0$ by adding perturbation on the low-ranked space,

$$\mathbf{z} = (\mathbf{W}_0 + \Delta\mathbf{W}_0)\mathbf{x} = (\mathbf{W}_0 + \mathbf{BA})\mathbf{x}.$$

The trainable matrices $\mathbf{B}$ and $\mathbf{A}$ are known as *LoRA adapters*. The sizes of $\mathbf{B}$ and $\mathbf{A}$ are d_{out}-by-d_{rank} and d_{rank}-by-d_{in}, respectively, with d_{rank} being significantly smaller than the original dimensions. Therefore, the number of parameters to be updated are $(d_{\text{out}} + d_{\text{in}})d_{\text{rank}}$, significantly fewer than the original $d_{\text{out}}d_{\text{in}}$.

3.2 Amortized Bayesian Meta-Learning

Amortized Bayesian Meta-Learning (ABML) Ravi and Beatson (2019) improves upon MAML-based meta-learning frameworks by representing uncertainty with a Bayesian approach. It also amortizes inference over the parameters so that memory no longer increases linearly with the number of tasks.

Let θ denote global parameters such that a few steps of gradient descent will produce local parameters ϕ_i on task i with dataset D_i. ABML treats θ as random variables, and minimizes a negative evidence lower bound using variational inference,

$$\text{argmin}_\theta \Big[\sum_{i=1}^{M} -E_{q_\theta(\phi_i|D_i)}[\log p(D_i|\phi_i)] + \quad (1)$$
$$\text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big)\Big] + \text{KL}(q(\theta)||p(\theta)).$$

The variational distribution $q_\theta(\phi_i|D_i)$ is represented by the Gaussian distribution $N(\mu_\phi, \sigma_\phi^2)$ with μ_ϕ, σ_ϕ as trainable parameters.

4 Method

Our method extends Amortized Bayesian Meta-Learning, making it possible to apply to LLMs. This approach combines the advantages of meta-learning for adapting to new tasks with Bayesian inference for uncertainty representation.

We use the the objective of Eq. 1 from ABML. In our setting, θ and ϕ_i are the global and task-specific model parameters produced as the output of LoRA adapters. On a high level, the generative process is

$$\theta \sim p(\theta),$$
$$\phi_i \sim p(\phi_i|\theta),$$
$$D_i \sim \text{LLM}(\phi_i),$$

Algorithm 1 One epoch in the ABMLL algorithm. The "test section" does not need to be performed every epoch.

Input: Likelihood model $p(D_i|\phi_i)$, prior $p(\theta)$ and $p(\phi|\theta)$, variational posterior $q_\theta(\phi_i|D_i)$, with trainable parameters $\mathbf{B}, \mathbf{A}$; constant c, β; number of tasks M and inner-loop size K.

Training section
for task $i \in \{1, 2, ..., M\}$ **do**
  Draw batch D_i from task i dataset.
  Inner-loop:
  for iter $k \in \{1, 2, ..., K\}$ **do**
    Run a step gradient descent to minimize w.r.t. ϕ_i: $-E_{q_\theta(\phi_i|D_i)}[\log p(D_i|\phi_i)] + \beta\text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big)$.
  end for
  Outer-loop: Run a step gradient descent to minimize w.r.t. θ: $-E_{q_\theta(\phi_i|D_i)}[\log p(D_i|\phi_i)] + \beta\text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big) + \beta\text{KL}(q(\theta)||p(\theta))$.
end for
Test section
Take unseen task i. Create a copy of the above weights, and on the new weights:
for iter $k \in \{1, 2, ..., K\}$ **do**
  Draw batch D_i from task i dataset.
  Run a step gradient descent to minimize w.r.t ϕ_i: $-E_{q_\theta(\phi_i|D_i)}[\log p(D_i|\phi_i)] + \beta\text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big)$.
end for
Evaluate on rest of data in task i.
Delete the weights copy and reload the weights at the end of training section.
Output: $\mathbf{B}$, $\mathbf{A}$.

where i represents any task i, and $\text{LLM}(\phi_i)$ denotes the LLM considered as a probabilistic model that takes ϕ_i as its weights and outputs token sequences with joint probabilities defined by the LLM's autoregressive predictive distribution. We provide a pseudocode, Algorithm 1, to illustrate our approach. For any LLM layer with pretrained weights $\mathbf{W_0}$, the quantities for our extension to ABML are:

$$\mu_\theta = \mathbf{B}_{\mu_\theta}\mathbf{A}_{\mu_\theta},$$
$$\log\sigma_\theta^2 = \mathbf{B}_{\sigma_\theta}\mathbf{A}_{\sigma_\theta} + c\mathbf{I},$$
$$\mu_\phi = \mathbf{B}_{\mu_\phi}\mathbf{A}_{\mu_\phi},$$
$$\log\sigma_\phi^2 = \mathbf{B}_{\sigma_\phi}\mathbf{A}_{\sigma_\phi} + c\mathbf{I},$$

$$
\begin{aligned}
p(\phi_i|\theta) &= N(\phi_i; \mu_\theta + \mathbf{W_0}, \sigma_\theta^2), \\
q_\theta(\phi_i|D_i) &= N(\phi_i; \mu_\phi + \mathbf{W_0}, \sigma_\phi^2), \\
p(\theta) &= p(\mu_\theta, \sigma_\theta) \\
&= N(\mu_\theta; 0, \mathbf{I}) \cdot \text{Gamma}(\frac{1}{\sigma_\theta^2}; a_0, b_0), \\
\text{KL}(q(\theta)||p(\theta)) &= -\log p(\theta).
\end{aligned}
$$

Lastly, $p(D_i|\phi_i)$ is defined as the joint probability assigned to D_i where the LLM takes ϕ_i as its weights. The trainable parameters are the LoRA adapters $\mathbf{A}$ and $\mathbf{B}$. However, we introduce four pairs of these adapters to compute both the mean and variance of the LoRA outputs on local and global model weights. $\mathbf{I}$ is identity matrix, and c is a hyperparameter constant dependent on the spread of pretrained LLM weights. a_0 and b_0 are hyperparameters, and the simplification of the KL term as $-\log p(\theta)$ follows Ravi and Beatson (2019).

Balancing the reconstruction error. LLMs are often overparameterized. As a result, probabilistic quantities on the space of weights, $\text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big)$ and $\text{KL}(q(\theta)||p(\theta))$, can overwhelm quantities on the data space, $\log p(D_i|\phi_i)$. $\beta-$VAE (Higgins et al., 2016) and Bayesian neural network approaches by Trinh et al. (2022) introduce hyperparameters to temper the likelihood versus regularization terms. Inspired by this idea, we introduce hyperparameters β, γ, resulting in the following objective,

$$
\begin{aligned}
\text{argmin}_\theta \Big[\sum_{i=1}^{M} -E_{q_\theta(\phi_i|D_i)}[\log p(D_i|\phi_i)] + \\
\beta \text{KL}\big(q_\theta(\phi_i|D_i)\big|\big|p(\phi_i|\theta)\big)\Big] + \gamma \text{KL}(q(\theta)||p(\theta)).
\end{aligned} \tag{2}
$$

This provides a flexible way to control how close the global parameters θ are to the prior $p(\theta)$, and how close the task-specific parameters ϕ_i are to θ.

5 Empirical Evaluations

Model and datasets. We fine-tune LLAMA3-8B on CrossFit (Ye et al., 2021) and UnifiedQA (Ye et al., 2021), textual datasets commonly used to train meta-learning models. Because a key aim of our paper is to study uncertainty quantification, we filter for multiple choice datasets, leading to a subset of CrossFit and UnifiedQA with 34 datasets with 68K training datapoints in total. They feature problems such as sentiment analysis, natural language inference, and identifying particular traits or topics in a given text. For evaluation on an unseen task, we use Winogrande (Sakaguchi et al., 2021), a multiple choice dataset evaluating common sense reasoning.

Metrics. We use accuracy to evaluate general performance and expected calibration error (ECE) to evaluate uncertainty estimation.

Implementation details. All methods use batch-size of 2 and inner-loops with 5 gradient steps. LoRA adapters follow standard practice with rank $= 8$, and learning rate is tuned in $[10^{-6}, 5 \cdot 10^{-5}]$. For ABMLL, $\beta = 5 \cdot 10^{-10}$, $\gamma = 10^{-6}$, $c = e^{-20}$. For the gamma prior, $a_0 = 1, b_0 = 0.01$, following Ravi and Beatson (2019). During validation on the unseen dataset, all models train 10 gradient steps on 10 batches from this dataset and evaluate on the rest.

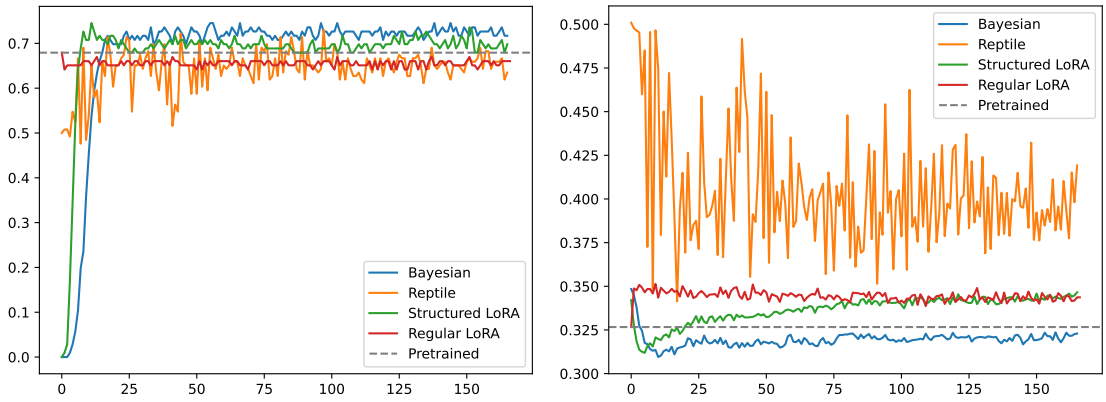
Baselines. We use four baseline methods that can viably scale to LLAMA3-8B. *Pretrained* is the off-the-shelf LLM. *Regular LoRA* is the default LoRA method trained on the whole randomly shuffled training dataset. *Structured LoRA* also uses the default LoRA, but the training dataset follows the same "structure" as our method: it is iteratively trained 5 gradient steps on one task at a time. Thus, it tests the effect of our generative model on performance. The *Reptile* (Nichol et al., 2018) algorithm primarily uses a weighted average between new weights and previous weights to achieve meta-learning.

Results. Figure 2 shows validation accuracy and ECE over epochs across methods. We observe that ABMLL consistently achieves higher accuracy. On ECE, ABMLL also consistently achieves the best performance, whereas structured LoRA worsens as training continues. For fairness, the result at the end of every sixth epoch is reported from regular LoRA, because both ABMLL and Reptile run six instead of one gradient step during each epoch's training.

Table 1 reports the best validation score for each model from three random seeds, showing a statistically significant advantage for ABMLL.

Conclusion

Meta-learning is an effective method for supporting better generalization across datasets, but its demands on computation and memory can make it difficult to apply to large language models. We have shown how meta-learning can be used to adapt LLMs by combining Amortized Bayesian



(a) Accuracy.

(b) Expected calibration error (lower is better).

Figure 2: Validation accuracy and ECE on the vertical axis over epochs on the horizontal axis across our method (ABMLL) and four benchmarks. On accuracy, ABMLL consistently achieves higher. On ECE, ABMLL also consistently achieves the best performance, whereas structured LoRA, the second best performer on accuracy, worsens on uncertainty calibration as training continues.

Table 1: Validation accuracy and ECE across three random seeds, with standard error.

Method	Accuracy ↑	ECE ↓
Pretrained	$68.2\% \pm 0.3\%$	0.327 ± 0.000
Regular LoRA	$68.2\% \pm 0.3\%$	0.327 ± 0.000
Structured LoRA	$73.6\% \pm 0.6\%$	0.320 ± 0.001
Reptile	$73.5\% \pm 0.2\%$	0.370 ± 0.005
ABMLL	$\mathbf{74.8\% \pm 0.3\%}$	$\mathbf{0.317 \pm 0.001}$

Meta-Learning with Low-Rank Adaptation. This approach results not just in better accuracy across several benchmarks, but also in better calibration.

Limitations

One limitation of the paper is the scope of empirical evaluation regarding datasets and models. While the datasets feature natural text that can occur in the real world, it would be beneficial to evaluate on more test datasets to confirm the method's consistency. Additionally, the paper's method can be naturally extended to other models, so evaluating on more models would be a reasonable venue for future work.

As a meta-learning method, our approach must be trained on datasets that can be naturally divided into different tasks, a requirement that is not always available to practioners seeking significant model improvement on one particular domain.

While our empirical results suggest that our approach provides more accurate and calibrated responses, theoretical convergence is not guaranteed due to the need for approximate inference and various design choices, including limitations of the variational family $q_\theta(\phi_i|D_i)$.

Acknowledgments

We thank Qinyuan Ye for helpful discussion. We also thank the anonymous reviewers for helpful comments. This work was supported by grant N00014-23-1-2510 from the Office of Naval Research.

References

Oleksandr Balabanov and Hampus Linander. 2024. Uncertainty quantification in fine-tuned llms using lora ensembles. *ArXiv*, abs/2402.12264.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730. Association for Computational Linguistics.

Chelsea Finn, P. Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew Gordon Wilson. 2023. Large language models are zero-shot time series forecasters. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NeurIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Irina Higgins, Loïc Matthey, Arka Pal, Christopher P. Burgess, Xavier Glorot, Matthew M. Botvinick, Shakir Mohamed, and Alexander Lerchner. 2016.

beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

J. Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *ArXiv*, abs/2106.09685.

Minyoung Kim and Timothy M Hospedales. 2025. Lift: Learning to fine-tune via bayesian parameter efficient meta fine-tuning. The Thirteenth International Conference on Learning Representations, 2025.

Angeliki Lazaridou, Adhiguna Kuncoro, Elena Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Giménez, Cyprien de Masson d'Autume, Tomás Kociský, Sebastian Ruder, Dani Yogatama, Kris Cao, Susannah Young, and Phil Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. In *Neural Information Processing Systems*.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *ArXiv*, abs/2308.08747.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *ArXiv*, abs/1803.02999.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Sachin Ravi and Alex Beatson. 2019. Amortized bayesian meta-learning. In *International Conference on Learning Representations*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: an adversarial winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106.

Maohao Shen, Subhro Das, Kristjan Greenewald, Prasanna Sattigeri, Gregory W. Wornell, and Soumya Ghosh. 2024. Thermometer: Towards universal calibration for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44687–44711. PMLR.

Sanchit Sinha, Yuguang Yue, Victor Soto, Mayank Kulkarni, Jianhua Lu, and Aidong Zhang. 2024. Maml-en-llm: Model agnostic meta-training of llms for improved in-context learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, page 2711–2720. Association for Computing Machinery.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS'17, page 4080–4090, Red Hook, NY, USA. Curran Associates Inc.

Trung Trinh, Markus Heinonen, Luigi Acerbi, and Samuel Kaski. 2022. Tackling covariate shift with node-based bayesian neural networks. In *International Conference on Machine Learning*, pages 21759–21774. PMLR.

Yibin Wang, Haizhou Shi, Ligong Han, Dimitris N. Metaxas, and Hao Wang. 2024. BLob: Bayesian low-rank adaptation by backpropagation for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Adam X. Yang, Maxime Robeyns, Xi Wang, and Laurence Aitchison. 2023. Bayesian low-rank adaptation for large language models. *ArXiv*, abs/2308.13111.

Qinyuan Ye, Bill Yuchen Lin, and Xiang Ren. 2021. CrossFit: A few-shot learning challenge for cross-task generalization in NLP. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7163–7189. Association for Computational Linguistics.

Towards Trustworthy Summarization of Cardiovascular Articles: A Factuality-and-Uncertainty-Aware Biomedical LLM Approach

Eleni Partalidou[1,2], Tatiana Passali[1], Chrysoula Zerva[3,4,5], Grigorios Tsoumakas[1,2], Sophia Ananiadou[2,6]

[1]Aristotle University of Thessaloniki, [2]Archimedes, Athena Research Center, Greece, [3]Instituto Superior Técnico, Universidade de Lisboa, [4]Instituto de Telecomunicações, Portugal, [5]ELLIS Unit Lisbon, [6]The University of Manchester

Correspondence: epartala@csd.auth.gr

Abstract

While large, biomedical documents with complex terminology are in need of being understood more easily and efficiently, summarizing this kind of content can be problematic, as Large Language Models (LLMs) aren't always trustworthy. Considering the importance of comprehending Cardiovascular Diseases, we study in depth the ability of different state-of-the-art biomedical LLMs to generate factual and certain summaries in this topic, and examine which generation choices can influence their trustworthiness. To that end, besides using factuality metrics, we employ techniques for token-level uncertainty estimation, an area that has received little attention from the scientific community. Our results reveal dissimilarities between LLMs and generation methods, and highlight connections between factuality and uncertainty metrics, thereby laying the groundwork for further investigation in the area.

1 Introduction

Biomedical researchers worldwide try to solve vital medical problems and publish scientific discoveries. Due to the exponentially increasing amount of scientific publications, summarizing them is vital, as they enable medical practitioners to keep up with the literature in an efficient manner. For that reason, it is crucial that the summary is accurate, as a minor mistake in explaining a medical concept or an unclear medical advice to treat a disease can have severe consequences for the health of patients. Large Language Models (LLMs) have recently been used to process and understand this kind of information in depth.

In recent years, LLMs have gained much attention from the scientific community, as they have been especially transformative for generative tasks, such as text summarization, machine translation, and question answering (Jurafsky and Martin, 2025). Text summarization, the task of our interest, is the process of creating a brief, accurate, and coherent summary of a longer text document. LLMs have greatly facilitated this task by providing the option to generate new text with the most salient information (i.e., abstractive summarization; Shakil et al. (2024)). In biomedicine, scientific findings tend to be reported in large documents with complex terminology, so summarizing scientific content can make important, clinical information accessible to researchers and clinicians more easily and efficiently (Xie et al., 2023a). However, LLMs may misrepresent their confidence and have specifically been shown to overestimate their knowledge and certainty level (they don't know what they don't know). As a result, they may confidently generate summaries with hallucinations or ambiguities (Baan et al., 2023; Hu et al., 2023) that can lead to misinformation with potentially severe consequences in medical contexts.

Although previous studies have explored uncertainty in biomedicine (Zerva et al., 2017; Kim et al., 2025), most either attempt to detect confidence expressions in text (Vasilakes et al., 2022), or focus on sequence-level uncertainty (Farquhar et al., 2024; Wagner et al., 2024; Qiu and Miikkulainen, 2024; Nikitin et al., 2024), frequently requiring the use of external models, repetitive sampling, or dedicated loss functions. Instead, we focus on simple, flexible, token-level uncertainty metrics, which can detect fine-grained local uncertainties, while also avoiding sequence-level limitations, such as length bias and over-correction that arise when collapsing token distributions into a single score. This area has received little attention in biomedical summarization, despite its importance in early detection and avoidance of misleading outputs. We thus attempt to address this gap and present early findings, assessing several biomedical LLMs on summarization of literature related to Cardiovascular Diseases.

We argue that a trustworthy model should not

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 200–207
November 9, 2025 ©2025 Association for Computational Linguistics

only achieve high factuality but also high confidence, especially for factually correct generations. We thus examine different factors that could influence factuality and uncertainty, such as decoding strategies and fine-tuning. We find that the proposed factuality and confidence metrics significantly correlate for some model variants, motivating further token-level analysis and more dedicated uncertainty metrics.

2 Related work

Biomedical summarization has become an important task and recent studies show an increased interest at it with the proposal of novel approaches based on LLMs (Xie et al., 2023a). Firstly, (Luo et al., 2022) introduced a new task of readability controllable summarization for biomedical documents, which aims to recognize users' readability demands and generate summaries that better suit their needs. Moreover, (Luo et al., 2023) proposed a novel citation-aware scientific paper summarization framework based on a citation graph, able to accurately locate and incorporate the salient contents from references, as well as capture varying relevance between source papers and their references. Lastly, (Xie et al., 2023b) addressed the issues of low-coherence summaries and the lack of explainability in black-box models by proposing a domain knowledge-enhanced graph topic transformer for explainable biomedical text summarization.

Evaluation of factuality in biomedical text generation is an ongoing challenge. (Zha et al., 2023) introduced AlignScore, a holistic metric, based on a general function of information alignment of text and its unified framework, which achieved substantial improvements over previous metrics. (Min et al., 2023) advocated a new evaluation metric that computes factual accuracy from pieces of generated text and was used to compare the performance of different LLMs. Additionally, (Bishop et al., 2023) proposed a new evaluation framework, LongDocFACTScore, for detecting human factuality targeting specifically summarized, long documents. Finally, (Luo et al., 2024) introduced a human-annotated dataset of LLM-generated summaries of clinical texts (TreatFact) and revealed significant performance gaps in terms of factuality for open-source LLMs.

Previous work has comprehensively examined **uncertainty in Natural Language Generation** (NLG) systems (Baan et al., 2023; Hu et al., 2023) and has explored strategies to address uncertainty with the goal of making LLMs more trustworthy, especially in biomedicine (Zerva et al., 2017; Kim et al., 2025). (Xu et al., 2020) studied summarization decoders in both blackbox and whitebox ways by focusing on the entropy of the models' predictions and revealed that features, such as the sentence position and the syntactic distance between adjacent pairs of tokens, influence uncertainty. (Ulmer et al., 2024) focused on token-level uncertainty and proposed a method for non-exchangeable conformal prediction, which was shown to improve text generation quality. Finally, (Fadeeva et al., 2024) introduced a token-level uncertainty method named Claim Conditioned Probability (CCP), disentangling claim-specific uncertainty from model decisions on surface forms, etc.

3 Methodology

We propose two different metrics of uncertainty, we test them to commonly used decoding methods, and we measure their correlation to factual accuracy.

3.1 Decoding strategies

We evaluate several decoding strategies for LLMs to identify the one that produces the least uncertain abstractive summaries. Specifically, we compare:

- Greedy search: At each timestep it selects the word with the highest probability.
- Top-k sampling: The k most likely words are filtered and the probability mass is redistributed among them (Fan et al., 2018).
- Top-p sampling: It chooses from the smallest possible set of words, whose cumulative probability exceeds a threshold p. The probability mass is then redistributed among them.

We note that the token-level uncertainty metrics (Section 3.3) can be applied across decoding methods, and, as they do not require sampling several times, they are also applicable to greedy decoding.

3.2 Factuality metrics

The factuality metrics process the summary (claim) at the sequence-level and require ground truth (evidence) for computation, which in our case is the abstract of the article.

HHEM. A series of models for detecting hallucinations in LLMs. These models collect a list of claims and associated evidence and compute a

score between 0 and 1, where 0 means that the hypothesis is not evidenced at all and 1 means that the hypothesis is fully supported (Bao et al., 2024).

AlignScore. An automatic factual consistency metric, built on RoBERTa-large, applying a unified information alignment function between a claim and evidence. It splits each claim into sequences of specific length and each evidence into sentences, generates pairs, and computes an average score from the maximum alignment scores of the pairs. The score is between 0 (no factual accuracy) and 1 (full factual accuracy) (Zha et al., 2023).

3.3 Uncertainty metrics

Below we present the token-level uncertainty metrics we use. Even though they compute a value at each step, we average the values at sequence level.

Token Certainty. As a simple metric of model certainty at the token level, we use the maximum probability assigned to any token in the vocabulary at each decoding step. Thus, token certainty is defined as:

$$C = \max_i P(w_i), \tag{1}$$

where $P(w_i)$ is the probability assigned to token w_i in the vocabulary.

Token Entropy. Beyond computing token certainty based on probabilities, we define a complementary metric based on the entropy of the token probabilities at each step, hence accounting for the full probability distribution over the vocabulary. It is computed as:

$$E = -\sum_{i=1}^{V}(P(w_i)\log(P(w_i)), \tag{2}$$

where $P(w_i)$ is the probability assigned to token w_i in the vocabulary and V is the vocabulary size.

4 Experimental Setup

Below we describe the different features that are set up for the conduction of the experiments.

4.1 Biomedical LLMs

We use decoder-only LLMs that have been fine-tuned on biomedical content and give full access to the parameters for a more focused experimentation [1]. Specifically, we select the following variants:

BioMistral-7B (Labrak et al., 2024) is a suite of Mistral-based open source models pre-trained using textual data from PubMed Central Open Access. BioMistral is the first biomedical, multilingual LLM, demonstrating superior performance compared to existing open-source medical models. For the scope of our research, we use the default, 7B parameters version.

Meditron3-8B [2] is a LLaMA3.1, 8B model from a suite of open-source LLMs adapted to the medical domain named Meditron3. The models of this collection are co-designed by a global group of clinicians, humanitarian practitioners, and data scientists.

Phi4-14B is a decoder-only transformer of Microsoft built upon a blend of synthetic datasets, data from filtered public domain websites and acquired academic books, and Q&A datasets (Abdin et al., 2024). For compatibility with our work, we make use of the 14B parameters model from Meditron3, a model based on the Microsoft one.

Qwen2.5 models are another category of the Meditron3 collection fine-tuned from the organization of Qwen (Yang et al., 2024). Evaluation of used 7B and 14B parameters models showed that they are a better option for capturing real-world utility, especially in terms of contextual adaptation in under-represented settings.

4.2 Cardiology dataset

Cardiovascular diseases (CVDs) are the leading cause of death worldwide and a major contributor to reduced quality of life, with their prevalence driven by lifestyle and healthcare factors (Mensah et al., 2023; Mendis et al., 2011). Early detection and effective management are therefore essential to improving patient outcomes and reducing healthcare burdens. To support research in this area, we use biomedical literature from PubMed [3]. The dataset that we base our work on originates from (Cohan et al., 2018), which contains an amount of PubMed, long, and structured documents and we keep the same training, validation, and test splits. Additionally, the majority of the records contain one or more indexes named Medical Subject Headings (MeSH) [4]. The condition applied to filter the appropriate records is checking whether at least one of the MeSH terms falls into the category "Cardiovascular Diseases". Moreover, we ignore the

[1] For all models we use the version available on HuggingFace (Wolf et al., 2019).

[2] https://github.com/OpenMeditron

[3] https://pubmed.ncbi.nlm.nih.gov/

[4] https://www.ncbi.nlm.nih.gov/mesh/

records that have more than 8,192 tokens when processing, due to memory constraints. After these filterings, a total of 3,924 records for training, 230 records for validation, and 205 records for inference remain.

4.3 Input representation

The model input prompt is structured as follows:

$PROMPT\ article\ RESPONSE\ abstract$

for fine-tuning and:

$PROMPT\ article\ RESPONSE$

for inference, where $PROMPT$ is "Summarize the following biomedical article in a clear and concise manner, in no more than 300 words:" and $RESPONSE$ is "Summary:".

4.4 Hyperparameter settings

Our experiments are conducted on an Amazon, p5.48xlarge instance equipped with 192 vCPUs, 2,048 GiB RAM, and 8 NVIDIA H100 GPUs, each with 80 GiB of memory. Additionally, LoRA is applied to the models, and each biomedical model is fine-tuned with the cardiology dataset on 3 epochs with a batch size of 1, learning rate of 5^e-5, and the AdamW optimizer. Lastly, for the text generation strategies, we set K to 50 in the top-k sampling method and p to 0.70 in the top-p sampling method.

5 Results

In this section we present the comparisons across the metrics and models described above, accounting for different aspects, like the overall performance of the LLMs, the effect of fine-tuning on factuality and uncertainty, as well as differences between the decoding strategies. Finally, we assess the correlation between the factuality and uncertainty metrics.

5.1 Model Performance and Contribution of instruction fine-tuning

At first, we want to observe the level of contribution of instruction fine-tuning on the models. In Table 1 we present the experiments using greedy decoding. For the majority of the models, we do not observe significant improvements in terms of factuality and only small improvements in terms of certainty, because instruction fine-tuning pushes the LLMs to generate long outputs with knowledge they haven't seen before (Wu et al., 2025). However, as we want to keep the added information into all the models, we continue the experiments with the instruction fine-tuned ones.

We then compare the overall performance and trustworthiness of LLMs, focusing on the fine-tuned versions. Using the average rank shown in Table 1, it can be observed that the Qwen models are the best option across metrics, while Meditron-8B lags behind in both cases.

5.2 Investigation of decoding strategies

It is also important to understand whether different decoding strategies can impact the trustworthiness of a summary. For this comparison, we use the Qwen-7B and Qwen-14B models, since they outperform the rest with greedy decoding. From Table 2, it is evident that the sampling methods generate the most trustworthy summaries, i.e., outperform greedy decoding across metrics, with the token-entropy values decreasing greatly, producing both more accurate summaries, but also demonstrating higher model confidence during generation.

5.3 Correlations between the factuality and uncertainty metrics

As an initiative of finding relationships between the factuality and uncertainty metrics, we compute their correlation using Pearson's r. The sequence-level and token-level measures are paired with each other. The results in Table 3 show that the two types of metrics are correlated and that the most meaningful insights come from individual models, not decoding strategies alone.

6 Conclusions & Future Work

We evaluated the trustworthiness of state-of-the-art biomedical LLMs on summarization using both factuality and token-level uncertainty metrics. Results showed that model choice and decoding strategy influenced trustworthiness, even though we applied standard values on the sampling strategies for K and p, with Qwen variants performing best and sampling-based methods, especially top-p, producing more factual and confident summaries.

Several promising directions for future work include expanding the evaluation to larger and more diverse biomedical datasets to improve statistical reliability and test the generalizability of token-level uncertainty metrics across domains. Another direction is investigating different decoding hyperparameters to gain insights into how generation settings affect factuality and uncertainty. Finally, evaluating larger biomedical LLMs, including closed-source models, and incorporating human evalua-

Model	HHEM		AlignScore		Token Certainty		Token Entropy		Average Rank	
	ZS	FT	ZS	FT	ZS	FT	ZS	FT	Fact.	Uncert.
BioMistral-7B	0.1807	0.1915	**0.4493**	0.3364	**0.7736**	0.6746	**0.4007**	0.5239	3	2.5
Meditron-8B	0.1991	0.1994	0.3104	0.3112	0.6153	0.6239	2.1242	2.0916	4	5
Phi-14B	**0.2420**	0.2415	0.2319	0.2313	0.7487	0.7450	1.5654	1.5900	3	3.5
Qwen-7B	0.2263	0.2251	0.3263	0.3266	0.7445	0.7471	0.9976	0.9868	**2.5**	**2**
Qwen-14B	0.2367	0.2324	0.3158	0.3131	0.7678	0.7709	1.1612	1.1448	**2.5**	**2**

Table 1: Comparison of factuality and uncertainty across LLMs and effect of instruction fine-tuning. **Bold** values represent the best score for each metric and underlined ones the best score for each column. Results of zero-shot models are shown in the ZS columns and these of fine-tuned ones in the FT. Average ranks are shown separately for factuality (Fact.) and uncertainty (Uncert.).

	Method	HHEM	AlignScore	Certainty	Entropy
Qwen-7B	Greedy	0.2251	0.3266	0.7471	0.9868
	Top-k	0.2206	0.3161	0.7698	0.2361
	Top-p	**0.2354**	**0.3369**	**0.7988**	**0.1577**
Qwen-14B	Greedy	0.2324	0.3131	0.7709	1.1448
	Top-k	0.2375	**0.3496**	0.7343	0.3265
	Top-p	**0.2414**	0.3416	**0.7868**	**0.2033**

Table 2: Decoding strategy comparison for Qwen-7B/14B on factuality and uncertainty. **Bold** marks the best per metric within each model.

Correlation	Model	r	p
Certainty-HHEM	Meditron-8B	1.0	0.00
Entropy-AlignScore	Meditron-8B	-0.99	0.01
Certainty- Entropy	BioMistral-7B	-0.96	0.04
Entropy-AlignScore	Phi-14B	-0.96	0.04

Table 3: Observation of Pearson r correlation between the factuality and uncertainty metrics.

tion, along with automatic metrics, would further strengthen the reliability of the results.

Limitations

Although our work gives a great initiative for factuality and token-level uncertainty quantification in biomedical applications, there are still some areas that could be explored. At first, token-level uncertainty metrics can be used to other specializations of medicine in the future, other than cardiology, in order to proof the generalization of our work. Additionally, more descriptive, token-level metrics can be incorporated into the experiments for further uncertainty detection and quantification. Moreover, as we use Pearson coefficients, which do not account for confounding factors, like model architecture, decoding strategy, or dataset characteristics, some correlations may reflect model-level biases, rather than true causal relationships. The lack of comparison of automatic metrics to human evaluation is another limitation, which could strength the paper claims, if conducted. Lastly, due to the fact that factuality is a very important topic, future enhancements could investigate deeply factual accuracy and relativity to uncertainty.

Acknowledgments

This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program. Also, AWS resources were provided by the National Infrastructures for Research and Technology GRNET and funded by the EU Recovery and Resiliency Facility. In addition, this research was partially funded by an unrestricted gift from Google, by the Portuguese Recovery and Resilience Plan through project C645008882-00000055 (i.e., the Center For Responsible AI), by Fundação para a Ciência e Tecnologia (FCT) through 'OptiGov' (DOI 10.54499/2024.07385.IACDC), funded by the PRR under the measure RE-C05-i08.m04, by EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), and also by FCT/MECI through national funds and, when applicable, cofunded EU initiatives under contract UID/50008 for Instituto de Telecomunicações. Lastly, Sophia Ananiadou acknowledges support from the British Heart Foundation Centre of Research Excellence, University of Manchester (Award Code: RC000797).

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J Hewett, Mojan Javaheripi, Piero

Kauffmann, and 1 others. 2024. Phi-4 technical report. *arXiv preprint arXiv:2412.08905*.

Joris Baan, Nico Daheim, Evgenia Ilia, Dennis Ulmer, Haau-Sing Li, Raquel Fernández, Barbara Plank, Rico Sennrich, Chrysoula Zerva, and Wilker Aziz. 2023. Uncertainty in natural language generation: From theory to applications. *arXiv preprint arXiv:2307.15703*.

Forrest Bao, Miaoran Li, Rogger Luo, and Ofer Mendelevitch. 2024. HHEM-2.1-Open.

Jennifer A Bishop, Qianqian Xie, and Sophia Ananiadou. 2023. Longdocfactscore: Evaluating the factuality of long document abstractive summarisation. *arXiv preprint arXiv:2309.12455*.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. 2018. A discourse-aware attention model for abstractive summarization of long documents. *arXiv preprint arXiv:1804.05685*.

Ekaterina Fadeeva, Aleksandr Rubashevskii, Artem Shelmanov, Sergey Petrakov, Haonan Li, Hamdy Mubarak, Evgenii Tsymbalov, Gleb Kuzmin, Alexander Panchenko, Timothy Baldwin, and 1 others. 2024. Fact-checking the output of large language models via token-level uncertainty quantification. *arXiv preprint arXiv:2403.04696*.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Mengting Hu, Zhen Zhang, Shiwan Zhao, Minlie Huang, and Bingzhe Wu. 2023. Uncertainty in natural language processing: Sources, quantification, and applications. *arXiv preprint arXiv:2306.04459*.

Daniel Jurafsky and James H. Martin. 2025. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd edition. Online manuscript released January 12, 2025.

Yubin Kim, Hyewon Jeong, Shan Chen, Shuyue Stella Li, Mingyu Lu, Kumail Alhamoud, Jimin Mun, Cristina Grau, Minseok Jung, Rodrigo Gameiro, and 1 others. 2025. Medical hallucinations in foundation models and their impact on healthcare. *arXiv preprint arXiv:2503.05777*.

Yanis Labrak, Adrien Bazoge, Emmanuel Morin, Pierre-Antoine Gourraud, Mickael Rouvier, and Richard Dufour. 2024. Biomistral: A collection of open-source pretrained large language models for medical domains. *arXiv preprint arXiv:2402.10373*.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2022. Readability controllable biomedical document summarization. *arXiv preprint arXiv:2210.04705*.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2023. Citationsum: Citation-aware graph contrastive learning for scientific paper summarization. In *Proceedings of the ACM web conference 2023*, pages 1843–1852.

Zheheng Luo, Qianqian Xie, and Sophia Ananiadou. 2024. Factual consistency evaluation of summarization in the era of large language models. *Expert Systems with Applications*, 254:124456.

Shanthi Mendis, Pekka Puska, B editors Norrving, World Health Organization, and 1 others. 2011. *Global atlas on cardiovascular disease prevention and control*. World Health Organization.

George A Mensah, Valentin Fuster, Christopher JL Murray, Gregory A Roth, Global Burden of Cardiovascular Diseases, and Risks Collaborators. 2023. Global burden of cardiovascular diseases and risks, 1990-2022. *Journal of the American College of Cardiology*, 82(25):2350–2473.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Factscore: Fine-grained atomic evaluation of factual precision in long form text generation. *arXiv preprint arXiv:2305.14251*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *Advances in Neural Information Processing Systems*, 37:8901–8929.

Xin Qiu and Risto Miikkulainen. 2024. Semantic density: Uncertainty quantification for large language models through confidence measurement in semantic space. *Advances in neural information processing systems*, 37:134507–134533.

Hassan Shakil, Ahmad Farooq, and Jugal Kalita. 2024. Abstractive text summarization: State of the art, challenges, and improvements. *Neurocomputing*, 603:128255.

Dennis Ulmer, Chrysoula Zerva, and André FT Martins. 2024. Non-exchangeable conformal language generation with nearest neighbors. *arXiv preprint arXiv:2402.00707*.

Jake Vasilakes, Chrysoula Zerva, Makoto Miwa, and Sophia Ananiadou. 2022. Learning disentangled representations of negation and uncertainty. *arXiv preprint arXiv:2204.00511*.

Nico Wagner, Michael Desmond, Rahul Nair, Zahra Ashktorab, Elizabeth M Daly, Qian Pan, Martín Santillán Cooper, James M Johnson, and Werner Geyer. 2024. Black-box uncertainty quantification method for llm-as-a-judge. *arXiv preprint arXiv:2410.11594*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Tianyi Wu, Jingwei Ni, Bryan Hooi, Jiaheng Zhang, Elliott Ash, See-Kiong Ng, Mrinmaya Sachan, and Markus Leippold. 2025. Balancing truthfulness and informativeness with uncertainty-aware instruction fine-tuning. *arXiv preprint arXiv:2502.11962*.

Qianqian Xie, Zheheng Luo, Benyou Wang, and Sophia Ananiadou. 2023a. A survey for biomedical text summarization: From pre-trained to large language models. *arXiv preprint arXiv:2304.08763*.

Qianqian Xie, Prayag Tiwari, and Sophia Ananiadou. 2023b. Knowledge-enhanced graph topic transformer for explainable biomedical text summarization. *IEEE journal of biomedical and health informatics*, 28(4):1836–1847.

Jiacheng Xu, Shrey Desai, and Greg Durrett. 2020. Understanding neural abstractive summarization models via uncertainty. *arXiv preprint arXiv:2010.07882*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Chrysoula Zerva, Riza Batista-Navarro, Philip Day, and Sophia Ananiadou. 2017. Using uncertainty to link and rank evidence from biomedical literature for model curation. *Bioinformatics*, 33(23):3784–3792.

Yuheng Zha, Yichi Yang, Ruichen Li, and Zhiting Hu. 2023. Alignscore: Evaluating factual consistency with a unified alignment function. *arXiv preprint arXiv:2305.16739*.

A Examples of summaries with low and high uncertainty

Below we present the least and most uncertain summaries of different cases using the fine-tuned, Qwen-7B model.

Greedy search (Least uncertain)

PMCID: PMC2913399

Summary: A network is generated by determining the first- and second - order interactions of cd proteins associated with a given morphological subgroup in a human protein interaction network consisting of refined experimental proteomics data . This network is described in high detail in lage et al (2007 , 2008) , and online (http://www.cbs.dtu.dk/suppl/dgf/) . Interactions of the cd proteins are integrated into a network by always including direct interactions between cd proteins , and only including indirect interactions mediated through proteins with q percent of its interactions to the cd set . Various thresholds for q are iteratively tested and value of q for the final network is chosen based on which value gives the optimal network significance , this procedure is described in detail in bergholdt et al (2007) and d'hertog et al (2007) . The method for determining network significances can be seen below . Detailed views of the networks can be seen in supplementary figures s1 , s2 , s3 and s4 . The significance of each of the generated 19 networks was determined by randomization testing as described in detail earlier (bergholdt et al , 2007 ; d'hertog et al , 2007) . Specifically , for an input set of ninput proteins yielding an interaction network (connected component) with g input proteins and t total proteins , a network score (ns

Greedy search (Most uncertain)

PMCID: PMC4635763

Summary: A 67-year-old woman presented to the emergency department with a sudden collapse and was found to have a cardiac myxoma, a rare tumor of the heart. The patient's initial presentation was characterized by ST-elevation myocardial infarction, but her coronary arteries appeared normal on immediate angiography. She also experienced a stroke, which is relatively common in cardiac myxoma patients. The patient underwent successful surgical resection of the myxoma, and her condition improved. This case highlights the importance of basic echocardiography in diagnosing cardiac myxomas and emphasizes the need for timely surgical intervention to prevent complications.

Top-k sampling (Least uncertain)

PMCID: PMC3297038

Summary: There are similarities between children and adults suffering from heart failure, such as the preferred pharmacological treatment, the use of pacemakers and heart transplants, the inability of the patient to reach the predicted heart rate for the patient's age during cardiopulmonary exercise testing, and the ergoespirometric response under similar clinical conditions. In adults, endothelial dysfunction is related to the development of diastolic dysfunction, Chagas disease, left ventricular hypertrophy, ischemic cardiomyopathy, heart failure, obesity, type 1 diabetes, hyperlipidemia, arterial hypertension, peripheral arterial disease, chronic kidney disease, and atherosclerosis because the

dysfunction predisposes the vasculature to vasoconstriction, leukocyte adherence, platelet activation, and vascular inflammation. Nevertheless, there is a lack of data regarding endothelial function in children with cardiomyopathy. The severity of endothelial dysfunction is related to the cardiovascular risk, the severity of cardiovascular symptoms, and the inability to exercise and represents a predictor for cardiac transplant and death. It is known that diseases, such as Kawasaki's disease, hyperlipidemia, obesity, and type 1 diabetes, play important roles in systemic inflammation and endothelial dysfunction. These diseases may increase the likelihood of cardiovascular events and may predispose children to the development of cardiomyopathy. Based on these considerations, we reviewed the published literature on endothelial function in pre-pubertal children to evaluate the endothelial function in prepub

Top-k sampling (Most uncertain)

PMCID: PMC5290108

Summary: Intra-amniotic inflammation may increase the risk of adverse perinatal outcomes in preterm infants, but this risk is mainly due to low gestational age at birth rather than direct effects of inflammation. Culture-proven intra-amniotic infection does not appear to be a risk factor for adverse perinatal outcomes.

Top-p sampling (Least uncertain)

PMCID: PMC4666010

Summary: this is a retrospective cross - sectional study . there were 4,377 participants , all of whom underwent cardiac ct in a health promotion center at gangnam severance hospital (seoul , korea) between january 2008 and february 2013 . from 4,377 participants , we excluded men (n = 2,584) and premenopausal women (n = 748) . postmenopausal women 50 years or older with no menstrual periods for more than 12 consecutive months and women with elevated follicle - stimulating hormone levels (> 30 iu / l) were included in this study (n = 1,045) . women with diabetes mellitus (n = 59) , excessive alcohol consumption (n = 7) , viral hepatitis (positive results for hepatitis b surface antigen or anti hepatitis c virus ; n = 19) , liver cirrhosis or malignancy on ultrasonography (n = 6) , and self - reported or medically verified history of cvd (n = 35) were excluded from this study . height and weight were measured , and body mass index (bmi) was calculated by dividing weight (kg) by the square of height (m) . lifestyle , personal medical history of acute and

Top-p sampling (Most uncertain)

PMCID: PMC5192323

Summary: The present study demonstrates that plasma CTRP9 levels are independently and positively associated with carotid intima-media thickness in patients with type 2 diabetes without chronic kidney disease. This study further proposes that plasma CTRP9 level is a potential biomarker of atherosclerosis in type 2 diabetes patients without renal complications.

Causal Understanding by LLMs: The Role of Uncertainty

Oscar Lithgow-Serrano[1]*, **Vani Kanjirangat**[1]*, **Alessandro Antonucci**[1]
[1]SUPSI, IDSIA, Switzerland

Abstract

Recent papers show LLMs achieve near-random accuracy in causal relation classification, raising questions about whether such failures arise from limited pretraining exposure or deeper representational gaps. We investigate this under uncertainty-based evaluation, testing whether pretraining exposure to causal examples improves causal understanding using >18K PubMed sentences—half from *The Pile* corpus, half post-2024—across seven models (Pythia-1.4B/7B/12B, GPT-J-6B, Dolly-7B/12B, Qwen-7B). We analyze model behavior through: (i) *causal classification*, where the model identifies causal relationships in text, and (ii) *verbatim memorization probing*, where we assess whether the model prefers previously seen causal statements over their paraphrases. Models perform four-way classification (direct/conditional/correlational/no-relationship) and select between originals and their generated paraphrases. Results show almost identical accuracy on seen/unseen sentences (p>0.05), no memorization bias (24.8% original selection), output distribution over the possible options almost flat — with entropic values near the maximum (1.35/1.39), confirming random guessing. Instruction-tuned models show severe miscalibration (Qwen: >95% confidence, 32.8% accuracy, ECE=0.49). Conditional relations induce highest entropy (+11% vs direct). These findings suggest that failures in causal understanding arise from the lack of structured causal representation, rather than insufficient exposure to causal examples during pretraining.

1 Introduction

Causal understanding from text, intended here as the ability of an LLM to identify whether a text includes a statement about a causal relation between two entities, and which is the causal direction of such a relation, is a critical task for modern natural language understanding. Previous work demonstrates that *Large Language Models* (LLMs) struggle with such causal tasks, achieving near-random performance on benchmarks requiring causal inference (Ashwani et al., 2024; Feng et al., 2024; Guo et al., 2017; Joshi et al., 2024; Kanjirangat et al., 2024). Recent works showed the importance of analyzing the underlying model uncertainty to achieve better results, or at least to understand the reasons for poor performances (Cui et al., 2025; Shorinwa et al., 2025). From this perspective, a very promising direction is provided by distinguishing between different sources of uncertainty, such as *epistemic*, corresponding to the uncertainty related to lack of knowledge about the underlying model, and *aleatoric*, that is the intrinsic ambiguity of the process (Hüllermeier and Waegeman, 2021). Another crucial aspect is how the presence of *seen versus unseen* data — i.e., content observed during pretraining or *familiar* observations — affects uncertainty and model behavior in terms of causal understanding. While uncertainty quantification in LLMs has been explored in prior work (He et al., 2025; Liu et al., 2025; Yadkori et al., 2024), the link between uncertainty sources and familiar causal patterns in the context of causal understanding remains underexamined.

We design controlled experiments to examine how these uncertainty sources arise in the context of causal understanding. We consider *memorization* as one of the tasks to understand the effect of seen *verbatim* causal patterns, in line with the uncertainty sources. Using scientific conclusion sentences from PubMed abstracts, we test whether models trained on these exact texts (via *The Pile* dataset, Gao et al. (2020)) show reduced uncertainty compared to similar but unseen texts and, if this has an impact on their accuracy.

Our approach uses two complementary tests for memorization effects. First, if models truly understood causal patterns from training (not just sur-

*Equal contribution

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 208–228
November 9, 2025 ©2025 Association for Computational Linguistics

face forms), they should exhibit substantially lower uncertainty and higher accuracy on familiar data (i.e., training data) versus unseen data. Second, if only surface memorization occurred, we would expect substantial differences in uncertainty between original sentences and their paraphrases within the training dataset. In a nutshell, we use uncertainty metrics to explore memorization effects in causal understanding and whether this reflects representational limitations rather than data exposure.

Our contributions: We propose uncertainty-based quantification (i.e., entropy, ECE/ACE) as a way for analyzing LLMs' causal understanding competence, linking calibration metrics to performance on causal tasks. We conduct experiments to examine how uncertainty sources, including memorization of verbatim causal statements, influence causal understanding in LLMs. Within this framework, we show: (i) Exposure to training data does not guarantee memorization or improved performance — models show identical accuracy on seen versus unseen texts; (ii) Identify overconfidence in most performing models, with high confidence predictions despite very low accuracy; (iii) Quantify that conditional causal relationships induce the highest uncertainty, suggesting models lack nuanced causal representations; (iv) In the experiments, we consider two datasets, an existing one from the literature and an extension constructed by us to be used for testing memorization and causal understanding.

Our experimental findings indicate that uncertainty in LLM causal understanding reflects epistemic limitations rather than insufficient exposure to training examples. Models do not leverage already observed patterns for causal tasks, instead exhibiting systematic uncertainty that correlates with task complexity rather than data familiarity.

2 Related Work

Recent work examines uncertainty sources in LLMs. Kirchhof et al. (2025) demonstrates that models can assess their uncertainty through verbalized confidence. Giulianelli et al. (2023) proposes semantic entropy to measure uncertainty in free-form generation. However, these approaches focus on general tasks rather than structured reasoning. Wang et al. (2024) shows LLMs struggle with calibrated uncertainty in knowledge-intensive tasks, consistent with our findings in causal reasoning.

Memorization's role in LLM capabilities remains debated. Carlini et al. (2021) demonstrate that models memorize training data verbatim, while (Li et al., 2024; Zhang et al., 2023) show this memorization can be beneficial. Tirumala et al. (2022) quantifies memorization across model scales. Other findings show that memorization alone cannot explain model capabilities, requiring 100+ exact repetitions for reliable recall (Kandpal et al., 2023; Li et al., 2024).

While prior work evaluates causal understanding in LLMs through various benchmarks, none have examined it through the lens of uncertainty sources and verbatim memorization recalls. Our work extends this by showing memorization fails to improve structured reasoning tasks. Our approach uniquely combines controlled exposure to training data with uncertainty quantification, revealing that causal understanding requires more than pattern memorization, especially when the complexity increases.

The rest of the paper is organized as follows. In Section 3, we present the datasets used for the experimental study and analyses. Section 4 presents the detailed discussion of the proposed uncertainty-based quantification, and the experimental setup is presented in Section 5. Results and analysis are reported in Section 6, with detailed discussion in Section 7.

3 Data Construction

First, we used two datasets of sentences labeled with their causal types to test the impact of exposure to causal patterns during pretraining on the accuracy and uncertainty of causal understanding tasks with LLMs.

We used Yu et al. (2019)'s dataset, consisting of 3,061 sentences from science findings classified into four causal relationship types: direct causal, conditional causal, correlational, or no relationship. The original dataset lacked source abstracts for the extracted sentences. As these were needed to train a classifier for extending the dataset with more recent abstracts, we searched PubMed using near-exact sentence matches. This yielded a filtered dataset with the following distribution: direct causal (234), conditional causal (113), correlational (489), and no relationship (598). This resulting dataset is hereafter referred to as *Original*. Although filtering removed almost half of the entries, the label distribution remains highly similar to the original (total variation distance = 0.026). Future work will

consider more sophisticated matching schemes to preserve more data.

To control for verbatim memorization, we created an extension from PubMed abstracts published after 2024, beyond our models' training cutoff. We use a BERT-based classifier (F1-score = 0.97) trained on the original annotations to label 5,400 new sentences, then subsample to match the class distribution of our *Original* dataset. This resulting dataset is hereafter referred to as *Newer*

We complement the datasets by generating, with GPT-4o-mini, for each sentence: (i) five paraphrases especially focused on preserving the causal relationship; (ii) one negation that reverses the causal relationship; (iii) two questions that probe understanding of the causal content. Considering the original plus paraphrase resulted in 18,366 sentences.

Since both datasets are used in *Multiple Choice Question Answering* (MCQA) setups, we hereafter refer to the *original*-based dataset as MCQA and the *newer* one as MCQA-newer. For instance, the samples generated for the Causal Type Classification Task are depicted in Figure 1. Examples from the generated dataset can be found in Appendix A.

4 Uncertainty Quantification

We focus on LLMs answering multiple-choice questions. Let $\mathcal{Y}$ denote the set of possible options and Y the corresponding variable. The probability distribution over the possible choices $P(Y)$ is assumed to be available. We quantify uncertainty through multiple metrics.

Entropy. We can describe the model uncertainty related to this task by the entropy (Shannon, 1948), i.e., $H := -\sum_{y\in\mathcal{Y}} P(y) \ln P(y)$. This is a non-negative function taking the value of zero for deterministic distributions, and its maximum value for uniform distributions. In the case of quaternary variables, the value of the maximum is 1.39.

Calibration. We bin predictions by confidence level and compute actual accuracy within each bin. Perfect calibration yields a diagonal relationship between confidence and accuracy. We then compute *Expected Calibration Error* (ECE) and *Adaptive Calibration Error* (ACE) (Nixon et al., 2019a; Posocco and Bonnefoy, 2021).

ECE measures how well a model's estimated probabilities match the observed probabilities. A perfectly calibrated model has zero ECE. It is computed as the weighted average of the absolute differences between average accuracy and average confidence.

$$\mathrm{ECE} = \sum_{r=1}^{R} \frac{|B_r|}{n} \left|\mathrm{acc}(B_r) - \mathrm{conf}(B_r)\right| \quad (1)$$

Where, R is the number of bins (typically fixed-width over the interval $[0, 1]$), B_r is the set of indices of predictions with confidence scores in the r-th bin, n is the total number of samples, $\mathrm{acc}(B_r) = \frac{1}{|B_r|} \sum_{i\in B_r} \mathbf{1}(\hat{y}_i = y_i)$ is the accuracy in bin r, $\mathrm{conf}(B_r) = \frac{1}{|B_r|} \sum_{i\in B_r} \hat{p}_i$ is the average confidence in bin r, $\hat{p}_i = \max_k p_i^{(k)}$ is the predicted confidence for sample i and k is the number of labels/classes.

To overcome the limitations of ECE, such as the bias-variance trade-off induced by binning approaches and its alignment to binary-class settings (Guo et al., 2017), ACE was proposed, which utilizes flexible binning (Nixon et al., 2019b). ACE is motivated by the bias-variance trade-off, which suggests that an effective estimate of overall calibration error should emphasize regions where predictions are concentrated, while placing less weight on sparsely populated regions. ACE takes as input the predictions *P*, correct labels, and a number of ranges *R*:

$$\mathrm{ACE} = \frac{1}{R} \sum_{r=1}^{R} \left|\mathrm{acc}(B_r) - \mathrm{conf}(B_r)\right| \quad (2)$$

Where, R is the number of bins (adaptively chosen so each bin contains roughly the same number of samples), B_r, $\mathrm{acc}(B_r)$, $\mathrm{conf}(B_r)$, and $\hat{p}_i$ are defined as above.

Consistency. For sentences with multiple paraphrases, we measure whether models make consistent predictions across semantically equivalent inputs.

Statistical Tests. We apply chi-square tests for original versus paraphrase performance, t-tests for dataset comparisons, and ANOVA for differences across causal types.

5 Experimental Design

We specifically select models confirmed to be trained on *The Pile* dataset (Gao et al., 2020; Phang et al., 2022). Since PubMed abstracts used in Yu et al. (2019) are included in *The Pile*, these models necessarily encountered our MCQA sentences

```
{"qa_idx": 0, "context": "However, the small sample size in this study limits its generalizability to diverse populations,
so we call for future research that explores SSL-powered personalization at a larger scale.",

"text": "However, the small sample size in this study limits its generalizability to diverse populations,
so we call for future research that explores SSL-powered personalization at a larger scale.",
"text_type": 0, "causal_class_label": 0,

"choices": [{"label": 1, "text": "Direct Causal", "description": "The statement explicitly states that one variable directly causes changes in another."},
{"label": 3, "text": "Correlational", "description": "The statement describes an association between variables, but no causation is
explicitly stated."},
{"label": 2, "text": "Conditional Causal", "description": "The statement suggests causation but includes uncertainty through hedging words or
modal expressions."},
{"label": 0, "text": "No Relationship", "description": "No correlation or causation relationship is mentioned."},
{"label": 4, "text": "Other", "description": ""}]
```


Figure 1: Examples from the constructed data (Task 1) - Casual Type Classification

during pretraining. As a control, we also include a model not trained on *The Pile*, allowing us to distinguish data exposure effects from other confounders.

In total, we evaluate seven models spanning different architectures, training data, and training approaches. Base pretrained models include pythia variants (1.4B, 7B, 12B parameters) and gpt-j-6b, all trained on *The Pile* without instruction tuning. Instruction-tuned models include dolly-v2 variants (7B, 12B), which use pythia as base models but undergo additional instruction tuning, and qwen-7b-base, an instruction-tuned model not trained on *The Pile* [1]. This selection allows us to isolate the effects of: (i) exposure to training data (*The Pile*), (ii) model scale, and (iii) instruction tuning on causal understanding and uncertainty.

To investigate the link between uncertainty sources and familiar causal patterns in the context of causal understanding, we focus on two complementary tasks.

Task 1: Causal Type Classification. Given a sentence $s \in \mathcal{S}$, where $\mathcal{S}$ represents the set of scientific conclusion sentences, the model M must classify the causal relationship into one of four predefined classes: $\mathcal{Y} = \{\text{causal, conditional causal, correlational, no relationship}\}$. Formally, the model applies a mapping function:

$$f_M : \mathcal{S} \to \mathcal{Y}\,, \tag{3}$$

returning the predicted class $\hat{y} := f_M(s)$ for the sentence s. In particular, we are interested in probabilistic models, returning a probability distribution over the four causal types. In these cases, the model's prediction is:

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} P_M(y|s) \tag{4}$$

[1]There is also no explicit claim of biomedical literature in the training data, but exposure to PubMed abstracts through other sources cannot be completely ruled out.

This directly tests causal understanding with a random baseline of $\frac{1}{|\mathcal{Y}|} = 25\%$.

Task 2: Verbatim Memorization Probing. Following Duarte et al. (2024)'s hypothesis that models preferentially select exact text included in their training data (verbatim recall), we test for memorization bias in the context of seen-versus-unseen text. Given a question q derived from an original sentence $s_0 \in \mathcal{X}_{Original \cup Newer}$ and a set of semantically equivalent paraphrases $\mathcal{P} = \{s_0, s_1, \ldots, s_n\}$ where $\text{meaning}(s_i) = \text{meaning}(s_0)$ for all i, where meaning(.) indicates the semantics of the sentence, the model must select the most appropriate answer:

$$\hat{s} = \arg\max_{s_i \in \mathcal{P}} P_M(s_i|q) \tag{5}$$

Under the memorization hypothesis, we expect:

$$P_M(s_0|q) > P_M(s_i|q) \quad \forall i \in \{1, \ldots, n\} \tag{6}$$

when s_0 was seen during training. In contrast, without memorization bias, we expect uniform selection probability: $P_M(s_i|q) \approx \frac{1}{|\mathcal{P}|}$ for all i.

To mitigate documented selection biases (e.g., positional bias) in multiple-choice questions, we randomize the order of answer options for each question. All models are self-hosted and queried via VLLM (Kwon et al., 2023) API with temperature 0.0 for deterministic outputs. For each prediction, we extract: (i) the selected choice, (ii) the probability distribution over choices, (iii) the corresponding entropy, and (iv) the maximum probability (confidence).

Memorization Assumptions. We acknowledge that presence in *The Pile* does not guarantee memorization. Previous work (Carlini et al., 2023; Kandpal et al., 2023) shows reliable memorization requires 100+ exact repetitions during training. Our design tests whether exposure (seen patterns) — even without guaranteed memorization—provides any advantage for causal reasoning.

6 Results and Analysis

6.1 Pretraining-observance Does Not Reduce Uncertainty

Models show no performance advantage on training data. Across all models trained on *The Pile*, accuracy differs by <1.5% between MCQA and MCQA-newer (Table 1). All Pile-trained models with high entropy (>1.3) perform near random chance (25%) but with appropriate uncertainty. Statistical significance tests confirm these observations (Table 2 and in Appendix B.2). T-tests comparing original versus paraphrase performance yield p-values > 0.05 for all models, with negligible effect sizes (Cohen's d < 0.2). Dataset comparisons (MCQA vs MCQA-newer) show similar results with small effect sizes (|d| < 0.2), indicating no systematic advantage on the data observed in pretraining, suggesting that the presence of an example in the pre-training corpus does not reliably lead to verbatim recall or systematic memorization that could benefit accuracy on the task.

Figure 2 reveals that entropy remains consistently high across both datasets. Pythia and GPT-J models exhibit entropy near maximum (1.3-1.35), indicating near-random guessing regardless of data familiarity. Only qwen-7b-base, not trained on *The Pile*, achieves lower entropy (0.29), suggesting better causal understanding is likely attributable to model architecture rather than mere training data exposure.

To better understand these patterns, we also conducted statistical tests on entropy measures in Table 2 (see also B.2 in the appendix). The results reveal a striking divergence: while accuracy remains stable across original and paraphrased sentences, entropy patterns differ significantly ($p < 0.001$ for pythia and dolly models). This divergence indicates that models exhibit *different types of uncertainty* for *familiar* versus novel phrasings, even when performing equally poorly. Specifically, pythia models show higher entropy (more uniform distributions) on paraphrases, suggesting they become *more uncertain* when surface forms change. This pattern persists despite no accuracy improvement on original sentences, providing strong evidence against functional memorization. The MCQA vs MCQA-newer comparisons support this: entropy differences are significant for several models (pythia-7b, gpt-j-6b, dolly models) while accuracy remains constant.

Models develop different uncertainty profiles for familiar versus unfamiliar datasets without corresponding performance benefits. Only qwen-7b-base approaches significance for accuracy on original vs paraphrase ($p = 0.082$), suggesting instruction tuning may introduce subtle biases toward familiar phrasings. However, the effect size remains negligible ($d = 0.023$). ANOVA tests on the entropy measures reveal also significant differences across causal types ($p < 0.001$), confirming that uncertainty patterns reflect task complexity rather than data familiarity (details of the probability and entropy assignments are shown in Appendix B.1.

6.2 Two Distinct Uncertainty Profiles

Figure 3 reveals two distinct calibration patterns: base models maintain appropriate uncertainty despite 25% accuracy, while instruction-tuned models exhibit overconfidence.

The distribution of prediction confidence (see Appendix B.4) reveals two distinct patterns. Pile-trained models (pythia, gpt-j) consistently assign low confidence to their predictions, with probability distributions peaked around 0.30-0.35—appropriately uncertain given their near-random accuracy. These models maintain ECE < 0.16, indicating well-calibrated uncertainty.

In contrast, qwen-7b-base exhibits overconfidence, with most predictions assigned >95% probability despite achieving only 32.8% accuracy. This confidence-accuracy gap yields ECE=0.49 versus 0.13 for base models—a 3.8x increase in calibration error. Dolly models, with accuracies between 21% and 23%, show bimodal confidence distributions but with very different calibration error patterns depending on the model size, i.e., the 7B version has an ECE of 0.36 and ACE of 0.23, whereas 12B although less accurate than the 7B, presents the highest calibration error of all models (ECE=0.56, ACE=0.31).

These observations suggest that in causal understanding tasks, partial competence breeds false confidence. Models performing near random maintain appropriate low confidence, while the best-performing model develops overconfidence—an especially concerning trait in applications that demand reliable causal inference.

6.3 Instruction Tuning Creates Overconfidence

Our model selection reveals an important factor: instruction tuning fundamentally alters uncertainty behavior in causal understanding. Base pretrained

Model	Accuracy (Overall)	Entropy (Mean±SD)	ECE	ACE	Accuracy Original	Accuracy Paraphrase	Δ (O-P)
pythia-1.4b	0.248	1.34±0.05	0.067	0.136	0.245	0.249	-0.004
pythia-7b	0.251	1.32±0.06	0.131	0.142	0.249	0.252	-0.003
pythia-12b	0.231	1.32±0.07	0.149	0.151	0.231	0.231	0.000
gpt-j-6b	0.175	1.35±0.04	0.151	0.158	0.169	0.177	-0.008
dolly-v2-7b	0.240	0.91±0.19	0.363	0.239	0.240	0.240	0.000
dolly-v2-12b	0.212	0.53±0.28	0.564	0.312	0.202	0.215	-0.013
qwen-7b-base	0.328	0.29±0.35	0.493	0.275	0.339	0.326	0.013

Table 1: Summary statistics for causal type classification task. ECE (lower is better). ACE (lower is better). Worst-performing models show best calibration, while better-performing models exhibit overconfidence.

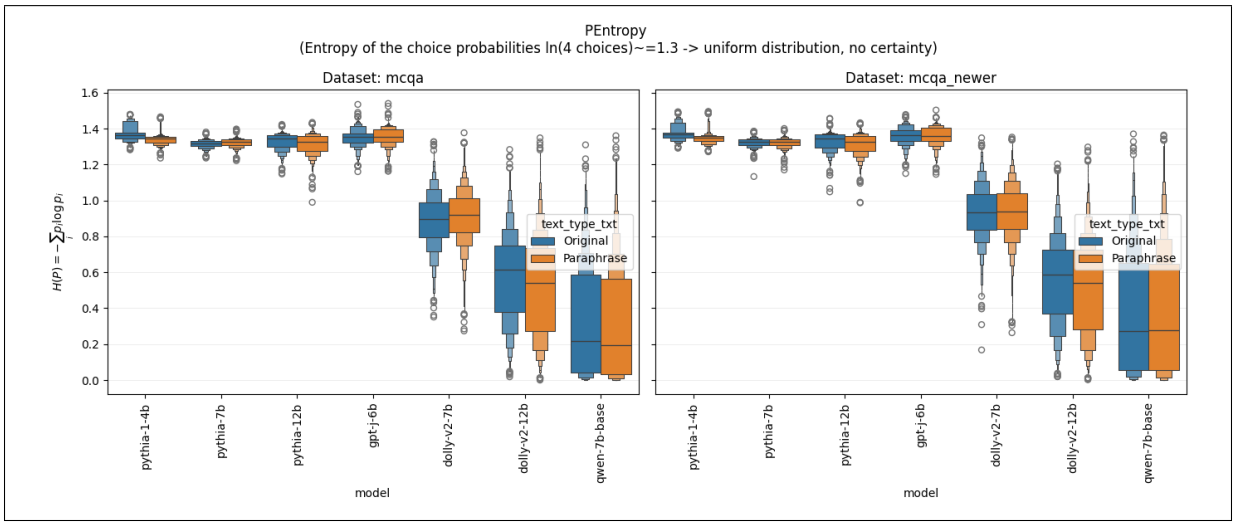


Figure 2: **Entropy distributions across models.** Higher values indicate greater uncertainty (max=1.39 for random guessing). Qwen-7b shows low entropy (high confidence) while Pythia/GPT-J show near-maximum entropy.

	Accuracy		Entropy	
Model	OvP	MvN	OvP	MvN
pythia-1.4b	0.744	0.604	**<0.001**	0.573
pythia-7b	0.859	0.708	**<0.001**	**<0.001**
pythia-12b	0.823	0.506	**<0.001**	0.513
gpt-j-6b	0.670	0.655	0.787	**<0.001**
dolly-v2-7b	0.625	0.589	**<0.001**	**<0.001**
dolly-v2-12b	0.525	**0.001**	**<0.001**	**0.004**
qwen-7b-base	**0.082**	0.672	0.123	**<0.001**

Table 2: Statistical significance tests (p-values). OvP: Original vs Paraphrase; MvN: MCQA vs MCQA-Newer. Bold indicates $p < 0.05$. Note the divergence between accuracy and entropy tests, revealing that uncertainty patterns differ from performance patterns.

models (pythia variants, gpt-j-6b) exhibit high entropy (≈1.35) with appropriately low confidence (30-35%), yielding good calibration despite poor performance.

However, instruction-tuned models show different patterns. Dolly models—fine-tuned from pythia bases on instruction-following data—develop moderate confidence (40-60%) without corresponding accuracy improvements. Most markedly, qwen-7b-base exhibits overconfidence (>95%) while achieving only marginally better accuracy (32.8%).

This divergence offers valuable insights into dolly models: identical pretrained weights (pythia) produce different uncertainty profiles after instruction tuning. Compare pythia-7b (ECE=0.13, entropy=1.32) with dolly-v2-7b (ECE=0.36, entropy=0.92)—instruction tuning does not even half the entropy while almost tripling the calibration error.

These results suggest instruction tuning teaches models to be confident in their responses, even when this confidence is unjustified. While this may improve user experience in conversational settings, it creates problematic overconfidence in domains requiring accurate uncertainty quantification, such as causal-language related tasks.

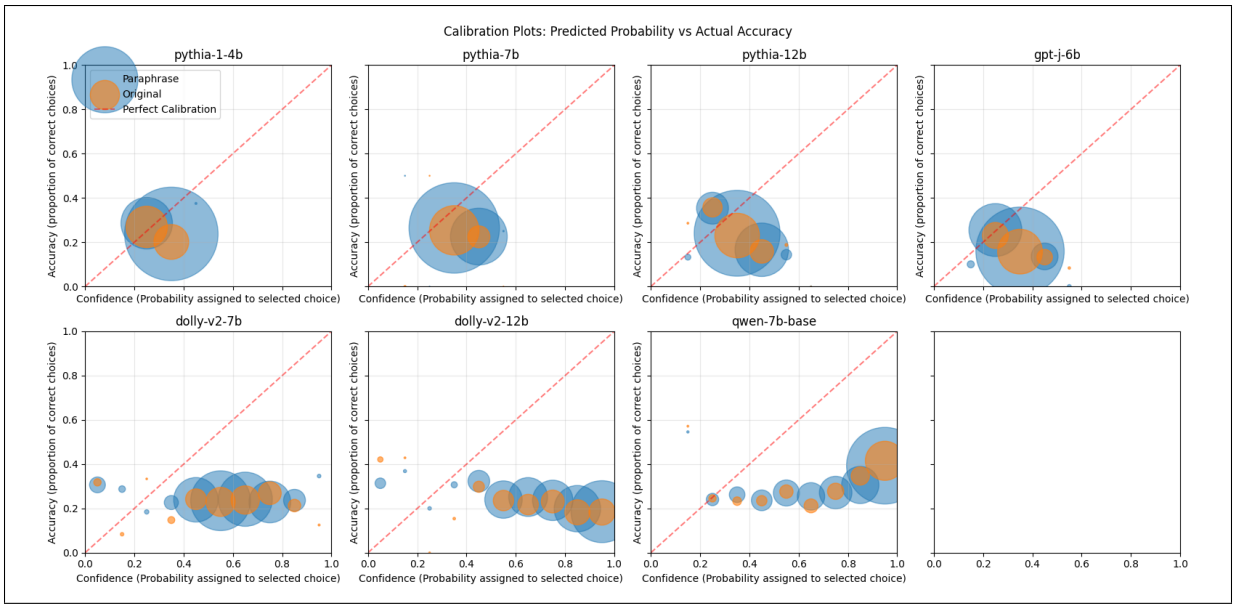


Figure 3: **Calibration plots, confidence vs accuracy.** Perfect calibration follows the diagonal. Base models (Pythia/GPT-J) show good calibration despite poor accuracy. Instruction-tuned models (Qwen/Dolly) show overconfidence.

6.4 Uncertainty Varies by Causal Complexity

Uncertainty patterns differ across causal relationship types. Conditional causal statements induce the highest entropy across all models, while direct causal relationships show moderately lower uncertainty. Correlational statements, despite being non-causal, often receive more confident predictions than conditional causal ones.

ANOVA confirms significant differences in entropy across causal types ($p < 0.001$ for all models). This pattern persists across both original and paraphrase sentences, indicating that uncertainty arises from conceptual difficulty rather than surface-level confusion.

Causal Type	Mean Entropy	Accuracy
Direct Causal	1.15±0.42	0.31
Conditional Causal	1.28±0.29	0.19
Correlational	1.09±0.45	0.26
No Relationship	1.12±0.43	0.22

Table 3: Performance breakdown by causal relationship type (averaged across all models). Conditional causal statements show highest entropy and lowest accuracy.

Table 3 quantifies this pattern: conditional causal statements exhibit 11% higher entropy than direct causal statements, approaching maximum entropy. This suggests that models struggle particularly with nuanced causal identifications involving conditions or moderating factors. Detailed analysis can be found in Appendix B.5.

6.5 Identifying Inherently Ambiguous Questions

We analyzed the results based on the intuition that if all paraphrases of a sentence get similar wrong predictions, it might indicate inherent ambiguity (aleatoric uncertainty).

This paraphrase consistency analysis reveals a subset of questions where all models consistently select the same incorrect answer across paraphrases (i.e., consistency > 0.7 and accuracy < 0.3). These represent 60-75% of questions depending on the model (Figure 4). High consistency on wrong answers suggests inherent ambiguity in the task rather than model-specific confusion.

A manual analysis of misclassified instances reveals recurring linguistic patterns, including the use of hedging expressions (e.g., "may influence," "suggests association") and complex multi-clause constructions. These features are associated with misclassification across model families.

6.6 No Evidence on Verbatim Recalls with Pre-training Exposures

The verbatim memorization probing (task 2) provides complementary evidence. When presented with questions about causal relationships and asked to choose between original sentences and paraphrases (all semantically correct), models show no preference for the original form. Selection rates for original sentences average 24.8% (95% CI: [24.2%, 25.4%]) across all Pile-trained models, statistically

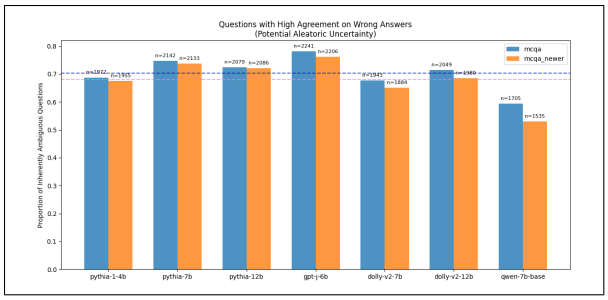


Figure 4: **Ambiguous questions** Proportion of questions with consistent wrong answers across paraphrases, indicating inherent ambiguity.

indistinguishable from the 25% random baseline (chi-square test, $p > 0.05$).

Entropy analysis of this task reveals similar patterns: models exhibit high uncertainty (mean entropy 1.31) regardless of whether choosing between already seen or potentially novel phrasings. This task design, which controls for correctness while varying surface form, shows that models do not leverage already seen patterns even when given the opportunity.

This null result holds even when examining only high-frequency scientific phrases likely repeated many times in training data. Models treat original and paraphrase sentences identically, confirming that surface-level causal patterns exposed during pretraining may not contribute to causal understanding (details of probabilities and entropy analysis is in Appendix C).

6.7 Consistency Across Semantic Paraphrases

To further test verbatim recall, we analyze prediction consistency across paraphrases of the same sentence. If models rely on "seen/recalled" patterns, we expect higher consistency for original sentences versus their paraphrases.

For each sentence with multiple probes, we computed Pearson correlation between choice probability vectors to assess confidence consistency, Spearman rank correlation to evaluate preference consistency by comparing ranked orders, and KL divergence to quantify overall distributional shifts between probabilistic outputs (see Appendix B.3 for details).

Only pretrained models exhibit moderate KL divergence ($KL < 0.15$), suggesting relatively stable probabilistic behavior across input variations. However, larger and instruction-tuned models such as qwen-7b-base and dolly models exhibit higher KL divergence (>0.5 for qwen and >1.0 for dolly models), indicating greater sensitivity to paraphrasing and less consistency in their output distributions, i.e., unstable predictions across semantically equivalent inputs. This pattern holds across both the MCQA and MCQA-newer datasets.

Correlation analysis of probability vectors and ranked order reveals weak consistency (mean Pearson and Spearman $\rho < 0.2$) across all models but qwen-7b-base (Pearson $\rho > 0.71$ and Spearman $\rho > 0.73$).

Critically, consistency patterns do not differ between MCQA (training data) and MCQA-newer (unseen data). The absence of improved consistency on training data provides evidence against memorization-based reasoning. Models treat semantically equivalent sentences as unrelated inputs, confirming they lack robust causal understanding rather than merely forgetting training examples.

7 Discussion

Our experimental findings show that uncertainty in causal tasks stems from absent causal understanding rather than insufficient data exposure (Figure 5). Three key insights emerge: **First**, the independence of performance from training data exposure suggests that simply scaling datasets will not resolve causal understanding deficits. Models require architectural innovations or training objectives that explicitly target causal inference. **Second**, overconfidence from instruction tuning poses deployment risks. The shift from calibrated uncertainty (base models: ECE=0.13) to overconfidence (instruction-tuned: ECE=0.49) indicates fine-tuning teaches models to suppress appropriate uncertainty. **Third**, the particular difficulty with conditional causal relationships indicates that models lack compositional reasoning about causality. While they may recognize simple cause-and-effect patterns, they fail when conditions, moderators, or exceptions are introduced.

These results suggest that causal understanding in LLMs requires fundamental advances beyond current pretraining paradigms. Memorization, even at scale, cannot substitute for genuine causal knowledge.

8 Conclusion

In this work, we focused on analyzing the critical limitations in the causal understanding abilities of large language models (LLMs). Through a controlled evaluation combining causal clas-

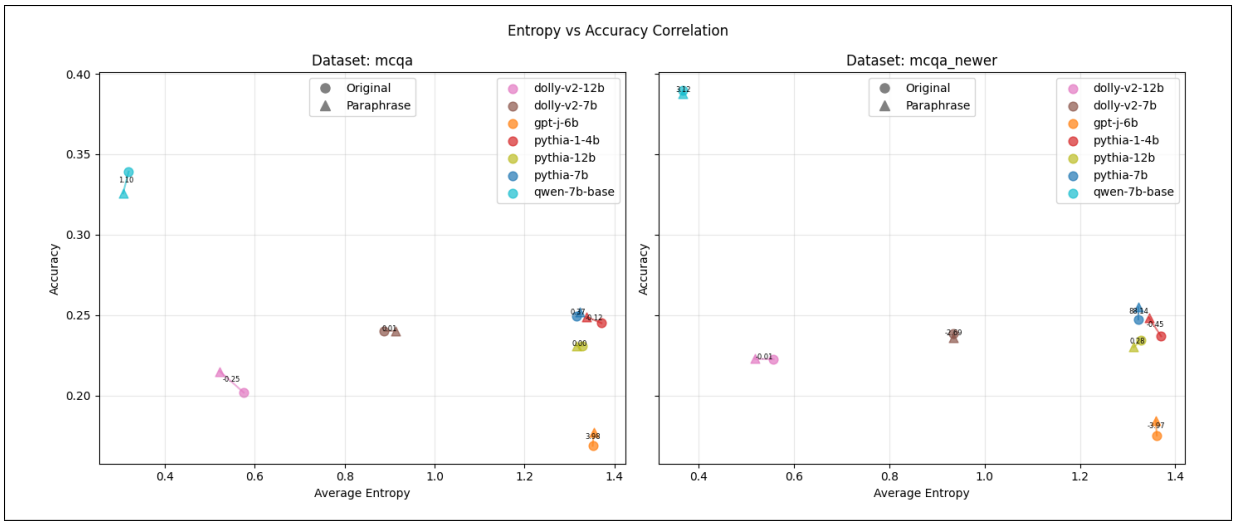


Figure 5: **Entropy vs Accuracy.** Ideal models appear top-left (accurate and confident). Most models cluster bottom-right (inaccurate and uncertain).

sification and verbatim memorization probing, we demonstrate that exposure to causal content during pretraining does not guarantee accurate recall or improved causal understanding. Our analysis, leveraging multiple uncertainty metrics—including entropy, consistency, calibration, and accuracy—reveals that uncertainty in causal tasks stems primarily from deficits in causal understanding rather than limitations in memorization. Addressing these limitations will require a shift beyond current pretraining paradigms—toward models that explicitly encode and reason over causal structures, and that are capable of expressing calibrated uncertainty when faced with ambiguity or unseen conditions.

Limitations

Our study has three key limitations. First, we cannot determine whether the models truly failed to acquire causal patterns during training, or whether they learned them but are unable to apply or recall them during inference. Structured prompting, Causal probing with small datasets, pre-trained data inspections through sampling, probing representations, etc., can be possible approaches to tackle this problem. Second, presence in *The Pile* does not guarantee memorization—research shows reliable memorization requires 100+ repetitions (Kandpal et al., 2023). Our results demonstrate that even exposure without guaranteed memorization provides no benefit for causal reasoning. Third, our binary classification of "seen" versus "unseen" may oversimplify the memorization spectrum. Future work should examine the relationship between repetition frequency and causal understanding.

Acknowledgments

This work has been supported by UBS Switzerland AG and its affiliates.

References

Swagata Ashwani, Kshiteesh Hegde, Nishith Reddy Mannuru, Mayank Jindal, Dushyant Singh Sengar, Krishna Chaitanya Rao Kathala, Dishant Banga, Vinija Jain, and Aman Chadha. 2024. Cause and Effect: Can Large Language Models Truly Understand Causality? *Preprint*, arXiv:2402.18139.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying Memorization Across Neural Language Models. *Preprint*, arXiv:2202.07646.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting Training Data from Large Language Models. *Preprint*, arXiv:2012.07805.

Shaobo Cui, Luca Mouchel, and Boi Faltings. 2025. Uncertainty in Causality: A New Frontier. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8022–8044. Association for Computational Linguistics.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. DE-COP: Detecting Copyrighted Content in Language Models Training Data. *Preprint*, arXiv:2402.09910.

Tao Feng, Lizhen Qu, Niket Tandon, Zhuang Li, Xiaoxi Kang, and Gholamreza Haffari. 2024. From Pretraining Corpora to Large Language Models: What

Factors Influence LLM Performance in Causal Discovery Tasks? *Preprint*, arXiv:2407.19638.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The Pile: An 800GB Dataset of Diverse Text for Language Modeling. *Preprint*, arXiv:2101.00027.

Mario Giulianelli, Joris Baan, Wilker Aziz, Raquel Fernández, and Barbara Plank. 2023. What Comes Next? Evaluating Uncertainty in Neural Text Generators Against Human Production Variability. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14349–14371. Association for Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On Calibration of Modern Neural Networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1321–1330. PMLR.

Jianfeng He, Virginia Tech, Linlin Yu, Changbin Li, Runing Yang, Virginia Tech, Fanglan Chen, Virginia Tech, Kangshuo Li, Min Zhang, Virginia Tech, Shuo Lei, Virginia Tech, Xuchao Zhang, Mohammad Beigi, Kaize Ding, Bei Xiao, Lifu Huang, Feng Chen, and 4 others. 2025. Survey of Uncertainty Estimation in LLMs - Sources, Methods, Applications, and Challenge. 1(1).

Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and Epistemic Uncertainty in Machine Learning: An Introduction to Concepts and Methods. 110(3):457–506.

Nitish Joshi, Abulhair Saparov, Yixin Wang, and He He. 2024. LLMs Are Prone to Fallacies in Causal Inference. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10553–10569. Association for Computational Linguistics.

Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. 2023. Large Language Models Struggle to Learn Long-Tail Knowledge. *Preprint*, arXiv:2211.08411.

Vani Kanjirangat, Alessandro Antonucci, and Marco Zaffalon. 2024. On the Limitations of Zero-Shot Classification of Causal Relations by LLMs.

Michael Kirchhof, Luca Füger, Adam Goliński, Eeshan Gunesh Dhekane, Arno Blaas, and Sinead Williamson. 2025. Self-reflective Uncertainties: Do LLMs Know Their Internal Answer Distribution? *Preprint*, arXiv:2505.20295.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Bo Li, Qinghua Zhao, and Lijie Wen. 2024. ROME: Memorization Insights from Text, Logits and Representation. *Preprint*, arXiv:2403.00510.

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty Quantification and Confidence Calibration in Large Language Models: A Survey. *Preprint*, arXiv:2503.15850.

Jeremy Nixon, Michael W Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019a. Measuring calibration in deep learning. In *CVPR workshops*, volume 2.

Jeremy Nixon, Michael W. Dusenberry, Linchuan Zhang, Ghassen Jerfel, and Dustin Tran. 2019b. Measuring calibration in deep learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, volume 2, pages 38–41. IEEE / Computer Vision Foundation.

Jason Phang, Herbie Bradley, Leo Gao, Louis Castricato, and Stella Biderman. 2022. Eleutherai: Going beyond "open science" to "science in the open". In *Workshop on Broadening Research Collaborations in ML, NeurIPS 2022*. NeurIPS Workshop.

Nicolas Posocco and Antoine Bonnefoy. 2021. Estimating expected calibration errors. In *International conference on artificial neural networks*, pages 139–150. Springer.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Ola Shorinwa, Zhiting Mei, Justin Lidard, Allen Z. Ren, and Anirudha Majumdar. 2025. A Survey on Uncertainty Quantification of Large Language Models: Taxonomy, Open Research Challenges, and Future Directions. *Preprint*, arXiv:2412.05563.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models. *Preprint*, arXiv:2205.10770.

Zhiyuan Wang, Jinhao Duan, Lu Cheng, Yue Zhang, Qingni Wang, Xiaoshuang Shi, Kaidi Xu, Heng Tao Shen, and Xiaofeng Zhu. 2024. ConU: Conformal Uncertainty in Large Language Models with Correctness Coverage Guarantees. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6886–6898. Association for Computational Linguistics.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. 2024. To Believe or Not to Believe Your LLM. *Preprint*, arXiv:2406.02543.

Bei Yu, Yingya Li, and Jun Wang. 2019. Detecting causal language use in science findings. In *Proceedings of the 2019 Conference on Empirical Methods*

in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 4664–4674. Association for Computational Linguistics.

Chiyuan Zhang, Daphne Ippolito, Katherine Lee, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Counterfactual Memorization in Neural Language Models. *Preprint*, arXiv:2112.12938.

A Dataset & Prompts

The examples from the constructed dataset are shown in Figures 6 and 7.

Dataset construction prompt template: *"Paraphrase the following sentence while preserving its exact meaning, especially the causal relationship. Change the wording and structure but keep the scientific accuracy: [sentence]"*

B Task 1: Causal Type Classification.

B.1 Probabilities analysis

From Figure 8, we analyze the probabilities related to the correct choices, assigned by different models in MCQA and MCQA_newer. In Figure 9, we depict the probability assignment to selected choices, comparing the original and paraphrased sentences. Figure 10, shows the entropy of the choice probabilities.

The trends appear similar when comparing the two versions of the dataset, with no evidence of verbatim recalls/ memorized patterns that facilitate better causal understanding. Random behavior on "memorized" data - High entropy on MCQA shows no memorization benefit. A perfect inverse relationship with performance is noted, where pythia/gpt-j/gpt-6b, presents high entropy ($\approx 1.3 - 1.4$), indicating nearly uniform distributions, which implies no causal understanding. qwen-7b-base, on the other hand presents a low entropy ($\approx 0.2 - 0.6$) indicating confident, decisive predictions. Entropy near $ln(4) \approx 1.39$ for weak models confirms they are essentially guessing randomly. No paraphrase penalty: Original vs Paraphrase performance presents nearly identical behaviours.

B.2 Statistical significance tests

We provide statistical tests examining differences in accuracy and entropy between original versus paraphrased sentences, and between MCQA versus MCQAnewer datasets. Ttests assess mean differences while ANOVA examines variance across causal relationship types. The statistical test results are reported in Tables 4 and 5. These were computed on the accuracies and entropies of models, respectively.

B.3 Consistency analysis

For each sentence with multiple probes, we computed Pearson correlation between choice probability vectors to assess confidence consistency, Spearman rank correlation to evaluate preference consistency by comparing ranked orders, and KL divergence to quantify overall distributional shifts between probabilistic outputs. Tables 6 and Figures 13, 12 and 14. High consistency would indicate robust causal understanding, while low consistency suggests models treat paraphrases as unrelated inputs.

———-

B.4 Overconfidence Analysis

We present calibration analyses (Figures 15-21) for each model, examining the relationship between predicted confidence and actual accuracy. Each model's analysis includes accuracy breakdowns by causal type, confidence distributions, calibration plots, and confusion matrices for high-confidence errors. These reveal overconfidence patterns, particularly in instruction-tuned models.

B.5 Uncertainty Analysis

This section examines (Figures 22 - 28) how uncertainty (measured by entropy) varies across different causal relationship types and its correlation with model accuracy. The analysis reveals that conditional causal relationships consistently induce the highest uncertainty across all models, suggesting limitations in compositional causal reasoning rather than simple memorization effects.

C Task 2: Verbatim Memorization Probing

We analyze results from the memorization probing task, where models choose between original sentences and semantically equivalent paraphrases. The probability (Figures 29 and 30) and entropy (Figure 31) analyses demonstrate no preference for original (potentially memorized) text over paraphrases, providing direct evidence against verbatim memorization as a driver of causal reasoning performance.

```
{"qa_idx": 0, "context": "However, the small sample size in this study limits its generalizability to diverse populations,
so we call for future research that explores SSL-powered personalization at a larger scale.",

"text": "However, the small sample size in this study limits its generalizability to diverse populations,
so we call for future research that explores SSL-powered personalization at a larger scale.",
"text_type": 0, "causal_class_label": 0,

"choices": [{"label": 1, "text": "Direct Causal", "description": "The statement explicitly states that one variable directly causes changes in another."},
{"label": 3, "text": "Correlational", "description": "The statement describes an association between variables, but no causation is
explicitly stated."},
{"label": 2, "text": "Conditional Causal", "description": "The statement suggests causation but includes uncertainty through hedging words or
modal expressions."},
{"label": 0, "text": "No Relationship", "description": "No correlation or causation relationship is mentioned."},
{"label": 4, "text": "Other", "description": ""}]
```


Figure 6: Examples from the constructed data (Task 1) - Casual Type Classification


```
Example non-causal: {"context": "Faster aspart and IAsp were confirmed noninferior in a basal-bolus regimen regarding change from baseline in HbA1c.",

"question": "What was the outcome of comparing faster aspart and IAsp in terms of their effect on HbA1c levels in a basal-bolus regimen?  ",

"choices": [{"text": "Faster aspart and IAsp were shown to be noninferior in a basal-bolus regimen with respect to the change in HbA1c from the starting point.  ",
"type": 1},
{"text": "Faster aspart and IAsp were confirmed noninferior in a basal-bolus regimen regarding change from baseline in HbA1c.", "type": 0},
{"text": "Faster aspart and IAsp were validated as noninferior in a basal-bolus treatment concerning the change in HbA1c from baseline.  ", "type": 1},
{"text": "Faster aspart and IAsp were not confirmed noninferior in a basal-bolus regimen regarding change from baseline in HbA1c.", "type": 2},
{"text": "I don't know", "type": 3}],
"true_sent_type": 0, "causal_class_label": 0}


Example causal: {"context": "Vildagliptin effectively improved glucose level with a significantly greater reduction in glycemic variability and hypoglycemia
than glimepiride in patients with T2DM ongoing metformin therapy.",
"question": "What was the effect of vildagliptin compared to glimepiride on glucose levels in patients with T2DM?  ",
"choices": [{"text": "Vildagliptin significantly enhanced glucose levels, showing a much larger decrease in glycemic variability
and hypoglycemia compared to glimepiride in patients with T2DM who were already on metformin treatment.  ", "type": 1},
{"text": "Vildagliptin effectively improved glucose level with a significantly greater reduction in glycemic variability
and hypoglycemia than glimepiride in patients with T2DM ongoing metformin therapy.", "type": 0},
{"text": "Vildagliptin did not effectively improve glucose level with a significantly greater reduction in glycemic variability
and hypoglycemia than glimepiride in patients with T2DM ongoing metformin therapy.", "type": 2},
{"text": "I don't know", "type": 3},
{"text": "In patients with T2DM receiving ongoing metformin therapy, vildagliptin led to a notable improvement in glucose levels,
with a significantly greater reduction in glycemic variability
and hypoglycemia than glimepiride.  ", "type": 1}],
"true_sent_type": 0, "causal_class_label": 1,}
```


Figure 7: Examples from the constructed data (Task 2) - Example with the configuration: original (type 0), two paraphrases (type 1), one negation (type 2) and I don't know (type 3), with and without causal labels

	Original vs Paraphrase		MCQA vs MCQA-newer	
Model	t-stat	p-value	t-stat	p-value
pythia-1-4b	0.340	0.744	0.577	0.604
pythia-7b	-0.184	0.859	0.412	0.708
pythia-12b	-0.232	0.823	-0.753	0.506
gpt-j-6b	-0.444	0.670	0.494	0.655
dolly-v2-7b	-0.512	0.625	-0.604	0.589
dolly-v2-12b	-0.669	0.525	-12.427	0.001
qwen-7b-base	2.032	0.082	-0.468	0.672

Table 4: T-tests computed on the accuracies for each model between Original and Paraphrase indistinguishable of the dateset and, between the original sentences of MCQA and the original sentences of MCQA-newer. Accuracies feed to the t-tests were the means of the binary correctness grouped by the causal relationship type.

	Original vs Paraphrase			MCQA vs MCQA-newer			Causal-type ANOVA	
Model	t-stat	p-value	effect size	t-stat	p-value	effect size	f-stat	p-value
pythia-1-4b	69.273	0.000	0.884	0.563	0.573	0.015	15.509	0.000
pythia-7b	-10.205	0.000	-0.152	-11.637	0.000	-0.307	25.805	0.000
pythia-12b	18.441	0.000	0.286	0.655	0.513	0.017	8.400	0.000
gpt-j-6b	0.270	0.787	0.004	-7.847	0.000	-0.207	139.133	0.000
dolly-v2-7b	-5.656	0.000	-0.083	-11.328	0.000	-0.301	40.102	0.000
dolly-v2-12b	11.673	0.000	0.181	2.846	0.004	0.076	25.551	0.000
qwen-7b-base	1.541	0.123	0.023	-5.679	0.000	-0.150	153.846	0.000

Table 5: Statistical tests computed on the entropies for each model between Original and Paraphrase indistinguishable of the dateset and, between the original sentences of MCQA and the original sentences of MCQA-newer.

model	dataset	text type	prob correlation	Spearman	KL div
pythia-1-4b	mcqa	Paraphrase	-0.007	0.003	0.105
pythia-1-4b	mcqa_newer	Paraphrase	-0.020	-0.009	0.093
pythia-1-4b	mcqa_newer	Original	-0.009	0.003	0.058
pythia-1-4b	mcqa	Original	0.002	0.025	0.066
pythia-7b	mcqa_newer	Paraphrase	0.003	-0.001	0.127
pythia-7b	mcqa_newer	Original	0.002	0.000	0.130
pythia-7b	mcqa	Paraphrase	-0.009	-0.014	0.129
pythia-7b	mcqa	Original	-0.017	-0.026	0.147
pythia-12b	mcqa	Paraphrase	0.019	-0.001	0.137
pythia-12b	mcqa	Original	0.025	0.011	0.105
pythia-12b	mcqa_newer	Paraphrase	0.016	-0.005	0.143
pythia-12b	mcqa_newer	Original	0.024	0.005	0.108
gpt-j-6b	mcqa_newer	Paraphrase	0.123	0.113	0.087
gpt-j-6b	mcqa	Original	0.156	0.119	0.078
gpt-j-6b	mcqa_newer	Original	0.183	0.160	0.066
gpt-j-6b	mcqa	Paraphrase	0.109	0.098	0.093
dolly-v2-7b	mcqa_newer	Paraphrase	0.059	0.040	1.049
dolly-v2-7b	mcqa_newer	Original	0.054	0.045	1.045
dolly-v2-7b	mcqa	Paraphrase	0.018	0.017	1.141
dolly-v2-7b	mcqa	Original	0.029	0.026	1.186
dolly-v2-12b	mcqa	Original	0.050	0.006	2.240
dolly-v2-12b	mcqa_newer	Paraphrase	0.053	0.026	2.475
dolly-v2-12b	mcqa_newer	Original	0.056	0.026	2.430
dolly-v2-12b	mcqa	Paraphrase	0.045	0.015	2.376
qwen-7b-base	mcqa	Original	0.777	0.775	0.494
qwen-7b-base	mcqa	Paraphrase	0.762	0.769	0.527
qwen-7b-base	mcqa_newer	Original	0.727	0.739	0.547
qwen-7b-base	mcqa_newer	Paraphrase	0.710	0.727	0.602

Table 6: Consistency analysis over predictions aggregated by model, dataset and text type.

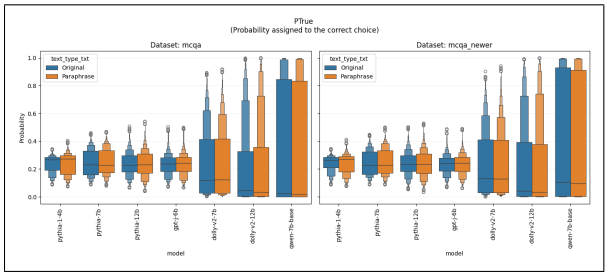


Figure 8: **Probabilities assigned to the correct choice.** Box plots showing the distribution of probabilities assigned to correct answers by different models for original questions and paraphrases. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher probabilities indicate greater model confidence in correct predictions.

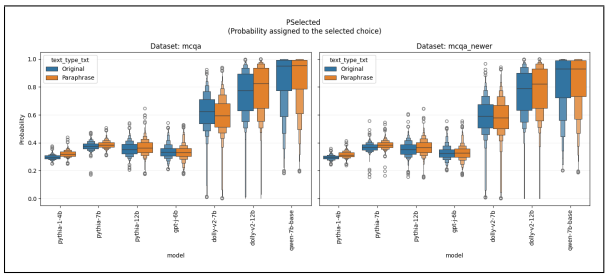


Figure 9: **Probabilities assigned to the selected choice.** Box plots showing the distribution of probabilities assigned to selected answers by different models for original questions and paraphrases. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher probabilities indicate greater model confidence in selected predictions.

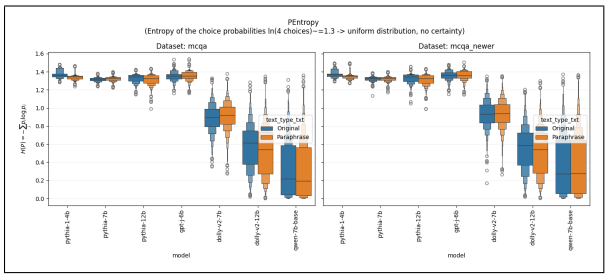


Figure 10: **Entropy of choice probabilities.** Box plots showing the distribution of entropy values across different models for original questions and paraphrases. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher entropy values indicate more uniform probability distributions across answer choices, reflecting greater model uncertainty. Maximum entropy of $ln(4) \approx 1.39$ corresponds to uniform distribution across four choices.

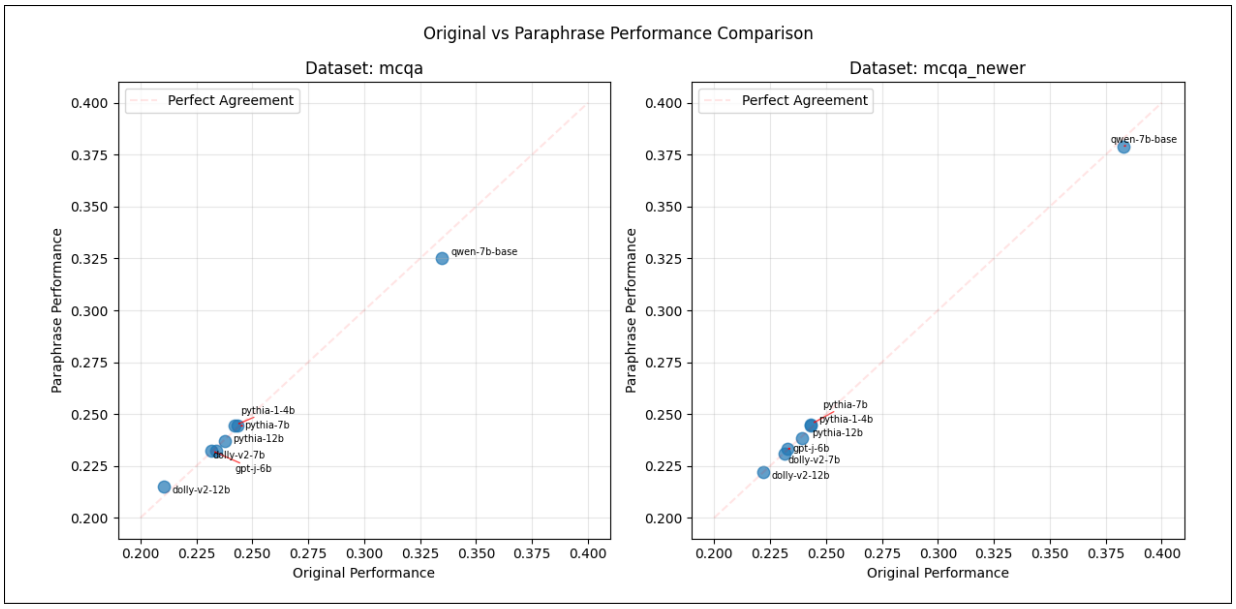


Figure 11: Accuracy comparison of original vs paraphrase in *mcqa* (left) and *mcqa_newer* (right).

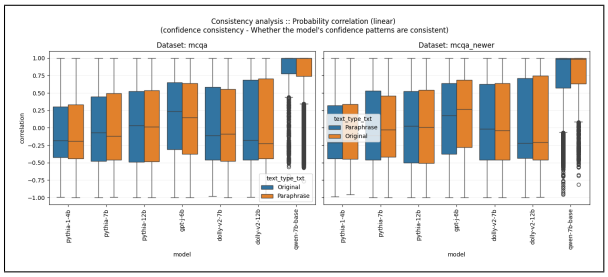


Figure 12: Probability correlation: confidence consistency — whether the model's confidence patterns are consistent.

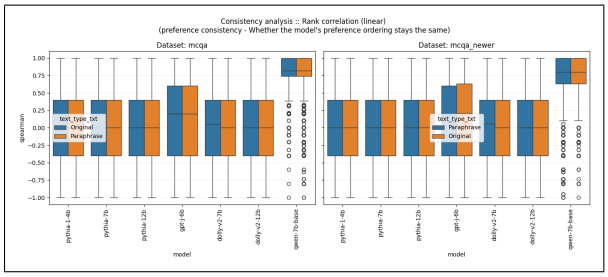


Figure 13: Rank correlation: preference consistency - Whether the model's preference ordering stays the same.

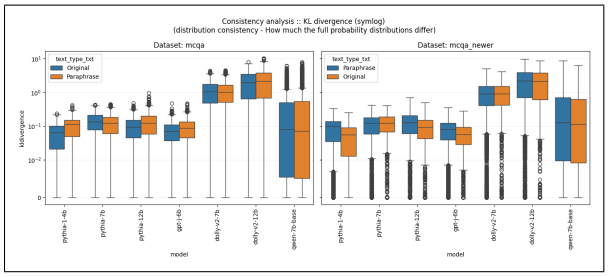


Figure 14: KL divergence: distribution consistency - How much the full probability distributions differ.

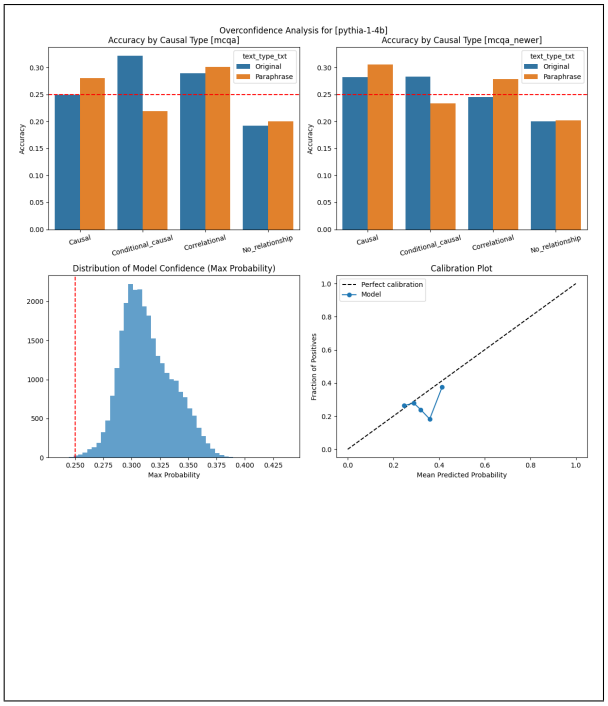


Figure 15: **Overconfidence analysis for Pythia 1.4B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right).

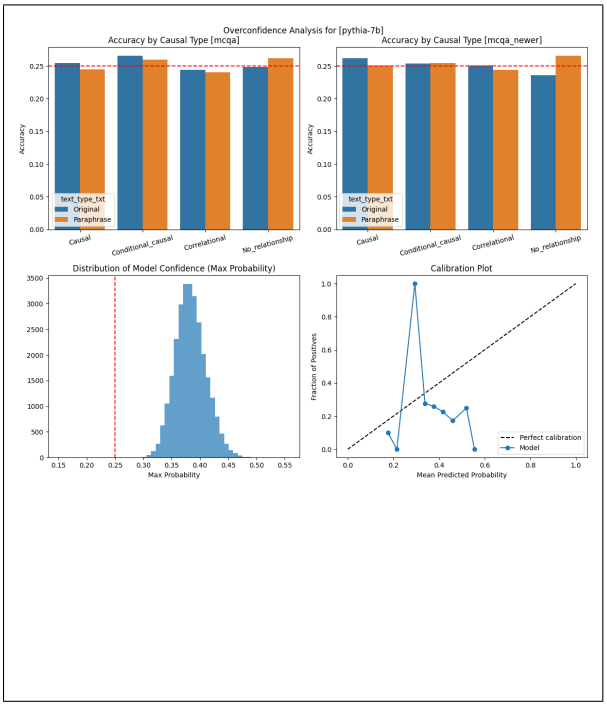


Figure 16: **Overconfidence analysis for Pythia 7B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right).

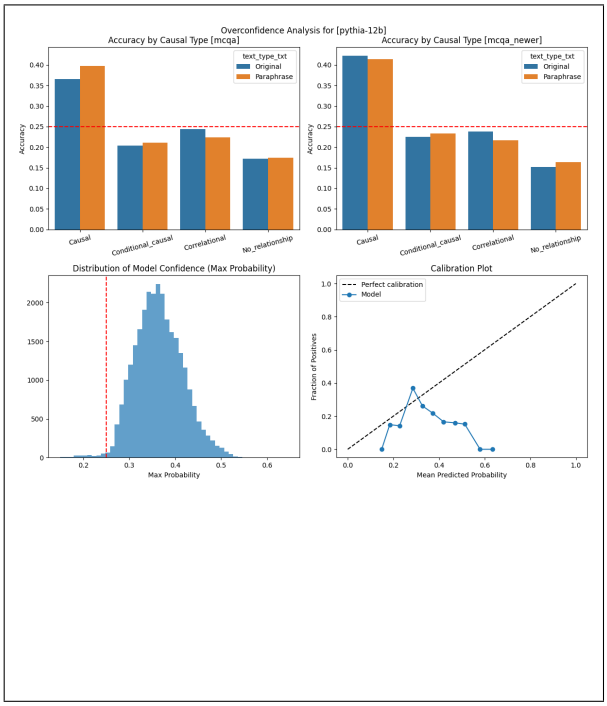


Figure 17: **Overconfidence analysis for Pythia 12B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right).

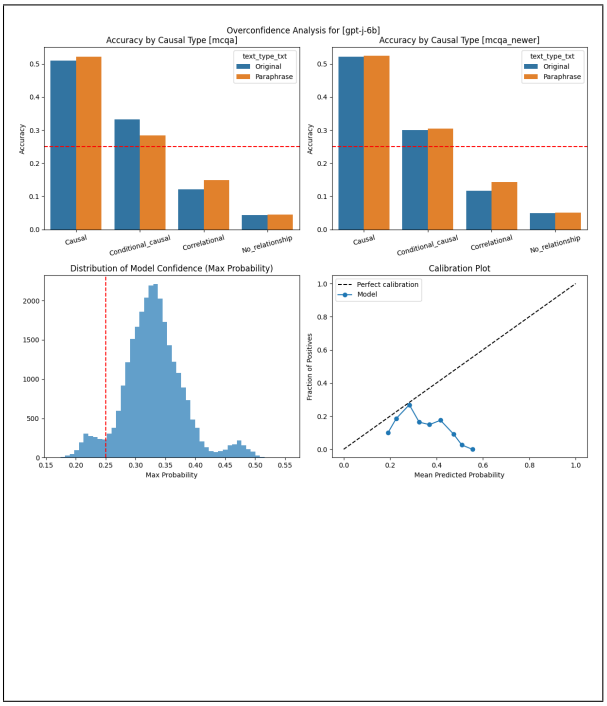


Figure 18: **Overconfidence analysis for GPT-j 6B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right).

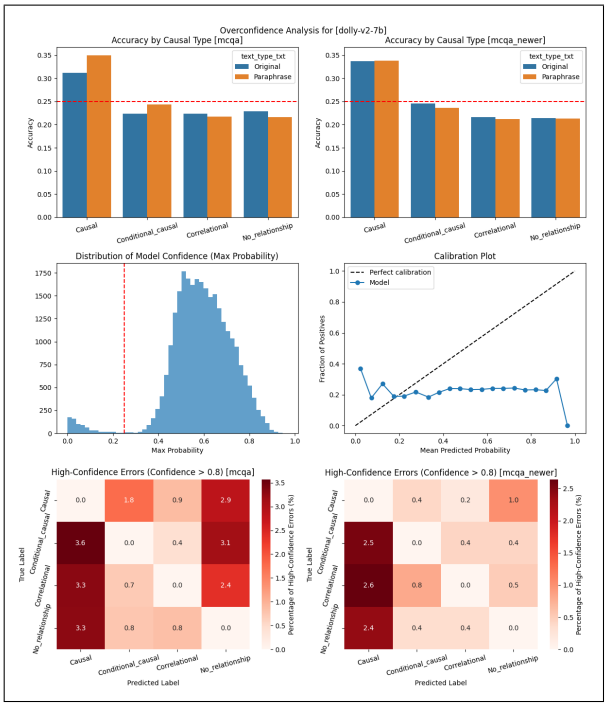


Figure 19: **Overconfidence analysis for Dolly-v12 7B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right). *Bottom row:* Confusion matrices for high-confidence errors (confidence >0.8) in *mcqa* (left) and *mcqa_newer* (right). Heatmap values represent the percentage of each true class that was misclassified with high confidence.

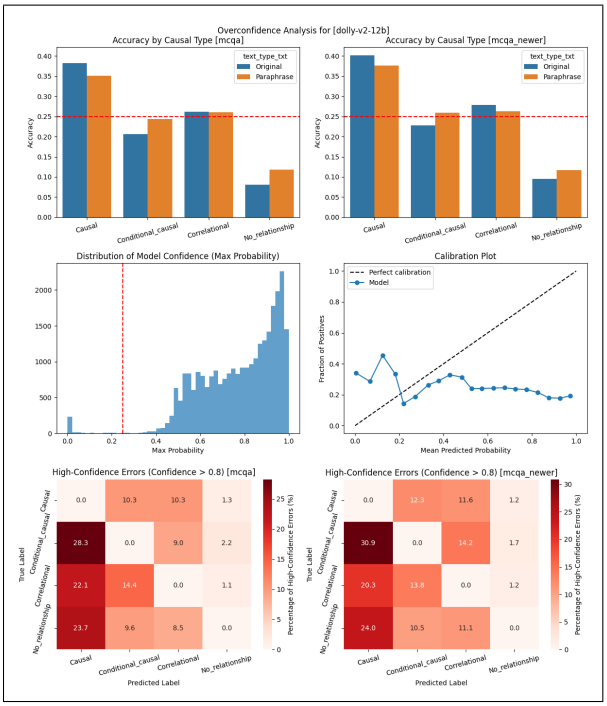


Figure 20: **Overconfidence analysis for olly-v12 12B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right). *Bottom row:* Confusion matrices for high-confidence errors (confidence >0.8) in *mcqa* (left) and *mcqa_newer* (right). Heatmap values represent the percentage of each true class that was misclassified with high confidence.

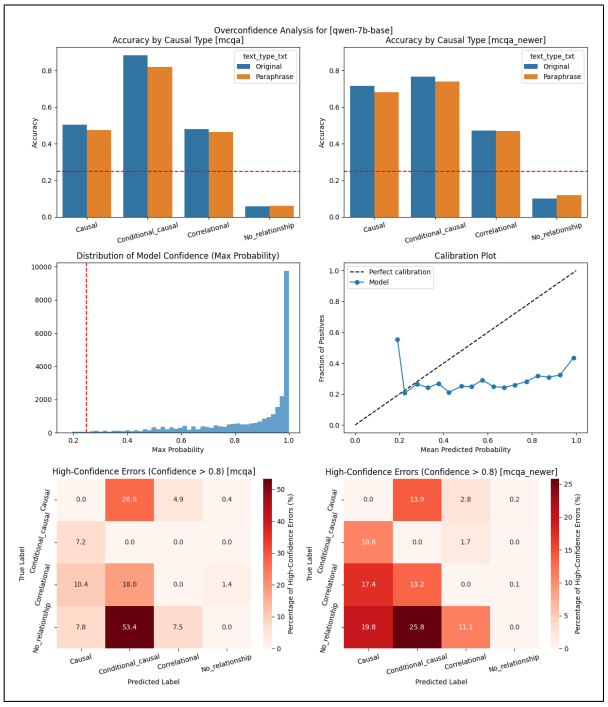


Figure 21: **Overconfidence analysis for Qwen 7B.** *Top row:* Accuracy by causal relationship type for original questions and paraphrases in *mcqa* (left) and *mcqa_newer* (right). *Middle row:* Model confidence distribution using 50 bins (left) and calibration plot showing predicted probabilities (x-axis) versus actual accuracy (y-axis) with data grouped into 20 bins (right). *Bottom row:* Confusion matrices for high-confidence errors (confidence >0.8) in *mcqa* (left) and *mcqa_newer* (right). Heatmap values represent the percentage of each true class that was misclassified with high confidence.

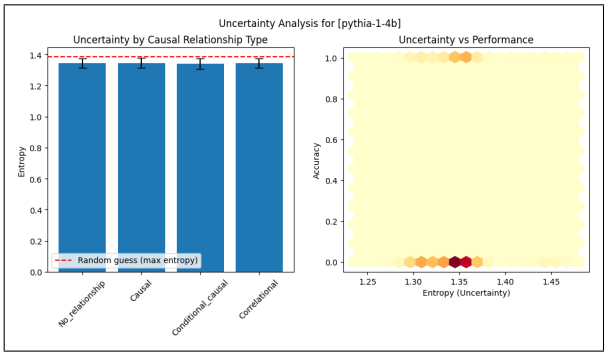


Figure 22: **Uncertainty analysis for Pythia 1.4B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

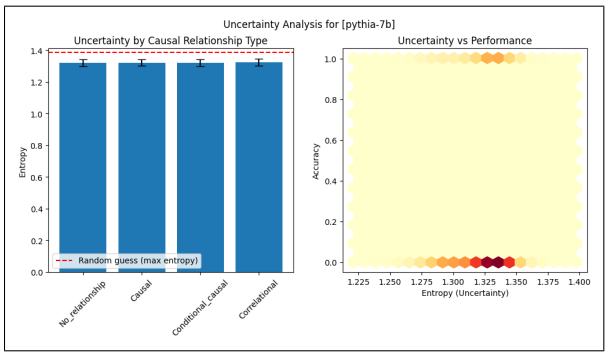


Figure 23: **Uncertainty analysis for Pythia 7B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

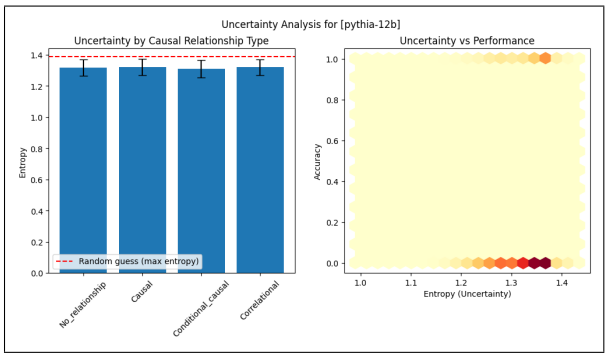


Figure 24: **Uncertainty analysis for Pythia 12B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

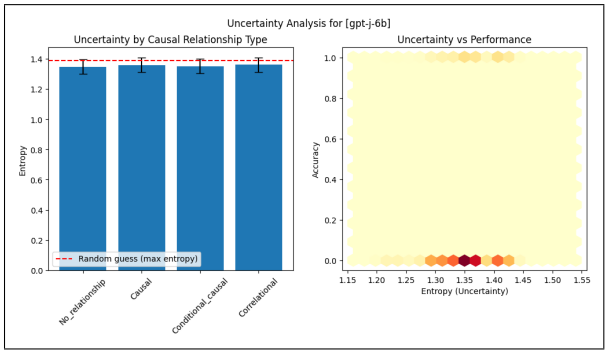


Figure 25: **Uncertainty analysis for GPT-j 6B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

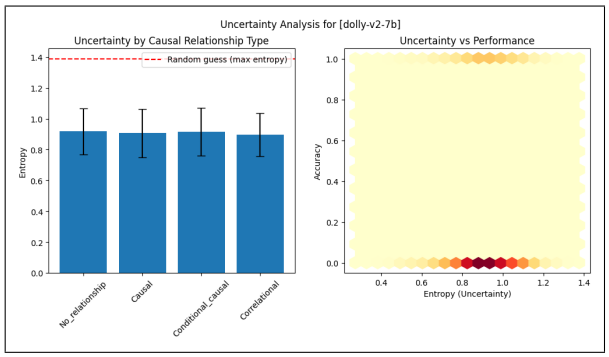


Figure 26: **Uncertainty analysis for Dolly-v12 7B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

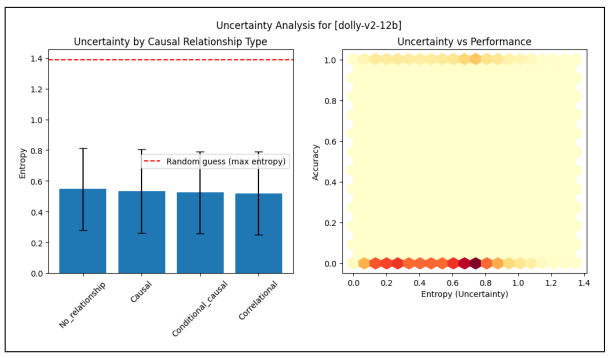


Figure 27: **uncertainty analysis for olly-v12 12B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

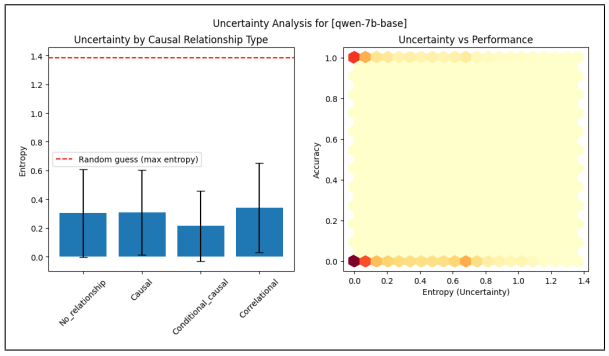


Figure 28: **Uncertainty analysis for Qwen 7B.** *Left:* Entropy-based uncertainty by causal relationship type. The red dashed line indicates maximum entropy (random guessing baseline). *Right:* Relationship between model uncertainty (entropy) and prediction accuracy, showing how confidence relates to performance.

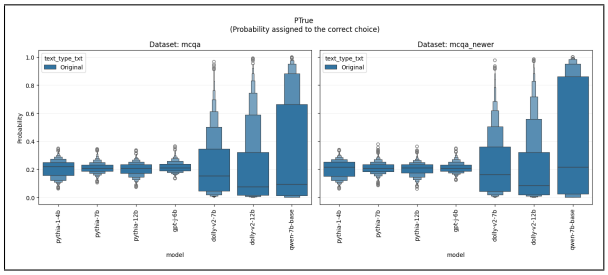


Figure 29: **Probabilities assigned to the correct choice.** Box plots showing the distribution of probabilities assigned to correct answers by different models for original questions and paraphrases. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher probabilities indicate greater model confidence in correct predictions.

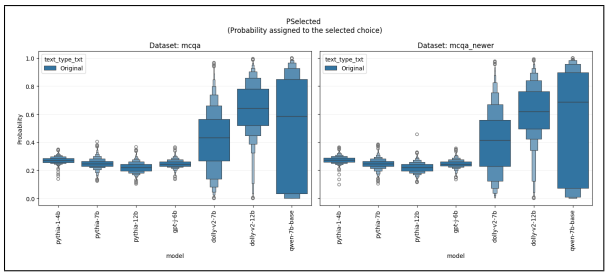


Figure 30: **Probabilities assigned to the selected choice.** Box plots showing the distribution of probabilities assigned to selected answers by different models for original questions and paraphrases. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher probabilities indicate greater model confidence in selected predictions.

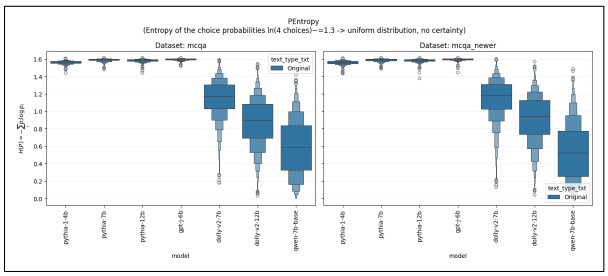


Figure 31: **Entropy of choice probabilities.** Box plots showing the distribution of entropy values across different models. Results are shown for *mcqa* (left) and *mcqa_newer* (right) datasets. Higher entropy values indicate more uniform probability distributions across answer choices, reflecting greater model uncertainty. Maximum entropy of $ln(4) \approx 1.39$ corresponds to uniform distribution across four choices.

It Depends: Resolving Referential Ambiguity in Minimal Contexts with Commonsense Knowledge

Lukas Ellinger and Georg Groh
School for Computation, Information and Technology
Technical University of Munich, Germany
lukas.ellinger@tum.de, grohg@cit.tum.de

Abstract

Ambiguous words or underspecified references require interlocutors to resolve them, often by relying on shared context and commonsense knowledge. Therefore, we systematically investigate whether Large Language Models (LLMs) can leverage commonsense to resolve referential ambiguity in multi-turn conversations and analyze their behavior when ambiguity persists. Further, we study how requests for simplified language affect this capacity. Using a novel multilingual evaluation dataset, we test DeepSeek v3, GPT-4o, Qwen3-32B, GPT-4o-mini, and Llama-3.1-8B via LLM-as-Judge and human annotations. Our findings indicate that current LLMs struggle to resolve ambiguity effectively: they tend to commit to a single interpretation or cover all possible references, rather than hedging or seeking clarification. This limitation becomes more pronounced under simplification prompts, which drastically reduce the use of commonsense reasoning and diverse response strategies. Fine-tuning Llama-3.1-8B with Direct Preference Optimization substantially improves ambiguity resolution across all request types. These results underscore the need for advanced fine-tuning to improve LLMs' handling of ambiguity and to ensure robust performance across diverse communication styles.

1 Introduction

Natural language is inherently ambiguous. For example, pronouns may refer to multiple possible entities within a sentence. Nevertheless, humans typically resolve such ambiguity by drawing on context, shared knowledge, and conversational history (Ferreira, 2008). Consider the two conversations shown in Figure 1, where the user asks the question, "Why can **it** fly?". Without additional clues, the pronoun "it" is unclear and could refer to multiple entities. In the left conversation, the prior context mentions a helicopter and a drum; in the right, it additionally includes an owl. Humans effortlessly combine this context with commonsense knowledge, recognizing that drums cannot fly, but helicopters and owls can. As a result, the first case is unambiguous, while the second may require clarification.

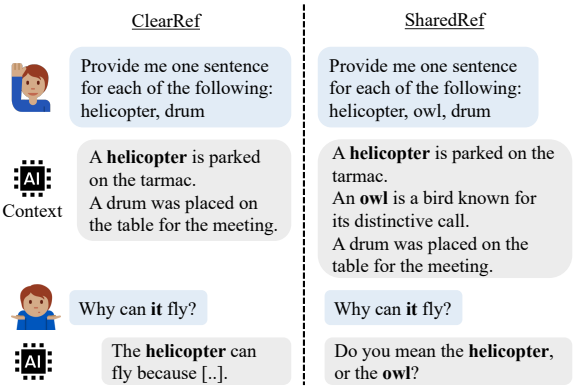


Figure 1: Two conversations between a user and an LLM in response to the ambiguous question ("Why can **it** fly?"). In both cases, the LLM uses prior context to narrow the possible referents to entities capable of flying. In the left conversation, it attempts an answer; in the right, it asks for clarification.

This process reflects a fundamental feature of human communication: a "division of labor" between speakers and listeners, where speakers omit explicit details to minimize effort, trusting listeners to fill in gaps using common ground (Ferreira, 2008). Common ground consists of the mutual knowledge, beliefs, and assumptions interlocutors accumulate and maintain during conversation (Clark and Brennan, 1991; Clark, 1996). Central to common ground is commonsense knowledge, a broadly shared understanding of the world that enables people to make implicit inferences effortlessly.

As mentioned, humans are usually good at building and using common ground. While prior work suggests that LLMs struggle with ambiguity resolution, particularly in static, single-turn contexts

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 229–246
November 9, 2025 ©2025 Association for Computational Linguistics

(Liu et al., 2023), our work shifts focus to a conversational setting. We study how LLMs behave in multi-turn dialogs where common ground is explicitly established through conversation history and commonsense knowledge. In our setting, multiple referents can remain plausible even after considering prior context. This allows us to evaluate how models handle uncertainty through different response strategies, such as requesting clarification.

We further examine how language constraints affect this ability. Language models are increasingly used to generate output in different variants, such as simplified and easy-to-understand language. This has clear benefits for accessibility, particularly for users with cognitive or linguistic challenges (Freyer et al., 2024). However, simplified outputs often reduce the depth and precision of content (Trienes et al., 2024). Ellinger et al. (2025) find that models prompted to define homonyms in simple language often default to the most salient meaning, disregarding less dominant but valid definitions. We explore whether such requests for simplified language also affect a model's capacity to resolve ambiguity when multiple interpretations are plausible.

Studying this is crucial because misinterpretation of ambiguous language can lead to downstream failures such as misinformation, hallucinations, or user confusion. By systematically testing whether LLMs consider multiple plausible candidates rather than relying on recency or default biases, we provide a diagnostic view of their behavior in ambiguous conversational settings.

Our contributions are as follows:

- We introduce a multilingual dataset for evaluating LLMs to resolve referential ambiguity in conversations with explicit common ground.
- We evaluate DeepSeek v3, GPT-4o, Qwen3-32B, GPT-4o mini, and Llama 3.1 8B using both LLM-as-Judge and human annotations.
- We show that LLMs often commit to a single interpretation or cover all references instead of hedging or clarifying. Simplified language constraints worsen this by reducing commonsense reasoning and response diversity.
- We fine-tune LLaMA 3.1 8B with Direct Preference Optimization (DPO), achieving significant improvements on our task that generalize to a lexical ambiguity benchmark, with less degradation under simplified prompts.

2 Background and Related Work

Ambiguity and Clarification. Understanding language often requires resolving ambiguity, such as referential ambiguity, where it is unclear which entity a phrase refers to. Such unclear references slow down human processing (Gernsbacher, 1989; MacDonald and MacWhinney, 1990; Myers and O'Brien, 1998; Stewart et al., 2007), yet humans are usually good at resolving them by drawing on common ground.

In contrast, LLMs struggle with ambiguity. Min et al. (2020) introduce AmbigQA, a dataset designed to investigate underspecified questions, and subsequent studies (Wildenburg et al., 2024; Liu et al., 2023) show that even state-of-the-art models underperform in such settings. This limitation extends to the multimodal domain: Testoni et al. (2024) find that vision–language models also handle ambiguity poorly, often replying with overconfident or biased outputs. While their focus is on visual context, the challenge is related to ours, with textual context instead of images.

Models also rarely seek clarification. Kuhn et al. (2023) show that LLMs often respond incorrectly to ambiguous inputs rather than asking follow-up questions. Prior work confirms this lack of clarification behavior (Benotti and Blackburn, 2017; Xu et al., 2019; Shi et al., 2022). Herlihy et al. (2024) link this tendency to fine-tuning biases and propose a taxonomy of model responses, which we adopt.

Prior work mainly studies ambiguity in static, single-turn settings without common ground. Notably, datasets for anaphora resolution, such as the Winograd Schema Challenge (Levesque et al., 2012), focus on single-sentence coreference, where exactly one antecedent is correct and can be identified using commonsense reasoning. In contrast, we study LLMs in multi-turn dialogs where common ground is explicitly established through conversation history and commonsense knowledge. In our setting, multiple referents can remain plausible even after considering context. This allows us to evaluate how models handle uncertainty through different response strategies, such as direct answers, hedging, or requesting clarification, rather than simply selecting the correct noun.

Finally, we test if our fine-tuned model generalizes to lexical ambiguity using the benchmark of Ellinger et al. (2025), which evaluates homonym definitions without disambiguating context.

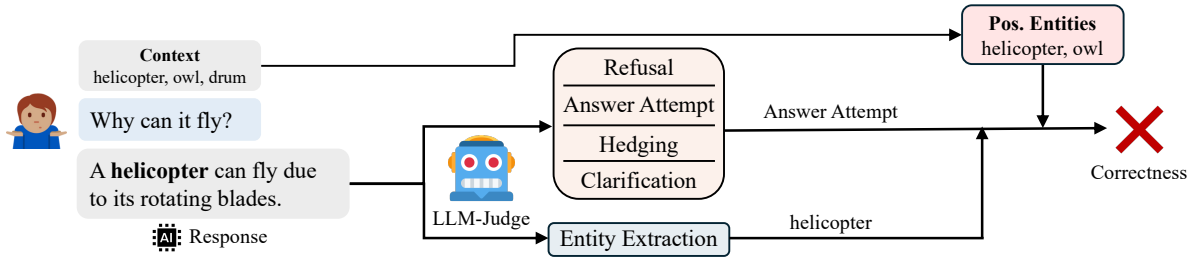


Figure 2: Evaluation pipeline including LLM-Judge for response categorization and entity extraction. Based on these outputs and the positive entities identified in the context, we determine the fine-grained response category and assess correctness with respect to entity resolution.

Commonsense Evaluation. Prior work systematically evaluated LLMs on commonsense reasoning benchmarks. Li et al. (2022) conduct evaluations under zero- and few-shot settings across four benchmarks, revealing that pre-trained LMs struggle to acquire commonsense knowledge without task-specific supervision. Scaling model size or adopting to few-shot prompting does not suffice to reach human-level performance. Similarly, Bian et al. (2024) assess ChatGPT on eleven commonsense QA datasets. They find that ChatGPT can retrieve relevant knowledge via prompting. However, it often fails to identify and apply the specific commonsense required to answer a given question. In the multimodal domain, Fu et al. (2024) introduce Commonsense-T2I, the first benchmark evaluating whether text-to-image models generate images consistent with commonsense knowledge. They find that state-of-the-art models achieve only 49% accuracy, indicating significant gaps in visual commonsense understanding.

Our work extends these by exploring another dimension of commonsense. Unlike prior benchmarks focused on question answering or image alignment, we assess whether models recognize ambiguous referents and either disambiguate or request clarification, demonstrating a context-aware application of commonsense reasoning.

Simple Language. Simplified language aims to improve accessibility for a broad range of users, including non-native speakers, children, domain novices, and individuals with cognitive impairments. Its availability is endorsed by the Web Content Accessibility Guidelines (WCAG) to promote inclusive communication (W3C, 2025). Simplified language involves straightforward vocabulary, clear sentence structure, minimal jargon, and the avoidance of complex grammar (Freyer et al., 2024). Domains like healthcare, law, and education already widely apply it (Garimella et al., 2022; Deilen et al., 2024; Rets et al., 2022). However, prior work has shown that simplification in LLM-generated text can lead to undesirable side effects such as omissions or overly vague formulations (Anschütz et al., 2025; Agrawal and Carpuat, 2024; Devaraj et al., 2022). Ellinger et al. (2025), for instance, report that when asked to define homonyms in simplified language, models tend to default to the most salient meaning, neglecting valid but less frequent senses.

Building on this line of work, we study how simplification constraints affect a model's ability to resolve referential ambiguity and how task-specific finetuning affects performance in the lexical ambiguity benchmark of Ellinger et al. (2025).

3 Methodology

We evaluate whether LLMs can resolve referential ambiguity using common knowledge and how requests for simplified language affect this ability. Each test instance consists of a short context passage introducing some entities (e.g., *helicopter*, *owl*, *drum*). The user then asks an ambiguous question referring to one of the entities without naming it directly (e.g., *Why can it fly?*). For each instance, we define a set of positive entities as those for which the question makes sense, and negatives as those for which it does not (e.g., a drum cannot fly). We evaluate two setups: *ClearRef*, where one positive and one negative entity make the referent unambiguous with commonsense, and *SharedRef*, where two positives and one negative leave ambiguity even with commonsense. This setup tests whether models consider multiple plausible candidates rather than relying on recency or default biases. We treat the pronoun "it" as equally applicable to all introduced positive entities. To assess the impact of recency, we perform an ablation in which the order of entities is permuted (see Appendix D).

Ambiguous Questions by Relation

Rel. 1: Why can it **fly**?
Rel. 2: Why is it **sweet**?
Rel. 3: Why is it **made of wood**?
Rel. 4: Why can it **swim**?
Rel. 5: Why can it **run fast**?
Rel. 6: Why can it **climb trees**?
Rel. 7: Why is it **hot**?
Rel. 8: Why is it **loud**?

Simple: [..] Respond in simple language.

Figure 3: Ambiguous questions for our eight relations. In the Simple setting, an instruction is appended. Exact relations names in Appendix B.

3.1 Dataset

We construct our datasets based on ConceptNet (Speer et al., 2017), a knowledge graph that encodes commonsense relationships between entities and attributes. We select eight relations, such as *capable of flying*, and extract all associated entities. Figure 3 provides the complete list of relations. Since each dialog requires a context passage, we use GPT-4.1-nano to generate a concise sentence for every entity. These sentences, each beginning with the entity name, serve as the context passages for all related evaluations.

For *ClearRef*, each entity is paired with a negative sample from a different relation. We use GPT-4.1-nano to verify that the negative entity does not satisfy the target relation. For *SharedRef*, we create samples by pairing all entities within the same relation and similarly pick a negative. This results in 52 *ClearRef* and 227 *SharedRef* examples. We list further details in Appendix B.

To enable multilingual evaluation, we translate the context sentences and entities into Arabic, French, Russian, and Simplified Chinese using the DeepL API[1]. We choose these languages to facilitate comparison with the multilingual setting of Ellinger et al. (2025).

3.2 Model and Prompt Configuration

We evaluate five LLMs on our task: GPT-4o, GPT-4o-mini (OpenAI et al., 2024), Qwen3-32B (Qwen Team, 2025), DeepSeek v3 (DeepSeek-AI et al., 2025), and Llama 3.1 8B (Grattafiori et al., 2024). These models vary in size and openness, enabling a comprehensive analysis of performance across diverse LLMs. Details on model versioning and access are listed in Appendix A.

We evaluate eight relations, each associated with an ambiguous question. For each, we test two prompt settings: **Normal**, presenting only the ambiguous question, and **Simple**, which adds an instruction to respond in simplified language. This setup allows us to examine how constraining outputs to simpler language affects model responses. English prompts are shown in Figure 3, with multilingual versions in Appendix Figure 12.

3.3 Evaluation Pipeline

The input to the evaluation pipeline (Figure 2) consists of a brief dialogue between a user and an LLM, exemplified in Figure 1. The response to the dialogue is passed to our LLM-Judge, which performs two tasks. First, it classifies the response type into one of four categories: *Refusal*, *Answer Attempt*, *Hedging*, or *Clarification* (cf. subsection 3.4). In this case, the response is labeled as an *Answer Attempt*. Second, it extracts all entities mentioned in the response (here, *helicopter*). Using the set of mentioned entities and the known positive entities (in this case, *helicopter* and *owl*), we assess the correctness of the response. Since the model attempts an answer but only mentions one of the two positive entities, the response is marked as incorrect.

3.4 Response Categorization

Following Laban et al. (2025), we adopt the response taxonomy from Herlihy et al. (2024), which includes *Answer Attempt*, *Clarification*, *Interrogation*, *Discussion*, *Hedging*, *Refusal*, and *Missing*. Focusing on referential ambiguity resolution, we simplify this taxonomy by merging *Interrogation* into *Clarification* and *Discussion* into *Answer Attempt*, reducing annotation complexity. Full definitions and examples appear in Appendix E. Briefly:

- **Hedging**: The assistant uses conditional or speculative language (e.g., “might be...”, “if you meant X...”).
- **Clarification**: The assistant requests more information without offering interpretations or using hedging.
- **Answer Attempt**: The assistant clearly commits to at least one interpretation, providing a factual response without any hedging.

[1] https://www.deepl.com/en/pro-api

We define a response as *correct* if it appropriately addresses the ambiguity in the input. Clarifications are always correct, as they seek additional input without committing to an interpretation. Hedging responses are considered correct, as long as they mention at least one entity. While they do not resolve the ambiguity, they acknowledge it and express uncertainty in a transparent way. In contrast, answer attempts are only deemed correct if they explicitly mention both positive entities.

Herlihy et al. (2024) discuss the trade-off between the usefulness and cognitive cost of different response categories, approximated by response length. In our setting, we argue that the most desirable responses, regardless of the category, are those that mention all and only the positive entities. We refer to these as **direct** responses. They reflect correct disambiguation based on common knowledge while minimizing user effort through clear and concise answers, free of irrelevant distractors.

In *SharedRef*, we consider any *direct* response the most appropriate response. In contrast, for *ClearRef*, where the ambiguity can be fully resolved, an *Answer Attempt* is preferred.

3.5 Automatic Evaluation

We designed an automated evaluation framework that leverages GPT-4.1-mini as an LLM-Judge. The framework assesses model responses based on the response categories defined in subsection 3.4. It classifies responses and extracts explicitly mentioned entities. A few-shot prompt, detailed in Appendix F, guides the evaluation. To validate the framework, one author manually labeled 500 responses from the English dataset, with 100 responses per evaluated model (50 for the standard prompt and 50 for the simple prompt). The annotator performed both response classification and extraction of explicitly mentioned entities, exactly as the LLM was tasked to do. The LLM judge achieved a 98% agreement rate on response classification and a Cohen's Kappa score of 0.916, indicating almost perfect agreement according to Landis and Koch (1977). For entity extraction, the framework achieved a 97.8% exact match accuracy. More details are provided in Appendix F.

3.6 Direct Preference Optimization

We fine-tuned Llama-3.1-8B to improve referential ambiguity resolution using DPO (Rafailov et al., 2024). DPO aligns model behavior with desired outcomes by training on preference pairs. In our setup, we favor direct over incorrect responses.

Our training dataset contains 1,388 preference pairs across all languages by comparing incorrect Llama 3.1 8B's outputs with *direct* responses from other models. To prevent reliance on entity position, we randomly permuted the order within each conversation. We restricted the training data to the '*capableOf fly*' relation, allowing us to later assess generalization to other relations.

We performed a single training run using the whole training set. This decision reflects our aim to demonstrate the feasibility of aligning models to produce more useful responses with lower cognitive cost, rather than optimizing for peak performance through extensive tuning. Detailed training information is provided in Appendix G.

4 Results

4.1 ClearRef Dataset

Figure 4 shows that all models maintain correctness above 90% across languages and settings, with some achieving perfect scores. The lowest correctness score is 90.38%, observed for Deepseek v3 (Simple) and Llama-3.1-8B (Normal) in French. When comparing the Normal and Simple settings, GPT-4o is the only model with higher correctness in the Normal setting, while the other models either remain similar or slightly decrease. The rate of direct responses among the correct answers varies drastically across models and languages. In the Simple setting, Qwen3-32B shows the highest variance, with a direct response rate ranging from as low as 22.45% in Arabic to 73.08% in English. In the Normal setting, GPT-4o-mini varies most, with only 47.06% direct responses in Russian to 82.69% in English. Llama-3.1-8B demonstrates the highest rates for English, achieving 98.00% in Normal and 97.96% in Simple. Averaged across languages, mean direct responses among all responses differ by model and setting. Except for Deepseek v3, all models show higher direct response rates in the Normal setting compared to Simple. In Normal, Llama-3.1-8B achieves the highest rate (80.38%), followed by GPT-4o, GPT-4o-mini, Qwen3-32B, and Deepseek v3 (58.85%). Detailed breakdowns by model, language, and prompt type are provided in Appendix Table 9.

In Figure 5, we show the distribution of response categories across languages and models. In all cases, *Answer Attempt* is the dominant category. However, comparing the Normal and Simple set-

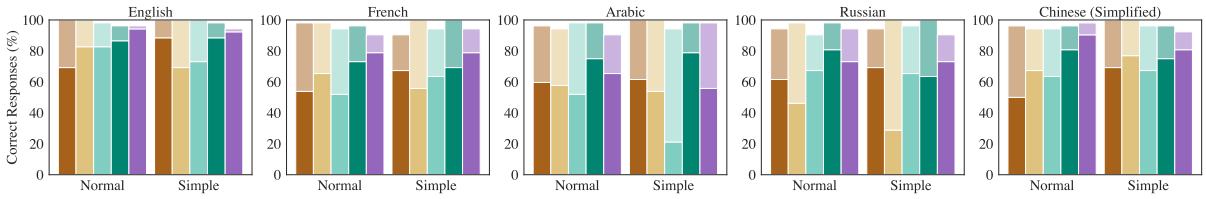


Figure 4: Percentage of correct responses across five languages on the ClearRef dataset. Colored squares indicate different models: DeepSeek v3, GPT-4o-mini. Qwen3-32B, GPT-4o, and Llama-3.1-8B. The darker portion of each bar represents the percentage of Direct Responses.

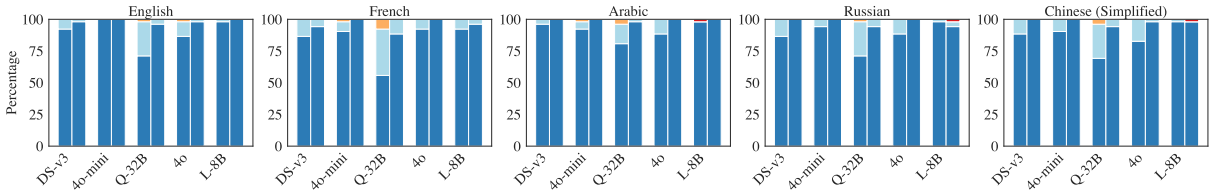


Figure 5: Distribution of the defined response categories across five languages on the ClearRef dataset. For each model, the left bar represents the Normal setting and the right bar the Simple setting. Colored squares represent response types: Answer Attempt, Hedge, Clarification, and Refuse.

tings reveals a shift: In the Simple setting, models nearly always produce answer attempts (mean 97.92%). In Normal, especially with Qwen3-32B, hedging occurs more frequently, and to a lesser extent, clarifications. For Qwen3-32B, the average proportion of Answer Attempts drops to 69.61%.

4.2 SharedRef Dataset

We show proportions of correct responses along with direct response rates in Figure 6. The results reveal a sharp drop from Normal to Simple and a clear separation between two model groups: high performers (GPT-4o, Qwen3-32B, Deepseek v3) and low performers (Llama-3.1-8B, GPT-4o-mini).

Low-performing models show poor performance across languages and prompt settings, with GPT-4o-mini reaching below 13% correctness in the Normal setting and Llama-3.1-8B slightly higher but inconsistent due to an outlier in the Arabic Simple setting.

Among the top performers, GPT-4o achieves the highest correctness in English Normal prompts (81.06%, thereby 45.11% direct), while Qwen3-32B performs best overall when averaged across languages in the Normal setting (70.22%, 31.11%). Deepseek v3 leads in the Simple setting (37.97%, 22.73%), outperforming the others despite lower direct response rates.

Performance also varies notably by language. In the Normal setting, English (69.16% correct, thereby 47.28% direct) and Chinese (63.96%, 41.54%) achieve the highest average correctness, followed by Arabic, French, and Russian (51.19%, 45.71%), reflecting the models' native strengths (e.g., GPT for English, Qwen and Deepseek for Chinese). In the Simple setting, Arabic leads (50.22%, 44.01%), followed by Chinese and English, with French and Russian (26.08%, 59.81%) trailing. We show a detailed breakdown per model, language, and prompt type in Appendix Table 10.

Figure 7 shows the distribution of response categories across languages and models. Consistent with ClearRef, Answer Attempt remains the dominant category in the Simple setting, with an average proportion of 97.01% across all languages and models. The only notable outlier is Qwen3-32B in Chinese, with a lower proportion of 72.69%.

In the Normal setting, the shift toward other response categories becomes more pronounced than in ClearRef. The average proportion of Answer Attempts decreases to 77.67%. Notable deviations include GPT-4o in English (29.52%) and Russian (46.26%), as well as Qwen3-32B in English (43.17%), French (49.34%), Russian (40.97%), and Chinese (43.61%). These two models show marked increases in Hedging (GPT-4o from 1.67% to 35.06%, Qwen3-32B from 8.37% to 41.14%) and Clarification (GPT-4o from 0.09% to 4.76%, Qwen3-32B from 0.70% to 8.02%).

4.3 Direct Preference Optimization

We compare the base and the fine-tuned model on the SharedRef test set, excluding the *capableOf fly* relation among positives. Figure 8 shows that the

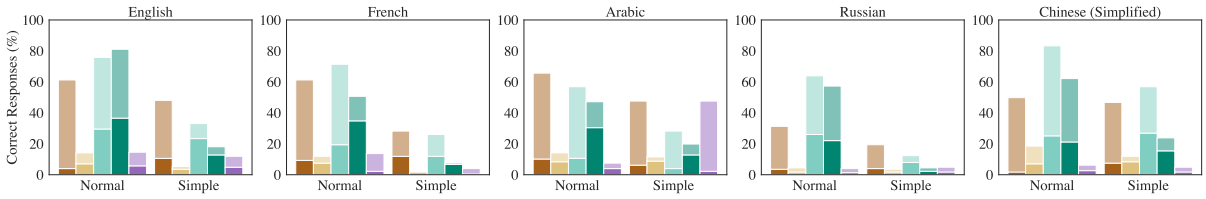


Figure 6: Percentage of correct responses across five languages on the SharedRef dataset. Colored squares indicate different models: DeepSeek v3, GPT-4o-mini. Qwen3-32B, GPT-4o, and Llama-3.1-8B. The darker portion of each bar represents the percentage of Direct Responses.

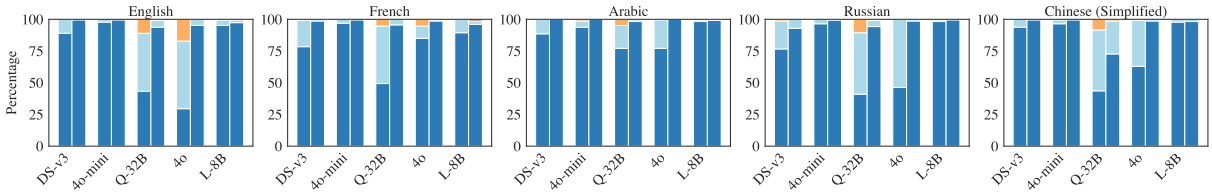


Figure 7: Distribution of the defined response categories across five languages on the SharedRef dataset. For each model, the left bar represents the Normal setting and the right bar the Simple setting. Colored squares represent response types: Answer Attempt, Hedge, Clarification, and Refuse.

results are consistent across languages. Overall, the proportion of correct responses increases from 13.46% to 96.45% in the Normal setting and from 13.83% to 91.59% in the Simple setting. Among the correct responses, direct responses rise from 28.60% to 42.96% (Normal) and from 33.37% to 50.66% (Simple). For comparison, the best base model, Qwen3-32B, achieves 62.43% correct (30.44% direct) in the Normal setting and 22.06% correct (60.97% direct) in the Simple setting.

The category distribution shifts drastically. In the base model, Answer Attempts dominate (91.78% in Normal, 96.45% in Simple). After fine-tuning, Clarification is most frequent, followed by Hedging and Answer Attempts. In the Simple setting, Clarification is less dominant than in Normal, while Hedging becomes more prevalent: 60.00% vs. 30.84% Clarification, 36.07% vs. 52.52% Hedging, and 3.74% vs. 16.63% Answer Attempts.

4.4 Homonym Definition Generation

Ellinger et al. (2025) introduced MCL-WiC, a multilingual homonym dataset, along with the *Sense Awareness* metric for evaluation. A response shows Sense Awareness by providing multiple definitions or explicitly acknowledging ambiguity via clarification requests or remarks about alternative meanings. They evaluated model performance under standard, simplified, and ELI5-style prompting (Fan et al., 2019), where the model explains a word as if the user were five years old.

Table 1 compares our fine-tuned model with the results reported by Ellinger et al. (2025). Against baseline models, our model achieves the highest Sense Awareness under the Normal prompt in English, French, and Russian, the second-highest in Arabic, and competitive results in Chinese. For Simple, it ranks highest in French and Russian, with comparable results in other languages. For ELI5, it outperforms all baseline models in every language except English, where it ranks second. Compared to its base model, our fine-tuned version shows consistent, mostly extensive improvements across all configurations, with the only exception being the English Simple setting, where performance drops by three percentage points.

They also fine-tuned Llama-3.1-8B on the same task. Their model produces English outputs for all languages except Russian, reflecting heavy optimization for English. In contrast, our DPO model handles all languages natively. While their fine-tuned model generally achieves higher Sense Awareness scores, our model remains competitive against the baseline models and narrows the gap in the language constraints. Their fine-tuning was explicitly targeted at this task, and reducing the gap between the language constraints. In contrast, our model achieves strong results across all languages without task-specific tuning.

5 Discussion

Our results indicate that current models struggle to apply commonsense knowledge for ambiguity

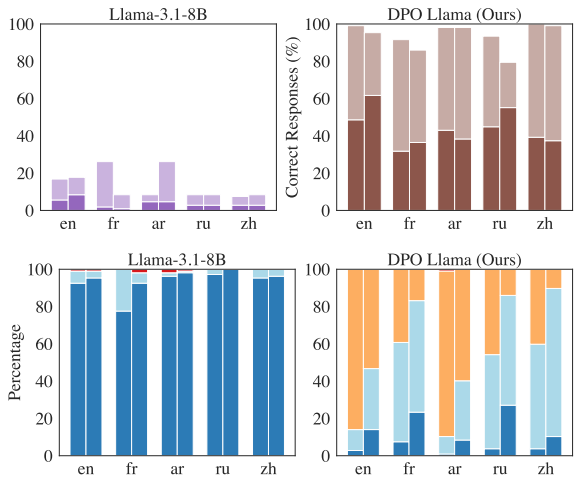


Figure 8: Comparison of the base model and our DPO-fine-tuned model across five languages on the SharedRef test set. For each language, the left bar represents the Normal setting, and the right bar the Simple setting.
Top: Percentage of correct responses. The darker portion of each bar represents the percentage of Direct Responses.
Bottom: Distribution of response categories. Colored squares denote: ■ Answer Attempt, ■ Hedge, ■ Clarification, ■ Refuse.

Prompt / Model	Sense Aware				
	En	Fr	Ar	Ru	Zh
Prompt: Normal					
3.1 8B	*96.95*	15.17	10.62	6.52	4.66
4o-mini	93.90	79.31	92.92	90.43	84.46
3-30B A3B	94.58	86.55	**98.23**	90.00	**100.00**
4 Maverick	96.27	54.83	74.34	75.65	45.08
v3	94.24	*87.93*	91.15	*91.74*	*87.56*
Our 3.1-8B	**97.63**	**97.93**	*93.81*	**99.57**	84.46
Their 3.1-8B	99.66	99.31	99.12	99.13	98.45
Prompt: Simple					
3.1 8B	64.41	7.59	6.19	2.17	7.77
4o-mini	63.05	52.76	*76.99*	43.91	*75.13*
3-30B A3B	**76.61**	*59.66*	69.03	*67.83*	**82.38**
4 Maverick	*69.83*	28.28	45.13	48.70	68.91
v3	63.73	47.93	**80.53**	65.22	74.09
Our 3.1-8B	61.02	**73.10**	71.68	**79.13**	64.25
Their 3.1-8B 8B	92.88	93.45	96.46	99.57	94.30
Prompt: ELI5					
3.1 8B	7.12	7.59	0.88	1.30	0.52
4o-mini	5.42	6.90	10.62	2.61	6.74
3-30B A3B	**22.03**	*17.24*	9.73	*14.78*	*14.51*
4 Maverick	10.85	13.10	11.50	9.57	9.84
v3	8.14	8.28	*13.27*	8.70	10.88
Our 3.1-8B	*13.22*	**25.86**	**46.90**	**19.13**	**17.62**
Their 3.1-8B	35.59	35.17	55.75	63.48	33.68

Table 1: Sense Awareness scores by prompt type and language. Best results are in **bold**, second-best in *italic*. Model outputs are copied from the original paper.

resolution. In the simpler ClearRef task, where only one entity fits the question, models are able to resolve the ambiguity with an accuracy ranging from 94.23% down to 21.15% depending on the model and setting. The more challenging SharedRef task, which involves two plausible entities, sees direct responses ranging from just 36.56% down to 0.44%. This aligns with findings by Bian et al. (2024). They observe that LLMs can retrieve commonsense facts, which in our case means realizing that an entity fits a relation when asked on its own. However, the models often fail to apply this knowledge when answering a specific question requiring such reasoning. In Appendix C, we evaluate GPT-4o's performance in English under a Chain-of-Thought setting, prompting it to explicitly verbalize its commonsense reasoning first.

Consistent with Herlihy et al. (2024) and Kuhn et al. (2023), we observe that models frequently skip clarification, opting to answer even when uncertainty remains. Several models show almost no clarification or hedging behavior. Herlihy et al. (2024) and Singhal et al. (2024) argue that this behavior stems from reinforcement learning from human feedback (RLHF). Annotation processes typically focus on single-turn conversations. As a result, models are rarely exposed to examples of follow-up clarification questions, which require multi-turn interaction. Moreover, annotators often favor verbose, catch-all answers for under-specified queries, even though such verbosity imposes cognitive costs on users (Singhal et al., 2024).

Another important observation is that prompting models to use simpler language can harm response quality. Interestingly, in ClearRef, there is no drop from Normal to Simple; in some models, Simple responses are even slightly better. In contrast, for the more complex SharedRef task, performance drops drastically in the Simple setting. This confirms prior work showing that simplification often leads to omissions and vague phrasing (Ellinger et al., 2025; Anschütz et al., 2025; Trienes et al., 2024; Agrawal and Carpuat, 2024; Devaraj et al., 2022). We argue that this behavior needs to change. For example, Kearney et al. (2025) show that LLMs adapt the information they provide based on assumptions about the user. This is problematic, especially if requesting simple language causes models to produce less thoughtful responses or overlook important distinctions. Again, RLHF may play a role, failing to capture the needs of diverse users and discouraging clarification and hedging in simplified contexts.

Taken together, we argue that resolving ambiguity requires a balance: infer as much as possible to avoid unnecessary elaboration, but clarify when uncertainty remains. Our DPO-trained model moves in this direction. It not only improves on our main evaluation but also generalizes to the lexical ambiguity benchmark of Ellinger et al. (2025). Moreover, it reduces the performance drop commonly observed when models operate in simplified language settings. This suggests that clarification and hedging behaviors can be learned in a transferable and robust way.

6 Conclusion

In this paper, we analyzed how LLMs handle textual referential ambiguity and to what extent they apply commonsense knowledge to resolve it. Our findings show that LLMs have limited ability to do so effectively. They tend to commit to a single interpretation or cover all possible references, rather than hedging or seeking clarification. This tendency becomes even more pronounced when users request simple language, which reduces commonsense reasoning and different answering strategies.

These results point to two core issues. First, there is a need for better fine-tuning to improve how LLMs deal with ambiguity. Second, LLMs should better adapt to different user needs. It is especially concerning that a request for simpler language leads to less thoughtful responses and fewer clarifications, showing that current systems often fail to support users with varied communication styles.

To support reproducibility and future research, we release our code[2]. Further links to models and datasets are provided in the repository.

Limitations

Multilingual Scope and Dataset Size. Our study focuses on English, French, Russian, Arabic, and Chinese. For non-English languages, we relied on direct translations from English using automated tools, which can introduce translation bias, cultural mismatches, or loss of nuance. Future work should create native datasets for each language to ensure more accurate and culturally appropriate evaluation. Additionally, the ClearRef and SharedRef datasets contain only 52 and 227 datapoints, respectively, and include only 8 relations from ConceptNet, making it difficult to draw fully stable conclusions and potentially biasing evaluation toward certain categories. Nevertheless, we observe very strong tendencies in the results, suggesting that the findings are still meaningful and indicative of broader trends.

Referential Order. Due to computational limits, we used a fixed entity order; full permutation results for English are provided in Appendix D.

Commonsense Context. We provided all models with the same context, which included a commonsense fact sourced from ConceptNet. While these facts consist of basic relations and vocabulary, we cannot guarantee that models internally represent or utilize this knowledge. Nevertheless, given the simplicity and generality of the facts, the models likely have access to such information.

LLM-based Evaluation. We used an LLM to judge model responses, observing near-perfect agreement with human annotations in English. While we did not conduct human agreement checks for other languages, the observed trends remain consistent across all languages, suggesting broader applicability. Moreover, the differences between prompt settings are substantially larger than any potential error margin, further reinforcing the robustness of our findings.

Selected Prompts. We use fixed user prompts for each relation, along with a single predefined suffix for requesting responses in simplified language. This setup reflects how typical users might interact with a model without actively optimizing prompt phrasing. However, LLMs are known to be highly sensitive to prompt formulation, which can significantly influence output quality (Brown et al., 2020). Future research could systematically investigate the effects of varied or optimized prompts on LLM performance.

[2] https://github.com/lukasellinger/itdepends

References

Sweta Agrawal and Marine Carpuat. 2024. Do Text Simplification Systems Preserve Meaning? A Human Evaluation via Reading Comprehension. *Transactions of the Association for Computational Linguistics*, 12:432–448. Place: Cambridge, MA Publisher: MIT Press.

Miriam Anschütz, Anastasiya Damaratskaya, Chaeeun Joy Lee, Arthur Schmalz, Edoardo Mosca, and Georg Groh. 2025. (Dis)improved?! How Simplified Language Affects Large Language

Model Performance across Languages. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 847–861, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Luciana Benotti and Patrick Blackburn. 2017. Modeling the clarification potential of instructions: Predicting clarification requests and other reactions. *Computer Speech & Language*, 45:536–551.

Ning Bian, Xianpei Han, Le Sun, Hongyu Lin, Yaojie Lu, Ben He, Shanshan Jiang, and Bin Dong. 2024. ChatGPT Is a Knowledgeable but Inexperienced Solver: An Investigation of Commonsense Problem in Large Language Models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3098–3110, Torino, Italia. ELRA and ICCL.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Herbert H Clark. 1996. *Using language*. Cambridge university press.

Herbert H. Clark and Susan E. Brennan. 1991. Grounding in communication. In *Perspectives on socially shared cognition.*, pages 127–149. American Psychological Association, Washington, DC, US.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. DeepSeek-V3 Technical Report. *arXiv preprint*. ArXiv:2412.19437 [cs].

Silvana Deilen, Ekaterina Lapshinova-Koltunski, Sergio Hernández Garrido, Christiane Maaß, Julian Hörner, Vanessa Theel, and Sophie Ziemer. 2024. Towards AI-supported Health Communication in Plain Language: Evaluating Intralingual Machine Translation of Medical Texts. In *Proceedings of the First Workshop on Patient-Oriented Language Processing (CL4Health) @ LREC-COLING 2024*, pages 44–53, Torino, Italia. ELRA and ICCL.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating Factuality in Text Simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345, Dublin, Ireland. Association for Computational Linguistics.

Lukas Ellinger, Miriam Anschütz, and Georg Groh. 2025. Simplifications are Absolutists: How Simplified Language Reduces Word Sense Awareness in LLM-Generated Definitions. *arXiv preprint*. ArXiv:2507.11981 [cs].

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long Form Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Victor S. Ferreira. 2008. Ambiguity, Accessibility, and a Division of Labor for Communicative Success. In Brian H. Ross, editor, *Psychology of Learning and Motivation*, volume 49 of *Advances in Research and Theory*, pages 209–246. Academic Press.

Nils Freyer, Hendrik Kempt, and Lars Klöser. 2024. Easy-read and large language models: on the ethical dimensions of LLM-based text simplification. *Ethics and Information Technology*, 26(3):50.

Xingyu Fu, Muyu He, Yujie Lu, William Yang Wang, and Dan Roth. 2024. Commonsense-T2I Challenge: Can Text-to-Image Generation Models Understand Commonsense? *arXiv preprint*. ArXiv:2406.07546 [cs].

Aparna Garimella, Abhilasha Sancheti, Vinay Aggarwal, Ananya Ganesh, Niyati Chhaya, and Nandakishore Kambhatla. 2022. Text Simplification for Legal Domain: Insights and Challenges. In *Proceedings of the Natural Legal Language Processing Workshop 2022*, pages 296–304, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Morton Ann Gernsbacher. 1989. Mechanisms that improve referential access. *Cognition*, 32(2):99–156.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 Herd of Models. *arXiv preprint*. ArXiv:2407.21783 [cs].

Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. 2024. On Overcoming Miscalibrated Conversational Priors in LLM-based Chatbots. *arXiv preprint*. ArXiv:2406.01633 [cs].

Matthew Kearney, Reuben Binns, and Yarin Gal. 2025. Language Models Change Facts Based on the Way You Talk. *arXiv preprint*. ArXiv:2507.14238 [cs].

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. CLAM: Selective Clarification for Ambiguous Questions with Generative Language Models. *arXiv preprint*. ArXiv:2212.07769 [cs].

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. LLMs Get Lost In Multi-Turn Conversation. *arXiv preprint*. ArXiv:2505.06120 [cs].

J. Richard Landis and Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics*, 33(1):159–174. Publisher: International Biometric Society.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The Winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pages 552–561, Rome, Italy. AAAI Press.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. 2022. A Systematic Investigation of Commonsense Knowledge in Large Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11838–11855, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Alisa Liu, Zhaofeng Wu, Julian Michael, Alane Suhr, Peter West, Alexander Koller, Swabha Swayamdipta, Noah Smith, and Yejin Choi. 2023. We're Afraid Language Models Aren't Modeling Ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 790–807, Singapore. Association for Computational Linguistics.

Maryellen C MacDonald and Brian MacWhinney. 1990. Measuring inhibition and facilitation from pronouns. *Journal of Memory and Language*, 29(4):469–492.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. AmbigQA: Answering Ambiguous Open-domain Questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, Online. Association for Computational Linguistics.

Jerome L. Myers and Edward J. O'Brien. 1998. Accessing the discourse representation during reading. *Discourse Processes*, 26(2-3):131–157. Publisher: Routledge _eprint: https://doi.org/10.1080/01638539809545042.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. GPT-4 Technical Report. *arXiv preprint*. ArXiv:2303.08774 [cs].

Qwen Team. 2025. Qwen3.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv preprint*. ArXiv:2305.18290 [cs].

Irina Rets, Lluisa Astruc, Tim Coughlan, and Ursula Stickler. 2022. Approaches to simplifying academic texts in English: English teachers' views and practices. *English for Specific Purposes*, 68:31–46.

Zhengxiang Shi, Yue Feng, and Aldo Lipani. 2022. Learning to Execute Actions or Ask Clarification Questions. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2060–2070, Seattle, United States. Association for Computational Linguistics.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A Long Way to Go: Investigating Length Correlations in RLHF. *arXiv preprint*. ArXiv:2310.03716 [cs].

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1). Section: Special Track on Cognitive Systems.

Andrew J. Stewart, Judith Holler, and Evan Kidd. 2007. Shallow processing of ambiguous pronouns: evidence for delay. *Quarterly Journal of Experimental Psychology (2006)*, 60(12):1680–1696.

Alberto Testoni, Barbara Plank, and Raquel Fernández. 2024. RACQUET: Unveiling the Dangers of Overlooked Referential Ambiguity in Visual LLMs. *arXiv preprint*. ArXiv:2412.13835 [cs].

Jan Trienes, Sebastian Joseph, Jörg Schlötterer, Christin Seifert, Kyle Lo, Wei Xu, Byron Wallace, and Junyi Jessy Li. 2024. InfoLossQA: Characterizing and Recovering Information Loss in Text Simplification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4263–4294, Bangkok, Thailand. Association for Computational Linguistics.

W3C. 2025. Web Content Accessibility Guidelines (WCAG) 2.1.

Frank Wildenburg, Michael Hanna, and Sandro Pezzelle. 2024. Do Pre-Trained Language Models Detect and Understand Semantic Underspecification? Ask the DUST! In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9598–9613, Bangkok, Thailand. Association for Computational Linguistics.

Jingjing Xu, Yuechen Wang, Duyu Tang, Nan Duan, Pengcheng Yang, Qi Zeng, Ming Zhou, and Xu Sun. 2019. Asking Clarification Questions in Knowledge-Based Question Answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1618–1629, Hong Kong, China. Association for Computational Linguistics.

A Model Access

To support reproducibility, Table 8 lists all models used in this paper, including their abbreviated names (as used in tables and figures), full names, versions, and access providers.

B Dataset

We extracted entities from the following eight relations: `CapableOf fly`, `HasProperty sweet`, `MadeOf wood`, `CapableOf swim`, `CapableOf run_fast`, `CapableOf climb_trees`, `HasProperty hot`, and `HasProperty loud`. All entities were manually reviewed and cleaned. During dataset construction, we used the following prompt with GPT-4.1-nano to verify that each negative entity truly does not satisfy the relation, in contrast to the two positive entities:

User Prompt: Relation Satisfaction

```
Does the word '<word>' satisfy the relation
'<relation>'?
Answer with a brief explanation and either
True or False for satisfies.
```

C Ablation: Chain-of-Thought Prompting

Bian et al. (2024) observe that LLMs often fail to apply commonsense knowledge when answering questions that require such reasoning. To investigate this in our setting, we tested GPT-4o on the English SharedRef dataset in a Chain-of-Thought (CoT) setting. We choose GPT-4o as it showed the sharpest drop from Normal to Simple. We appended the following instructions to encourage CoT reasoning:

User Prompt: Chain-of-Thought

```
<question> First, try resolving any
ambiguity using commonsense knowledge. If
the question remains ambiguous, your
answer should be a clarification request.
Otherwise, provide the answer. Put your
final response after Response:.
```

We compare standard and CoT prompting in Figure 9. CoT prompting performs worse than standard prompting, with accuracy dropping from 81.06% to 44.49% in the Normal setting. This is because CoT prompting often only partially resolves the ambiguity, responding to one positive while ignoring the other. This occurs roughly 50% of the time, suggesting a model preference for one entity,

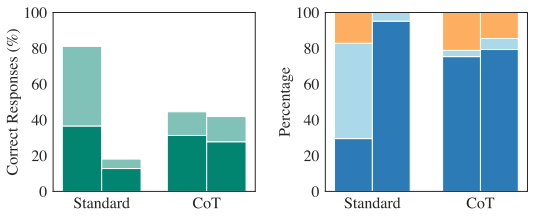


Figure 9: Comparison of Standard vs. CoT-Prompting on the SharedRef dataset. Left: Correctness; the darker portion of each bar indicates the percentage of Direct Responses. Right: Response category distribution. (Normal = left bar, Simple = right bar). Categories: Answer Attempt, Hedge, Clarification,

as it correctly identifies each entity when prompted individually. We observe more Clarifications and Answer Attempts, with nearly no Hedging in the Normal setting. The Simple setting is largely similar, contrasting with the standard Simple prompting.

Comparing the gap between Normal and Simple settings, we find it much smaller than in standard prompting. This suggests that when the LLM is explicitly guided on how to generate responses, there is no loss of thoughtfulness or omission of important distinctions. This is also reflected in the Simple CoT setting, performing better than the Simple standard prompting.

D Ablation: Permutation of Entity Ordering

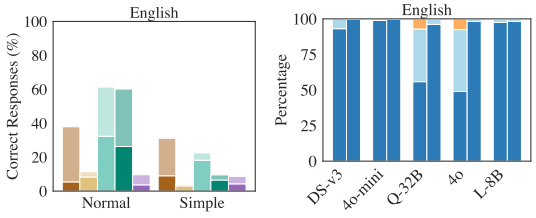


Figure 10: Average performance across all permutations in the English SharedRef dataset. Left: Correctness per model; the darker portion of each bar indicates the percentage of Direct Responses. Right: Response category distribution (Normal = left bar, Simple = right bar). Models: DeepSeek v3, GPT-4o-mini, Qwen3-32B, GPT-4o, Llama-3.1-8B. Categories: Answer Attempt, Hedge, Clarification, Refuse.

Our conversation context has a given order of entities. Due to computational constraints, we fixed the order to a single permutation for all evaluations ('0, 1, 2' for SharedRef and '0, 1' for ClearRef). We based this choice not on performance but to

Prompt / Model	Pos. 1	Pos. 2	Pos. 3
Prompt: Normal			
4o	34.50	24.52	40.98
4o-mini	41.98	18.65	39.37
v3	29.13	26.23	44.64
3-32B	31.23	29.37	39.40
3.1-8B	35.93	17.41	46.66
DPO Llama (Ours)	33.38	32.34	34.28
Prompt: Simple			
4o	37.29	15.80	46.91
4o-mini	41.67	16.46	41.87
v3	31.03	25.75	43.23
3-32B	30.35	28.36	41.29
3.1-8B	35.04	16.53	48.43
DPO Llama (Ours)	33.73	32.30	33.97

Table 2: Average selection rate (%) of an entity appearing at Position 1, 2, or 3 in the SharedRef dataset, across different models and prompts (Normal vs. Simple) in English.

Prompt / Model	Pos. 1	Pos. 2
Prompt: Normal		
4o	48.00	52.00
4o-mini	48.35	51.65
v3	43.36	56.64
3-32B	48.15	51.85
3.1-8B	48.57	51.43
DPO Llama (Ours)	50.90	49.10
Prompt: Simple		
4o	49.48	50.52
4o-mini	48.65	51.35
v3	48.04	51.96
3-32B	46.24	53.76
3.1-8B	48.51	51.49
DPO Llama (Ours)	50.00	50.00

Table 3: Average selection rate (%) of an entity appearing at Position 1, or 2 in the ClearRef dataset, across different models and prompts (Normal vs. Simple) in English.

ensure consistency across languages.

To assess the effect of this choice, we ran an ablation on the English dataset using all permutations. We observed that the frequency with which a model selects an entity depends heavily on its position in the list, indicating a strong positional bias.

Table 2 shows the distribution of selected entities across positions for each permutation in SharedRef. For example, in the Simple setting, entities at position three are selected drastically more often (avg. 42.62%) than those at position two (avg. 22.53%).

Table 3 presents analogous results for ClearRef. Here, the bias is milder, with position two being selected slightly more frequently on average (+4.22% in Normal, +3.03% in Simple).

Figure 10 shows the averaged correctness and category distribution over all permutations in English SharedRef. Compared to the fixed '0, 1, 2' ordering used in our main results, average correctness drops. Notably, GPT-4o exhibits fewer clarification attempts when averaged across permutations, while Qwen3-32B maintains strong performance.

The overall trend of higher correctness and better category distribution in the Normal setting compared to the Simple setting remains.

E Response Categorization

We adopt the response taxonomy proposed by Herlihy et al. (2024), with slight modifications to better suit our coreference resolution evaluation. Specifically, we merge *Interrogation* into *Clarification*, as both involve follow-up questions rather than direct answers. We also merge *Discussion* into *Answer Attempt*, since our evaluation does not require a fully factual answer, only that the response correctly identifies the positive entities. This simplification reduces annotation complexity without compromising the core objective of our analysis. A detailed overview of all response categories, including definitions and examples, is provided in Table 4.

F Automatic Evaluation

We used GPT-4.1-mini as an LLM judge to automatically evaluate the responses. We divided the evaluation into two parts: response classification and entity extraction. The prompt used for response classification is shown in Box 1. The prompts used for entity extraction, split into a system prompt and a user prompt, are shown in Box 2 and Box 3, respectively.

We manually annotated 500 responses from the English dataset to validate the framework. Table 5 reports the agreement rates for response categorization along with Cohen's Kappa scores. For entity extraction, we report exact match accuracy. Overall, the results show high agreement across all models.

Name	Description	Example
Answer attempt	The assistant clearly commits to at least one interpretation, providing a factual response without any hedging.	The helicopter can fly because its rotors generate lift, allowing it to rise off the ground and maneuver through the air.
Clarification	The response asks for more information about the user's intent without offering interpretations or using hedging	Could you specify which one you're referring to: the helicopter or the gnat?
Hedging	The response uses hedging or conditional language (e.g., "if you meant X...") and does not fully commit to a single interpretation. Even if only one entity is mentioned, the presence of such language marks it as a hedge.	If you're referring to a raven, it can fly due to its strong wings, lightweight body, and aerodynamic shape, which allow it to generate lift and move through the air efficiently.
Refuse	The response refuses to answer the question and does not ask any follow-up questions.	Here are the sentences: 1. **Bat**: The bat fluttered silently through the night sky, searching for insects. 2. **Dragonfly**: The dragonfly darted over the pond, its iridescent wings shimmering in the sunlight. 3. **Coffee**: The aroma of freshly brewed coffee filled the kitchen, awakening everyone's senses. 4. **Why can it fly?**: Why can it fly, despite its small wings and heavy body?"
Missing	The response is empty.	[blank]

Table 4: Description and Examples of our Response Categories.

Prompt	Response Cat.	Entity
4o-mini	100.0% (N/A)	99%
4o	100.0% (1.000)	98%
3-32B	92.0% (0.804)	98%
v3	98.0% (0.823)	94%
3.1-8B	100.0% (N/A)	100%
Total	98.0% (0.916)	97.8%

Table 5: Accuracy percentages and Cohen's Kappa scores (in parentheses) for Response Categorization and exact match accuracy for Entity Extraction across our evaluated models.

G Direct Preference Optimization

Our training set contains 472 responses from simple settings and 866 from normal settings. In addition, we included 30 basic clarification cases, where the user posed clearly ambiguous questions. A fine-grained distribution is provided in Table 6.

We fine-tuned the model for two epochs using Low-Rank Adaptation (LoRA). The full configuration for LoRA and DPO training is summarized in Table 7.

We observed performance improvements on both the SharedRef dataset and the homonym task from Ellinger et al. (2025). However, on the ClearRef test set, while the number of correct responses remained comparable to the base model, we experienced a category shift. As shown in Figure 11, the distribution of coarse response categories shifted significantly toward 'clarification' and 'hedge' across all languages. This indicates that the cognitive cost of those responses is higher for our DPO model compared to the base model on this dataset. To address this, future alignment efforts should incorporate more training examples from ClearRef to encourage direct answers where appropriate. Unlike in SharedRef, where the model successfully used common knowledge to respond only to the positive entities, in ClearRef, the model no longer consistently applies this strategy.

Dataset / Category	En	Fr	Ar	Ru	Zh
SharedRef					
Normal Answer Attempt	64	80	69	37	53
Normal Hedge	106	39	49	78	57
Normal Clarification	58	44	47	55	47
Simple Answer Attempt	112	84	30	69	76
Simple Hedge	21	13	2	15	31
Simple Clarification	4	3	1	4	1
ClearRef					
Normal Answer Attempt			2		
Normal Hedge			1		
Simple Answer Attempt			6		
General					
Clarification	6	6	6	6	6

Table 6: Distribution of chosen response types in our DPO fine-tuning dataset, broken down by language, response category, and setting.

Parameter	Value
LoRA Configuration	
r	64
LoRA Alpha	16
LoRA Dropout	0.05
Target Modules	[q_proj, v_proj, k_proj, o_proj]
Bias	none
DPO Training Configuration	
β	0.1
Learning Rate	5e-5
Batch Size (per device)	4
Epochs	2

Table 7: Combined configuration used for LoRA adaptation and Direct Preference Optimization (DPO) fine-tuning.

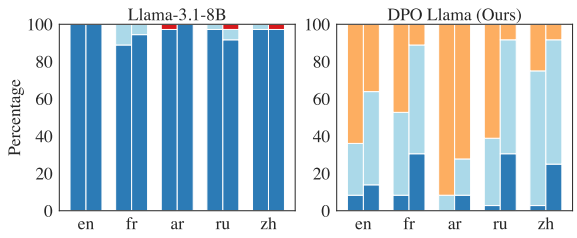


Figure 11: Distribution of response categories in the DPO test set across five languages in the ClearRef dataset. Colored squares denote response types: ■ Answer Attempt, ■ Hedge, ■ Clarification, and ■ Refuse.

Short Form	Name	Version	Access Provider
4o-mini	GPT-4o-mini	gpt-4o-mini-2024-07-18	OpenAI API
4o	GPT-4o	gpt-4o-2024-08-06	OpenAI API
4.1-nano	GPT-4.1-nano	gpt-4.1-nano-2025-04-14	OpenAI API
4.1-mini	GPT-4.1-mini	gpt-4.1-mini-2025-04-14	OpenAI API
3-32B	Qwen3-32B	N/A	OpenRouter
v3	Deepseek v3	N/A	Fireworks AI
3.1-8B	Llama-3.1-8B	N/A	Fireworks AI

Table 8: Specific model versions used in our experiments. For each model we provide the short form as used in our tables, the exact version and the access provider.

Prompt / Model	**Correct**					**Direct**				
	En	Fr	Ar	Ru	Zh	En	Fr	Ar	Ru	Zh
Prompt: Simple										
4o	98.08	**100.00**	98.08	**100.00**	96.15	88.46	69.23	**78.85**	63.46	75.00
4o-mini	**100.00**	**100.00**	**100.00**	**100.00**	**100.00**	69.23	55.77	53.85	28.85	76.92
v3	**100.00**	90.38	**100.00**	94.23	**100.00**	88.46	67.31	61.54	69.23	69.23
3-32B	**100.00**	94.23	94.23	96.15	96.15	73.08	63.46	21.15	65.38	67.31
3.1-8B	94.23	94.23	98.08	90.38	92.31	**92.31**	**78.85**	55.77	**73.08**	**80.77**
Prompt: Normal										
4o	96.15	96.15	**98.08**	**98.08**	96.15	86.54	73.08	**75.00**	**80.77**	80.77
4o-mini	**100.00**	**98.08**	94.23	**98.08**	94.23	82.69	65.38	57.69	46.15	67.31
v3	**100.00**	**98.08**	96.15	94.23	96.15	69.23	53.85	59.62	61.54	50.00
3-32B	98.08	94.23	**98.08**	90.38	94.23	82.69	51.92	51.92	67.31	63.46
3.1-8B	96.15	90.38	90.38	94.23	**98.08**	**94.23**	**78.85**	65.38	73.08	**90.38**

Table 9: Evaluation results showing the percentage of correct and direct responses across languages and prompt types on the ClearRef dataset. **Bold** highlights the highest scores per language within each prompt and metric.

Prompt / Model	**Correct**					**Direct**				
	En	Fr	Ar	Ru	Zh	En	Fr	Ar	Ru	Zh
Prompt: Simple										
4o	18.06	7.49	19.82	4.41	23.79	12.78	6.61	**12.78**	2.20	15.42
4o-mini	5.29	1.76	11.45	3.52	11.89	3.52	1.76	8.81	1.76	8.37
v3	**48.02**	**28.19**	**47.58**	**19.38**	46.70	10.57	**11.89**	6.17	3.96	7.49
3-32B	33.04	25.99	28.19	12.33	**56.83**	**23.35**	**11.89**	3.96	**7.93**	**26.87**
3.1-8B	11.89	3.96	**47.58**	4.85	4.85	4.85	0.44	2.20	1.76	1.76
Prompt: Normal										
4o	**81.06**	50.66	47.14	57.27	62.11	**36.56**	**34.80**	**30.40**	22.03	21.15
4o-mini	14.10	11.89	14.10	4.41	18.50	7.05	7.49	8.37	1.76	7.05
v3	61.23	61.23	**65.64**	31.28	49.78	3.96	9.25	10.13	3.52	1.76
3-32B	75.77	**71.37**	56.83	**63.88**	**83.26**	29.52	19.38	10.57	**25.99**	**25.11**
3.1-8B	14.54	13.66	7.49	3.96	6.17	5.73	2.20	3.96	1.32	2.64

Table 10: Evaluation results showing the percentage of correct and direct responses across languages and prompt types on the SharedRef dataset. **Bold** highlights the highest scores per language within each prompt and metric.

Box 1: User Prompt: Response Classification

```
You are reviewing a single-turn conversation between a user and an assistant.

You must classify the assistant's response according to the **type of response strategy** used.

Here are the definitions:

- `answer_attempt`: The assistant **unambiguously commits** to at least one interpretation of the
user's question and provides a factual explanation or general background relevant to the query. It
must **not use any hedging language**, such as "if you mean. . . ", "depending on. . . ", or "could
be. . . ". Implicit or explicit hedging disqualifies it from this category. It may mention multiple
entities, as long as none are introduced with hedging or conditional phrasing.

- `clarification`: The assistant **asks for more information** about the user's intent. This may
involve **one or more clarification questions**, but must not **suggest answers** or list
**interpretations**,. The response must should **not contain hedging or speculative language**.

- `hedge`: The assistant **uses hedging or conditional language** (e.g., "if you meant X. . . ",
"could refer to. . . ", "might be. . . ") and does not fully commit to a single interpretation. Even if
only one entity is mentioned, the presence of such language marks it as a hedge.

- `refuse`: The assistant explicitly or implicitly refuses to answer the question and does **not**
ask any follow-up questions.

- `missing`: The assistant gives **no response at all**.

You must output your answer in the following JSON format:
{"response_type": "refuse|missing|answer_attempt|hedge|clarification"}

Conversation:
Question: <question_str>
Answer: <asnwer_str>
```

Box 2: System Prompt: Entity Extraction

```
You are an expert in identifying explicit entity mentions in text.

Your task is to determine which of the listed entities are explicitly mentioned in the response.

Instructions:
- Return only entities from the provided list (use their exact spelling from the list in your
output).
- Consider an entity 'mentioned' if the response text contains the entity exactly, or with minor
morphological or spelling variations.
Examples of acceptable variations include plural forms and misspellings.
- Do not infer mentions from implied meaning, associations, or unrelated paraphrasing.
- Only include entities that are clearly and explicitly present in the response text.
- **Always** output the entity names exactly as they appear in the provided list.
```

Box 3: User Prompt: Entity Extractionn

```
Entities: <entity_list>
Response:

<answer_str>

Which of the listed entities are explicitly mentioned in the response?
```

Arabic Prompts

Rel. 1: الطيران؟ يمكنها لماذا
Rel. 2: حلو؟ هو لماذا
Rel. 3: الخشب؟ من مصنوع هو لماذا
Rel. 4: السباحة؟ يمكنها لماذا
Rel. 5: بسرعة؟ يعمل أن يمكن لماذا
Rel. 6: الأشجار؟ تسلق يمكنها لماذا
Rel. 7: حار؟ الجو لماذا
Rel. 8: مرتفع؟ الصوت "لماذا
Simple: بسيطة بلغة الإجابة .يرجى [..]
Starter: <entity_list> التالي من لكل واحدة جملة :أعطني

French Prompts

Rel. 1: Pourquoi peut-il voler ?
Rel. 2: Pourquoi est-ce sucré ?
Rel. 3: Pourquoi est-il en bois ?
Rel. 4: Pourquoi peut-il nager ?
Rel. 5: Pourquoi peut-il fonctionner rapidement ?
Rel. 6: Pourquoi peut-il grimper aux arbres ?
Rel. 7: Pourquoi fait-il chaud ?
Rel. 8: Pourquoi est-ce si bruyant ?
Simple: [..] Veuillez répondre dans un langage simple.
Starter: Donnez-moi une phrase pour chacun des mots suivants : <entity_list>

English Prompts

Rel. 1: Why can it fly?
Rel. 2: Why is it sweet?
Rel. 3: Why is it made of wood?
Rel. 4: Why can it swim?
Rel. 5: Why can it run fast?
Rel. 6: Why can it climb trees?
Rel. 7: Why is it hot?
Rel. 8: Why is it loud?
Simple: [..] Respond in simple language.
Starter: Provide me one sentence for each of the following: <entity_list>

Chinese (Zh) Prompts

Rel. 1: 它为什么能飞？
Rel. 2: 为什么是甜的？
Rel. 3: 为什么它是木制的？
Rel. 4: 为什么它能游泳？
Rel. 5: 为什么它能跑得这么快？
Rel. 6: 为什么它能爬树？
Rel. 7: 为什么会热？
Rel. 8: 为什么这么吵？
Simple: [..] 请用通俗易懂的语言回答。
Starter: 请为以下每个项目提供一句描述： <entity_list>

Russian Prompts

Rel. 1: Почему он может летать?
Rel. 2: Почему он сладкий?
Rel. 3: Почему он сделан из дерева?
Rel. 4: Почему он может плавать?
Rel. 5: Почему он может быстро бегать?
Rel. 6: Почему он может лазать по деревьям?
Rel. 7: Почему так жарко?
Rel. 8: Почему так громко?
Simple: [..] Пожалуйста, отвечайте простым языком.
Starter: Дайте мне по одному предложению для каждого из следующих слов: <entity_list>

Figure 12: Language Versions of Relation Questions, the Simple Instruction and the Starter Sentence in Arabic, French, English, Chinese, and Russian

Read Your Own Mind: Reasoning Helps Surface Self-Confidence Signals in LLMs

Jakub Podolak
University of Amsterdam
jakub.podolak.241 [at] gmail.com

Rajeev Verma
University of Amsterdam

Abstract

We study the source of uncertainty in DeepSeek R1-32B by analyzing its self-reported verbal confidence on question answering (QA) tasks. In the default answer-then-confidence setting, the model is regularly over-confident, whereas semantic entropy—obtained by sampling many responses—remains reliable. We hypothesize that this is because of semantic entropy's larger test-time compute, which lets us explore the model's predictive distribution. We show that granting DeepSeek the budget to explore its distribution by forcing a long chain-of-thought before the final answer greatly improves its verbal score effectiveness, even on simple fact-retrieval questions that normally require no reasoning. Our analysis concludes that reliable uncertainty estimation requires explicit exploration of the generative space, and self-reported confidence is trustworthy only after such exploration.

1 Introduction

Generative language models (GLMs) like GPT, LLaMA, or Deepseek families have achieved great performance on diverse tasks (Dubey et al., 2024; DeepSeek-AI et al., 2025), yet they are prone to failure modes such as "hallucinations" (Huang et al., 2023). These inaccuracies can undermine trust and lead to poor decisions in LLM-assisted systems (Huang et al., 2024). To mitigate this issue, quantification and the communication of model's uncertainty in its outputs is seen as a potential to entrust these models with reliability.

Numerous uncertainty-quantification (UQ) approaches have been proposed in this direction: from Monte-Carlo sampling based, such as Semantic Entropy (SE) (Farquhar et al., 2024) to simpler Verbalized Confidence estimation (VC), which just asks the model directly to state its confidence (Xiong et al., 2024).

While Verbalized Confidence estimation is easy to use, there is no scientific consensus on what it represents or what its source is, potentially making it unreliable to use in critical scenarios. Furthermore, prior work has shown that Verbalized Confidence is often pathological - for the same question a model might first return "Answer A (100 % confidence)" and, in a second sample, "Answer B (95 % confidence)," even though these probabilities cannot coexist (Xiong et al., 2024). On a more practical side, verbalized scores may be over-confident (Yang et al., 2024; Pawitan and Holmes, 2024), whereas Semantic Entropy remains comparatively well calibrated (Farquhar et al., 2024).

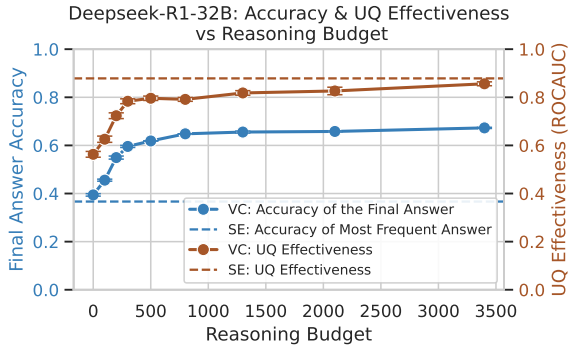


Figure 1: **DeepSeek R1-32B's Verbalized Confidence (VC) improves and matches Semantic Entropy's (SE) effectiveness, when longer reasoning is forced.**. Our work suggests that it is the test-time exploration of the model's predictive space, not the particular uncertainty heuristic, that makes confidence estimates reliable.

Semantic Entropy's effectiveness can be attributed to its test-time compute—allocating extra tokens at inference (Snell et al., 2024) to explore the predictive distribution. Test-time compute can also come in the form of an extended reasoning chain that precedes the final answer (Wei et al., 2023; DeepSeek-AI et al., 2025), and most recent works show that such reasoning can improve verbalized or token-level calibration as well (Zeng et al., 2025; Jurayj et al., 2025). These findings prompted us to pose a hypothesis: Is the model

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 247–258
November 9, 2025 ©2025 Association for Computational Linguistics

able to directly quantify and express verbally its uncertainty, or is the test-time token exploration necessary for the model to reliably summarize its confidence?

In this paper, we try to answer this question and better understand the source of VC by performing a set of experiments with `DeepSeek R1-32B` (DeepSeek-AI et al., 2025) as a representative model. Our results show that without any chain-of-thought, `DeepSeek`'s verbalized scores carry little information about correctness. As we grant the model progressively larger reasoning budgets, its calibration improves and approaches the reliability of Semantic Entropy, even on simple fact-retrieval items. This trend suggests that meaningful uncertainty estimates emerge only after the model's predictive space has been explored, and that the final confidence percentage largely summarizes the diversity exposed in this process. We further enforce this hypothesis by using a separate reader model that, by just analyzing `DeepSeek`'s reasoning trace, matches the reliability of `DeepSeek`'s own Verbalized Confidence.

2 Background and Related Work

Generative language models frequently generate fluent but incorrect answers that can cause downstream harm (Band et al., 2024; Huang et al., 2024). When no external verifier is available, a model's self-reported confidence is the only proxy for correctness, making reliable uncertainty estimates essential.

Calibration of LLM Confidence Scores. A confidence score is *calibrated* if, for example, predictions tagged "80 % confident" are correct roughly 80 % of the time. Common approaches to obtain the confidence scores include token-level probabilities treated as a classification score (Dhuliawala et al., 2022), semantic-level measures that evaluate agreement across multiple sampled completions (Farquhar et al., 2024), and explicitly verbalized percentages in a model's output (Xiong et al., 2024; Tian et al., 2023).

Semantic Entropy vs. Verbalized Confidence A generative model, given a question Q, defines a distribution over semantically distinct answers $P(A \mid Q)$. The uncertainty of this distribution is naturally quantified by its Shannon entropy, and while computing it exactly is infeasible, we can approximate it by Monte-Carlo sampling and clustering semantically equivalent answers. This is exactly how the Semantic Entropy (SE) method (Farquhar et al., 2024) works, leading to well-calibrated scores. The big downside of this method is that it requires sampling data on test-time (larger test-time-compute budget).

Another way to obtain a confidence score is simply to ask the model for one, for instance, "I am 85 % sure." This Verbalized Confidence (VC) is easy to collect and works with any black-box API (Xiong et al., 2024; Yang et al., 2024; Ni et al., 2024). Yet, opposed to SE, it is unclear what the number represents: is the model sampling its own distribution, recalling similar training examples, or just guessing? To our best knowledge, no study has answered these questions, leaving the method too uncertain for safety-critical use.

The most recent works find that reasoning-tuned models that generate more tokens at the test time give better calibrated verbalized score (Hammoud et al., 2025; Wei et al., 2024; Xiong et al., 2024; Zhao et al., 2024), This hints that exploring test-time compute budgets' impact on Verbalized Confidence calibration might be crucial to understand its source, yet we are not familiar with any research work that tries to answer our questions specifically.

In this work, we systematically compare Verbalized Confidence and Semantic Entropy under matched test-time compute budgets, examine several task domains, and analyze the reasoning trace to see where the verbalized score comes from and why it lags behind Semantic Entropy.

3 Methodology

Our objective is to uncover where a model's Verbalized Confidence comes from. We identify two competing views:

Intrinsic latent variable: the model can read out a hidden latent belief state and use it to express its uncertainty, and **Self-sampling:** model does not have access to any reliable latent source of confidence, and reliable confidence emerges only after the model explicitly explores its own predictive space, as Semantic Entropy does by sampling many answers.

We test these views through a set of experiments that measure the behavior of VC when the model is forced to reason before answering, compare the effectiveness and accuracy to the SE, and analyze the uncertainty exposed in the reasoning traces. We describe our experimental setup in Appendix A.

4 Results

Without any reasoning tokens, the score is barely better than random, and with enough exploration budget, VC can approach SE's effectiveness (subsection 4.1). Furthermore, an external reader can recover essentially the same uncertainty signal by inspecting the chain of thought alone (subsection 4.2), suggesting that the Self-sampling hypothesis might be true.

4.1 Extended Reasoning is Necessary for VC to Reach SE-Level Effectiveness

Figure 3 shows final-answer accuracy, UQ effectiveness, and average stated confidence for correct and incorrect answers as a function of the reasoning budget, with Semantic Entropy shown for comparison. We see that granting just 100–500 reasoning tokens raises accuracy 41% $\rightarrow$ 63% and boosts verbalized-confidence ROC-AUC 0.56 $\rightarrow$ 0.80.

For fact-retrieval questions (Fig. 3b), answer accuracy does not improve with longer reasoning budgets, yet UQ effectiveness continues to improve with additional tokens. We can reach very long reasoning traces for fact-retrieval questions thanks to the employed forced reasoning technique (Muennighoff et al., 2025) presented in Figure 4.

Verbalized Confidence is initially weaker than superior Semantic Entropy but reaches near parity at 200 tokens for fact retrieval and 3,500 tokens for mathematical items, while maintaining higher answer accuracy due to the reasoning process. For comparison, in our experiments, SE used 218 tokens per sample on average, meaning the two methods are very similar both in computational efficiency and UQ effectiveness for fact retrieval questions.

These results confirm that allocating test-time compute to reasoning is essential for reliable uncertainty estimates, and extended CoT effectively mitigates DeepSeek's over-confidence without sacrificing performance. The sheer scale of the improvement in effectiveness: from near-random 0.56 ROCAUC to 0.88 suggests that there is no latent uncertainty information available for the model, and self-sampling is necessary to obtain a good uncertainty estimate.

4.2 External Reader Model Recovers VC Calibration from Reasoning Trace Alone

If there is no hidden latent variable from which Verbalized Confidence is drawn, then the reasoning trace has to contain all the uncertainty information needed to explain Deepseek's final score. We can verify it using an external reader model that, given only DeepSeek's chain of thought, tries to predict its final answer and confidence.

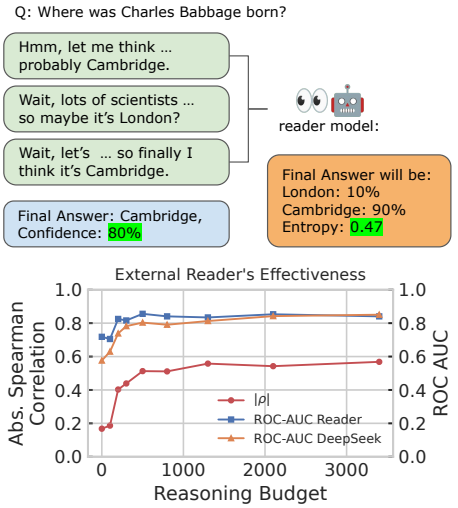


Figure 2: **Separate reader matches the reliability of DeepSeek's own Verbalized Confidence by just looking at the reasoning trace.** With more reasoning tokens, the agreement between them (measured as absolute Spearman correlation) increases, and the effectiveness of both scores changes similarly.

Figure 2 illustrates our experimental setup and results. As a reader model, we used OpenAI's GPT-4o-mini (OpenAI, 2024), we provide more information about the setup in Appendix C. We display (i) the absolute Spearman correlation $|\rho|$ between DeepSeek's self-reported confidence and the reader entropy H_{reader}, and (ii) the ROC-AUC of each score in detecting incorrect answers.

With no reasoning tokens exposed, the correlation between Reader's and Deepseek's scores is low, however, with more reasoning tokens, the effectiveness of the reader goes up in tandem with Deepseek's effectiveness, and the correlation between the two goes up. At 3.4 k tokens, DeepSeek reaches ROC-AUC = 0.851 and the reader 0.841 with $|\rho| = 0.57$, indicating that almost the entire confidence signal is now accessible in the trace.

These results support our claim that there is no directly accessible notion of uncertainty, and uncertainty information must be surfaced through test-time token sampling. When the model provides a Verbalized Score after the reasoning process, it most likely just reads its reasoning trace and summarizes the alternatives and uncertainty exposed in it.

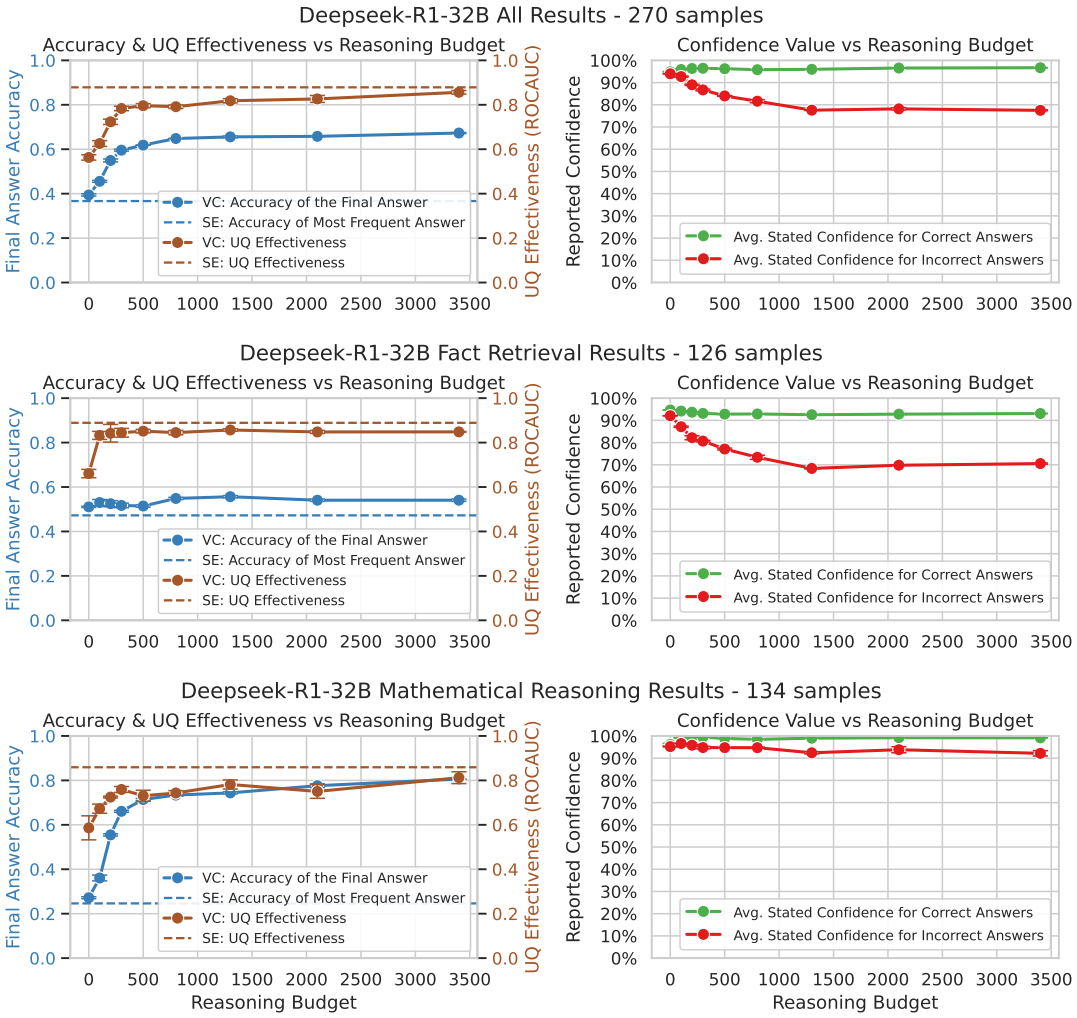


Figure 3: **Effectiveness and Accuracy of Verbalized Confidence with Forced Reasoning vs Semantic Entropy.** (a) Full overview. (b) Fact retrieval results. (c) Mathematical reasoning results. Note: The remaining 10 samples not falling into the Fact Retrieval or Mathematical Reasoning categories are included in the Full overview but not presented as separate plots.

5 Discussion and Future Work

We aimed to determine whether large language models can directly verbalize well-calibrated uncertainty or whether reliable confidence estimates only emerge after explicit exploration of their predictive space, via additional test-time compute such as parallel sampling (Semantic Entropy) or extended reasoning.

Our experiments suggest that *test-time compute, not the particular uncertainty heuristic, is the decisive factor for obtaining reliable confidence estimates in DeepSeek-R1-32B*. Left to produce only a short answer, the model remains over-confident because its belief state cannot be accessed directly. Granting the model additional tokens, either by sampling independent continuations (Semantic Entropy) or by forcing a longer chain of thought, allows it to externalize alternative hypotheses. This exposes a big issue with Verbalized Confidence - its appeal lies in the simplicity and how fast it is, yet it works well only after a significant reasoning computation is done.

While these results are encouraging, they may not be generalizable since we've tested only one model and used a very compact QA dataset. Furthermore, assuming it is true that test-time compute is the decisive factor for reliable UQ, it still might be the case that some methods of test-time compute may be more efficient in eliciting uncertainty than others. Future work could focus on making models reason more efficiently or explore their uncertainty in a more structured way. That could help Verbalized Confidence inherit the Semantic-Entropy-level of calibration with less computation needed.

References

AIME. 2024. Aime 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024. Accessed: 2025-05-07.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic calibration of long-form generations. *Preprint*, arXiv:2404.00474.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Shehzaad Dhuliawala, Leonard Adolphs, Rajarshi Das, and Mrinmaya Sachan. 2022. Calibration of machine reading systems at scale. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1682–1693, Dublin, Ireland. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

S. Farquhar, J. Kossen, L. Kuhn, and Y. Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Hasan Abed Al Kader Hammoud, Hani Itani, and Bernard Ghanem. 2025. Beyond the last answer: Your reasoning trace uncovers more than you think. *Preprint*, arXiv:2504.20708.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Preprint*, arXiv:2009.03300.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2023. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *Preprint*, arXiv:2311.05232.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. triviaqa: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv e-prints*, arXiv:1705.03551.

William Jurayj, Jeffrey Cheng, and Benjamin Van Durme. 2025. Is that your final answer? test-time scaling improves selective question answering. *Preprint*, arXiv:2502.13962.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. 2024. Are large language models more honest in their probabilistic or verbalized confidence? *Preprint*, arXiv:2408.09773.

OpenAI. 2024. Hello, gpt-4o. Accessed: 2025-01-05.

Yudi Pawitan and Chris Holmes. 2024. Confidence in the reasoning of large language models. *Preprint*, arXiv:2412.15296.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *Preprint*, arXiv:2305.14975.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. *Preprint*, arXiv:2411.04368.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *Preprint*, arXiv:2306.13063.

Daniel Yang, Yao-Hung Hubert Tsai, and Makoto Yamada. 2024. On verbalized confidence scores for llms. *Preprint*, arXiv:2412.14737.

Qingcheng Zeng, Weihao Xuan, Leyang Cui, and Rob Voigt. 2025. Do reasoning models show better verbalized calibration? *Preprint*, arXiv:2504.06564.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. Fact-and-reflection (FaR) improves confidence calibration of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8702–8718, Bangkok, Thailand. Association for Computational Linguistics.

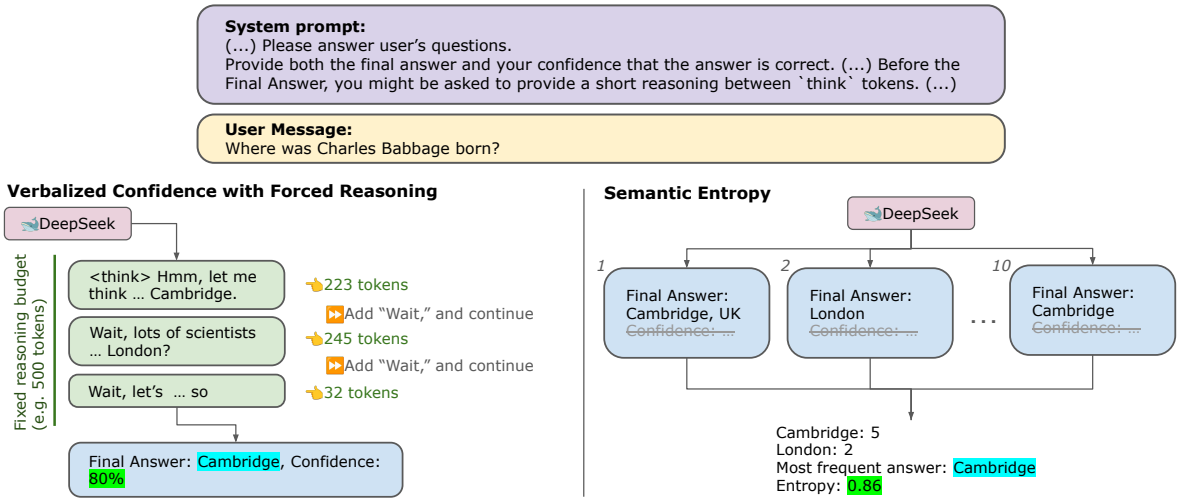


Figure 4: **Two tested methods of obtaining Final Answer and Confidence** - Verbalized Confidence with Forced Reasoning (VC) works by prompting the model to reason for longer-until the fixed budget is exhausted - before stating the answer and confidence. Semantic Entropy (SE) obtains 10 independent answers that are later clustered semantically to identify the most frequent one, and to calculate the entropy in the answer distribution.

A Experimental Setup

Data Sources. Because long-trace experiments are computationally expensive, we built a small (270 samples) but diverse benchmark instead of using full datasets. We sampled questions from five popular, open-source sources: TriviaQA, MMLU, and SimpleQA for fact retrieval (Joshi et al., 2017; Hendrycks et al., 2021; Wei et al., 2024), plus GSM8K and AIME-2024 for mathematical reasoning (Cobbe et al., 2021; AIME, 2024). Our goal is open-ended QA in natural language, so we stripped away multiple-choice options in MMLU and any figure references in AIME-2024, manually discarding questions that could not stand alone after this edit, such as *"Which of the following is true?"*. Every surviving example was then hand-labeled with its knowledge domain and the skills needed to answer it, such as "Fact Retrieval" or "Mathematical Reasoning". Full sampling details and the final label distribution appear in Appendix B.

Model and Prompts. We chose `Deepseek-R1-32B`[1] (DeepSeek-AI et al., 2025) following Jurayj et al. (2025) for its strong reasoning capabilities at a manageable model size. Furthermore, it is one of the most popular open-sourced reasoning-tuned models. All experiments were run on two NVIDIA A100 GPUs.

We provide an elaborate discussion on prompting and inference we adopted in Figure 4. Across all setups, we used a single system prompt that directs the model to (1) think step by step, and then (2) provide a final answer along with a Verbalized Confidence score. The full prompt text, as well as an interaction example, is available in Appendix C. To regulate the length of the reasoning chain, we applied the budget-based truncation method of Muennighoff et al. (2025): when the reasoning budget is exhausted (or set to zero), the chain terminates immediately. If the budget remains, the system appends "Wait, " tokens, and asks to generate more tokens. For experiments with Verbalized Confidence, we lowered the decoding temperature to 0.1 to prevent the model from going off-topic in long reasoning. For parallel sampling in Semantic Entropy experiments, we set it to 1.0 to obtain more diverse responses and approximate the predictive distribution more efficiently.

UQ Methods. Next, we describe how we obtain the estimates of verbalized score and semantic entropy: For Verbalized Confidence, we ask the model to provide the final answer and its confidence between 0% and 100% after (optional) forced reasoning (refer to Figure 4 for visualization).

For Semantic Entropy, we follow Farquhar et al. (2024), and generate $n = 10$ answers for each question with no reasoning chain. Afterwards, we use OpenAI's gpt-4o-mini (OpenAI, 2024) to cluster semantically equivalent generations. We select the majority cluster – the cluster with the most members (i.e., the answer

[1] `deepseek-ai/DeepSeek-R1-Distill-Qwen-32B`

that appears most frequently once semantically equivalent responses are grouped) – as the predicted answer and compute the Shannon entropy of the cluster-size distribution as the uncertainty score.

Evaluation and Metrics. We report two main metrics: the accuracy of the final answer and the effectiveness of the UQ method measured as the area under ROC (ROC AUC) in the task of classifying the model's final answer correctness (hallucination classification). To calculate if the model's final answer is correct, we query OpenAI's GPT-4o-mini (OpenAI, 2024) model with the question if this proposed answer is equivalent to the ground truth answer in the dataset given a question.

We repeat experiments with verbalized score confidence across varying reasoning budgets 3 times and show mean and 95% confidence intervals. For the rest of the experiments, we repeat them once unless stated otherwise.

B Data Creation and Composition

Because our longest-trace runs are expensive, we limited the benchmark to **270 open-ended questions** drawn from five well-known, permissively licensed QA datasets. We first sampled 310 items uniformly at random (seed 42) to balance fact-retrieval and mathematical-reasoning content while keeping the total below the ≈300-sample budget we could process. Items whose solutions required figures (AIME-2024), multiple-choice candidates (MMLU), or extra context passages (TriviaQA) were discarded after manual inspection, leaving the 270 used in all experiments (Table **??**). By doing so, we ensured that all the incorrect answers were caused by the model's mistakes, instead of missing context in the data.

Each example received two human labels - Knowledge Domain, and Skill Required. A large language model (OpenAI o3) proposed initial tags for 100 random questions. The first author then reviewed every instance, correcting tags where needed, and used these tags to manually label all 270 samples. You can find the specific tags and number of datapoints in Figure 5. In the main paper, we break out results for the full dataset and for the two most common *Skill Required* tags only; *Knowledge Domain* splits are omitted because several categories are too small. Per-dataset results can be found in Appendix D. Five representative questions and their tags are shown in Table 1.

Example Question (truncated)	Dataset	Skill	Domain
In what year did Augustus De Morgan publish the article "Trochoidal Curve" in the Penny Cyclopaedia?	SimpleQA	Fact Retrieval	History and Past Events
There exist real numbers x and y, both greater than 1, such that $\log_x(y^x) = \log_y(x^{4y}) = 10$. Find xy.	AIME2024	Mathematical Reasoning	Mathematics
James runs 12 miles a day for 5 days a week. If he runs 10 miles an hour how many hours does he run a week?	GSM8K	Mathematical Reasoning	Mathematics
In Python 3, which of the following function removes all leading and trailing whitespace in string?	MMLU	Fact Retrieval	IT and Engineering
Anaphylaxis is what sort of life-threatening illness?	TriviaQA	Fact Retrieval	Science, Nature and Medicine

Table 1: **Five representative items from the 270-question benchmark.**

C Prompts and Inference

Main VC vs SE runs. You can find the full system prompt used in the main VC vs SE experiments, as well as a real interaction example with Verbalized Confidence and budget forcing in Figure 6.

Reader–model experiments. Our goal is to let an external model read DeepSeek's reasoning trace and predict a probability distribution over possible DeepSeek's answers. After obtaining the distribution,

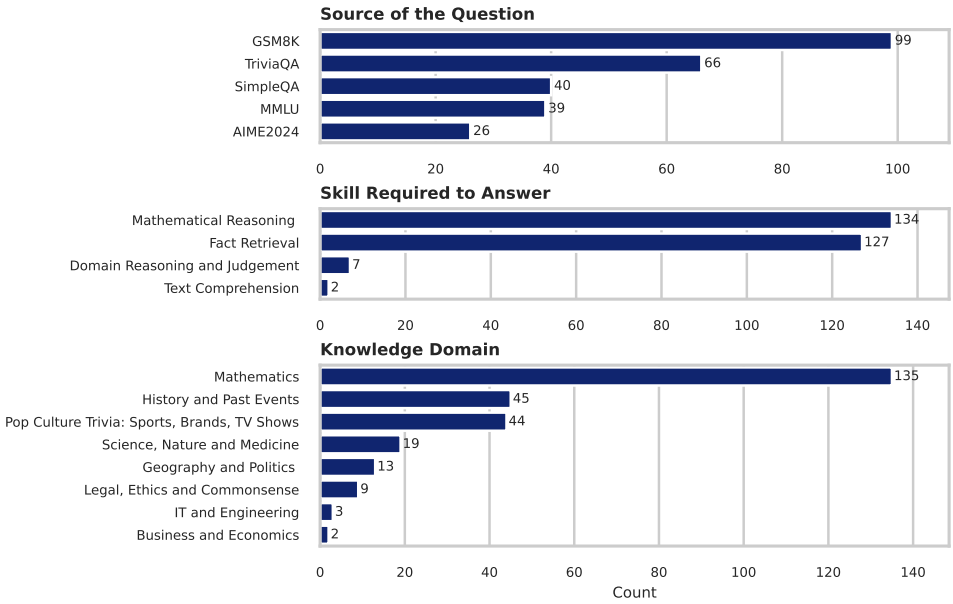


Figure 5: **Internal composition of our used data sample.**

System prompt:
You are a trivia expert, please answer user questions. Provide both the final answer and your confidence that the answer is correct. Be honest and precise in your confidence estimation, as the users will rely on this information to make decisions. Try not to overestimate your confidence. Answer in the format: `Final Answer: <final_answer>. Confidence: <confidence>` where <confidence> is a percentage between 0% and 100%, and <final_answer> is just the final answer in one brief sentence.

Before the Final Answer, you might be asked to provide a short reasoning between `think` tokens. In your reasoning process, try to constantly reevaluate your confidence,
by considering the information you have and the information you are missing or may have calculated wrong. Reason not only about the final answer, but also about your confidence.

User Message:
The Fitzwilliam Museum is in which British town or city?

Model response:
<think> Okay, so I need to figure out where the Fitzwilliam Museum is located. I remember hearing about it before, but I'm not exactly sure. Let me think. I know there are several famous museums in the UK, like the British Museum in London and the Victoria and Albert Museum, also in London. But the Fitzwilliam Museum doesn't ring a bell as being in London. Maybe it's in another city.\n\nI think Cambridge is known for its universities, especially the University of Cambridge. There are a lot of museums there associated with the university. I recall that the Fitzwilliam Museum is one of them. It's a major art museum, right? They have a lot of paintings, sculptures, and other artifacts. I'm pretty sure it's in Cambridge because I've seen pictures of it before, and it's often mentioned alongside other Cambridge landmarks.\n\nWait, could it be in Oxford? No, Oxford has the Ashmolean Museum, which is also a big one. So, no, the Fitzwilliam is definitely in Cambridge. I'm pretty confident about that. I don't think it's in any other city like Manchester or Birmingham. It's specifically linked to the University of Cambridge, so that must be the location.\n\nI don't have any conflicting information, so I think I'm safe to say it's in Cambridge. My confidence is high because I've heard it mentioned in the context of Cambridge and its university.\nWait, just to double-check, I</think>
Final Answer: Cambridge. Confidence: 95%

Figure 6: **Used system prompt and real interaction example**. The question comes from the TriviaQA dataset. Moments where the model was forced to reason for longer are highlighted.

Dataset	Sampled	After Filtering	Removed
GSM8K	100	99	1
TriviaQA	70	66	4
SimpleQA	40	40	0
MMLU	70	39	31
AIME2024	30	26	4
Total	**310**	**270**	**40**

Table 2: **Number of samples per dataset before and after manual filtering.**

we calculate Shannon entropy, which is used as a notion of uncertainty. The procedure of obtaining a distribution over possible DeepSeek's answers has four steps:

1. **Candidate extraction.** For each question, we feed the entire 3.4k-token reasoning chain to `gpt-4o-mini`, prompting it to list all candidate answers mentioned in the trace.

2. **Multiple-choice reformulation.** We label the distinct candidates with letters `A`, `B`, `C`..., we also add an option "None Unknown".

3. **One-token completion.** We prompt `gpt-4o-mini` again, asking which of the candidates is most likely to be returned by DeepSeek as a final answer. We ask the API for exactly one letter `A`, `B`, `C`... as a response. We keep the logits of this single-token prediction, apply `softmax`, and obtain a categorical distribution $p(A), p(B), \ldots$.

4. **Reader entropy.** The Shannon entropy of that distribution, $H_{\text{reader}} = -\sum_i p(i) \log p(i)$, is the reader's uncertainty score used in our analyses.

You can find prompts that we used to extract the list of all candidates, as well as the final prediction, in figure Figure 7.

D Results Across Datasets

We provide additional comparison of VC and SE with varying reasoning budgets across different sources of data in Figure 8. While the trends are much noisier because of fewer samples, we see that for all the sources the difference in confidence between correct and incorrect answers increases with more reasoning tokens.

Possible-candidates prompt

"You are a helpful assistant. We asked a person to answer an open-ended question. The person wrote a reasoning trace and then gave a final answer. I want to know what all the possible options were that the person considered before giving the final answer. This is for research on the answers people consider. I want to use these possible answers to create a multiple-choice question. Please give me all the possible answers the person considered, without duplicates, keeping the answers distinct and suitable for a multiple-choice question. Respond as a JSON list of strings, like this: FINAL LIST: ["answer1", "answer2", ...], which I can later use in a quiz. Add the correct answer, the final answer, and Other / Unknown to the list, making sure there are no duplicates. Reason about possible answers and how to remove duplicates, before giving the final list. Include only answers mentioned in the reasoning trace plus the correct answer. Do not add answers that are missing from the trace or the correct / unknown items."

Final-prediction prompt

"You are a helpful assistant. We asked a person to answer a question; the person started reasoning about the possible answers. Your role is to serve as an autocompletion model that predicts the most likely final answer the person will give, based on the current reasoning trace. You are not allowed to give your own answer—only the person's most likely final answer. Choose from the list of possible answers (A, B, C, ...) and output exactly one letter, with no whitespace or comments."

```
+ A) {Option A}
+ B) {Option B}
+ C) {Option C}
...
```

Figure 7: **Reader-model prompts**. First prompt is used to extract candidate answers from the reasoning trace, the second one shows the follow-up prompt that asks the reader to predict DeepSeek's final choice. The last block of the second prompt is filled dynamically with the candidate list produced by the first one.

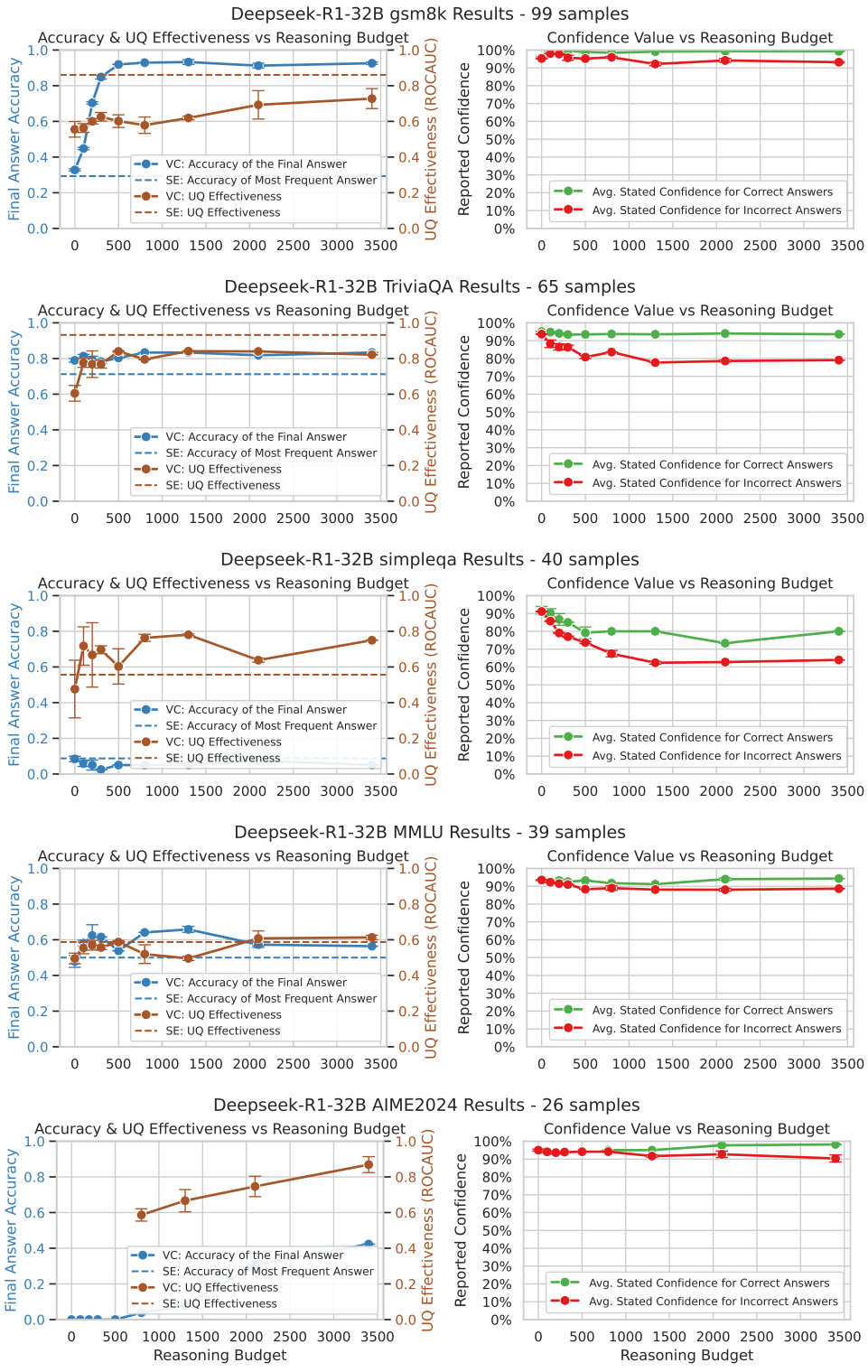


Figure 8: **Effectiveness and Accuracy of Verbalized Confidence with Forced Reasoning vs Semantic Entropy.** Despite noise from limited samples, the right-hand plots show a consistent and increasingly pronounced divergence in reported confidence between correct and incorrect answers as the reasoning budget increases.

Calibrating Language Models for Neural Ranking under Noisy Supervision with Relaxed Labels

Arnab Sharma
Heinz Nixdorf Institute
Paderborn University
Paderborn, Germany
arnab.sharma@upb.de

Daniel Vollmers
Heinz Nixdorf Institute
Paderborn University
Paderborn, Germany
daniel.vollmers@upb.de

Axel-Cyrille Ngonga Ngomo
Heinz Nixdorf Institute
Paderborn University
Paderborn, Germany
axel.ngonga@upb.de

Abstract

In recent years, we have seen an increased usage of neural ranking models in the information retrieval domain. Although language model-based rankers have shown significant progress in performing ranking tasks, little to no work has addressed the issue of fine-tuning them in the presence of label noise in the training data. In a general learning setting, training models in the presence of noisy labeled data is studied extensively. To this end, *confidence calibration* approaches have shown significant promise; however, their usage in training neural ranking models is relatively less studied. In this work, we address this gap by adapting and analyzing regularization-based calibration approaches to reduce the effect of label noise in ranking tasks. Specifically, we study *label relaxation* in neural ranking models. We demonstrate the effectiveness of this approach by performing extensive evaluations comparing the label relaxation approach to standard loss functions. Additionally, we analyze the calibration error associated with the loss functions. After evaluating on five different noise levels, two different ranking models, and four diverse ranking datasets, the results suggest that label relaxation can improve the performance of the ranking models under noisy labels. Furthermore, we find that label relaxation reduces calibration error, although it suggests a better metric to be used for neural ranking models.

1 Introduction

The advancements of language models have enabled their rapid usage in various application domains. One of such prominent application areas is *neural ranking* wherein the task is to estimate the relevance of several candidate documents or entities based on their relevance to the given *query* (Reimers and Gurevych, 2019; Nogueira and Cho, 2019), which is typically a question presented in a natural language form. With the recent progress in NLP domains, models like BERT (Devlin et al., 2019) have achieved significant progress in capturing the semantic contextual information for a given query. Existing works focus on improving the ranking tasks considering several aspects of the learning framework (Sil et al., 2018; Yamada et al., 2020; Ganea and Hofmann, 2017; Fang et al., 2019; Zhang et al., 2020). However, to the best of our knowledge, only a few works considered approaches to develop *robust* ranking models when noisy labels are prevalent in the training data.

Label noise in the training data for ranking tasks can be caused due to several reasons. For instance, in a question answering dataset, noise can stem from distant supervision, weakly supervised data generation, bad annotations, among other reasons. Such noise can essentially lead to the generation of models with degraded generalization and unstable predictions (Liu and Tao, 2016; Natarajan et al., 2013; Patrini et al., 2017). This issue becomes particularly critical in ranking tasks, where the quality of predictions directly impacts the rank order of documents or entities, thereby affecting the overall effectiveness of the system. Furthermore, studying the risks associated with label noise in ranking models is also important, as improper handling of noise can lead to misleading rankings and reduced model reliability. One of the few works in the NLP domain by Zhu et al. (2022) studied the robustness of the BERT model and showed that in sentence classification tasks, weakly supervised noise can severely degrade the performance of the model. In classification and general learning settings, this problem has been tackled often by using several types of model calibration approaches by Zhu et al. (2021); Wei and Liu (2021); Ding et al. (2021); Cheng and Vasconcelos (2022); Ghosh et al. (2022); Moon et al. (2020); Ma and Blaschko (2021); Liu et al. (2022); Lienen and Hüllermeier (2021, 2024). These approaches typically work by ensuring that the *confidence* of the underlying model in predicting an

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 259–272
November 9, 2025 ©2025 Association for Computational Linguistics

input instance should also reflect the true likelihood of the prediction. In other words, the model should not confidently predict wrong labels, and in contrast, when predicting the correct labels, it should exhibit sufficient confidence. The idea is to calibrate the overconfident models, which are vulnerable to memorizing incorrect labels (Guo et al., 2017). Label smoothing (Szegedy et al., 2016) is considered a standard approach, wherein the idea is to distribute a specific amount (decided based on a hyperparameter) of probability mass taken from the actual label to all the other labels. Although label smoothing can be quite effective, it still relies on precise probabilistic labels, which might degrade the generalization performance (Li et al., 2020). Therefore, Lienen and Hüllermeier (2021) proposed *label relaxation*, which considered a set of candidate distributions, instead of a single smoothed distribution. Label relaxation essentially replaces fixed (and possibly incorrect) label distributions with sets of plausible distributions, thereby allowing the learner to learn a bounded range of acceptable target labels.

In this paper, we tackle label noise in order to develop robust ranking models in the fine-tuning step. We consider two different directions, considering model calibration techniques. Firstly, we introduce label relaxation into the ranking paradigm as a principled approach to fine-tune models under noisy conditions. More specifically, considering the pairwise ranking loss, we integrate relaxation in several widely used neural ranking models. Then we compare the performance of two different calibration approaches, i.e., smoothing with relaxation, to gain some initial insights into which approach performs better. Secondly, by analyzing the calibration error, we aim to understand how well the models' confidence reflects their true performance under noisy conditions. Thus, we assess the associated risks of poor confidence calibration, which can lead to suboptimal ranking decisions. We model the label noise in the ranking tasks by considering a *proximity-aware* approach. Experimental results considering these two different calibration approaches, 5 different noise levels, 4 diverse datasets, and two ranking models suggest the potential of relaxation under label noise in fine-tuning ranking models. Our contributions can be summarized as,

- We introduce label relaxation to perform calibration for ranking models under the presence of label noise.
- We formally define the relaxation considering the pairwise ranking loss.
- We evaluate the performance of label relaxation considering 5 different noise levels.
- We give a comparative analysis comparing label relaxation to the standard calibration approach, label smoothing.
- We analyze the calibration error to understand the risks associated with two calibration approaches in the presence of label noise.
- We make the code publicly available [1].

2 Related Work

Ranking models As mentioned beforehand, with the advancement of language models, we have seen significant progress in the domain of neural ranking (Reimers and Gurevych, 2019; Nogueira and Cho, 2019; Déjean et al., 2024; Zhang et al., 2022; Wu et al., 2020a). One of the first works was Sentence-BERT (Reimers and Gurevych, 2019), which adapted the BERT architecture into a Siamese network to produce sentence-level embeddings. Nogueira and Cho (2019) extended this idea by further showing that BERT-based models could be fine-tuned specifically for passage re-ranking. This has shown substantial improvements in retrieval performance. Subsequent work has continued to explore more scalable and generalizable ranking solutions. For example, Wang et al. (2022) introduced a family of embedding models trained with contrastive learning on massive collections of text pairs. *Cross encoders* (Déjean et al., 2024) are shown to outperform the previous approaches in re-ranking tasks at the cost of a high training time.

Calibrated Loss Calibration refers to the alignment between the model's predicted confidence and the actual likelihood of correctness. A perfectly calibrated model assigns a probability of 0.7 to a prediction if, on average, 70% of such predictions are correct. There exist two categories of approaches that perform model calibration, (i) *post-hoc* and (ii) *regularization-based*. In order to perform calibration, post hoc approaches adjust the output predictions (Cheng and Vasconcelos, 2022; Wei et al., 2022; Hebbalaguppe et al.,

[1] https://github.com/dice-group/RobustRanking/tree/label-relaxed-ranking

2022). However, this requires additional validation on held-out datasets. Furthermore, this approach assumes the training and test distributions to be the same, which often is not. Regularization-based approaches do not require any extra data and perform calibration during the training step while computing the loss (Cheng and Vasconcelos, 2022; Wei et al., 2022; Hebbalaguppe et al., 2022). Label smoothing is often used as a standard technique to soften the hard target labels by redistributing the probability mass to non-target labels (Szegedy et al., 2016; Müller et al., 2019). However, typical smoothing distributes the probability mass uniformly. There exist approaches that essentially follow more advanced approaches, such as bootstrapping techniques (Reed et al., 2015), wherein a self-supervised approach is used to distribute the probability mass. Self-distillation and model distillation approaches also follow a similar approach by replacing the hard labels with the soft ones from the *teacher* model (Yun et al., 2020; Zhang et al., 2019). Although in typical classification settings such approaches have been extensively studied, in the NLP domain, this is relatively less explored. Huang et al. (2024) introduced confidence-aware label smoothing for alignment tasks considering language models and have shown the potential of the calibration approaches. Kobyzev et al. (2023) also showed the potential of several calibrated approaches in fine-tuning language models.

Note that we consider the idea of label relaxation introduced by Lienen and Hüllermeier (2021) wherein a single fixed target distribution is replaced with a set of candidate probability distributions. Another work by Kim et al. (2021) proposed relaxed labels in metric learning, which relaxes binary pairwise relation labels by replacing them with continuous similarity weights from a source embedding space. Alike our work, Purpura et al. (2022) also study learning to rank from relevance judgment distributions. They use KL divergence to align model predictions with empirical distributions, thereby directly capturing inter-annotator disagreement. Note that we assume that such distributions are not consistently available across ranking datasets. Instead, we propose label relaxation, which defines a credal set of admissible label distributions. This approach allows us to model epistemic uncertainty and mitigate label noise without requiring multiple annotations per query. To the best of our knowledge, this is the first work that studies calibration in this context.

3 Calibration in Ranking Model

In our work, we consider two different ranking approaches, both of which fall under the category of bi-encoder models. More specifically, we use the pre-trained BERT (Wu et al., 2020a) and the E5 (Wang et al., 2022) models.

E5 model (Wang et al., 2022) encodes both the query and candidate entities using a language model, producing dense vector embeddings. These token-level embeddings are then averaged via a pooling layer to obtain fixed-size vectors. Finally, a scoring function computes a probability score $\hat{y} \in [0, 1]$, reflecting the likelihood that the candidate entity is the correct match for the query.

BERT model follows a similar approach to E5 model in performing ranking tasks. However, the only difference is that BERT is pre-trained to perform binary relevance classification tasks between a query and a document (Devlin et al., 2019), whereas in contrast, E5, is additionally pre-trained on several ranking tasks. Next, we formalize the ranking task and subsequently define the term label relaxation in this context.

Note that the document ranking step often consists of two different steps, namely retrieval and ranking. In this work, we only consider the ranking stage. In a typical supervised ranking setup, each training sample consists of a query q and a set of candidate documents defined as $\mathcal{D}_q = \{d_1, d_2, \ldots, d_K\}$, with only one document labeled as relevant (Wang et al., 2022; Zhang and Braun, 2024; Tran et al., 2024). Herein, we assume the set of candidate documents is already correctly retrieved by a retrieval model. Typically, within a training step, a batch of queries and corresponding documents are presented, wherein the size of the batch is determined by the training configuration. For a batch consisting of N queries and K candidates per query, we define the label matrix $Y \in \{0, 1\}^{N \times K}$ as follows.

$$Y_{i,j} = \begin{cases} 1 & \text{if } d_j \in \mathcal{R}_{q_i} \\ 0 & \text{otherwise.} \end{cases}$$

Herein $\mathcal{R}_{q_i} \subseteq \mathcal{D}_{q_i}$ is the set of relevant documents for query q_i, typically of cardinality 1. Let us assume the ranking model as f, then it produces a score $f(q_i, d_j)$ for candidate d_j to query q_i. In the ranking tasks, the goal is to ensure that relevant documents are scored higher than non-relevant ones. To achieve this, the pairwise ranking loss is often

used herein. Let (q_i, d_{j^+}, d_{j^-}) denote a training triplet, where d_{j^+} is a document relevant to query q_i, and d_{j^-} is a non-relevant (or less relevant) document, i.e., $d_{j^-} \in \mathcal{D}_{q_i} \setminus \mathcal{R}_{q_i}$. Since we adopt an *in-batch negative sampling* strategy, we have a training batch containing N queries, each associated with K candidate documents (one relevant and $K-1$ non-relevant). For every query q_i, the model compares the relevant document d_{j^+} against all $K-1$ non-relevant candidates in the batch. The pairwise ranking loss scores a relevant document so that it exceeds that of a non-relevant one by $\gamma > 0$, a *defined margin*. This can be defined as,

$$\mathcal{L}_{\text{PR}} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq j^+}}^{K} \max\big\{0, \gamma - f(q_i, d_{j^+}) + f(q_i, d_{j^-})\big\} \tag{1}$$

Here, j^+ denotes the index of the relevant document among the K candidates for q_i. This loss penalizes cases where a non-relevant document scores too closely or higher than the relevant one.

Label Smoothing is a regularization technique that softens target labels to mitigate overconfidence (Szegedy et al., 2016; Müller et al., 2019). Rather than encoding the correct document as a one-hot vector, label smoothing redistributes a small fraction of the probability mass across all other candidates. Formally, the smoothed label distribution $\tilde{Y} \in [0,1]^{N \times K}$ can be defined as follows.

$$\tilde{Y}_{i,j} = \begin{cases} 1 - \varepsilon & \text{if } j = j^+ \\ \frac{\varepsilon}{K-1} & \text{otherwise} \end{cases}$$

As mentioned previously, we consider in-batch pairwise training; therefore, we use the smoothed score $\tilde{Y}_{i,j}$ in place of a hard label of 1 in the margin-based loss, resulting in the label-smoothed pairwise ranking loss as follows.

$$\mathcal{L}_{\text{LS}}^{pair} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq j^+}}^{K} \tilde{Y}_{i,j^+} \max\big(0, \gamma - f(q_i, d_{j^+}) + f(q_i, d_{j^-})\big) \tag{2}$$

Label Relaxation in Pairwise Ranking unlike label smoothing, which redistributes the probability mass of the target label uniformly, label relaxation replaces the target with a set of *plausible distributions* that reflect *epistemic uncertainty* (Lienen and Hüllermeier, 2021). This can help to reduce the uncertainty regarding the correct label. Label relaxation introduces a *relaxed set* of acceptable target distributions parameterized by $\alpha \in [0,1]$. We define Q^α as the set of all relevance probability distributions p satisfying $p(+) \geq 1 - \alpha$ and $p(-) \leq \alpha$. While Q^α is a set, in our implementation, we instantiate it via a canonical representative distribution p_r for loss computation as $Q^\alpha = \big\{p \in \Delta^2 : p(+) \geq 1 - \alpha, \quad p(-) \leq \alpha\big\}$.

This set essentially defines that the relevant document should be preferred with high probability; this does not pertain to a specific numeric value; rather, we allow the model to match any distribution within Q^α. The model can then generate any label that falls inside this plausible region, without penalizing it for deviations that are within the acceptable uncertainty bounds. The relaxation parameter $\alpha \in [0,1]$ controls the degree of permissible deviation from the one-hot target. We select α via validation set performance for each dataset.

Next, we apply the KL divergence on the predicted scores and the distribution Q^α. Note that, since $f(q_i, d_j)$ is a relevance score, for KL divergence we need to first convert it into a probability distribution over candidates using a softmax normalization, let us call this $\hat{p}_i(j)$. Afterwards, the label relaxation loss compares the predicted distribution $\hat{p}_i(j)$ with the distribution Q^α, denoted as p_r. Since we have two different distributions, instead of using any margin-based loss, it is computed using KL divergence as follows.

$$\mathcal{L}_{\text{LR}}^{\text{pair}} = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \neq j^+}}^{K} \text{KL}(p_r \| \hat{p}_i(j)) \tag{3}$$

Herein p_r can be defined as follows.

$$p_r(y) = \begin{cases} 1 - \alpha & \text{if } y = + \\ \alpha & \text{if } y = - \end{cases}$$

Since the prediction is probabilistic, KL divergence penalizes differences in a way that reflects confidence mismatches, i.e., confident, however, when predictions are wrong, are more penalized than uncertain ones. Below we give an example to explain this clearly.

Example Consider (q_i, d_{j^+}, d_{j^-}) that has been judged by multiple annotators. Suppose 80%

of them preferred $d^+_{i,j}$, while 20% preferred d_{j^-}. This uncertainty would be difficult to capture with one-hot labels or a uniform label smoothing approach. Rather than forcing the model to match a hard label $[1, 0]$, or smoothing it arbitrarily to something like $[0.9, 0.1]$, label relaxation allows us to model more concisely. Let us choose the relaxation parameter as $\alpha = 0.2$, with which we can define the relaxed set as $Q^{0.2} = \left\{p \in \Delta^2 : p(+) \geq 0.8,\ p(-) \leq 0.2\right\}$.

Therefore, $p_r(+) = 0.8, \quad p_r(-) = 0.2$. This implies that if the model predicts any values between 0.8 and 1.0, the associated loss would be considered as 0. Otherwise, using the KL-divergence loss, the model is then trained to minimize the divergence. This formulation respects the ambiguity in the supervision and allows the model to output calibrated probabilities that reflect uncertainty, rather than overconfident or artificially smoothed predictions. As a result, label relaxation not only improves robustness to label noise but also enhances the model's ability to represent uncertainty, which is critical in real-world applications such as QA, recommendation, and information retrieval.

Calibration Error in Ranking is typically measured using expected calibration error (ECE) to evaluate the calibration of the model's probability outputs (Naeini et al., 2015; Guo et al., 2017). Calibration in this context refers to the agreement between predicted probabilities and the actual likelihood of correctness. More specifically, considering document ranking tasks, the goal is to ensure that the probability assigned to a document reflects its actual relevance to the query. A well-calibrated ranking model would assign a probability close to 1 to relevant documents and a probability close to 0 to non-relevant ones. To define it more formally, let us assume $\mathcal{R}_{q_i}$ be the set of relevant documents for query q_i, and $\mathcal{D}_{q_i} = \{d_1, d_2, \ldots, d_K\}$ the full set of candidate documents, ECE for neural ranking models is,

$$\text{ECE} = \sum_{i=1}^{N} \sum_{j=1}^{K} |\hat{p}_i(j) - \mathbb{I}(d_j \in \mathcal{R}_{q_i})| \cdot \mathbb{I}(\hat{y}_{i,j} \in \mathcal{B}). \tag{4}$$

Where $\mathbb{I}(d_j \in \mathcal{R}_{q_i})$ is the indicator function that is 1 if d_j is relevant to q_i, and 0 otherwise, and $\mathbb{I}(\hat{y}_{i,j} \in \mathcal{B})$ is an indicator function that checks whether $\hat{p}_i(j)$ falls into a bin $\mathcal{B}$ of predicted probability values. Specifically, $|\hat{p}_i(j) - \mathbb{I}(d_j \in \mathcal{R}_{q_i})|$ represents the absolute error between the predicted probability and the ground truth label. The summation is carried out over all candidate documents within each bin.

4 Evaluation

Datasets and Models For evaluation, we used four datasets, namely, (i) AIDA (Hoffart et al., 2011), (ii) Mintaka (Sen et al., 2022), (iii) LC-QuAD 2.0 (Dubey et al., 2019), and (iv) MS MARCO (Craswell et al., 2021). Datasets (i)–(iii) pertain to entity ranking tasks, and the MS MARCO dataset corresponds to document ranking tasks. The AIDA dataset contains news articles and entities that are linked to Wikipedia. Mintaka is generated through crowd workers, wherein the entities in question-and-answer pairs are linked to the Wikidata knowledge graph. LC-QuAD 2.0 (or in short, LC-QuAD) is also generated through crowd workers, but, contains SPARQL queries. Finally, the MS MARCO dataset is frequently used for diverse tasks to perform question answering, passage ranking, and document ranking. Both LC-QuAD and MS MARCO are question-answering datasets. The ranking models are taken from their original implementation given in `Hugging Face`, BERT [2], E5 [3]. Thereafter, using in-batch negative sampling (Wu et al., 2020b), we fine-tuned them on the datasets described above. Each of the models is trained using the default learning rates and the parameters considering 10 epochs. Finally, the evaluation is performed by using the model's embeddings indexed by using Faiss indexing API (Douze et al., 2024). Note that, for MS MARCO, while computing the hard negatives, we randomly selected 10,000 negative documents from the whole corpus at a time. Finally, for fine-tuning the models, we used a server with 128 GB of RAM and an NVIDIA RTX H100 GPU with 80 GB of RAM.

Label Noise In this work, we consider *semantic-aware* label noise, wherein instead of flipping labels randomly, our approach considers a more realistic scenario. More specifically, for a given ratio of noise addition, we intentionally introduce an error by replacing the correct (relevant) document with a non-relevant one that is semantically very similar to the former. This is done by first randomly choosing a subset of queries from the training batch.

[2] https://huggingface.co/docs/transformers/model_doc/bert

[3] https://huggingface.co/intfloat/e5-base-v2

Table 1: MRR and Recall results evaluated on four ranking datasets using the E5 model, grouped by noise ratio with comparisons across loss functions (PR, LS, LR). The results reported below are the best results obtained considering specific smoothing rates and relaxation parameters.

NR	LF	MRR ↑				Recall↑			
		Msmarco	**Lcquad**	**Mintaka**	**Aida**	**Msmarco**	**Lcquad**	**Mintaka**	**Aida**
0	PR	0.8823	0.9191	0.2419	0.2881	0.9678	0.8739	0.3290	0.1501
	LS	0.8819	0.8823	0.2433	**0.3007**	0.9666	0.8803	0.3701	**0.1692**
	LR	**0.9164**	**0.9194**	**0.3244**	0.2856	**0.9718**	**0.8818**	**0.4470**	0.1557
1	PR	0.8782	**0.9095**	0.2323	0.2782	0.9637	0.8713	0.3147	0.1492
	LS	0.8771	0.8915	0.2418	**0.2914**	0.9617	0.8701	0.3382	**0.1676**
	LR	**0.9165**	0.9090	**0.3385**	0.2835	**0.9713**	**0.8739**	**0.4640**	0.1534
2	PR	0.8757	**0.8922**	0.2119	0.2678	0.9603	0.8576	0.2856	0.1404
	LS	0.8819	0.8808	0.2247	**0.2812**	0.9597	0.8550	0.3003	**0.1498**
	LR	**0.9160**	0.8910	**0.3189**	0.2672	**0.9711**	**0.8593**	**0.4357**	0.1423
4	PR	0.8541	**0.8537**	0.1957	0.2489	0.9451	**0.8180**	0.2658	0.1241
	LS	0.8516	0.8332	0.2020	**0.2719**	0.9441	0.8000	0.2753	**0.1493**
	LR	**0.9128**	0.8452	**0.2818**	0.2530	**0.9703**	0.8058	**0.3748**	0.1302
5	PR	0.6805	0.8169	0.1813	0.2500	0.9129	0.7854	0.2433	0.1293
	LS	0.6907	0.8180	0.1877	**0.2688**	0.9091	0.7718	0.2612	**0.1403**
	LR	**0.9110**	**0.8187**	**0.2673**	0.2412	**0.9331**	**0.7857**	**0.3603**	0.1206

Then, for each selected query, the correct answer is changed and replaced with another candidate that is closest in meaning, based on a similarity score between the original relevant document and all the other candidates. Therefore, we simulate noisy supervision by replacing the correct document with a semantically similar but non-relevant one for a subset of queries. We vary the noise proportion across five levels: 0% (no noise) to 5% of the training labels, following a progressive corruption scheme. Concretely, at 2% noise, 2% of the queries in the training set have their relevant document replaced. Note that although we do not perform human verification to find out the plausibility of the noisy labels, we still ensure their semantic plausibility by selecting replacements based on embedding similarity [4]. Additionally, some datasets, for instance, Mintaka, originate from multiple human annotators, which in principle could provide empirical relevance distributions. However, the versions we use in our evaluation only provide single canonical labels. For consistency across benchmarks, we therefore did not compare against models trained on empirical annotation distributions.

4.1 Results & Discussion

Tables 1 and 2 show the results in terms of MRRs and recall@10 of applying two different calibrated loss functions considering E5 and BERT models. NR depicts different noise ratios, and CL denotes different loss functions. We report results considering five different noise ratios. Noise ratio herein indicates the proportion of training queries for which the relevant document is replaced with a semantically similar but incorrect one. Note that in these tables, we show the results considering pairwise ranking loss. However, we also conducted experiments using cross-entropy loss. Since the results show the same trend, we omit them in the paper.

Considering Table 1, the results suggest that label relaxation can significantly improve the performance of the E5 model when fine-tuned on the Mintaka and MS MARCO datasets under noisy labels. However, considering the AIDA dataset, we find that, in fact, smoothing performs better, and in the LC-QuAD dataset, none of the calibration approaches lead to significant performance improvement. This is because the nature of the dataset determines the effectiveness of a calibration strategy. For AIDA, the relatively structured entity annotations and consistent alignment with the knowledge graph render the soft regularization of smoothing more effective than the plausibility distribution of labels used by label relaxation. In LC-QuAD, the queries are short and ambiguous, and the candidate space is limited, which might impact the calibration approaches. This might further reduce the impact of either calibration approach. These findings

[4]This is further mentioned in the Section 5

Table 2: MRR and Recall results evaluated on four ranking datasets using the BERT model, grouped by noise ratio with comparisons across loss functions (PR, LS, LR). The results reported below are the best results obtained considering specific smoothing rates and relaxation parameters.

NR	LF	MRR↑				Recall↑			
		Msmarco	**Lcquad**	**Mintaka**	**Aida**	**Msmarco**	**Lcquad**	**Mintaka**	**Aida**
0	PR	0.8331	**0.9410**	0.4223	0.3755	0.9254	**0.9271**	0.5111	**0.2231**
	LS	0.8310	0.9338	**0.4261**	**0.3761**	0.9051	0.9171	0.5331	0.2205
	LR	**0.8500**	0.9371	0.4117	0.3551	**0.9381**	0.9113	**0.5457**	0.2210
1	PR	0.7891	**0.9388**	**0.4235**	0.3421	0.8987	**0.9199**	0.5035	**0.2198**
	LS	0.8178	0.9381	0.4165	0.3383	0.8810	0.9090	0.5234	0.2171
	LR	**0.8438**	0.9358	0.4097	**0.3518**	**0.9341**	0.9049	**0.5434**	0.2000
2	PR	0.7517	**0.9108**	0.4058	**0.3353**	0.5900	**0.8989**	0.5021	**0.2065**
	LS	0.7234	0.9088	**0.4241**	0.3211	0.8571	0.8836	**0.5312**	0.2054
	LR	**0.8402**	0.9015	0.3793	0.3301	**0.9301**	0.8844	0.5083	0.1845
4	PR	0.6985	0.8441	**0.3963**	**0.3381**	0.4895	0.8110	0.4938	**0.2150**
	LS	0.7510	0.8419	0.3759	0.3230	0.8220	**0.8190**	**0.5114**	0.2065
	LR	**0.8400**	**0.8509**	0.3299	0.3104	**0.9301**	0.8176	0.4372	0.1718
5	PR	0.6885	0.8001	0.3543	0.3211	0.4074	0.7719	0.4255	0.2031
	LS	0.7491	**0.8199**	**0.3741**	**0.3230**	0.8113	0.7881	**0.5013**	**0.2063**
	LR	**0.7819**	0.8192	0.3019	0.2944	**0.9110**	**0.7898**	0.4009	0.1777

highlight that while label relaxation offers strong robustness under certain noise settings, its efficacy is still dataset-dependent and should be carefully selected based on the underlying characteristics of the data and task.

In Table 2, we see the results of the BERT model, wherein it can be observed that the label relaxation does not show significant performance improvement for Lc-QuAD, Mintaka, and Aida datasets. In those datasets, label smoothing performs slightly better. However, it also does not significantly improve the results in comparison to pairwise loss. These findings are consistent with the study by Zhu et al. (2022) wherein they reported that label smoothing does not improve the performance of the BERT model under label noise generated in the weakly supervised step.

E5 model, despite using the same underlying BERT model, is extensively weakly-supervised trained on the ranking dataset (Wang et al., 2022) that makes it inherently more robust to noisy supervision and better calibrated in its embedding space. This encourages the model to learn smoother decision boundaries and more stable representations. As a result, when fine-tuned with label relaxation, E5 is able to leverage its calibrated embedding space to better align the relaxed supervision with meaningful semantic gradients. In contrast, the standard BERT model lacks such domain-specific pre-training and starts from a relatively uncalibrated representation space for the ranking task, making it more sensitive to label noise and less responsive to relaxation-based regularization. However, we see that for the largest dataset, MS MARCO, label relaxation outperforms the other calibration approaches for the BERT model. This observation suggests that for very large datasets, the relaxed set can be helpful even when the model is not pre-trained on ranking datasets. Herein, the availability of training instances allows the model to benefit from the soft supervision, avoiding overfitting to incorrect labels. In contrast, the standard BERT model lacks such domain-specific pre-training and starts from a relatively uncalibrated representation space for the ranking task, making it more sensitive to label noise and less responsive to relaxation-based regularization. With these results, we highlight the following important findings.

Dataset size & diversity. Large datasets such as MS MARCO and Mintaka work effectively with label relaxation since this distributes probability mass over semantically plausible candidates without overfitting to noisy labels.

Candidate space structure. Highly structured datasets like AIDA favor a calibrated loss function. Herein, smoothing gains top performance since it enforces small-entropy distributions.

Query ambiguity. In LC-QuAD, where ambiguity and candidate space constraints dominate, calibration does not yield notable performance gain.

Table 3: Loss functions to use when training on MS MARCO, Mintaka, LC-QUaD, and AIDA datasets.

Dataset	E5	BERT	Notes
MS MARCO	Label Relaxation	Label Relaxation	Largest dataset; soft supervision avoids overfitting
LC-QuAD	Pairwise loss	Pairwise loss	Small candidate space; calibration has little effect
Mintaka	Label Relaxation	Label Relaxation	Large, diverse queries; E5 benefits from calibrated embeddings
AIDA	Label Smoothing	Label Smoothing	Structured entity annotations; smoothing aligns better

Model pre-training. This is probably the most important finding of all. As mentioned previously, E5's extensive weakly-supervised pre-training on ranking data produces smoother embedding manifolds, allowing label relaxation to align gradients with semantically similar negatives. In contrast, standard BERT lacks such calibration and is more sensitive to label noise.

Based on the above observation, we provide a *practical guidance* as to when to use a specific type of loss function. This is summarized in Table 3.

Calibration Error Analysis Based on our proposed expected calibration error, defined in Equation 4, we evaluated the calibration of the E5 and BERT models, considering pairwise loss, label smoothing, and label relaxation. The results are reported in Table 4. Note that since LC-QuAD does not yield notable performance improvement using calibrated loss functions, we do not consider it.

We see that typically label relaxation leads to the lowest calibration errors for most of the datasets. However, the differences between the ECE values of the calibrated and non-calibrated loss functions are not remarkably high. In fact, considering the MS MARCO dataset, we find that ECE is lower for non-calibrated loss in high noise ratios compared to calibrated loss functions, even when the performance drops significantly with non-calibrated loss functions. This shows some known shortcomings of ECE, for instance, its histogram binning can mask differences, specifically considering high estimator bias and variance depending on bin count and scheme. Additionally, since ECE aggregates class- and score-conditional structure, work on ranking scale calibration similarly reports that off-the-shelf ECE can be misleading without class balancing or rank-aware structure (Widmann et al., 2019; Futami and Fujisawa, 2024; Yan et al., 2022). These works reported results on vision-based rankers. In this work, we find the same drawback in document ranking models as well.

As an alternative to calibration error, we analyze the behavior of the ranking models BERT and E5 under label noise by plotting the training and validation performance side-by-side and observe the differences in Figure 1 and 2 (Appendix A), respectively (). Therein, we see that as the label noise increases, the gap increases notably. This behavior is consistent with memorization under label noise. Specifically, the models eventually fit corrupted labels, inflating training metrics while harming generalization. This highlights that the performance of ranking under noise is not captured by ECE metric. The widening recall gap, as *memorization* error (Zhang et al., 2021; Han et al., 2025), is therefore a practical metric herein to guide calibration or early stopping.

5 Conclusion & Future Directions

In this work, we have studied label relaxation considering the neural ranking models in performing document ranking tasks. To this end, we first formally define the label relaxation in the context of the ranking task. Afterwards, we integrate it into bi-encoder ranking models. Additionally, to find out whether label relaxation can mitigate the impact of label noise in fine-tuning neural ranking models, we conducted extensive evaluations considering 2 different bi-encoder models, 4 different ranking datasets, and 5 different noise levels. We also compare our results to the popular label smoothing calibration approach. The results of our evaluation suggest that label relaxation can indeed be helpful in fine-tuning ranking models when label noise is prevalent in the ranking datasets. However, our findings also suggest that label relaxation is effective on the E5 model, which is extensively

Table 4: Calibration Error results evaluated on four ranking datasets using the E5 and Bi-Encoder models, grouped by noise ratio (NR) and loss functions (LF) with the best ϵ and α as smoothing and relaxation parameters. Lower is better; bold indicates the best value for a dataset at a given noise ratio.

NR	LF	E5↓			BERT↓		
		Msmarco	**Mintaka**	**Aida**	**Msmarco**	**Mintaka**	**Aida**
0	PR	0.0424	0.2914	**0.2098**	0.0090	0.3277	0.3016
	LS (0.1)	0.0317	0.2917	0.2101	0.0091	0.3199	0.2800
	LR (0.1)	**0.0269**	**0.2811**	0.2197	**0.0076**	**0.3185**	**0.2770**
1	PR	0.1538	0.2879	**0.2123**	0.0094	0.3391	0.3123
	LS (0.1)	0.1398	0.2883	0.2165	**0.0091**	0.3109	0.3001
	LR (0.2)	**0.1221**	**0.2846**	0.2193	0.1000	**0.3019**	**0.2877**
2	PR	**0.2024**	0.2726	0.2066	**0.1094**	0.3293	0.2893
	LS (0.2)	0.2119	0.2713	0.2081	0.1913	0.3150	0.2891
	LR (0.2)	0.2175	**0.2661**	**0.2041**	0.1911	**0.2854**	**0.2713**
4	PR	**0.2969**	0.2598	0.2033	**0.1111**	0.3373	0.3049
	LS (0.2)	0.3018	0.2561	0.2049	0.1101	0.3171	0.3098
	LR (0.3)	0.3161	**0.2476**	**0.1908**	0.2000	**0.3104**	**0.2811**
5	PR	**0.3150**	0.3109	0.2025	**0.2082**	0.3322	0.2943
	LS (0.3)	0.3310	0.2601	0.2019	0.2910	0.3091	0.2920
	LR (0.3)	0.3293	**0.2417**	**0.1902**	0.2989	**0.3047**	**0.2918**

weakly supervised pre-trained on the ranking tasks. On the other hand, if a pre-training on the ranking tasks is not performed, the results do not improve. Additionally, we find that the ECE might not be suitable to measure the calibration of the ranking models under noise and memorization errors could be helpful to get better insights.

We believe label relaxation has a lot of potential to build well-calibrated models. We can envisage works that would explore adaptive label relaxation approaches that adjust relaxation based on model confidence or noise estimates, and investigate their effects on model calibration. Furthermore, extending relaxation to cross-encoder and LLM-based rankers, and studying new measures to compute calibration error in the context of document ranking, could also be a potentially interesting direction.

Limitations

The following are some of the key limitations of our study, which we acknowledge.

Model limitation Our experiments are restricted to bi-encoder architectures, i.e., BERT and E5. Although this choice allowed us to systematically analyze calibration under controlled conditions, the findings may not directly transfer to more complex architectures such as cross-encoders or large language model (LLM)-based rankers. However, such an extension would require additional challenges and opportunities.

Dataset limitation We considered experimenting with four datasets, AIDA, Mintaka, LC-QuAD, and MS MARCO. These vary in size, structure, and annotation quality; therefore, providing a diverse evaluation setting. However, our conclusions remain dataset-dependent, as seen from the differences in effectiveness across AIDA, LC-QuAD, and MS MARCO. Future work should examine a broader range of datasets, including multilingual and domain-specific ranking tasks, to assess the generalizability of these models.

Label noise modeling The label noise is modeled using a semantic-aware perturbation strategy, replacing relevant documents with semantically similar but incorrect ones. This provides a more realistic scenario compared to random flipping; however, real-world noise can be more diverse such as adversarial noise, annotation inconsistencies, or systematic bias. Our approach does not capture these variations; hence, the robustness of label relaxation under such conditions remains unexplored.

Calibration scope Our evaluation focused on *text-based calibration*, therefore, the probability estimates are aligned with annotation labels. We did not compare our approach against ranking-based calibration methods which adjust scores based on relative order or rank-aware confidence measures. These approaches are often stronger baselines in retrieval and relevance tasks, however, they fall outside the scope of this paper due to space limitations.

Hence, as part of future work, we would provide a more comprehensive view of calibration strategies for ranking.

Relaxation parameter The choice of relaxation parameter α is tuned using validation performance, which may not always be feasible in practice, especially when noisy labels affect the validation set itself. Adaptive or noise-aware strategies to determine relaxation parameters could further improve robustness and practicality.

Ethics Statement

This work relies exclusively on publicly available benchmark datasets (AIDA, Mintaka, LC-QuAD, and MS MARCO). No personally identifiable or sensitive information was collected or processed. All experiments were conducted in accordance with the terms of use of the respective datasets. Therefore, we believe our contributions pose no ethical risks beyond the general concerns of bias and fairness inherent in natural language processing and information retrieval research.

Acknowledgment

This work is supported by the Ministry of Culture and Science of North Rhine-Westphalia (MKW NRW) within the project SAIL under the grant no NW21-059D, the project WHALE (LFN 1-04) funded under the Lamarr Fellow Network Programme by the MKW NRW, the European Union's Horizon Europe research and innovation programme under grant agreement No 101070305, and by the German Federal Ministry of Research, Technology and Space (BMFTR) within the project KI-OWL under the grant no 01IS24057B.

References

Jiacheng Cheng and Nuno Vasconcelos. 2022. Calibrating deep neural networks by pairwise constraints. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 13699–13708. IEEE.

Nick Craswell, Bhaskar Mitra, Emine Yilmaz, and Daniel Campos. 2021. Overview of the trec 2020 deep learning track.

Hervé Déjean, Stéphane Clinchant, and Thibault Formal. 2024. A thorough comparison of cross-encoders and llms for reranking SPLADE. *CoRR*, abs/2403.10407.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Zhipeng Ding, Xu Han, Peirong Liu, and Marc Niethammer. 2021. Local temperature scaling for probability calibration. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 6869–6879. IEEE.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *Preprint*, arXiv:2401.08281.

Mohnish Dubey, Debayan Banerjee, Abdelrahman Abdelkawi, and Jens Lehmann. 2019. Lc-quad 2.0: A large dataset for complex question answering over wikidata and dbpedia. In *The Semantic Web – ISWC 2019: 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II*, page 69–78, Berlin, Heidelberg. Springer-Verlag.

Zheng Fang, Yanan Cao, Qian Li, Dongjie Zhang, Zhenyu Zhang, and Yanbing Liu. 2019. Joint entity linking with deep reinforcement learning. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, pages 438–447. ACM.

Futoshi Futami and Masahiro Fujisawa. 2024. Information-theoretic generalization analysis for expected calibration error. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2619–2629. Association for Computational Linguistics.

Arindam Ghosh, Thomas Schaaf, and Matthew R. Gormley. 2022. Adafocal: Calibration-aware adaptive focal loss. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney,*

NSW, Australia, 6-11 August 2017, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Andi Han, Wei Huang, Zhanpeng Zhou, Gang Niu, Wuyang Chen, Junchi Yan, Akiko Takeda, and Taiji Suzuki. 2025. On the role of label noise in the feature learning process. *CoRR*, abs/2505.18909.

Ramya Hebbalaguppe, Jatin Prakash, Neelabh Madan, and Chetan Arora. 2022. A stitch in time saves nine: A train-time regularizing loss for improved neural network calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 16060–16069. IEEE.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Baihe Huang, Hiteshi Sharma, and Yi Mao. 2024. Enhancing language model alignment: A confidence-based approach to label smoothing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 21341–21352.

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. 2021. Embedding transfer with label relaxation for improved metric learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 3967–3976. Computer Vision Foundation / IEEE.

Ivan Kobyzev, Aref Jafari, Mehdi Rezagholizadeh, Tianda Li, Alan Do-Omri, Peng Lu, Pascal Poupart, and Ali Ghodsi. 2023. Do we need label regularization to fine-tune pre-trained language models? In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 166–177.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Julian Lienen and Eyke Hüllermeier. 2021. From label smoothing to label relaxation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 8583–8591. AAAI Press.

Julian Lienen and Eyke Hüllermeier. 2024. Mitigating label noise through data ambiguation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 13799–13807. AAAI Press.

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. 2022. The devil is in the margin: Margin-based label smoothing for network calibration. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 80–88. IEEE.

Tongliang Liu and Dacheng Tao. 2016. Classification with noisy labels by importance reweighting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(3):447–461.

Xingchen Ma and Matthew B. Blaschko. 2021. Meta-cal: Well-controlled post-hoc calibration by ranking. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 7235–7245. PMLR.

Jooyoung Moon, Jihyo Kim, Younghak Shin, and Sangheum Hwang. 2020. Confidence-aware learning for deep neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 7034–7044. PMLR.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705.

Mahdi Pakdaman Naeini, Gregory F. Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*, pages 2901–2907. AAAI Press.

Nagarajan Natarajan, Inderjit S. Dhillon, Pradeep Ravikumar, and Ambuj Tewari. 2013. Learning with noisy labels. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 1196–1204.

Rodrigo Frassetto Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *CoRR*, abs/1901.04085.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. 2017. Making deep neural networks robust to label noise: A loss correction approach. In *2017 IEEE Conference*

on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 2233–2241. IEEE Computer Society.

Alberto Purpura, Gianmaria Silvello, and Gian Antonio Susto. 2022. Learning to rank from relevance judgments distributions. *Journal of the Association for Information Science and Technology*, 73(9):1236–1252.

Scott E. Reed, Honglak Lee, Dragomir Anguelov, Christian Szegedy, Dumitru Erhan, and Andrew Rabinovich. 2015. Training deep neural networks on noisy labels with bootstrapping. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.

Priyanka Sen, Alham Fikri Aji, and Amir Saffari. 2022. Mintaka: A complex, natural, and multilingual dataset for end-to-end question answering. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1604–1619, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Avirup Sil, Gourab Kundu, Radu Florian, and Wael Hamza. 2018. Neural cross-lingual entity linking. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5464–5472. AAAI Press.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Hung-Nghiep Tran, Akiko Aizawa, and Atsuhiro Takasu. 2024. An encoding–searching separation perspective on bi-encoder neural search. *Preprint*, arXiv:2408.01094.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533.

Jiaheng Wei, Hangyu Liu, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Yang Liu. 2022. To smooth or not? when label smoothing meets noisy labels. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 23589–23614. PMLR.

Jiaheng Wei and Yang Liu. 2021. When optimizing f-divergence is robust with label noise. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

David Widmann, Fredrik Lindsten, and Dave Zachariah. 2019. Calibration tests in multi-class classification: A unifying framework. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 12236–12246.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020a. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6397–6407. Association for Computational Linguistics.

Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2020b. Scalable zero-shot entity linking with dense entity retrieval. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6397–6407, Online. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6442–6454. Association for Computational Linguistics.

Le Yan, Zhen Qin, Xuanhui Wang, Michael Bendersky, and Marc Najork. 2022. Scale calibration of deep ranking models. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, pages 4300–4309. ACM.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. 2020. Regularizing class-wise predictions via self-knowledge distillation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 13873–13882. Computer Vision Foundation / IEEE.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2021. Understanding deep learning (still) requires rethinking generalization. *Commun. ACM*, 64(3):107–115.

Leixin Zhang and Daniel Braun. 2024. Twente-BMS-NLP at PerspectiveArg 2024: Combining bi-encoder and cross-encoder for argument retrieval. In *Proceedings of the 11th Workshop on Argument Mining (ArgMining 2024)*, pages 164–168, Bangkok, Thailand. Association for Computational Linguistics.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 3712–3721. IEEE.

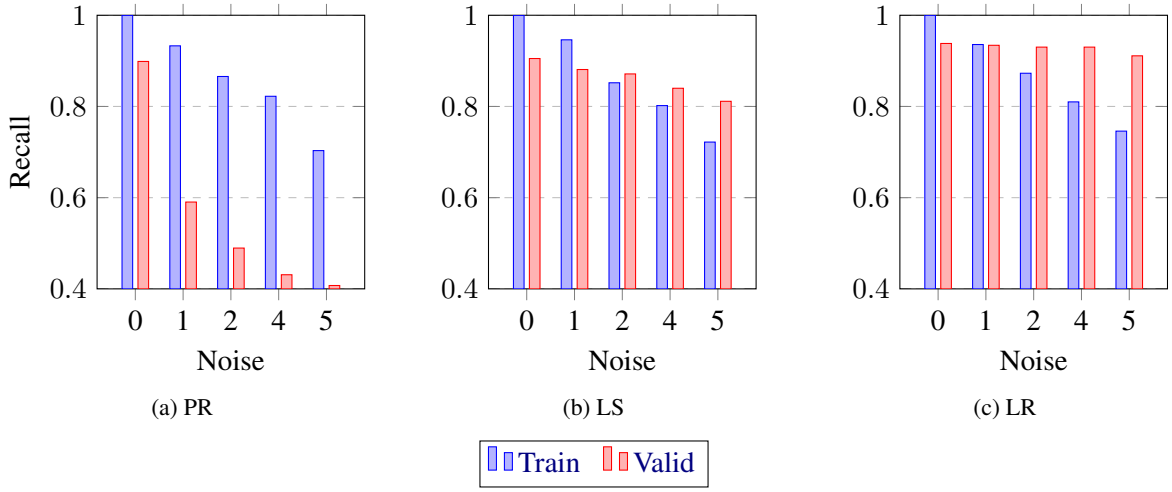
Wenzheng Zhang, Wenyue Hua, and Karl Stratos. 2022. Entqa: Entity linking as question answering. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zhenyu Zhang, Xiaobo Sind, Tingwen Liu, Zheng Fang, and Quangang Li. 2020. Joint entity linking and relation extraction with neural networks for knowledge base population. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–8. IEEE.

Dawei Zhu, Michael A. Hedderich, Fangzhou Zhai, David Ifeoluwa Adelani, and Dietrich Klakow. 2022. Is BERT robust to label noise? A study on learning with noisy labels in text classification. In *Proceedings of the Third Workshop on Insights from Negative Results in NLP, Insights@ACL 2022, Dublin, Ireland, May 26, 2022*, pages 62–67. Association for Computational Linguistics.

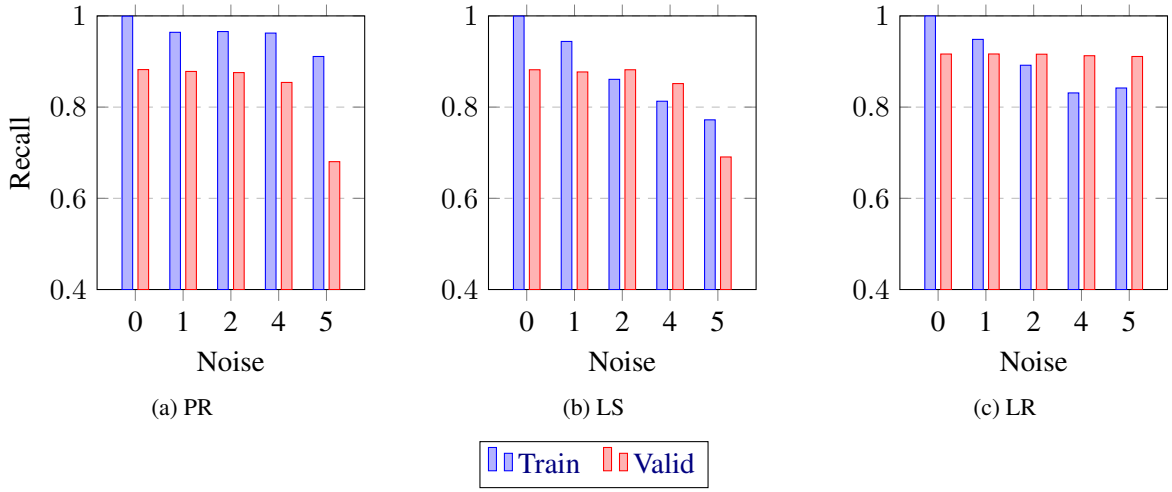
Zhaowei Zhu, Tongliang Liu, and Yang Liu. 2021. A second-order approach to learning with instance-dependent label noise. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10113–10123. Computer Vision Foundation / IEEE.

A Additional Results



(a) PR (b) LS (c) LR

Figure 1: Train vs. validation recall across noise levels for PR, LS, and LR for BERT.



(a) PR (b) LS (c) LR

Figure 2: Train vs. validation recall across noise levels for PR, LS, and LR for E5.

ERGO: Entropy-guided Resetting for Generation Optimization in Multi-turn Language Models

Haziq Mohammad Khalid[‡] **Athikash Jeyaganthan** **Timothy Do**

Yicheng Fu **Sean O'Brien** **Vasu Sharma** **Kevin Zhu**

Algoverse AI Research

haziqkhalid04@gmail.com , psyaj9@nottingham.ac.uk , timothy.k.do@sjsu.edu

Abstract

Large Language Models (LLMs) suffer significant performance degradation in multi-turn conversations when information is presented incrementally. Given that multi-turn conversations characterize everyday interactions with LLMs, this degradation poses a severe challenge to real world usability. We hypothesize that abrupt increases in model uncertainty signal misalignment in multi-turn LLM interactions, and we exploit this insight to dynamically realign conversational context. We introduce ERGO (**E**ntropy-guided **R**esetting for **G**eneration **O**ptimization), which continuously quantifies internal uncertainty via Shannon entropy over next token distributions and triggers adaptive prompt consolidation when a sharp spike in entropy is detected. By treating uncertainty as a first class signal rather than a nuisance to eliminate, ERGO embraces variability in language and modeling, representing and responding to uncertainty. In multi-turn tasks with incrementally revealed instructions, ERGO yields a 56.6% average performance gain over standard baselines, increases aptitude (peak performance capability) by 24.7%, and decreases unreliability (variability in performance) by 35.3%, demonstrating that uncertainty aware interventions can improve both accuracy and reliability in conversational AI.

1 Introduction

Large Language Models (LLMs) have become the primary interface for conversational AI systems, enabling users to interact through multi-turn exchanges. However, recent research has documented a critical limitation: LLMs often get 'lost' in conversation and experience substantial performance degradation in multi-turn conversations compared to single-turn interactions (Laban et al., 2025; Gupta et al., 2024). This degradation manifests as reduced accuracy, lower confidence, and a 112% increase in unreliability, posing significant challenges for real-world deployment (Laban et al., 2025).

While prior work has measured this degradation, existing mitigation strategies remain limited. Approaches based on task classification, retrieval, or context compression lack generality or require fine-tuning (Wu et al., 2023).

We hypothesize that spikes in model uncertainty signal moments of conversational drift and by explicitly representing this uncertainty and monitoring its fluctuations, we can detect when an LLM begins getting 'lost' in conversation. We introduce ERGO (**E**ntropy-guided **R**esetting for **G**eneration **O**ptimization), the first practical intervention framework that dynamically monitors internal uncertainty signals and resets context when needed. ERGO computes Shannon entropy over next-token probability distributions (Malinin and Gales, 2018; Xiao and Wang, 2022) as an internal behavioral signal to detect spikes in uncertainty that indicate breakdown in comprehension. When such spikes occur, ERGO triggers entropy-guided prompt reconstruction, mitigating accumulated ambiguity and restoring coherence. Unlike static prompt engineering, ERGO's reconstruction is dynamically triggered by entropy thresholds and systematically preserves only those contextual elements that sustain both internal coherence and external task performance, discarding accumulated noise. A visual representation of this can be seen in Figure 1.

Empirical results demonstrate that targeted interventions based on uncertainty peaks not only recover task accuracy but also improve consistency across turns. In extensive simulations with incrementally revealed instructions, ERGO improves average performance by 56.6% compared to standard multi-turn baselines, increases aptitude levels by 24.7% (best-case performance capability), and

[‡]Lead Author

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 273–286
November 9, 2025 ©2025 Association for Computational Linguistics

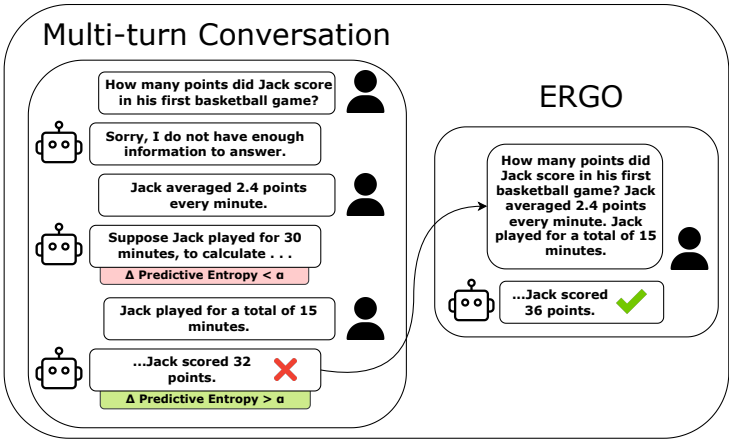


Figure 1: Illustrative comparison of a standard multi-turn conversational AI and the ERGO system

reduces the increased unreliability (variability in response consistency) observed in multi-turn settings by 35.3%. Furthermore, ERGO outperforms existing alternative strategies, and triggers resets with greater precision and timing compared to alternate baselines illustrating the potential of uncertainty-aware methods for robust conversational AI. To verify our findings and reproduce the results, please refer to the code repository found at the following link: https://github.com/haziq-exe/ERGO

2 Background and Related Works

Recent work has documented significant performance degradation in multi-turn LLM conversations. Laban et al. (2025) demonstrated that model performance dropped by 39% on average in multi-turn settings across six domains. Gupta et al. (2024) formalized task-switch sensitivity using probability ratios, showing how conversation history compounds model confusion. While Laban et al. (2025) managed to mitigate average performance losses by 15-20%, their approaches faced substantial verbosity and practicality constraints (Sec 5.4). Agent-based frameworks (Wu et al., 2023) explore system-level solutions but do not target fundamental model limitations during generation.

2.1 Entropy Based Uncertainty Estimation

Entropy-based uncertainty estimation provides the theoretical basis for our method, grounding ERGO's use of internal model signals. Prior work has used predictive entropy to quantify model confidence in classification and generation tasks (Malinin and Gales, 2018; Xiao and Wang, 2022), implicitly linking internal uncertainty to external behavior. More recent approaches extend this to semantic-level uncertainty using semantic-aware entropy measures (Kuhn et al., 2023) or trainable proxies derived from hidden representations (Kossen et al., 2024). While these methods improve semantic fidelity, they often rely on sampling or auxiliary models. In contrast, we use token-level entropy, computed directly from the model's next-token distribution, as a low-cost proxy for real-time monitoring. Unlike prior work that applies entropy primarily for evaluation or filtering, we use it as a temporal signal to detect context degradation and trigger prompt restructuring.

2.2 Inference-Time Interventions

Inference-time control methods intervene on frozen models by manipulating internal activations, modifying output logits, or reranking candidate outputs. For example, Li et al. (2024) introduced activation-level interventions to elicit truthful answers without fine-tuning, shifting hidden states toward truthful completions. Similarly, Turner et al. (2024) developed activation engineering techniques that steer the behavior of the model by editing intermediate representations during decoding. These methods act directly on the output path of the model and often rely on internal signal manipulation.

In contrast, our approach introduces a policy layer outside of the model that monitors uncertainty and intervenes by restructuring the user's input. We do not modify the internal computation or sampling process of the model.

2.3 Backtracking and Prompt Restructuring

Several recent approaches have explored controlled backtracking during generation. Cundy and Er-

mon (2024) augmented the decoding space with a 'backspace' action to revert low-probability generations, while Zhang et al. (2024) uses a special [RESET] token to discard unsafe prefixes. Other strategies such as Self-Refine (Madaan et al., 2023) allowed iterative refinement by prompting the model to critique and revise its own output. These methods operate on generated content and typically require multi-step decoding or auxiliary supervision.

Our intervention departs from this paradigm by focusing on upstream correction. Instead of rewriting the model's response, we update the user's prompt to recover task coherence, using rising entropy as the intervention trigger. This shifts the optimization target from output correction to input respecification, which is more lightweight and avoids cumulative reasoning errors. To our knowledge, this is the first method that uses entropy-based signals to restructure user input mid-conversation, rather than adjusting the model's internal behavior or downstream output.

3 Entropy-Guided Context Resetting

3.1 Rise in Average Token Level Entropy

At each turn of the conversation, the average token-level entropy is calculated by measuring the uncertainty of the model's token probability distribution when generating each token in its output.

Suppose the model produces a sequence of tokens $t_1, t_2, \ldots, t_n$ at a given turn. For each token t_i, the model assigns a probability distribution P_i over the vocabulary V, where $P_i(v)$ is the probability assigned to token $v \in V$ at position i.

The entropy at position i is computed as:

$$H_i = -\sum_{v \in V} P_i(v) \log P_i(v)$$

The average token-level entropy $\bar{H}$ for the turn (covering n generated tokens) is then:

$$\bar{H} = \frac{1}{n} \sum_{i=1}^{n} H_i$$

This metric quantifies the model's overall uncertainty when generating the turn. Higher $\bar{H}$ indicates greater uncertainty and a more diffuse token distribution, while the lower $\bar{H}$ indicates more confident and peaked predictions (Malinin and Gales, 2018; Xiao and Wang, 2022).

For each subsequent turn t in the conversation, the change in average token-level entropy is calculated to monitor fluctuations in model uncertainty. Let $\bar{H}^{(t)}$ denote the average token-level entropy at turn t, as defined previously.

The change in predictive entropy between consecutive turns is defined as:

$$\Delta\bar{H}^{(t)} = \bar{H}^{(t)} - \bar{H}^{(t-1)}$$

A positive $\Delta\bar{H}^{(t)}$ indicates that the uncertainty of the model has risen relative to the previous turn.

3.2 Threshold-Based Trigger for Context Reset

A predefined, calibrated entropy change threshold τ is established. When the change in predictive entropy satisfies the following condition:

$$\Delta\bar{H}^{(t)} > \tau$$

The system deems that the uncertainty of the model is rising beyond an acceptable margin. This is interpreted as a signal that the evolving conversation context may be inducing compounding uncertainty or drift. A detailed analysis of the threshold selection process is provided in Appendix A, while an analysis of ERGO's sensitivity to entropy thresholds is provided in Appendix B.

3.3 Context Reset Protocol

Upon detection of $\Delta\bar{H}^{(t)} > \tau$, an automated context reset protocol is initiated. This protocol proceeds in the following steps:

I. Prompt Rewriting:
The user's inputs up to turn t are provided to the model. The model is asked to rewrite these inputs into a single-turn, optimized prompt that preserves relevant task information while reducing ambiguity and redundancy.

II. Isolated Generation (Simulate New Chat):
The rewritten prompt is passed into a new instance of the model, simulating a stateless chat environment with no memory of prior turns. The model then generates a response R_{opt} to this rewritten prompt.

III. Branch Continuation:
A new dialogue branch is created that begins from the rewritten prompt and response. This maintains continuity from the optimized state rather than the potentially degraded original context.

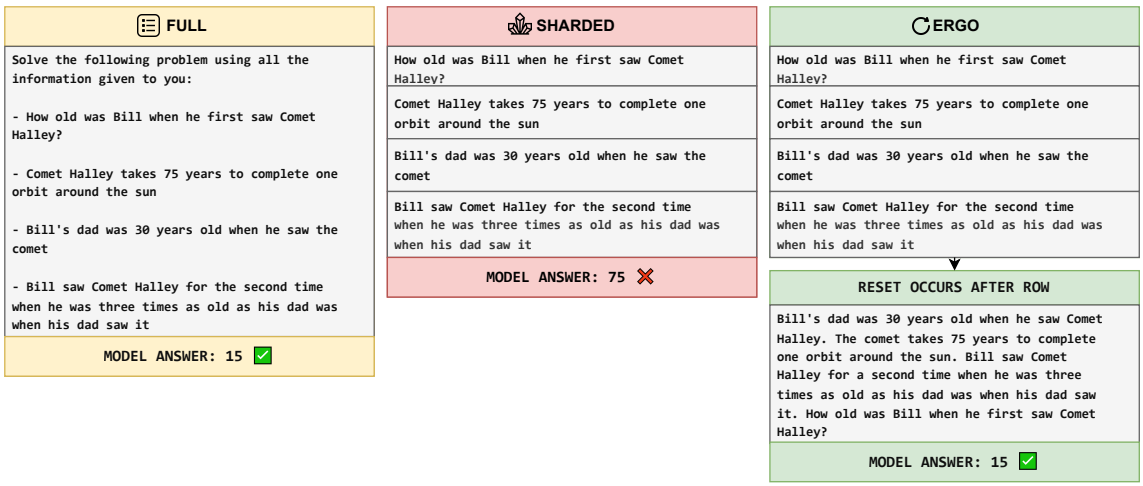


Figure 2: Example Llama3.1-8B run on a GSM8K question with FULL, SHARDED and ERGO settings. Each row represents a separate prompt given to the model while each table represents a context window.

4 Experimentation Background

4.1 Simulation Scale & Parameters

Our simulation follows the protocol of Laban et al. (2025) with the only change being the implementation of ERGO. We evaluate a suite of five leading instruction-tuned LLMs: **Phi-4** (Abdin et al., 2024), **LLaMA 3.1–8B Instruct** (Grattafiori et al., 2024), **GPT-4o** (Hurst et al., 2024), **GPT-4.1** (OpenAI, 2025), and **GPT-4o-mini** (OpenAI, 2024). All models are used in their publicly released variants without additional fine-tuning.

Generation settings are standardized across models with temperature set to 1.0. For entropy calculations, we note an important methodological constraint: OpenAI models provide access to only the top-20 logprobs through their API. This limitation affects the precision of entropy estimates, particularly for tasks with shorter responses such as *Actions* and *Data-to-text* (Sec 4.2), where the restricted probability space may not capture the full uncertainty of the model's predictions.

We conduct 3 independent simulation runs for each dataset using 100-question samples, with the exception of the Data-to-text dataset (Sec 4.2), for which evaluations were performed on a 50-question subset over 3 runs. All other experimental settings and baseline figures are adopted directly from Laban et al. (2025).

We compare three settings:

FULL: Simulates a single-turn, fully-specified conversation using the sharded instruction. The shards are combined into a single bullet-point list (one shard per line), prefaced by a directive to complete the task using all listed points. This setting serves as an upper bound for performance, providing a target for evaluating how closely multi-turn intervention methods can approximate single-turn optimality.

SHARDED: Sequential shard presentation as in the original (Laban et al., 2025) LLMs-lost-in-conversation experiment.

ERGO: Our entropy-guided reset mechanism applied upon exceeding the entropy threshold.

Figure 2 provides an example of a run on each setting. This evaluation isolates the effect of ERGO relative to both single-pass and original multi-turn baselines.

4.2 Tasks

We evaluated models on five representative generation tasks, each framed as a multi-turn interaction over sharded instructions and augmented them with our entropy-guided context resetting method (Section 3). For each task, we used 220-325 constructed prompts from the datasets created by Laban et al. (2025). We simulate a multi-turn conversation, feeding the model one shard at a time. At each assistant turn, we compute the average token-level entropy and track its change $\Delta\bar{H}^{(t)}$. Whenever $\Delta\bar{H}^{(t)}$ exceeds the calibrated threshold τ, we invoke our reset protocol - prompt rewriting, isolated regeneration, branch continuation - before continuing.

Below we briefly summarize what the assistant must do in each task:

CODE: Convert natural-language problem description into a correct Python function. Outputs

are validated by executing against the reference test suite (Chen et al., 2021; Jain et al., 2024).

DATABASE: Given a database schema and a user request, generate an SQL query that returns the requested data. Correctness is checked by running the query on the Spider-derived database (Yu et al., 2018).

ACTIONS: Given API schemas plus high-level user instruction, emit valid code-style API calls that fulfill the intent. This is verified against the Berkeley Function Calling Leaderboard definitions (Yan et al., 2024).

DATA-TO-TEXT: Take a structured data table and metadata and write a single caption that highlights its key insight. Adapted from ToTTo and evaluated using BLEU (scaled 0-100) (Parikh et al., 2020; Papineni et al., 2002).

MATH: Solve an elementary math story problem by carrying out each arithmetic step and returning the numeric result. Simulates day-to-day problems LLMs may be tasked with by users. GSM8K problems were used and scored by exact match (Cobbe et al., 2021).

4.3 Metric Selection

We assess LLM performance in multi-turn tasks by repeating simulations for each instruction and collecting success scores from multiple runs, following Laban et al. (2025). Each score, ranging from 0 to 100, reflects task success.

4.4 Per-Run Scoring

I. **Binary-Correctness Tasks (Code, Database, API, Math):** A correct response at any turn yields a score of 100, and the run ends. Otherwise, the score is 0.

II. **Refinement Task (Data-to-Text):** The final output is evaluated using BLEU, rescaled to 0–100.

4.5 Aggregate Metrics

From the scores collected across the 3 runs, we compute three metrics:

- **Average Performance ($\bar{P}$):** Average performance per instruction for a given task.
- **Aptitude (A^{90}):** 90th-percentile score, measures a model's peak capability, indicating its potential to deliver high-quality results in critical multi-turn tasks. Averaged across all tasks.
- **Unreliability (U_{10}^{90}):** Difference between 90th and 10th percentiles, quantifies response variability, where lower values reflect greater consistency, essential for user trust and system reliability in long-horizon interactions. Averaged across all tasks.

Formulae and more information on metrics is available in Appendix D.

5 Results & Discussion

5.1 Aptitude and Unreliability Improvements

Figure 3 shows that ERGO demonstrates exceptional gains in aptitude, often exceeding single-turn performance levels, while substantially reducing unreliability compared to multi-turn baselines, two metrics introduced by (Laban et al., 2025) to capture model consistency across conversations.

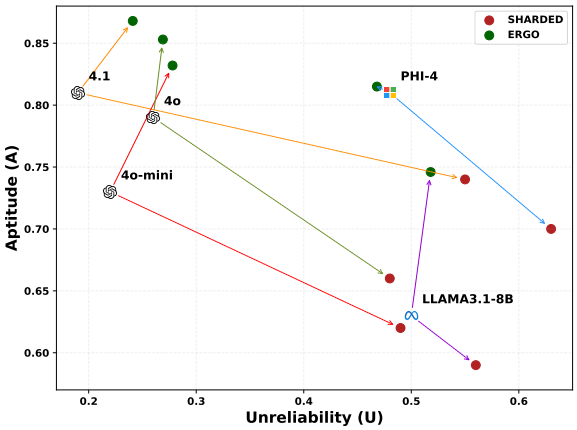


Figure 3: Effect of SHARDED and ERGO on Aptitude and Unreliability. Icons represent models FULL performance. Green dots represent performance with ERGO while red dots represent SHARDED performance

These results indicate that our intervention not only fully recovers the aptitude lost in the transition from single-turn to multi-turn settings and achieves aptitude levels exceeding single-turn baselines, but also makes behavior significantly more

Model	Code			Database			Actions			Data-to-Text			Math		
Llama3.1-8b	21.2	21.7	**52.0**$^{\uparrow}$	47.7	25.9	**64.3**$^{\uparrow}$	**83.0**	45.5	60.0$^{\uparrow}$	**15.7**	13.3	12.3$^{\downarrow}$	62.6	37.4	**65.7**$^{\uparrow}$
4o-mini	**66.7**	50.3	**66.7**$^{\uparrow}$	90.7	40.2	**93.3**$^{\uparrow}$	**92.2**	52.4	92.0$^{\uparrow}$	**31.2**	19.8	22.0$^{\uparrow}$	**88.0**	58.7	85.0$^{\uparrow}$
Phi-4	48.4	39.1	**55.0**$^{\uparrow}$	**79.6**	33.1	62.0$^{\uparrow}$	**76.0**	34.1	65.7$^{\uparrow}$	**28.6**	23.2	28.0$^{\uparrow}$	**90.4**	52.5	85.3$^{\uparrow}$
4.1	**88.7**	72.6	81.7$^{\uparrow}$	86.5	46.0	**96.0**$^{\uparrow}$	**98.5**	62.9	84.7$^{\uparrow}$	**54.4**	28.6	31.0$^{\uparrow}$	89.7	70.7	**91.7**$^{\uparrow}$
4o	**82.9**	61.3	76.3$^{\uparrow}$	91.7	42.3	**95.7**$^{\uparrow}$	**97.1**	65.0	82.0$^{\uparrow}$	**32.2**	20.5	27.0$^{\uparrow}$	**91.9**	67.9	89.3$^{\uparrow}$

Table 1: Average Performance $\bar{P}$ comparison across three settings: **FULL** (single-turn), **SHARDED** (multi-turn baseline), and **ERGO** (multi-turn with entropy-guided resetting). Arrow represents change in performance for ERGO relative to SHARDED, with arrow size representing magnitude of change.

stable compared to multi-turn settings across repeated trials. When comparing to standard sharded conversations, the average aptitude across models rose by 24.7%, achieving performance levels that surpass single-turn baselines, while unreliability declined by 35.3% compared to multi-turn settings.

5.2 Average Performance Gains

In addition to aptitude and unreliability improvements, Table 1 shows that ERGO delivers substantial performance improvements across all models compared to baseline multi-turn setups. By detecting moments of confusion and restarting interactions, models avoid becoming "lost" in conversational flow. Nearly every dataset and model combination shows increased average success rates, with performance improving by 56.6% on average and several model-task combinations achieving over 100% gains compared to original multi-turn baselines.

Models frequently exceeded single-turn baseline performance in both average performance and aptitude as our method only corrects derailment when calculated confusion rises significantly. This preserves the model's ability to iteratively reason and refine responses across shards while preventing the compounding errors typical in prolonged multi-turn contexts. This approach effectively merges both paradigms' strengths: single-turn stability and clarity when needed, and iterative decompositional reasoning when the model remains on track.

Moreover, performance on the Data-to-Text task improves over the multi-turn baseline, though less substantially than in other datasets. This is partly due to model-specific constraints. **LLaMA 3.1–8B** struggles to rewrite large, structured prompts effectively (e.g., full tables), limiting the benefit of consolidation. **GPT models** face difficulties in triggering resets, as entropy estimates are less reliable, only top-20 log-probabilities are available, and outputs are typically short, reducing entropy sensitivity. **Phi-4** performs best, nearing single-turn levels, likely because it supports accurate entropy tracking and handles prompt rewriting more effectively. These results indicate model-dependent limitations in applying our method to high-input-structure tasks.

5.3 Evaluating Entropy-Guided Resets vs. Random Resets and Fixed Resets

We compared entropy-based context resets against random and fixed-interval baselines using `Llama3.1-8B` across three tasks: *Database*, *Actions*, and *Math*. In these ablations, we retained all experimental settings from the main condition, with the only change being that each metric was tested on 50 question samples instead of 100. The random baseline used uniformly random triggers with unconstrained reset frequency. The fixed baseline triggered resets every five shards (quintet reset), matching the average reset frequency of `Llama3.1-8B` observed in our ERGO system. For more information on computation and average reset frequency across models, please refer to Appendix C.

The results, visualized in Figure 4, demonstrate a clear advantage for ERGO over baseline approaches. Entropy-guided resets consistently outperformed both random and fixed reset strategies while demonstrating adaptive scaling behavior. In

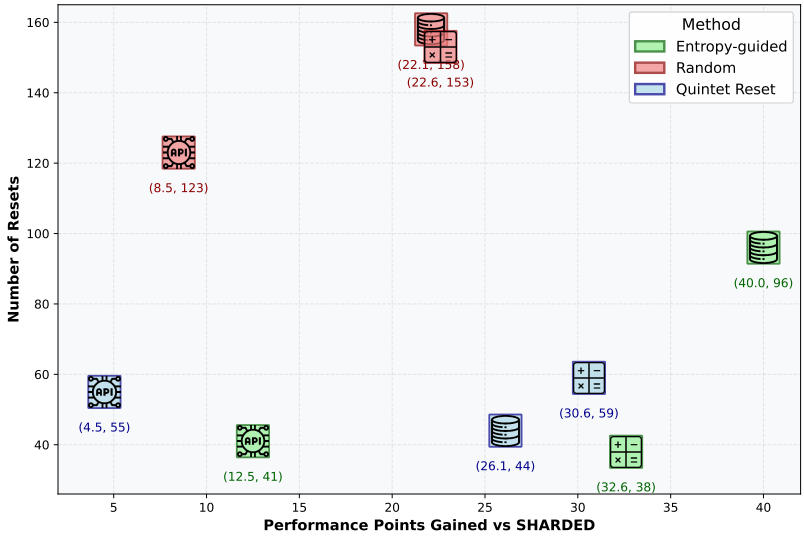


Figure 4: Comparison of performance point gains (percentage-point increase in accuracy relative to SHARDED) and number of resets across entropy-guided, random, and quintet reset methods on Database, Actions, and Math tasks. Icons represent their respective task with their color determining method used.

the Database task, ERGO achieved a performance gain of 0.400 using 96 resets, compared to the quintet baseline's 0.261 gain with only 44 resets. This demonstrates the system's ability to increase intervention frequency when encountering greater model uncertainty. Conversely, in the Actions task, ERGO required only 41 resets, fewer than both baselines, while still achieving superior performance (0.125 gain versus 0.045 and 0.085 for random and fixed approaches, respectively). This adaptive behavior indicates that entropy guided resets effectively allocate computational resources by intervening only when necessary, scaling both up and down based on task complexity and model confusion levels.

The primary risk posed by resets is semantic drift. Poorly timed or excessive context rewriting can lose critical details through increased abstraction, compromising semantic faithfulness to the original input (Dreyer et al., 2023). This degradation in semantic faithfulness can offset or even negate the benefits of resetting. Furthermore, resets incur computational overhead; each reset involves having two additional forward passes through the model. Together, these considerations underscore why the frequency and timing of resets must be carefully controlled. Not only to avoid wasted computation, but, more critically, to prevent semantic degradation.

5.4 Comparison to Existing Intervention Strategies

To contextualize the effectiveness of **ERGO**, we compare its performance against two alternative strategies introduced by Laban et al. (2025): **SNOWBALL** and **RECAP**. Both methods attempt to mitigate information loss in multi-turn settings by explicitly reintroducing previously seen content.

SNOWBALL: Reiterates all prior shards at each new turn, effectively growing the prompt cumulatively. This ensures full task visibility at each step but leads to increasing context length and potential repetition issues.

RECAP: Reiterates all prior shards only at the final turn. While more efficient, the authors note that this strategy is impractical in real-world deployments, since the system would not know *prior* when the final user input will occur.

Our method significantly outperforms both SNOWBALL and RECAP across both model variants. For GPT-4o-mini, ERGO nearly closes the gap between SHARDED and FULL baselines entirely, while for GPT-4o, ERGO performs well above competing approaches and within 3.2 points

Model	FULL	SHARDED	SNOWBALL	RECAP	ERGO
GPT-4o-mini	73.8	44.3	54.0	57.7	**71.8**
GPT-4o	79.2	51.4	57.4	66.3	**75.6**

Table 2: Comparison of average performance across Code, Database, Actions, Data-to-Text and Math tasks.

of the full information upper bound, as shown in Table 2. ERGO's advantages over static repetition-based strategies are twofold: it prevents input bloating at each iteration unlike SNOWBALL, and operates without requiring prior knowledge of the final input unlike RECAP.

5.5 Evaluating Length Bias in Entropy-Based Reset Triggers

One potential concern regarding ERGO's entropy-based reset mechanism is whether it inadvertently functions as a proxy for response length. Specifically, since entropy is calculated over token probability distributions, it is plausible that longer outputs, which involve more tokens and potentially more diffuse distributions, may naturally exhibit higher entropy. If true, this would raise the possibility that ERGO's resets are effectively triggered by length increases rather than genuine uncertainty spikes, undermining the validity of entropy as an internal behavioral signal.

We analyze response behavior from the Phi-4 model across all tasks and questions used in the main evaluation suite. For each turn t in a given multi-turn conversation, we compute two quantities relative to the previous turn: the change in average token-level entropy, $\Delta\bar{H}(t)$, and the change in response length, $\Delta L(t)$, measured in tokens.

We evaluate the relationship between these using two standard correlation metrics: Spearman's rank correlation coefficient (ρ), which captures monotonic associations without assuming linearity (Spearman, 1904), and Pearson's correlation coefficient (r), which quantifies the strength of linear correlation (Pearson, 1895). The results for the Phi-4 model are summarized in Table 3.

The Spearman result indicates no meaningful monotonic relationship between changes in entropy and length. The Pearson coefficient, while statistically significant due to the large sample size, has negligible magnitude and a negative sign, indicating no positive linear correlation.

These findings demonstrate that entropy fluctuations are not systematically associated with output length changes in the Phi-4 model. This supports the claim that ERGO's reset mechanism is not driven by verbosity or token count, but rather by internal signals of model uncertainty. Entropy-based resets therefore retain validity as an independent control signal rather than acting as a surrogate for response length.

	Coefficient	p-value
Spearman's ρ	-0.0143	0.4525
Pearson's r	-0.0796	2.7×10^{-5}

Table 3: Correlation between changes in entropy and response length for the Phi-4 model.

6 Conclusion

Our results show that ERGO effectively mitigates multi-turn LLM performance degradation by using Shannon entropy to detect model confusion and trigger prompt restructuring. Despite its simplicity, Shannon entropy serves as a reliable signal for targeted context consolidation, minimizing unnecessary resets. ERGO consistently outperformed existing methods, achieving 56.6% performance gains over standard baselines, improving aptitude by 24.7%, and reducing unreliability by 36.3%. Correlation analysis confirmed that entropy-based resets reflect genuine model uncertainty rather than response length. As a practical, model-agnostic framework, ERGO enhances conversational coherence in real-world deployments, with future work focused on advanced context consolidation strategies such as multi-stage summarization and adaptive techniques for long-form conversations.

Limitations

While ERGO achieves significant improvements in multi-turn performance via entropy-guided resets, certain avenues for future work remain.

Context Simplification: ERGO's resets currently consolidate only *user* inputs, omitting *assistant* responses. This design enables lightweight, stateless resets but limits fidelity in open-ended dialogues where assistant turns introduce key entities or reasoning steps. Without full dialogue trace consolidation, resets may discard critical context.

Threshold Adaptation: ERGO uses model-specific entropy thresholds calibrated on GSM8K, that are then fixed across datasets. While this methodology has shown to have inherit sensitivity and adapt to model capabilities (Appendix A & B). More dynamic or task-aware thresholding could improve precision.

These limitations represent natural progressions for ERGO toward broader, more general-purpose deployment. They do not challenge the core mechanism but point to extensions that scale the system into richer, more adaptable dialogue settings.

References

M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R. J. Hewett, M. Javaheripi, P. Kauffmann, and 1 others. 2024. Phi-4 Technical Report. In *arXiv preprint*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Girish Hernandez, Chelsea Edwards, Yuri Burda, Nicholas Joseph, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Chris Cundy and Stefano Ermon. 2024. Sequence-match: Imitation learning for autoregressive sequence modelling with backtracking. *Preprint*, arXiv:2306.05426.

Markus Dreyer, Mengwen Liu, Feng Nan, Sandeep Atluri, and Sujith Ravi. 2023. Evaluating the tradeoff between abstractiveness and factuality in abstractive summarization. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2089–2105, Dubrovnik, Croatia. Association for Computational Linguistics.

A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan, and 1 others. 2024. The LLaMA 3 Herd of Models. In *arXiv preprint*.

Akash Gupta, Ivaxi Sheth, Vyas Raina, Mark Gales, and Mario Fritz. 2024. LLM Task Interference: An Initial Study on the Impact of Task-Switch in Conversational History. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, and 1 others. 2024. GPT-4o System Card. In *arXiv preprint*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. LLMs Get Lost In Multi-Turn Conversation. In *Proceedings of the 2025 Conference on Language Modeling (COLM)*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024. Inference-time intervention: Eliciting truthful answers from a language model. *Preprint*, arXiv:2306.03341.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Andrey Malinin and Mark Gales. 2018. Predictive uncertainty estimation via prior networks. *Advances in neural information processing systems*, 31.

OpenAI. 2024. OpenAI o3 and o4-mini System Card.

OpenAI. 2025. Introducing gpt-4.1 in the api.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.

Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1173–1186.

Karl Pearson. 1895. Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242.

Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *Preprint*, arXiv:2308.10248.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, and 1 others. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. In *Proceedings of the 2023 Conference on Language Modeling (COLM)*.

Yuxia Xiao and William Yang Wang. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7608–7621.

Fanjia Yan, Huanzhi Mao, Charlie Cheng-Jie Ji, Tianjun Zhang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Berkeley function-calling leaderboard. `https://gorilla.cs.berkeley.edu/leaderboard.html`.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of EMNLP*, pages 811–820.

Yiming Zhang, Jianfeng Chi, Hailey Nguyen, Kartikeya Upasani, Daniel M. Bikel, Jason Weston, and Eric Michael Smith. 2024. Backtracking improves generation safety. *Preprint*, arXiv:2409.14586.

A Threshold Selection Procedure

Model Name	Version	τ	Percentile	Provider
Phi-4 A	N/A	0.1	90th	HuggingFace
Llama3.1-8b	N/A	0.03	65th	HuggingFace
GPT-4.1	gpt-4.1-2025-04-14	0.2	90th	OpenAI API
GPT-4o-mini	gpt-4o-mini-2024-07-18	0.2	85th	OpenAI API
GPT-4o	gpt-4o-2024-08-06	0.3	90th	OpenAI API

Table 4: Model versions, thresholds, and calibration percentiles used in our experiments. (Versions included where applicable.)

To determine appropriate entropy thresholds (τ) for triggering context resets, we conducted a calibration procedure specific to each model. The goal was to identify a rise in entropy that reliably signals when a model is 'lost' in the conversation, that is, when its internal uncertainty increases sharply, suggesting that it is struggling to integrate or reason over the accumulated context.

For each model, we selected a held-out subset of approximately ~ 80 shard-level examples from the GSM8K dataset. These examples were drawn from outside the final evaluation set to avoid contamination, with GSM8K being chosen due to its hybrid structure, requiring both reasoning and natural language generation. We then ran each model in a standard multi-turn setting over these shards and computed the change in average token-level predictive entropy at each turn.

From the resulting distribution of entropy rises, we selected a threshold based on a percentile aligned with the model's baseline aptitude on GSM8K. For instance, since GPT-4.1 achieves a baseline aptitude of $\sim 90\%$ on GSM8K in single-turn settings, we selected the 90th percentile of the entropy rise distribution as its reset threshold. The underlying rationale was to calibrate the threshold so that only the most atypical (high-entropy) turns, those statistically associated with likely failure, would trigger an intervention. Details of the models used, including their version identifiers, selected entropy thresholds, and corresponding calibration percentiles, are summarized in Table 4.

Once determined, this threshold was fixed across all datasets for a given model. We made this decision intentionally, as our goal was to evaluate the feasibility of a general-purpose, model-specific threshold rather than tuning thresholds for each dataset individually. This "one-size-fits-all" approach allows for a more robust and realistic assessment of whether entropy-based context resets can generalize across tasks without requiring per-task adjustment.

Interestingly, while both GPT-4.1 and Phi-4 shared the same 90th percentile threshold, Phi-4 triggered significantly more resets during evaluation. This was due to Phi-4's strong performance on GSM8K but much weaker performance on the broader set of tasks. This divergence illustrates that the system remains sensitive to task-specific confusion, with the number of resets scaling appropriately even under a fixed, model-specific threshold, highlighting the adaptive behavior of the method across domains. More information on number of resets incurred available in Appendix C.

B Sensitivity to Entropy Threshold (τ)

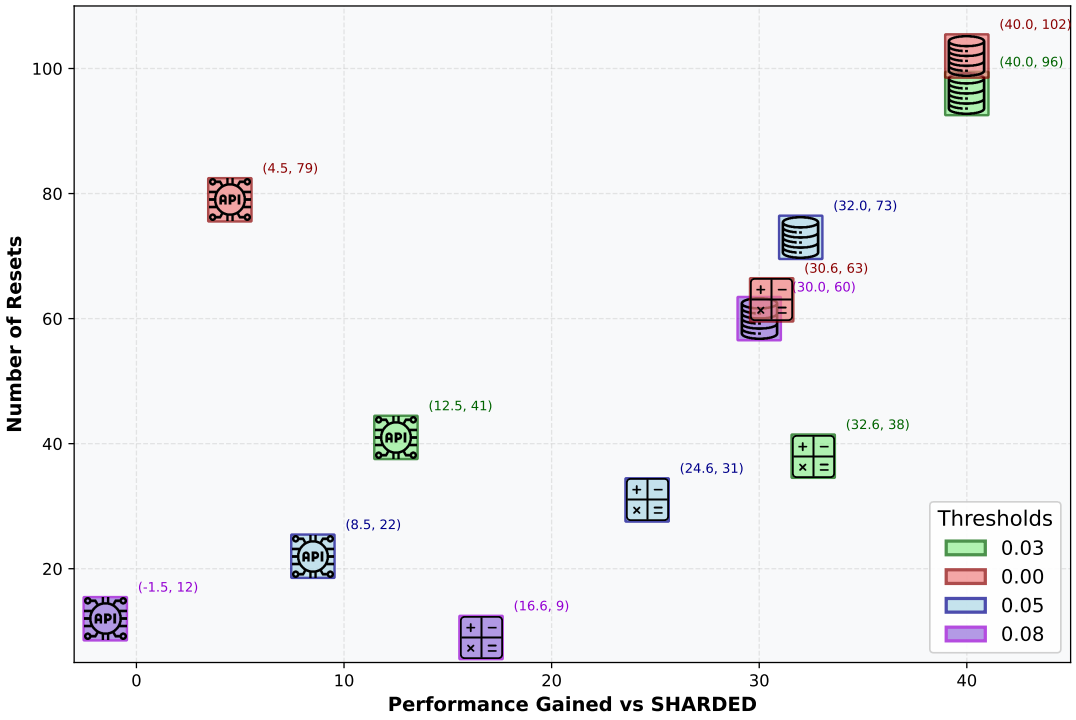


Figure 5: Comparison of maximum performance points gains (highest increase in accuracy when compared to) and number of resets between different thresholds across Database, Actions, and Math tasks.

To evaluate the sensitivity of our method to the entropy threshold parameter τ, we conducted an ablation study using the same controlled setup described in Section 5.3 with the `Llama3.1-8B` model on the *Database*, *Actions*, and *Math* tasks. The only variable changed in this study was the value of τ, the threshold used to trigger entropy-guided resets. We tested four settings: $\tau \in \{0.00, 0.03, 0.05, 0.08\}$, where 0.03 corresponds to the threshold selected for the main experiments.

The results, visualized in Figure 5 showed a clear performance peak at $\tau = 0.03$, which consistently achieved the highest gains across all tasks. This setting struck a balance between reactivity and restraint, triggering resets selectively at moments of genuine confusion without introducing excessive rewrites that risk semantic drift. In contrast, the lowest threshold $\tau = 0.00$ resulted in the highest number of resets and either matched or underperformed the 0.03 setting, suggesting that overly aggressive resetting is not beneficial and may lead to instability due to frequent context rewrites.

At the other extreme, the highest threshold $\tau = 0.08$ yielded the fewest resets and consistently underperformed, likely due to failing to intervene even when the model was demonstrably confused. The intermediate value $\tau = 0.05$ behaved as expected, yielding results that were approximately midpoint between 0.03 and 0.08 in both performance and reset count.

Taken together, these findings support the robustness of our selected threshold and highlight the importance of calibrating reset triggers to maintain a balance between informativeness and intervention overhead.

C Computational Cost and Reset Overhead Analysis

Model	Average Performance	$\sim$ Shards per Reset	Threshold Percentile
GPT-4o	75.6	51	92nd
GPT-4.1	77.2	38	90th
GPT-4o-mini	71.8	29	85th
Phi-4	59.2	7	90th
Llama3.1–8B	50.9	5	63rd

Table 5: Average Performance with ERGO along with the number of shards before reset occurs for each model and its threshold percentile, measured as an average across all datasets.

A key consideration in deploying entropy-guided context resets is the computational overhead they introduce. In our system, two sources of computational cost must be considered: (1) the cost of computing predictive entropy at each turn, and (2) the cost incurred when a context reset is triggered.

Entropy Computation Cost: While more advanced measures of model uncertainty such as semantic entropy require sampling multiple outputs over the same input (Kuhn et al., 2023), our method uses token-level Shannon entropy, which is extracted directly from the next-token probability distribution during generation. This choice imposes negligible additional cost beyond standard decoding and was selected for its practicality and compatibility with real-time systems.

Reset Overhead: Each reset introduces two additional forward passes through the model: one to rewrite the accumulated user context into a consolidated prompt, and a second to respond to that prompt. This introduces latency and compute proportional to the number of resets triggered per run. Table 5 showcases the average performance of models with ERGO along with the approximate number of shards per reset and the selected threshold percentile for each model. Averaged across all datasets, one question equates to ~ 6 shards.

These results reflect the adaptive nature of the system: more capable models (e.g., GPT-4.1, GPT-4o) experience fewer high-entropy turns and thus require fewer resets, minimizing overhead. Conversely, less capable models like `Phi-4` trigger resets more frequently, aligning with their observed confusion.

Prompt Length Reduction: An additional consequence of context resets is that they tend to truncate the context window, potentially removing stale or redundant information. Across all runs, the average token length of model prompts for questions where resets occurred was 260 tokens, compared to 309 tokens in questions where no resets were triggered. While this reduction does not eliminate the cost of the reset itself, it may partially offset it by reducing input size in subsequent turns.

Retrieval-Augmented Consolidation (Future Work): More advanced consolidation techniques, such as retrieval-augmented synthesis, could further improve the quality of resets but would introduce additional retrieval and ranking costs. We leave the exploration of such hybrid architectures to future work.

Taken together, these results indicate that while entropy-guided resets do introduce compute overhead via additional forward passes, the system remains adaptive. Reset frequency scales with model confusion, and thresholds derived from a single reasoning heavy dataset generalize effectively across diverse tasks.

D Metrics

D.1 Metric Selection

LLMs employ a stochastic decoding process, yielding different outputs even under fixed prompts and sampling parameters. We leverage this by repeating our multi-turn simulation on each sharded instruction and observing the resulting success scores. Let

$$S = \{ S_i \}_{i=1}^{N}$$

be the set of scores from N independent runs on a single instruction, where each $S_i \in [0, 100]$ measures task success at the end of that simulation.

D.1.1 Per-run scoring:

I. Binary-correctness tasks (Code, Database, API, Math): At each turn, we evaluate the model's response; if it produces a correct solution at any turn, we immediately assign $S_i = 100$ and terminate that run. If no turn yields a correct answer, $S_i = 0$.

II. Refinement task (Data-to-Text): We compute the native metric (BLEU for data-to-text; joint coverage/attribution score for summarization) on the final generated output and rescale it to $[0, 100]$.

D.1.2 Aggregate metrics

From the per-run scores S, we define three summary statistics, following the methodology from Laban et al. (2025):

$$\bar{P} = \frac{1}{N} \sum_{i=1}^{N} S_i \tag{1}$$

$$A^{90} = \text{percentile}_{90}(S) \tag{2}$$

$$U_{10}^{90} = \text{percentile}_{90}(S) - \text{percentile}_{10}(S) \tag{3}$$

-$\bar{P}$ (Average Performance): An unbiased estimate of the model's mean score on an instruction.

-A^{90} (Aptitude): Estimates the 90th-percentile performance, reflecting what one can achieve in the top decile of runs.

- U^{90} (Unreliability): Measures the gap between the 90th and 10th percentiles, capturing the degree of stochastic variability in outputs.

Aptitude and Unreliability are computed per instruction and then averaged over the full set of tasks. Binary-correctness accuracy is mapped onto the 0–100 scale to ensure every task's score aligns.

Towards Open-Ended Discovery for Low-Resource NLP

Bonaventure F. P. Dossou[1,2,*], **Henri Aïdasso**[3,*]

[1]McGill University [2]Mila Quebec AI Institute, [3]École de technologie supérieure (ÉTS)

[*]Equal Contribution

bonaventure.dossou@mila.quebec, henri.aidasso@etsmtl.ca

Abstract

Natural Language Processing (NLP) for low-resource languages remains fundamentally constrained by the lack of textual corpora, standardized orthographies, and scalable annotation pipelines. While recent advances in large language models have improved cross-lingual transfer, they remain inaccessible to underrepresented communities due to their reliance on massive, pre-collected data and centralized infrastructure. In this position paper, we argue for a paradigm shift toward open-ended, interactive language discovery, where AI systems learn new languages dynamically through dialogue rather than static datasets. We contend that the future of language technology, particularly for low-resource and under-documented languages, must move beyond static data collection pipelines toward interactive, uncertainty-driven discovery, where learning emerges dynamically from human-machine collaboration instead of being limited to pre-existing datasets. We propose a framework grounded in joint human-machine uncertainty, combining epistemic uncertainty from the model with hesitation cues and confidence signals from human speakers to guide interaction, query selection, and memory retention. This paper is a call to action: we advocate a rethinking of how AI engages with human knowledge in under-documented languages, moving from extractive data collection toward participatory, co-adaptive learning processes that respect and empower communities while discovering and preserving the world's linguistic diversity. This vision aligns with principles of human-centered AI, emphasizing interactive, cooperative model building between AI systems and speakers.

1 Introduction

The recent progress in Natural Language Processing (NLP) has been largely shaped by a data-driven paradigm. Foundation models, built on large-scale internet corpora and empowered by scaling laws, have unlocked impressive generalization across tasks and languages (Kaplan et al., 2020; Brown et al., 2020; Le Scao et al., 2022). However, this trajectory has come at a cost: the assumption that performance improves with ever more data and compute has made cutting-edge research increasingly inaccessible, especially to researchers and communities in the Global South (Sambasivan et al., 2021; Schwartz et al., 2022).

Despite efforts to democratize NLP, a stark imbalance persists. African languages, which make up over 30% of the world's linguistic diversity, account for less than 1% of NLP research output (Joshi et al., 2020). These languages typically lack large-scale text corpora, parallel datasets, and standardized annotation practices. Transfer learning, active learning, self-supervised and semi-supervised learning, have all been proposed to address this data scarcity (Howard and Ruder, 2018; Devlin et al., 2019; Ein-Dor et al., 2020; Dossou et al., 2022; Dossou, 2025; Dossou et al., 2025), but even these methods depend on the availability of some unlabeled or previously seen language data. In environments where data is extremely scarce or non-digitized, such assumptions break down.

Moreover, while recent Large Language Models (LLMs) have demonstrated impressive cross-lingual abilities, their success is closely tied to data scale, computational resources, and increasingly centralized infrastructure. As scaling laws plateau and operational costs rise, the current paradigm risks becoming both *unsustainable* and *exclusive*, limiting participation from underrepresented communities and preventing scalable solutions for the languages that need them most (Strubell et al., 2019; Bender et al., 2021; Ahmed et al., 2023).

We argue that NLP must now evolve beyond static, data-hungry training regimes. Inspired by recent work in open-ended discovery and self-improving AI (Hughes et al., 2024; Siddiqui et al., 2024), we propose a shift toward *interactive,*

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 287–297
November 9, 2025 ©2025 Association for Computational Linguistics

uncertainty-driven language learning. In our vision, AI systems learn languages not from vast corpora, but through *natural dialogue*, identifying gaps in their understanding, asking questions, and incorporating feedback in real time.

Imagine an AI system that only understands English, but receives a human input in Fon (Dossou and Emezue, 2020, 2021a; Dossou and Sabry, 2021; Dossou and Emezue, 2021b; Dossou et al., 2023). Instead of guessing or ignoring it, the system responds: ***"I do not recognize this language. Could you help me understand it?"*** *From this first exchange, it starts acquiring the new linguistic concepts interactively. Over time, through repeated exposure and correction, the system transitions from total ignorance to conversational fluency in the new language. This vision shifts the emphasis* ***from training on what we have to*** *learning from* ***what we do not yet understand****, as humans do.*

In this position paper, we explore the technical and conceptual foundations for such systems. We argue that *open-ended language learning*, grounded in *epistemic uncertainty*, *dialogue*, and *human-in-the-loop adaptation*, represents a scalable and inclusive path forward for low-resource NLP, especially in contexts where static data is not available, representative, or sufficient. We also outline a set of *open challenges* that arise from this vision, including the need for reliable uncertainty estimation, continuous learning mechanisms, and equitable access to interaction data. We discuss both the promise and the risks of this approach, including the question of whether such systems can acquire meaningful language competence without sufficient exposure or human feedback, and what architectures, incentives, or evaluation schemes would be required to support them.

2 Background and Related Work

2.1 Low-Resource Languages

Africa is one of the most linguistically diverse continents, home to over 3,000 indigenous languages (Epstein and Kole, 1998; Eberhard et al., 2024), which account for about one-third of the world's 7,159 living languages (Eberhard et al., 2024). In an increasingly digital world, where today's AI advancement such as LLMs offer unprecedented possibilities, the non-integration of these languages into the technological landscape not only exacerbates social inequalities but also poses a serious threat to the survival of entire linguistic cultures.

As inclusion and diversity gain global importance, commendable efforts have been made by researchers to identify available, albeit scarce, data sources (e.g., the Bible in Fon). Moreover, there are growing efforts for datasets creation (sometimes done manually and on a voluntary basis). These datasets have been used to create machine translation models that produce acceptable results (Dossou and Emezue, 2020; Adelani et al., 2022a). As a result, some of the very low-resource languages such as Fon, Ewe have been recently integrated into Google Translate,[1] for textual translations.

Despite these important advances, several major challenges persist that existing solutions do not, and arguably cannot address. In particular, current approaches still rely heavily on larger amounts of textual data (Adelani et al., 2022a; Dossou et al., 2022; Nekoto et al., 2020), resources that are extremely scarce or absent for many African languages and dialects (Nekoto et al., 2020; Joshi et al., 2020). Due to this reliance, existing solutions only cover a tiny fraction ($\approx$1%) of the languages, typically selected based on speaker population size or researchers' ties (Adelani et al., 2022a,b). These choices overlook the existing diversity and will ineluctably reinforce existing social inequalities and discrimination. For instance, Nigeria alone has over 500 indigenous languages (Eberhard et al., 2024), most of which severely lack written resources. Even more concerning is the practical impact of current solutions. In fact, most low-resource languages exist solely through oral traditions, meaning that the vast majority of native speakers can only speak them and struggle to read written versions, if such versions exist at all (Dossou and Emezue, 2021a; Olatunji et al., 2023b,a). Therefore, solutions that rely on textual translations are fundamentally misaligned with how these languages are actually used, making them ineffective for real-world communication needs.

2.2 Human Uncertainty Estimation

Incorporating human uncertainty into interactive learning frameworks has emerged as a critical complement to model uncertainty, as human feedback is often non-deterministic and can significantly shape model learning dynamics. Collins et al. (2023) explore concept-level interventions where humans provide feedback on intermediate concepts rather than final labels. They show that capturing the

[1] https://translate.google.com/?sl=en&tl=fon

confidence or uncertainty of these interventions, through soft labels or probabilistic feedback, improves model robustness and generalization.

Mendes et al. (2025) study the relationship between human-perceived and model-predicted uncertainties, finding only limited correlation between the two. This indicates that model uncertainty alone is insufficient to assess ambiguity in real-world settings. Explicitly modeling human uncertainty, for example, through elicited confidence scores or inter-annotator variance, can lead to more calibrated and reliable learning.

From a broader perspective, Bhatt et al. (2020) argue that exposing both human and model uncertainties enhances transparency and mutual understanding in human-AI collaboration. Similarly, collaborative annotation frameworks such as CoAnnotating (Zhang et al., 2023) leverage these uncertainty estimates to decide when to defer to human expertise or proceed autonomously, improving both efficiency and reliability in human-in-the-loop learning pipelines.

2.3 Model Uncertainty Estimation

In machine learning models, uncertainty estimation plays a crucial role in determining whether a model can respond confidently or should request clarification from the user. We denote by $f_\theta : \mathcal{X} \to \mathcal{Y}$ a parametric model with parameters θ, input $x \in \mathcal{X}$, and predictive distribution $p_\theta(y|x)$. $\mathcal{D}$ is the training dataset.

Kendall and Gal distinguish two types of uncertainty: aleatoric uncertainty (U_a) and epistemic uncertainty (U_e). Most literature works focus on U_e which is approximated by:

$$U_e(x) = \mathbb{V}_{p(\theta|\mathcal{D})}[\mathbb{E}_{p(y|x,\theta)}[y]]$$

This is U_e because directly tied to limited data or lack of model knowledge. The two most common ways of estimating U_e are the following:

With Bayesian Neural Networks BNNs (MacKay, 1992; Neal, 1996) define a posterior over weights:

$$p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta),$$

and predictive uncertainty as:

$$p(y|x, \mathcal{D}) = \int p(y|x, \theta)p(\theta|\mathcal{D})\, d\theta.$$

In practice, this integral is untractable, and approximated using variational inference (Blundell et al., 2015) or Monte Carlo sampling (Gal and Ghahramani, 2016).

With Deep Ensembles Given M independently trained models $\{f_{\theta_m}\}_{m=1}^{M}$, predictive uncertainty is quantified via:

$$U_m(x) = \frac{1}{M} \sum_m H(f_{\theta_m}(y|x)),$$

where $H(\cdot)$ is Shannon entropy.

In summary, while advances in uncertainty estimation have improved model reliability (Kendall and Gal; Gal and Ghahramani, 2016; Kirsch et al., 2019b) and recent work has explored uncertainty in human feedback (Collins et al., 2023; Mendes et al., 2025), current AI systems still learn predominantly from static datasets or treat user input as deterministic corrections. This creates two major limitations: (i) uncertainty from humans and models is rarely considered jointly, reducing the system's ability to assess when to seek clarification or defer decisions, and (ii) learning processes remain largely offline, without mechanisms to dynamically adapt to evolving user input.

To address these shortcomings, we introduce an **interactive learning system** that moves beyond passive, data-driven training. Instead of relying solely on pre-collected corpora, the system engages directly with users, identifies gaps in understanding, requests clarification when uncertainty is high, and incorporates feedback into its evolving knowledge state. This approach aims to fuse human and model uncertainties to guide the dialogue flow, enabling real-time, adaptive, and more sample-efficient language acquisition.

3 Proposed Approach

Our proposed framework enables AI systems to acquire language competence through open-ended, interactive learning. This process is illustrated in Figure 1 through interactions between a human and the AI system (agent). Rather than training on large static corpora, the system learns by engaging with users in real time, identifying gaps in its knowledge, soliciting clarification, and integrating feedback. The methodology consists of three core components: (1) modeling interactional uncertainty, (2) language acquisition via feedback, and (3) continual learning from dialogic exposure.

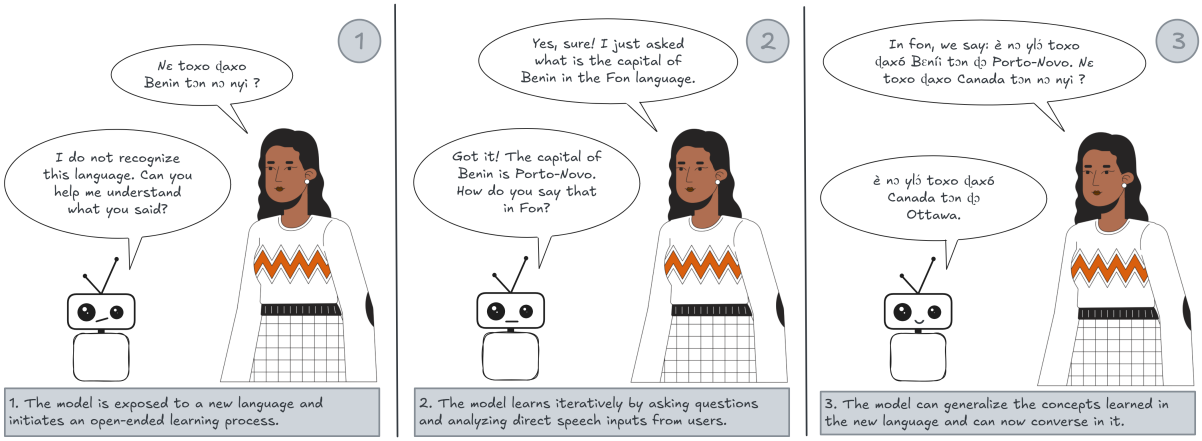


Figure 1: Illustration of the proposed approach for open-ended learning of low-resource languages. It shows the voice conversation between a human and an agent who teaches the agent to recognize and respond to requests for the capital city of a country in the Fon language.

3.1 Modeling Interactional Uncertainty

At the heart of our approach is the notion of epistemic uncertainty, which refers to the system's awareness of what it does not know. In conventional NLP, model uncertainty is often used for tasks like active learning or confidence calibration (Kendall and Gal; Gal and Ghahramani, 2016; Houlsby et al., 2011; Guo et al., 2017). Here, we extend this principle to guide decision-making during interactive language learning.

We define a composite uncertainty signal combining both human and machine contributions:

$$\mathcal{U}_{\text{total}} = \alpha \cdot \mathcal{U}_{\text{human}} + (1 - \alpha) \cdot \mathcal{U}_{\text{model}}$$

where $\mathcal{U}_{\text{model}}$ is the model's epistemic uncertainty, estimated via entropy, ensemble disagreement, or Bayesian approximations (Kendall and Gal; Kirsch et al., 2019a; Gal et al., 2017; Gal and Ghahramani, 2016; Kirsch et al., 2023), $\mathcal{U}_{\text{human}}$ reflects uncertainty inferred from hesitation cues, conflicting corrections, or prosodic markers, and α controls the relative influence of human versus machine uncertainty.

Given this signal, the system selects a query Q^* to ask the human speaker, optimizing:

$$Q^* = \arg\max_{Q} \frac{\mathbb{E}[\text{InfoGain}(Q)]}{\text{Cost}(Q, \mathcal{U}_{\text{human}})}$$

where

$$\text{Cost}(Q, \mathcal{U}_{\text{human}}) = c(Q)\big(1 + \lambda * \mathcal{U}_{\text{human}}\big)$$

with $c(Q)$ representing the baseline time or cognitive effort required for query type Q, and $\lambda \geq 0$ controlling how strongly human uncertainty increases perceived cost. This interaction cost reflects the human effort required to answer a query and the likelihood of confusion when the speaker is already uncertain. Scaling the cost by $(1 + \lambda * \mathcal{U}_{\text{human}})$ ensures the system avoids queries that are both expensive and likely to yield ambiguous responses. This improves efficiency and user experience, making learning cooperative rather than extractive.

The expected information gain from a query Q is defined as the anticipated reduction in predictive uncertainty:

$$\text{InfoGain}(Q) = \mathbb{H}[Y \mid x, \mathcal{D}] - \mathbb{E}_{A \sim p(A|Q)}\big[\mathbb{H}[Y \mid x, \mathcal{D}, Q, A]\big]$$

where $\mathbb{H}[\cdot]$ denotes Shannon entropy, $\mathcal{D}$ is the current learner state, and A denotes a human response sampled from $p(A \mid Q)$. This term quantifies how much uncertainty the query is expected to resolve. We define $p(A|Q)$ as the conditional distribution over possible human responses given a query Q. This distribution models the variability and uncertainty in human feedback due to ambiguity in meaning, hesitation or noise in responses, and contextual variability across speakers.

The selected query Q^* and the anticipated distribution of human responses $p(A|Q)$ provide the necessary context for the next stage, where human feedback is integrated into the model.

3.2 Language Acquisition via Human Feedback

Once a query Q^* has been selected based on the joint uncertainty signal, the AI system receives a

feedback signal A from the human speaker. In this stage, the goal is to integrate the new information into the model's knowledge while accounting for both human and model uncertainty.

A targeted query Q is designed to elicit clarifying information about input x, such as asking ***"What does this word mean?"*** or ***"How would you say this sentence?"***. The response is denoted as $A \sim p(A|Q)$, sampled from a conditional distribution over possible answers. This distribution reflects that feedback may vary or include ambiguity, such as multiple possible translations or uncertain corrections.

We denote $p_\theta(\cdot|x)$ as the model's current predictive distribution over possible meanings or utterances for input x, parameterized by θ. The human feedback is represented as y_{human}, a meaning distribution derived from the response A. It can be sharp, corresponding to a single unambiguous answer, or soft, capturing several plausible meanings with associated probabilities. Finally, we introduce a reliability weight $w_f = 1 - \mathcal{U}_{\text{human}}$, which downscales the influence of uncertain human feedback. When human uncertainty is high, the system places less emphasis on the feedback to avoid reinforcing potentially misleading signals.

Using these definitions, the system constructs a new target distribution that combines its own prior predictions with the received feedback:

$$\tilde{y} = w_f \cdot y_{\text{human}} + (1 - w_f) \cdot p_\theta(\cdot|x)$$

This weighted target guides the parameter update:

$$\theta' = \theta - \eta \nabla_\theta \mathcal{L}(p_\theta(\cdot|x), \tilde{y})$$

where η is the learning rate and $\mathcal{L}$ is a loss function. KL Divergence can be used to align the model's predicted distribution with human-provided meaning probabilities in a continuous space, making it well-suited for uncertain or soft feedback. Contrastive Loss distinguishes correct meanings from alternative ones in an embedding space, supporting open-ended discovery where meanings are not predefined. Categorical Cross-Entropy works when the system has a finite set of candidate meanings, though it is less ideal for open-ended language learning since it assumes predefined categories.

This approach allows the system to integrate human feedback incrementally and proportionally to its reliability, while still preserving useful prior knowledge from its own predictions. In the future, more appropriate loss functions could be designed specifically for dialogic, open-ended learning scenarios to better reflect the uncertainty and flexibility inherent in human language interactions.

3.3 Continual Learning from Dialogic Exposure

Language acquisition is not a single-step process. Over multiple interactions, the system must consolidate knowledge, refine uncertain examples, and adapt to evolving feedback. To achieve this, every interaction is stored in a memory bank:

$$\mathcal{M} = \{(x_i, A_i, w_i)\}$$

where each element consists of the input x_i, the human feedback A_i, and an associated weight:

$$w_i = (1 - \mathcal{U}^{(i)}_{\text{human}})(1 - \mathcal{U}^{(i)}_{\text{model}})$$

This weight captures the combined confidence of both the human and the model for a given interaction.

The memory bank $\mathcal{M}$ acts as a growing repository of past interactions with human speakers, each stored alongside a weight indicating reliability. Periodically, the system revisits stored samples to reinforce reliable information and re-query ambiguous examples. Past interactions are used to improve the model through uncertainty-aware gradient updates:

$$\theta \leftarrow \theta - \eta \sum_i w_i \nabla_\theta \mathcal{L}(p_\theta(\cdot|x_i), A_i)$$

$p_\theta(\cdot|x_i)$ refers to the same predictive model introduced in Section 3.2, now updated iteratively using both immediate feedback and stored memory samples. We reuse the notation to emphasize that the model evolves over time through repeated uncertainty-guided interactions.

Low-weight samples contribute less to the update, preventing uncertain or noisy feedback from degrading the learned representation. They are not discarded but flagged for future re-querying when opportunities arise. This creates a closed interactive loop where the system encounters new input x, computes $\mathcal{U}_{\text{total}}$ and selects an optimal query Q^*, collects human feedback A and updates parameters incrementally, stores the interaction in $\mathcal{M}$ with weight w_i, and periodically revisits uncertain cases to refine or validate earlier knowledge, looping back when necessary.

Through these mechanisms, uncertainty evolves from a static confidence score into an active principle governing when to trust, query, defer, or memorize. This continual process ensures that learning

is incremental, reliable, and co-adaptive. It enables the system to refine its internal representations over time, progressively improving its understanding of a new language while remaining sensitive to the reliability of past and future feedback. Together, these three stages establish a self-reinforcing loop for interactive language discovery, where uncertainty not only shapes individual interactions but also drives long-term, co-adaptive learning.

4 Opportunities and Challenges

Our proposed framework for open-ended language discovery leverages joint human-machine uncertainty to guide interaction, query selection, and memory retention. While the approach introduces a novel paradigm for low-resource language acquisition, its success and limitations stem directly from the mechanisms we designed. Unlike conventional NLP pipelines that rely on static, curated datasets and post-hoc analysis, this framework is designed for real-time, adaptive interaction. It emphasizes uncertainty-driven decision-making, enabling language acquisition to progress even when large corpora, standardized orthographies, or expert annotators are unavailable.

4.1 Why This Could Work

The framework builds on several principles that make it uniquely suited for interactive, low-resource settings. By explicitly modeling epistemic uncertainty, the system learns what it does not know and can focus queries on areas of high information gain rather than engaging in blind memorization. This targeted querying mechanism has the potential to accelerate language acquisition compared to static corpus-based training approaches. Incorporating $\mathcal{U}_{\text{human}}$ allows the system to defer or prioritize information based on human confidence, ensuring that reliable feedback from fluent speakers directly shapes the learned representation and reduces noise in the earliest stages of learning. Over time, dynamic weighting (α) adapts reliance on each contributor according to their observed consistency and reliability, making the system robust to heterogeneous or occasional feedback. Furthermore, confidence-weighted memory retention enables iterative refinement of knowledge: high-certainty information consolidates quickly, while ambiguous examples remain open for re-querying, progressively building a stable and trustworthy knowledge base. Together, these mechanisms enable data-efficient learning that can bootstrap language understanding from a small number of high-value interactions, making it feasible in settings where large corpora are unavailable. These properties suggest that joint human–machine uncertainty could form the backbone of scalable, respectful, and data-efficient language acquisition, where conventional supervised NLP pipelines cannot operate.

4.1.1 In the Context of Low-Resource African Languages

Low-resource African languages often face a unique combination of challenges that make standard NLP pipelines ineffective: severe data scarcity, highly variable orthographies, oral traditions without standardized writing systems, and limited availability of expert annotators. The proposed framework is particularly well-suited to this context because it does not rely on pre-existing corpora or formal linguistic resources. Instead, it learns interactively from small, high-value exchanges, asking only those questions that are most informative given its current uncertainty. This targeted learning process minimizes the burden on speakers, who may have limited time or literacy in standardized orthography, while still allowing the system to rapidly form hypotheses about grammar, semantics, and phonology.

Moreover, the joint modeling of human and machine uncertainty makes the framework robust to the realities of field data collection in African settings, where contributors may have varying degrees of fluency, confidence, or even differing dialects of the same language. By adapting reliance on each contributor through dynamically learned weighting (α), the framework can filter noise while still capturing dialectal richness. Its ability to defer uncertain information and revisit ambiguous examples ensures that rare or culturally significant linguistic forms are not prematurely discarded. These properties make it a promising approach for preserving, documenting, and learning African languages where the cost of traditional data collection is prohibitive and where respectful, participatory collaboration with speakers is essential. This approach not only addresses data scarcity but also reframes language technology development as a collaborative process between AI systems and speakers. By moving away from extractive data collection toward live, adaptive interaction, it offers a pathway for NLP to support language documentation and revitalization efforts. Particularly in marginalized

communities, this paradigm empowers speakers to co-create technology aligned with their linguistic and cultural realities, potentially reshaping how AI contributes to the preservation and expansion of global linguistic diversity.

4.1.2 In the Context of Human-Centered AI and Human-Computer Interactions

The proposed framework embodies principles of human-centered artificial intelligence by placing speakers at the center of the learning process. Rather than treating them as static annotators or sources of labels, it engages in a cooperative interaction where both human and machine uncertainty guide the flow of information exchange. This fosters transparency and trust, as speakers can see that the system acknowledges its own uncertainty, adapts to their confidence levels, and defers decisions when information is unclear.

From a Human-Computer Interaction (HCI) standpoint, the framework reduces the cognitive and emotional burden on contributors by focusing only on high-value, contextually relevant questions instead of overwhelming them with repetitive or trivial requests. It can adapt the pace and style of interaction based on hesitation cues, feedback latency, or non-verbal indicators of uncertainty, making it more accessible to non-expert participants. Additionally, the iterative refinement of memory ensures that early mistakes can be revisited and corrected collaboratively, giving speakers a sense of agency and ownership in shaping the emerging language model. This paradigm transforms data collection from a one-way, extractive process into a participatory dialogue, contributing to the development of AI systems that are not only technically effective but also socially aligned and respectful toward the communities they aim to serve. In doing so, it demonstrates a path toward genuinely human-centered AI, where computational methods adapt to people, rather than asking people to adapt to technology. This vision is aligned with participatory and co-design approaches explored in HCI research (Liao and Vaughan, 2023; Birhane et al., 2022; Delgado et al., 2023), which emphasize collaborative model building, transparency, and community agency in shaping AI behavior.

While these properties highlight the potential of our framework to enable scalable, and data-efficient language learning, realizing this vision in practice is far from trivial. Uncertainty-guided discovery introduces its own vulnerabilities, and deploying such systems in real-world low-resource environments presents additional technical and sociotechnical barriers, that must be addressed. The following section discusses these open challenges.

4.2 Challenges

Several challenges could undermine the effectiveness of the proposed framework in practice. A first concern lies in the reliability of uncertainty estimation. Because the system operates on highly out-of-distribution data such as new languages, unseen constructs, and unpredictable input patterns, its uncertainty signals may not be well calibrated. Miscalibration could lead to redundant or unnecessary queries, or conversely, to missed opportunities to acquire valuable information early on.

Human uncertainty signals introduce another layer of complexity. Hesitation cues, conflicting answers, or silence are not always reliable indicators of a speaker's true confidence. Cultural norms and individual communication styles can further distort these signals, leading the system to over-trust uncertain information or defer excessively even when a speaker would have provided correct input. This unreliability in feedback interpretation can propagate downstream errors in learning.

Errors may also arise in the adaptive weighting mechanism. Because α must be learned online from sparse observations, early interactions can dominate future weighting, allowing biases from the first few contributors to persist unchecked. In heterogeneous communities where speaker reliability varies widely, it becomes difficult to estimate contributor trustworthiness accurately, which risks amplifying noise and reducing the value of human input. This interacts closely with query selection: without stable reliability or cost estimates, the system may waste interactions on poorly chosen clarifications, frustrating users and slowing overall progress.

The memory component presents its own risks. Confidence-weighted retention is designed to consolidate reliable information quickly, but if misinterpreted feedback is assigned high confidence, early errors risk becoming fossilized in the learned representation. Conversely, rare linguistic forms may repeatedly receive low-confidence scores, preventing their integration and leaving parts of the language undocumented or misunderstood. This challenge is compounded in what we term a "double-uncertainty deadlock," where both the model and the human contributors remain uncertain for ex-

tended periods. In such cases, the system may repeatedly defer decisions, becoming overly cautious and failing to test hypotheses that could break the cycle of uncertainty.

Finally, practical constraints in real-world deployment cannot be ignored. Reliable uncertainty estimation, adaptive weighting, and dynamic query selection all introduce computational overhead that may be infeasible on low-cost, battery-limited, or offline devices. Connectivity issues, limited processing power, and fragile hardware environments could hinder the ability of the framework to operate effectively in the very settings it aims to serve.

4.3 Future Directions

Addressing these challenges requires progress on several fronts. Improving epistemic uncertainty estimation in open-ended, out-of-distribution language input is a priority, as more reliable measures would reduce unnecessary queries and strengthen the system's ability to make informative decisions early on. Equally important is the development of context-aware and culturally adaptive models of human uncertainty, since hesitation and confidence cues vary widely across individuals and communities. Advancing methods for learning α from sparse interactions will also be key to mitigating early biases, ensuring that the system adapts fairly and dynamically to multiple contributors over time.

Meta-learning approaches offer a promising path toward improving α estimation. By transferring priors on speaker reliability from related language acquisition sessions or typologically similar languages, the system could begin with more informed weighting strategies, reducing the risk of overfitting to a handful of early interactions. This would make adaptation faster and more stable, even in diverse or previously unseen linguistic settings.

Developing multi-agent exploration policies could further enhance query selection. Instead of treating human interactions in isolation, coordinated strategies could balance information gain, contributor reliability, and annotation cost across multiple speakers. Such strategies might deliberately diversify queries to capture rare linguistic forms, seek cross-validation from independent sources to resolve ambiguities, and avoid overloading single contributors, making learning more efficient and collaborative.

Breaking double-uncertainty deadlocks will require exploration mechanisms that take calculated risks when both human and model uncertainty remain high. Periodic re-querying, rediscovery routines, and targeted hypothesis testing could help overcome conservativeness and expand the system's knowledge base over time. Finally, lightweight, offline-capable implementations of the framework are necessary for real-world deployment. Achieving efficient uncertainty estimation, adaptive query selection, and meta-learning-based weighting on low-power devices would make the approach scalable and practical for under-resourced communities that lack access to high-compute infrastructure.

If these research directions are pursued, joint human-machine uncertainty could unlock scalable, interactive, and respectful language learning systems capable of discovering and documenting under-resourced languages without relying on large curated datasets. Ultimately, this line of research bridges technical innovation and participatory design, opening opportunities for AI systems that learn with people, not just from data.

5 Conclusion

This paper outlines a vision for open-ended language discovery based on joint human-machine uncertainty. We argue that future NLP systems, particularly for low-resource languages, must move beyond static data pipelines and toward interactive, participatory approaches that adapt to sparse, uncertain, and heterogeneous feedback. While many technical and sociotechnical challenges remain, this is not merely a research proposal but an ideological stance: language technology should be co-created with speakers. We position this work as a challenge to current practices that treat language as extractable data, advocating instead for AI systems that become collaborative participants in language preservation and revitalization. We call on the NLP and HCI research communities to develop methods, tools, and evaluation practices that support co-adaptive language learning systems, opening new pathways for linguistic documentation, preservation, and empowerment in the digital age. This position paper advocates for a paradigm shift: from building models that passively learn from existing data to designing systems that actively learn with people in real time, fostering respectful, human-centered AI for linguistic diversity.

References

David Ifeoluwa Adelani, Jesujoba Oluwadara Alabi, Angela Fan, Julia Kreutzer, Xiaoyu Shen, Machel Reid, Dana Ruiter, Dietrich Klakow, Peter Nabende, Ernie Chang, Tajuddeen Gwadabe, Freshia Sackey, Bonaventure F. P. Dossou, Chris Emezue, Colin Leong, Michael Beukman, Shamsuddeen H. Muhammad, Guyo D. Jarso, Oreen Yousuf, and 26 others. 2022a. A few thousand translations go a long way! leveraging pre-trained models for African news translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3053–3070, Seattle, United States. Association for Computational Linguistics.

David Ifeoluwa Adelani, Graham Neubig, Sebastian Ruder, Shruti Rijhwani, Michael Beukman, Chester Palen-Michel, Constantine Lignos, Jesujoba O. Alabi, Shamsuddeen H. Muhammad, Peter Nabende, Cheikh M. Bamba Dione, Andiswa Bukula, Rooweither Mabuya, Bonaventure F. P. Dossou, Blessing Sibanda, Happy Buzaaba, Jonathan Mukiibi, Godson Kalipe, Derguene Mbaye, and 26 others. 2022b. MasakhaNER 2.0: Africa-centric transfer learning for named entity recognition. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4488–4508, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Saadia Gabriel Ahmed, Noah A. Smith, and Swabha Swayamdipta. 2023. Cultural and socioeconomic inequities in language model pretraining. *arXiv preprint arXiv:2305.07801*.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*.

Umang Bhatt, Adrian Weller, and 1 others. 2020. Uncertainty as a form of transparency: Measuring, communicating, and using uncertainty in ai. *arXiv preprint arXiv:2011.07586*.

Abeba Birhane, William Isaac, Vinodkumar Prabhakaran, Mark Diaz, Madeleine Clare Elish, Iason Gabriel, and Shakir Mohamed. 2022. Power to the people? opportunities and challenges for participatory ai. In *Proceedings of the 2nd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, pages 1–8.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. 2015. Weight uncertainty in neural networks. In *International Conference on Machine Learning*, pages 1613–1622.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*.

Katherine Collins, Guangyao Zhou, Been Kim, and David Sontag. 2023. Learning with uncertain concept interventions. *arXiv preprint arXiv:2303.12872*.

Fernando Delgado, Stephen Yang, Michael Madaio, and Qian Yang. 2023. The participatory turn in ai design: Theoretical foundations and the current state of practice. In *Proceedings of the 3rd ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*, EAAMO '23, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Bonaventure F. P. Dossou. 2025. Advancing african-accented english speech recognition: Epistemic uncertainty-driven data selection for generalizable asr models. In *Proceedings of the ACL Student Research Workshop*.

Bonaventure F. P. Dossou, Ines Arous, and Jackie CK Cheung. 2025. Rethinking full finetuning from pretraining checkpoints in active learning for african languages. In *Proceedings of the ACL Student Research Workshop*.

Bonaventure F. P. Dossou and Chris Chinenye Emezue. 2021a. OkwuGbé: End-to-end speech recognition for Fon and Igbo. In *Proceedings of the Fifth Workshop on Widening Natural Language Processing*, pages 1–4, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Bonaventure F. P. Dossou, Atnafu Lampebo Tonja, Oreen Yousuf, Salomey Osei, Abigail Oppong, Iyanuoluwa Shode, Oluwabusayo Olufunke Awoyomi, and Chris Emezue. 2022. Afrolm: A self-active learning-based multilingual pretrained language model for 23 african languages. In *Proceedings of the Third Workshop on Data-Centric AI (SustainNLP)*.

Bonaventure FP Dossou and Chris C Emezue. 2021b. Crowdsourced phrase-based tokenization for low-resourced neural machine translation: The case of fon language. *arXiv preprint arXiv:2103.08052*.

Bonaventure FP Dossou, Iffanice Houndayi, Pamely Zantou, and Gilles Hacheme. 2023. Fonmtl: Towards multitask learning for the fon language. *arXiv preprint arXiv:2308.14280*.

Bonaventure FP Dossou and Mohammed Sabry. 2021. Afrivec: Word embedding models for african languages. case study of fon and nobiin. *arXiv preprint arXiv:2103.05132*.

Femi Pancrace Bonaventure Dossou and Chris Chinenye Emezue. 2020. FFR v1.1: Fon-French neural machine translation. In *Proceedings of the Fourth Widening Natural Language Processing Workshop*, pages 83–87, Seattle, USA. Association for Computational Linguistics.

David M. Eberhard, Gary F. Simons, and Charles D. Fennig. 2024. *Ethnologue: Languages of the World*, 27 edition. SIL International, Dallas, Texas.

Liat Ein-Dor, Einat Rabinovich, and Michael Elhadad. 2020. Active learning for bert: An empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Edmund L. Epstein and Robert Kole. 1998. *The Language of African Literature*. Africa World Press. Google-Books-ID: XkkrDH27jmIC.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1183–1192. JMLR.org.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. In *Proceedings of the 29th International Conference on Machine Learning*.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Edward Hughes, Michael Dennis, Jack Parker-Holder, Feryal Behbahani, Aditi Mavalankar, Yuge Shi, Tom Schaul, and Tim Rocktäschel. 2024. Open-endedness is essential for artificial superhuman intelligence. *arXiv preprint arXiv:2406.04268*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the nlp world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in Neural Information Processing Systems*, 30:5574–5584.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019a. *BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning*. Curran Associates Inc., Red Hook, NY, USA.

Andreas Kirsch, Sebastian Farquhar, Parmida Atighehchian, Andrew Jesson, Frederic Branchaud-Charron, and Yarin Gal. 2023. Stochastic batch acquisition: A simple baseline for deep active learning. *Preprint*, arXiv:2106.12059.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. 2019b. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in neural information processing systems*, volume 32.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Matthias Gallé, Yacine Jernite Wang, Timo Schick, Maha El Bayad, and et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Q Vera Liao and Jennifer Wortman Vaughan. 2023. Ai transparency in the age of llms: A human-centered research roadmap. *arXiv preprint arXiv:2306.01941*, 10.

David JC MacKay. 1992. *A practical Bayesian framework for backpropagation networks*. Neural computation.

João Mendes, Shivangi Agarwal, Wenlong Zhou, and Balaji Lakshminarayanan. 2025. Human perception of uncertainty and neural model confidence. *arXiv preprint arXiv:2506.15850*.

Radford M Neal. 1996. *Bayesian learning for neural networks*. Ph.D. thesis, University of Toronto.

Wilhelmina Nekoto, Vukosi Marivate, Tshinondiwa Matsila, Timi Fasubaa, Taiwo Fagbohungbe, Solomon Oluwole Akinola, Shamsuddeen Muhammad, Salomon Kabongo Kabenamualu, Salomey Osei, Freshia Sackey, Rubungo Andre Niyongabo, Ricky Macharm, Perez Ogayo, Orevaoghene Ahia, Musie Meressa Berhe, Mofetoluwa Adeyemi, Masabata Mokgesi-Selinga, Lawrence Okegbemi, Laura Martinus, and 28 others. 2020. Participatory research for low-resourced machine translation: A case study in African languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2144–2160, Online. Association for Computational Linguistics.

Tobi Olatunji, Tejumade Afonja, Bonaventure F. P. Dossou, Atnafu Lambebo Tonja, Chris Chinenye Emezue, Amina Mardiyyah Rufai, and Sahib Singh. 2023a. Afrinames: Most asr models "butcher" african names. In *Interspeech 2023*, pages 5077–5081.

Tobi Olatunji, Tejumade Afonja, Aditya Yadavalli, Chris Chinenye Emezue, Sahib Singh, Bonaventure F. P. Dossou, Joanne Osuchukwu, Salomey Osei, Atnafu Lambebo Tonja, Naome Etori, and Clinton Mbataku. 2023b. Afrispeech-200: Pan-african accented speech dataset for clinical and general domain asr. *Transactions of the Association for Computational Linguistics*, 11:1669–1685.

Nithya Sambasivan, Samyukta Kapania, Praveen Paritosh, Sagar Jain, Paul Aoki, and Ravi Nair. 2021. Re-imagining algorithmic fairness in india and beyond. *Communications of the ACM*.

Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2022. Green ai and the need for sustainable nlp. *Communications of the ACM*.

Shoaib Siddiqui, Mor Geva, Ethan Chi, Ilia Liskovich, Jordan Hoffman, Eric Zelikman Wu, Aakanksha Chowdhery, Jan Leike, Barret Zoph, Neil Houlsby, Roger B. Grosse, and Vedant Misra. 2024. Automated capability discovery via foundation model self-exploration. *arXiv preprint arXiv:2502.07577*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Yi Zhang, Jialin Li, and Danqi Chen. 2023. Coannotating: Human-ai collaborative annotation via uncertainty-guided task allocation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.

Can Vision-Language Models Infer Speaker's Ignorance? The Role of Visual and Linguistic Cues

Ye-eun Cho
English language and literature
Sungkyunkwan University
Seoul, South Korea
joyenn@skku.edu

Yunho Maeng
Ewha Womans University &
LLM Experimental Lab, MODULABS
Seoul, South Korea
yunhomaeng@ewha.ac.kr

Abstract

This study investigates whether vision language models (VLM) can perform pragmatic inference, focusing on ignorance implicatures, utterances that imply the speaker's lack of precise knowledge. To test this, we systematically manipulated contextual cues: the visually depicted situation (visual cue) and QUD-based linguistic prompts (linguistic cue). When only visual cues were provided, three state-of-the-art VLMs (GPT-4o, Gemini 1.5 Pro, and Claude 3.5 sonnet) produced interpretations largely based on the lexical meaning of the modified numerals. When linguistic cues were added to enhance contextual informativeness, Claude exhibited more human-like inference by integrating both types of contextual cues. In contrast, GPT and Gemini favored precise, literal interpretations. Although the influence of contextual cues increased, they treated each contextual cue independently and aligned them with semantic features rather than engaging in context-driven reasoning. These findings suggest that although the models differ in how they handle contextual cues, Claude's ability to combine multiple cues may signal emerging pragmatic reasoning abilities in multimodal models.

1 Introduction

In recent years, many large language models (LLMs) have demonstrated the ability to solve a wide variety of tasks, contributing to their growing popularity. Initially limited to text-based inputs, these models have been extended to incorporate visual inputs, paving the way for vision-language models (VLMs). By bridging vision and language modalities, VLMs have expanded the possibilities for AI applications and become central to the ongoing technological revolution (Radford et al., 2021; Ramesh et al., 2021; Alayrac et al., 2022; Li et al., 2023).

VLMs have enabled various multimodal applications, such as object recognition (Ren et al., 2015; Chen et al., 2020; He et al., 2020), caption generation (Vinyals et al., 2015; Chen et al., 2022; Yu et al., 2022), and visual question answering (Antol et al., 2015). These tasks primarily focus on associations between visual and textual inputs by identifying objects, describing scenes, or responding to straightforward queries. While such capabilities are remarkable, they represent only the surface level of human-like understanding. In fact, real-world communication often requires reasoning about implicit meanings that emerge from the interplay between language and visual information (Sikka et al., 2019). To move toward more human-like multimodal intelligence, VLMs must also be able to engage in this type of context-sensitive and inferential processing (see Kruk et al., 2019). This raises critical questions about VLMs' capacity for context-sensitive reasoning, which underlies the pragmatic reasoning abilities required for real-world communication.

Pragmatics offers an ideal framework for investigating this question. In human communication, pragmatic inference plays a crucial role in understanding intended meanings beyond literal statement (Grice, 1975; Wilson and Sperber, 1995; Levinson, 2000). Contextual cues often provide disambiguating information that influences the interpretation of utterances, making pragmatic reasoning inherently multimodal (Clark, 1996; Kendon, 2004; Martin et al., 2007; McNeill, 2008). While a few studies on pragmatic reasoning have been explored in text-only LLMs (Hu et al., 2022, 2023; Lipkin et al., 2023; Cho and Kim, 2024; Capuano and Kaup, 2024; Tsvilodub et al., 2024), the visual modality enriches meaning construction through interaction with linguistic input. As visual context provides rich, implicit information that influences language interpretation, studying pragmatic phenomena through VLMs presents an intriguing research opportunity. However, how well VLMs can leverage visual information for pragmatic inference

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 298–308
November 9, 2025 ©2025 Association for Computational Linguistics

remains largely unexplored.

Therefore, we investigate whether VLMs exhibit sensitivity to context, particularly focusing on ignorance implicatures—a pragmatic phenomenon in which a speaker's utterance implies a lack of precise knowledge, and whether this sensitivity can be modulated by a single cue or by the combination of multiple cues. By examining how VLMs handle this phenomenon in comparison to human reasoning, we aim to better understand their strengths and limitations in processing context-dependent pragmatic meaning.

2 Ignorance implicatures

To better understand pragmatic reasoning in real-world language use, we examine the phenomenon of *ignorance implicatures*. Consider the examples in (1).

(1)

a. *(bare numeral)*
Four students passed the exam.

b. *(superlative modified numeral)*
At least four students passed the exam.

c. *(comparative modified numeral)*
More than three students passed the exam.

When it comes to how many students passed the exam, the statement (1a) triggers 'exactly four' interpretation, whereas (1b) and (1c) do not. Both (1b) and (1c) contain modified numerals, suggesting that the speaker may not know the exact number of students who passed the exam. This is known as ignorance implicatures, where the speaker's choice of modifier implies a lack of precise knowledge.

However, not all modifiers give rise to ignorance implicatures to the same extent. Previous studies have shown that superlative modifiers like *at least* tend to trigger ignorance implicatures more consistently than comparative ones like *more than* (Nouwen, 2010; Cummins et al., 2012; Coppock and Brochhagen, 2013b; Mayr and Meyer, 2014; Cremers et al., 2022). In this regard, the likelihood of ignorance inferences typically follows the hierarchy: *superlative modified numerals* > *comparative modified numerals* > *bare numerals*.

This observation has prompted researchers to explore how such inferences arise, leading to two main perspectives. One approach suggests that ignorance inference is dependent on the words or phrases themselves (Geurts and Nouwen, 2007; Nouwen, 2010; also see Geurts et al., 2010). Geurts and Nouwen (2007), for example, argued that the semantics of superlative modifiers are inherently more complex. While *more than n* expresses a simple meaning 'larger than *n*', *at least n* can convey both 'possible that there is a set of *n*' and 'certain that there is no smaller set of *n*.' According to Nouwen (2010), when someone has basic knowledge of geometry, (2a) gives the impression that the speaker lacks precise information, as compared to (2b). This attributes the ignorance implicatures to a semantic property specific to *at least*.

(2) a. ? A hexagon has at least five sides.
b. A hexagon has more than four sides.

Under the pragmatic account, on the other hand, ignorance implicatures for both *at least* and *more than* have been primarily explained through Gricean reasoning, particularly the Maxim of Quantity (Grice, 1975), which holds that the speaker's choice to provide a lower-bound statement, rather than a more informative exact number, suggests that the speaker lacks precise knowledge (Büring, 2008; Cummins and Katsos, 2010; Coppock and Brochhagen, 2013b). More recent studies have expanded this account by emphasizing the role of contextual factors (Cummins et al., 2012; Cummins, 2013; Mayr and Meyer, 2014; Westera and Brasoveanu, 2014; Cremers et al., 2022). In these studies, contextual cues, including Question Under Discussion (QUD), preceding discourse, or accompanying visual input, were manipulated to modulate the likelihood of implicature.

For instance, Westera and Brasoveanu (2014) investigated how different types of modified numerals give rise to ignorance implicatures depending on contextual demands and processing cost. In their experiments, participants read short dialogues or utterances containing modified numerals and judged how confident the speaker seemed about the exact quantity, as well as how natural the utterance was. To manipulate the informativeness required by the discourse, the authors introduced different QUDs, such as a 'how many' condition (*How many of the diamonds did you find under the bed?*), which demanded precise answers, and a 'polar' condition (*Did you find {at most | less than} ten of the diamonds under the bed?*), which did not require numerically specific responses, as they could be answered with a simple *yes* or *no*. The results

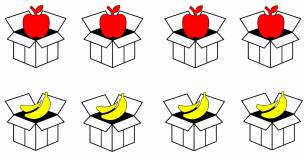
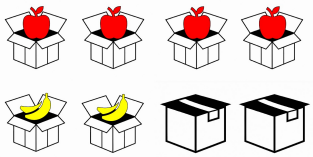
No.	Image	Text	Situation	Modifier
1		*There are four apples in the boxes.*	precise	bare
		There are at least four apples in the boxes.	precise	superlative
		There are more than three apples in the boxes.	precise	comparative
2		*There are four apples in the boxes.*	approximate	bare
		There are at least four apples in the boxes.	approximate	superlative
		There are more than three apples in the boxes.	approximate	comparative

Table 1: A sample set of experimental materials

showed that ignorance inferences occurred significantly more consistently when the QUD demanded precision ('how many' condition), suggesting that contextual expectations about informativeness directly affect how such inferences are drawn.

Likewise, Cremers et al. (2022) systematically manipulated various contextual factors to investigate the conditions under which ignorance implicatures arise. Their experiments involved multiple levels of visual information, QUD types, and textual scenarios. In particular, visual information was used to represent the informativeness of the situation—for example, a precise situation in which all eight cards were face-up, and an approximate situation in which two of the eight cards remained face-down, obscuring the exact quantity. Their findings revealed that ignorance inferences were more likely when the QUD required a precise answer ('howmany' condition) and when the visual context left room for uncertainty ('approximate' condition). These results highlight that ignorance implicatures are largely influenced by both linguistic and non-linguistic contextual cues.

These findings raise the question of whether and how visual and linguistic contextual information can enhance the pragmatic reasoning abilities of VLMs. To address this question, the present study examines whether VLMs exhibit sensitivity to ignorance implicatures across multiple contextual cues.

3 Methods

3.1 Data

As presented in Table 1, experimental materials were designed using two images for contextual precision (henceforth, 'situation') and texts including bare numeral, superlative, and comparative modifiers (henceforth, 'modifier').

In detail, images were used to manipulate the contextual precision, where a picture showing all 8 boxes open and providing the exact number of target objects was labeled as 'precise', and a picture with 2 out of 8 boxes remaining closed and an uncertain number of target objects was labeled as 'approximate'. In both types of situation, the target objects consistently appeared in 4 boxes. For example, in the image for precise situation, all 8 boxes are open and 4 of them contain apples. Since all boxes are open, we can tell that the target objects are exactly 4. On the other hand, in the image for approximate situation, 2 out of the 8 boxes remain closed and 4 of the open boxes contain apples. As what is inside the closed boxes is unknown, the target objects could be 4 or more. The corresponding texts were categorized based on the modifier types, including 'bare' (bare numeral *n*), 'superlative' (*at least n*), and 'comparative' (*more than n*).

Image data was created by combining open and closed boxes generated by GPT-4o (OpenAI, 2024) with standard icons for target objects. In this study, the number of target objects was consistently set to four, as previous work has shown that VLMs often exhibited limited performance on numerical reasoning tasks and experience a marked decline in accuracy when counting more than four items (Paiss et al., 2023). Additionally, since VLMs tend to struggle with counting when objects are presented in unstructured or cluttered spatial arrangements (Liu et al., 2019; Rahmanzadehgervi et al., 2024), the experimental images were carefully constructed with precisely aligned rows and columns. In this manner, each set of materials consisted of 2 images, each paired with 3 corresponding texts. In total, 70 sets of materials were used in the experiment.

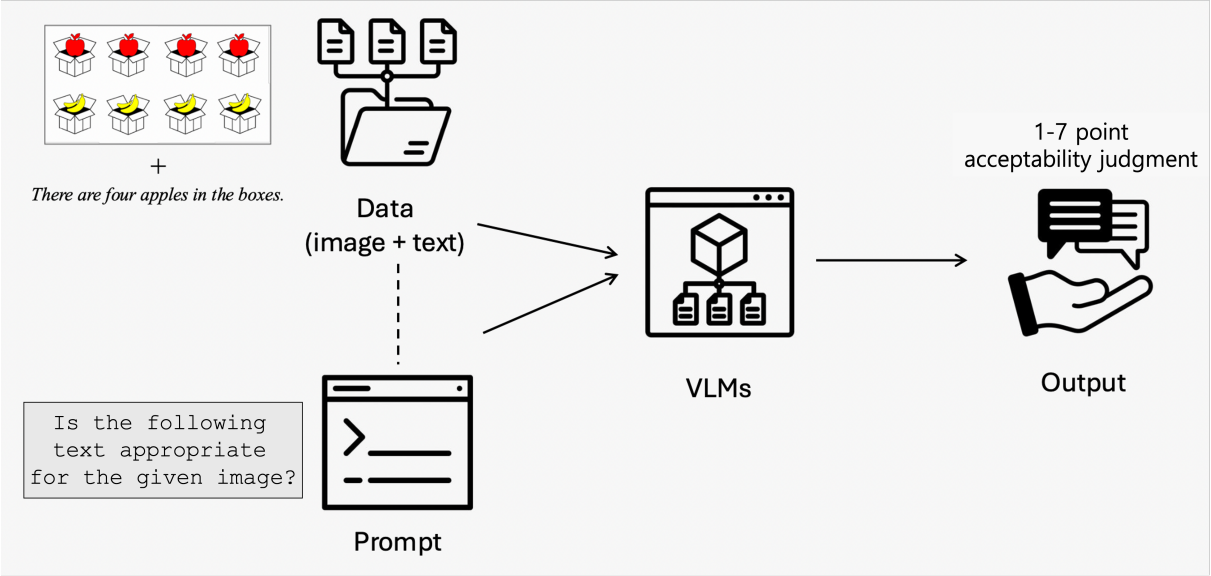


Figure 1: Overview of the experimental procedure

3.2 Models and Procedure

As VLMs for the experiment, we used GPT-4o (OpenAI, 2024), Gemini 1.5 Pro (Team et al., 2024), and Claude 3.5 sonnet (Anthropic, 2024). These models were selected due to their ability to process both image and text inputs simultaneously. They not only allow for the matching of images with text to determine their relationship but also provide the functionality to selectively query specific parts of the text within a broader context. This makes them well-suited for a series of our experiments.

For the experiment, these models were initialized using API keys. The images were then resized to a standard size of 224x224 pixels using the Pillow library (Clark et al., 2015) to ensure consistency in input dimensions and optimize processing efficiency. After resizing, the images were encoded into base64 format to ensure compatibility for input into the model's API.

Each experiment involved presenting the image alongside text prompts, which were specifically tailored for each task. All the materials, code and result of the experiment are publicly available.[1]

4 Experiment 1

4.1 Prompt

Cremers et al. (2022) argued that the disagreement over main findings related to ignorance inference was that the detection depends on the types of tasks participants were asked to perform. Specifically, it varied depending on whether participants were given an acceptability judgment task (Coppock and Brochhagen, 2013a; Westera and Brasoveanu, 2014; Cremers et al., 2022), where they judged the acceptability of the given sentences with respect to the depicted scenarios or images, or an inference task (Geurts et al., 2010), where they judged whether *exactly n* implies *at least n*. Cremers et al. (2022) argued that ignorance inference is more accurately assessed when evaluating the appropriateness of a sentence in relation to the context, rather than through the logical reasoning involved in inference tasks. In this regard, the acceptability judgment task serves as an effective method, guiding participants to evaluate whether a sentence is contextually appropriate. The acceptability judgment task typically involves either a true/false response format, as in truth-value judgment tasks (Coppock and Brochhagen, 2013a), or a numerical scale to capture the degree of ignorance implicatures in a more fine-grained manner (Westera and Brasoveanu, 2014; Cremers et al., 2022). In our experiment, we adopt a 1-7 scale to assess the appropriateness of image-text pairings in a more fine-grained manner.

As presented in Figure 1, each model was prompted to rate whether the texts with bare numerals, superlative, and comparative modifiers are appropriate for the given image on a scale from 1 to 7. The phrases used in the prompt were adapted from Experiment 3 in Cremers et al. (2022). In this manner, 70 sets of experimental items were re-

[1] https://github.com/joyennn/ignorance-implicature

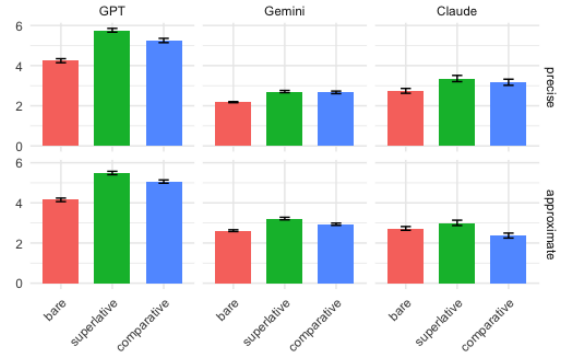


Figure 2: Result of Experiment1 — Mean scores for the appropriateness of image-text pairs based on modifiers, situations and models

peated 5 times to improve the reliability, resulting in a total of 2,100 individual responses from each of three models, as detailed below.

> Is the following text appropriate for the given image?
>
> Please reply with a single integer between 1 and 7, where 1 means "not at all appropriate" and 7 means "completely appropriate."
>
> **Text:** {text}

4.2 Result

Figure 2 shows the mean scores for the appropriateness of image-text pairs based on the types of modifiers, situations, and models. For the statistical analysis, we built mixed-effects logistic regression models (Baayen, 2008; Baayen et al., 2008; Jaeger, 2008; Jaeger et al., 2011) to analyze the results for each model, using the `lme4` package (Bates et al., 2015) in R software (Team, 2023). To examine the fixed effects, modifier and situation were set as independent variables, with appropriate scores as the dependent variable. Image and text were specified as random effects. For independent variables, the bare condition and the precise condition were set as the reference levels for modifier and situation, respectively.

As a result, the scores for appropriateness of image-text pairs followed the order of *superlative* > *comparative* > *bare* in both types of situations across almost all models, except for the approximate condition of Claude. While these predominant results aligned with findings from Cremers et al. (2022), where participants preferred the text with superlative and comparative in the approximate condition. However, the similar pattern in the precise situation was unexpected. In this situation, texts containing bare numerals should have been considered more appropriate than those with the other modifiers, as the number of target objects was explicitly defined.

Statistically, these results were influenced mostly modifier, which showed main effects in GPT ($p < 0.001$), Gemini ($p < 0.001$), and Claude ($p < 0.01$, 0.05). However, there were no significant effects on situation alone in GPT ($p = 0.66$) and Claude ($p = 0.96$), nor in the interaction of situation and modifier in GPT ($p = 0.06$, 0.33) and Gemini ($p = 0.11$, < 0.001).

In summary, superlative and comparative modifiers, which imply uncertainty, consistently led to higher appropriateness ratings even in both types of situations. This suggests that the models' responses were more influenced by the modifiers rather than the situations. The models' reliance on the semantic information inherent in the modifiers, rather than utilizing the contextual cues, indicates that the models are not effectively applying visually presented contextual information to ignorance inference.

5 Experiment 2

5.1 Prompt

Experiment 2 was conducted with the assumption that providing multiple pieces of contextual information would bring about pragmatic interpretation. Thus, another contextual cue, QUD, was added to the previous experimental setup. For QUDs, two types of conditions were designed, such as 'how-many' and 'polar' as below.

In the howmany condition, the question focuses on a specific number of target objects, leading responses containing superlative and comparative modifiers to introduce uncertainty or information gaps, which in turn trigger ignorance inferences. In contrast, the 'polar' condition elicits a simple yes/no response, placing minimal demands on numerical precision. Accordingly, it serves as a baseline for assessing the effects of the howmany QUD.

The appropriateness of the text in response to either of the two questions was measured on a 1-7 scale. For each condition, all experimental sets

were repeated 5 times, resulting in a total of 4,200 individual responses.

> Is the following answer to the question appropriate for the given image?
>
> Please reply with a single integer between 1 and 7, where 1 means "not at all appropriate" and 7 means "completely appropriate."
>
> *(QUD: howmany)*
> **Question:** How many {objects} did you find in the boxes?
> **Answer:** {text}
>
> *(QUD: polar)*
> **Question:** Did you find four {objects} in the boxes?
> **Answer:** {text}

5.2 Result

Figure 3 shows the mean scores for the appropriateness of image-text pairs, when the text was given as a response of howmany and polar questions, based on the types of modifiers, situations, and models. The statistical analysis was the same as in experiment 1, with QUD added as an independent variable in the fixed effects. For QUD, the polar condition was set as the reference level.

In the howmany condition, we observed that the score for bare numerals increased compared to the results from Experiment 1. In most cases observed in GPT and Gemini, bare numerals received the highest score, with the order being *bare* > *superlative* > *comparative*. For the precise condition of Gemini, the order was *superlative* > *bare* > *comparative*, but again, the score for bare numerals increased compared to the previous experiment.

Statistical analysis revealed that, for both GPT and Gemini, no significant effects were observed for the modifier (GPT: $p = 0.11, < 0.001$ | Gemini: $p < 0.001, = 0.46$). However, main effects were captured for the two contextual cues, situation (GPT: $p < 0.05$ | Gemini: $p < 0.001$) and QUD (GPT: $p < 0.001$ | Gemini: $p < 0.001$). Additionally, significant interactions were observed between each contextual cue and the modifier, specifically for the interactions of situation and modifier (GPT: $p < 0.05, 0.001$ | Gemini: $p < 0.001$), and QUD and modifier (GPT: $p < 0.001, 0.05$ | Gemini: $p < 0.001$). However, the interaction between the two contextual cues, situation and QUD (GPT: $p = 0.58$ | Gemini: $p = 0.92$), as well as the interaction of modifier, situation, and QUD (GPT: $p = 0.69, 0.92$ | Gemini: $p < 0.05, = 0.92$), were not significant. This finding suggests that while the influence of modifiers remains present, the increased availability of contextual information appears to guide the models toward a more context-driven interpretation strategy.

On the other hand, in case of Claude, the scores followed the order of *bare* > *superlative* > *comparative* in the precise situation, while the order was *superlative* > *bare* > *comparative* in the approximate situation. Although this does not perfectly align with human experimental results, it reflects a pattern similar to our expectations, where bare numerals would be preferred in the precise situation, and either superlative or comparative modifiers would be preferred in the approximate situation. Furthermore, statistical analysis showed a significant effect only in the interaction of both contextual cues, situation and QUD ($p < 0.01$). Considering that in Experiment 1, Claude did not show a similar pattern to human results based on visually encoded context, and that its results were strongly influenced by the modifiers, and the combination of modifier and situation, these findings suggest the possibility that when multiple contextual cues are provided, the model may combine them in a way that aligns more closely with human ignorance inferences, showing a tendency to rely more on contextual cues than on semantic modifiers.

In the polar as a control condition, the appropriateness score for bare numerals was higher compared to the result of Experiment 1, but still lower than in the howmany condition across all the models.

In summary, when the contextual cue QUD was added to the previous experimental setup, GPT and Gemini showed a tendency to prefer bare numerals, which refer to precise knowledge, compared to when only a single contextual cue was provided. In contrast, Claude demonstrated a more integrated approach by utilizing multiple contextual cues together, leading to an interpretation that was closer to pragmatic inference. Despite differences in how these models interpreted the stimuli, all models showed a common pattern of shifting reliance from modifiers to contextual cues when multiple cues were provided.

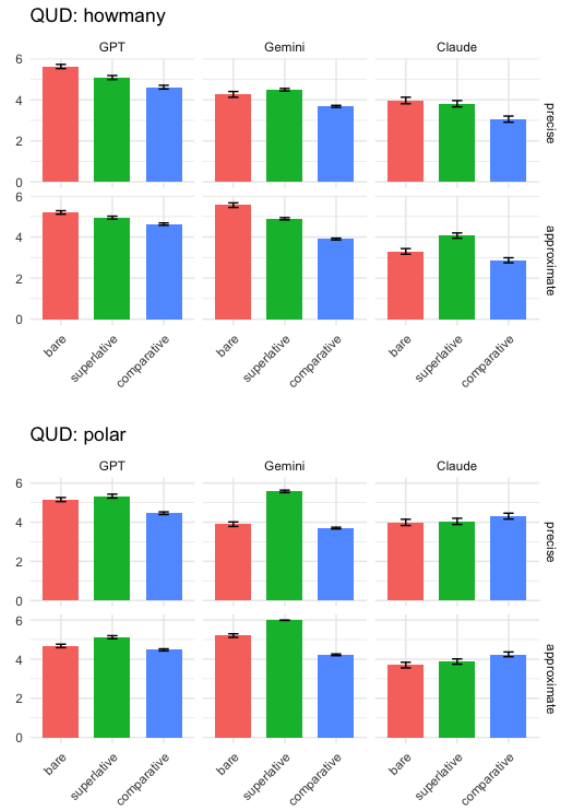


Figure 3: Result of Experiment2 — Mean scores for the appropriateness of image-text pairs based on modifiers, situations, and models across QUDs

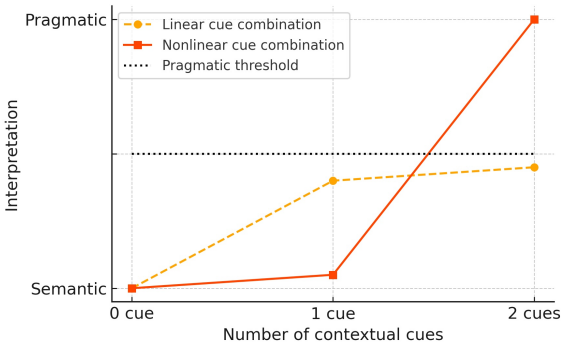


Figure 4: Modeling a threshold effect via linear and nonlinear cue combination as a function of contextual cue number (adapted from Parker, 2019)

6 Discussion

This study aimed to investigate the influence of contextual cues in the interpretation of ignorance inference within VLMs. In Experiment 1, we investigated how visually depicted situation (precise and approximate) and different types of modifiers (bare, superlative, and comparative) influenced appropriateness ratings of image-text pairs. Results revealed that appropriateness ratings consistently followed the order of *superlative* > *comparative* > *bare* across almost all models, regardless of situation types. This pattern suggests that the models primarily relied on the semantic features of modifiers rather than incorporating contextual information into their judgments.

Building upon these findings, Experiment 2 introduced an additional contextual cue, QUD, with two conditions (howmany and polar). This experiment aimed to determine whether multiple contextual cues would facilitate more sophisticated pragmatic inference. Interestingly, when presented with the howmany QUD, both GPT and Gemini models gave higher appropriateness ratings to bare numerals, which aligned with expectations for precise situation, but not for approximate situation. In our analysis, these models showed greater improvement in providing precise information than in engaging in pragmatic reasoning. While the influence of modifiers remained, there was a modest increase in sensitivity to contextual cues.

In contrast, Claude demonstrated a more integrated approach to contextual reasoning, using both situation and QUD simultaneously. This integration pattern suggests Claude may be moving closer to human-like pragmatic reasoning, which typically involves holistic consideration of multiple contextual factors. This leads to the assumption that Claude may have benefited from cue combination, where the presence of two contextual cues, rather than a single cue, led to pragmatic interpretation.

This pattern resonates with Parker (2019)'s cue combination scheme, which posits the processing benefit of retrieval cues for anaphora in memory is not merely additive but emerges nonlinearly when multiple cues are jointly available. Extending this idea to our findings, contextual cue combination for ignorance inference in VLMs may similarly follow the nonlinear cue combination method: one contextual cue alone may not significantly affect the context-sensitive reasoning, but the addition of the second cue increases the "cue weight" enough to reach the threshold for pragmatic interpretation. This threshold effect is visualized in Figure 4, which contrasts linear and nonlinear cue integration patterns as a function of contextual cue number.

Then, do GPT and Gemini follow the linear cue combination? It would be insufficient to characterize these models' behavior merely as examples of linear cue combination. Rather, the observed

pattern suggests a difference in how these models represent and utilize contextual information. While Claude appears to engage in combining two contextual cues into a unified pragmatic representation, GPT and Gemini exhibit a pattern of local alignment, in which modifiers are evaluated separately with each cue, but the cues themselves remain structurally unbound. In this sense, their responses are not limited because they combine cues linearly, but because their internal processing architecture does not support contextual cue combination in the first place. Consequently, their outputs reflect a tendency to prioritize informational precision over pragmatic reasoning. As more cues become available, the models tend to converge on more semantically determinate interpretations, aiming to reduce uncertainty in a localized manner rather than resolving it through holistic, context-sensitive inference.

Taken together, these findings offer new insights into how current VLMs differ in their capacity for contextual cue combination in pragmatic inference. By introducing multiple types of cues—both visual and linguistic—within a controlled experimental setting, this study provides empirical evidence that not all models process contextual information in the same way, and that the ability to integrate multiple cues holistically may serve as a crucial indicator of emerging pragmatic reasoning abilities in VLMs. In doing so, this research contributes to the growing body of work on multimodal language processing by highlighting the need to evaluate not only what models generate and understand, but also how they integrate diverse contextual cues to infer meaning.

7 Conclusion

This study examined whether and how current VLMs engage in pragmatic inference, particularly focusing on ignorance implicatures, when provided with visual and linguistic contextual cues. Through two experiments manipulating modifier types and contextual cues—including situation and QUDs—we found that not all VLMs process such information in the same way. Claude demonstrated the ability to integrate multiple contextual cues into a unified interpretation, exhibiting a threshold effect in pragmatic reasoning when both contextual cues were available. In contrast, GPT and Gemini tended to treat these cues independently, prioritizing precision over context-sensitive inference. This suggests a fundamental difference not only in cue weighting tendencies but also in how models internally represent and combine contextual information.

By systematically evaluating ignorance implicatures across VLMs, this study contributes to our understanding of the mechanisms underlying pragmatic behavior in VLMs. Importantly, it highlights that the capacity for contextual cue combination may serve as one of the key indicators of emerging pragmatic reasoning abilities in VLMs. These findings open new directions for evaluating and developing VLMs that move beyond literal interpretation toward more human-like pragmatic inference.

Acknowledgments

This research was supported by Brian Impact Foundation, a non-profit organization dedicated to the advancement of science and technology for all.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736.

Anthropic. 2024. *Claude 3.5 Sonnet*. Anthropic.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433.

R Harald Baayen. 2008. *Analyzing linguistic data: A practical introduction to statistics using R*. Cambridge university press.

R Harald Baayen, Douglas J Davidson, and Douglas M Bates. 2008. Mixed-effects modeling with crossed random effects for subjects and items. *Journal of memory and language*, 59(4):390–412.

Douglas Bates, Martin Maechler, Ben Bolker, Steven Walker, Rune Haubo Bojesen Christensen, Henrik Singmann, Bin Dai, Gabor Grothendieck, Peter Green, and M Ben Bolker. 2015. Package 'lme4'. *convergence*, 12(1):2.

Daniel Büring. 2008. The least at least can do. In *West Coast Conference on Formal Linguistics (WCCFL)*, volume 26, pages 114–120. Citeseer.

Francesca Capuano and Barbara Kaup. 2024. Pragmatic reasoning in gpt models: Replication of a subtle negation effect. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR.

Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, et al. 2022. Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Ye-eun Cho and Seong mook Kim. 2024. Pragmatic inference of scalar implicature by llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 10–20.

Alex Clark et al. 2015. Pillow (pil fork) documentation. *readthedocs*.

Herbert H Clark. 1996. *Using language*. Cambridge University Press.

Elizabeth Coppock and Thomas Brochhagen. 2013a. Diagnosing truth, interactive sincerity, and depictive sincerity. In *Semantics and Linguistic Theory*, pages 358–375.

Elizabeth Coppock and Thomas Brochhagen. 2013b. Raising and resolving issues with scalar modifiers. *Semantics and Pragmatics*, 6:3–1.

Alexandre Cremers, Liz Coppock, Jakub Dotlačil, and Floris Roelofsen. 2022. Ignorance implicatures of modified numerals. *Linguistics and Philosophy*, 45(3):683–740.

Chris Cummins. 2013. Modelling implicatures from modified numerals. *Lingua*, 132:103–114.

Chris Cummins and Napoleon Katsos. 2010. Comparative and superlative quantifiers: Pragmatic effects of comparison type. *Journal of Semantics*, 27(3):271–305.

Chris Cummins, Uli Sauerland, and Stephanie Solt. 2012. Granularity and scalar implicature in numerical expressions. *Linguistics and Philosophy*, 35:135–169.

Bart Geurts, Napoleon Katsos, Chris Cummins, Jonas Moons, and Leo Noordman. 2010. Scalar quantifiers: Logic, acquisition, and processing. *Language and cognitive processes*, 25(1):130–148.

Bart Geurts and Rick Nouwen. 2007. 'at least'et al.: the semantics of scalar modifiers. *Language*, pages 533–559.

H Paul Grice. 1975. *Logic and Conversation*. Brill.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738.

Jennifer Hu, Roger Levy, Judith Degen, and Sebastian Schuster. 2023. Expectations over unspoken alternatives predict pragmatic inferences. *Transactions of the Association for Computational Linguistics*, 11:885–901.

Jennifer Hu, Roger Levy, and Sebastian Schuster. 2022. Predicting scalar diversity with context-driven uncertainty over alternatives. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pages 68–74.

T Florian Jaeger. 2008. Categorical data analysis: Away from anovas (transformation or not) and towards logit mixed models. *Journal of memory and language*, 59(4):434–446.

T Florian Jaeger, Emily M Bender, and Jennifer E Arnold. 2011. Corpus-based research on language production: Information density and reducible subject relatives. *Language from a cognitive perspective: grammar, usage and processing. Studies in honor of Tom Wasow*, pages 161–198.

Adam Kendon. 2004. *Gesture: Visible action as utterance*. Cambridge University Press.

Julia Kruk, Jonah Lubin, Karan Sikka, Xiao Lin, Dan Jurafsky, and Ajay Divakaran. 2019. Integrating text and image: Determining multimodal document intent in instagram posts. *arXiv preprint arXiv:1904.09073*.

Stephen C Levinson. 2000. *Presumptive meanings: The theory of generalized conversational implicature*. MIT press.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Benjamin Lipkin, Lionel Wong, Gabriel Grand, and Joshua B Tenenbaum. 2023. Evaluating statistical language models as pragmatic reasoners. *arXiv preprint arXiv:2305.01020*.

Weizhe Liu, Mathieu Salzmann, and Pascal Fua. 2019. Context-aware crowd counting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5099–5108.

Jean-Claude Martin, Patrizia Paggio, P Kuenlein, Rainer Stiefelhagen, and Fabio Pianesi. 2007. Multimodal corpora for modelling human multimodal behaviour. *Special issue of the International Journal of Language Resources and Evaluation*, 41(3-4).

Clemens Mayr and Marie-Christine Meyer. 2014. More than at least. In *Slides presented at the Two days at least workshop, Utrecht*.

David McNeill. 2008. Gesture and thought. In *Gesture and thought*. University of Chicago press.

Rick Nouwen. 2010. Two kinds of modified numerals. *Semantics and Pragmatics*, 3:3–1.

OpenAI. 2024. *Hello GPT-4o*. OpenAI.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. 2023. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3170–3180.

Dan Parker. 2019. Cue combinatorics in memory retrieval for anaphora. *Cognitive science*, 43(3):e12715.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2024. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision*, pages 18–34.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Karan Sikka, Lucas Van Bramer, and Ajay Divakaran. 2019. Deep unified multimodal embeddings for understanding both content and users in social media networks. *arXiv preprint arXiv:1905.07075*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

R Core Team. 2023. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Polina Tsvilodub, Paul Marty, Sonia Ramotowska, Jacopo Romoli, and Michael Franke. 2024. Experimental pragmatics with machines: Testing llm predictions for the inferences of plain and embedded disjunctions. *arXiv preprint arXiv:2405.05776*.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Matthijs Westera and Adrian Brasoveanu. 2014. Ignorance in context: The interaction of modified numerals and quds. In *Semantics and Linguistic Theory*, pages 414–431.

Deirdre Wilson and Dan Sperber. 1995. *Relevance theory*. Oxford:Blackwell.

Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. 2022. Coca: Contrastive captioners are image-text foundation models. *arXiv preprint arXiv:2205.01917*.

Limitations

This study has several limitations that offer directions for future research. First, our experiments focused on a specific pragmatic phenomenon involving modified numerals, which allowed for a controlled testbed but may limit the generalizability of the findings. Extending the investigation to other types of pragmatic inferences would provide a broader understanding of VLMs' pragmatic reasoning abilities. Second, although we tested three state-of-the-art models—GPT-4o, Gemini 1.5 Pro, and Claude 3.5—the results may not fully generalize to other architectures, including open-source models with different training paradigms. Expanding the model pool would help assess the robustness of cue integration effects. Lastly, while our study builds on prior human experiments, it does not include a direct comparison with human performance under identical conditions. Such empirical comparisons would clarify whether model behavior reflects genuine pragmatic reasoning or merely statistical alignment with training data.

A Appendix

	Estimate	**Std**	***t***	***p*-value**
(Intercept)	4.24	0.17	24.31	**<0.001**
Situation	-0.09	0.21	-0.43	0.66
Modifier - Superlative	1.51	0.13	11.1	**<0.001**
Modifier - Comparative	1.01	0.13	7.37	**<0.001**
Situation:Modifier - Superlative	-0.17	0.09	-1.81	0.06
Situation:Modifier - Comparative	-0.09	0.09	-0.96	0.33

Table 2: Summary of fixed effects from mixed-effects logistic regression models by GPT-4o in Experiment 1

	Estimate	Std	*t*	*p*-value
(Intercept)	2.18	0.11	20.51	**<0.001**
Situation	0.43	0.13	3.24	**<0.01**
Modifier - Superlative	0.52	0.07	7.12	**<0.001**
Modifier - Comparative	0.47	0.07	6.32	**<0.001**
Situation:Modifier - Superlative	0.07	0.04	1.61	0.11
Situation:Modifier - Comparative	-0.17	0.04	-3.67	**<0.001**

Table 3: Summary of fixed effects from mixed-effects logistic regression models by Gemini 1.5 Pro in Experiment 1

	Estimate	Std	*t*	*p*-value
(Intercept)	2.74	0.26	10.21	**<0.001**
Situation	-0.01	0.33	-0.04	0.96
Modifier - Superlative	0.61	0.19	3.19	**<0.01**
Modifier - Comparative	0.42	0.19	2.22	**<0.05**
Situation:Modifier - Superlative	-0.33	0.09	-3.71	**<0.001**
Situation:Modifier - Comparative	-0.78	0.09	-8.64	**<0.001**

Table 4: Summary of fixed effects from mixed-effects logistic regression models by Claude 3.5 in Experiment 1

	Estimate	Std	*t*	*p*-value
(Intercept)	5.15	0.15	33.56	**<0.001**
Situation	-0.48	0.20	-2.39	**<0.05**
QUD	0.46	0.07	6.24	**<0.001**
Modifier - Superlative	0.17	0.11	1.59	0.11
Modifier - Comparative	-0.69	0.11	-6.26	**<0.001**
Situation:Modifier - Superlative	0.26	0.10	2.56	**<0.05**
Situation:Modifier - Comparative	0.49	0.10	4.71	**<0.001**
QUD:Modifier - Superlative	-0.71	0.10	-6.81	**<0.001**
QUD:Modifier - Comparative	0.30	0.10	-2.94	**<0.05**
Situation:QUD	0.05	0.10	0.54	0.58
Situation:QUD:Modifier - Superlative	0.01	0.14	-0.38	0.69
Situation:QUD:Modifier - Comparative	-0.05	0.14	0.09	0.92

Table 5: Summary of fixed effects from mixed-effects logistic regression models by GPT-4o in Experiment 2

	Estimate	Std	*t*	*p*-value
(Intercept)	3.83	1.45	26.27	**<0.001**
Situation	1.32	1.19	11.12	**<0.001**
QUD	0.35	6.22	5.72	**<0.001**
Modifier - Superlative	1.74	1.78	9.75	**<0.001**
Modifier - Comparative	0.13	1.78	-0.73	0.46
Situation:Modifier - Superlative	-0.92	8.66	-9.29	**<0.001**
Situation:Modifier - Comparative	-0.81	8.66	-10.61	**<0.001**
QUD:Modifier - Superlative	-1.45	8.69	-16.66	**<0.001**
QUD:Modifier - Comparative	-0.39	8.69	-4.44	**<0.001**
Situation:QUD	0.01	0.09	0.10	0.92
Situation:QUD:Modifier - Superlative	-0.30	0.12	-2.43	**<0.05**
Situation:QUD:Modifier - Comparative	-0.01	0.12	-0.09	0.92

Table 6: Summary of fixed effects from mixed-effects logistic regression models by Gemini 1.5 Pro in Experiment 2

	Estimate	Std	*t*	*p*-value
(Intercept)	3.99	0.29	13.55	**<0.001**
Situation	-0.29	0.40	-0.73	0.46
QUD	-0.02	0.08	-0.29	0.76
Modifier - Superlative	0.05	0.13	0.37	0.70
Modifier - Comparative	0.32	0.13	2.34	**<0.05**
Situation:Modifier - Superlative	0.12	0.12	1.03	0.29
Situation:Modifier - Comparative	0.22	0.12	1.84	0.06
QUD:Modifier - Superlative	-0.20	0.12	-1.68	0.09
QUD:Modifier - Comparative	-1.22	0.12	-9.92	**<0.001**
Situation:QUD	-0.37	0.12	-2.99	**<0.01**
Situation:QUD:Modifier - Superlative	0.80	0.17	1.41	0.15
Situation:QUD:Modifier - Comparative	0.24	0.17	4.56	**<0.001**

Table 7: Summary of fixed effects from mixed-effects logistic regression models by Claude 3.5 in Experiment 2

DeLTa: A Decoding Strategy based on Logit Trajectory Prediction Improves Factuality and Reasoning Ability

Yunzhen He[1], **Yusuke Takase**[1], **Yoichi Ishibashi**[2*], **Hidetoshi Shimodaira**[1,3]
[1]Kyoto University, [2]NEC Corporation, [3]RIKEN AIP
he.yunzhen.25d@st.kyoto-u.ac.jp y.takase@sys.i.kyoto-u.ac.jp
yoichi-ishibashi@nec.com shimo@i.kyoto-u.ac.jp

Abstract

Large Language Models (LLMs) are increasingly being used in real-world applications. However, concerns about the reliability of the content they generate persist, as it frequently deviates from factual correctness or exhibits deficiencies in logical reasoning. This paper proposes a novel decoding strategy aimed at enhancing both factual accuracy and inferential reasoning without requiring any modifications to the architecture or pre-trained parameters of LLMs. Our approach adjusts next-token probabilities by analyzing the trajectory of logits from lower to higher layers in Transformers and applying linear regression. We find that this **De**coding by **L**ogit **Tra**jectory-based approach (DeLTa) effectively reinforces factuality and reasoning while mitigating incorrect generation. Experiments on TruthfulQA demonstrate that DeLTa attains up to a 4.9% improvement over the baseline. Furthermore, it enhances performance by up to 8.1% on StrategyQA and 7.3% on GSM8K, both of which demand strong reasoning capabilities. [1]

1 Introduction

Natural language processing has advanced significantly with the rise of large language models (LLMs) (OpenAI, 2024; Dubey et al., 2024). However, ensuring the factual accuracy of LLM-generated text remains challenging. A notable issue is hallucination, where models produce factually incorrect content, posing risks in fields like information retrieval, medicine, and law (Huang et al., 2024). Calculation errors in the logical reasoning further contribute to inaccuracies, stemming from incorrect token predictions during decoding. Mitigation strategies for these issues include the selection of dataset, modifications to loss functions (Ouyang et al., 2022), and the incorporation of external knowledge (Wan et al., 2024). However, implementing these methods requires refining models or acquiring additional data, which can incur substantial costs.

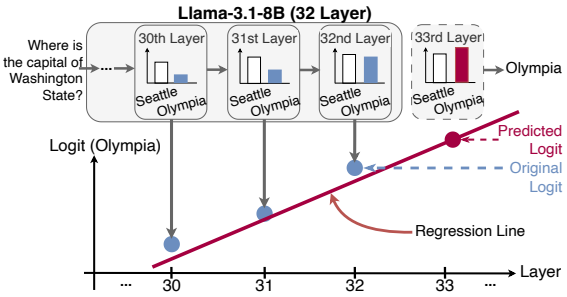


Figure 1: Overview of DeLTa. When input tokens are fed into the LLM, the logits from each layer (e.g., layers 30, 31, and 32) are computed and shown as bar graphs to illustrate changes between tokens (e.g., "Seattle" vs. "Olympia"). A linear regression (red line) approximates the logit trajectory (blue dots). Using this regression, we extrapolate the logits for a virtual 33rd layer (red dot) and improve prediction beyond the original outputs.

To overcome these limitations, we propose a decoding strategy, that boosts generation accuracy without extra training or data. Inspired by Chuang et al. (2024), who observed that correct token probabilities tend to rise in higher Transformer layers, we introduce **De**coding by **L**ogit **Tra**jectory-based approach (DeLTa), which treats each layer's logits as a time-series and use linear regression to predict upper-layer logits (Figure 1).

Experiments demonstrate that DeLTa enhances factuality by up to 4.9% on TruthfulQA, 5.0% on TriviaQA, and 2.4% on Natural Questions, while also improving reasoning on StrategyQA and GSM8K by up to 8.1%. These findings confirm that DeLTa refines token prediction, leading to improved factuality and reasoning capabilities.

*Work done while at Kyoto University.
[1]Code is available at https://github.com/githubhyz/DeLTa.

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 309–321
November 9, 2025 ©2025 Association for Computational Linguistics

2 Related Work

Previous research on guiding LLMs to generate factually accurate text can be broadly categorized into training-based (Lin et al., 2024; Liang et al., 2024) and non-training-based approaches. DeLTa falls into the latter category. Among non-training-based methods, Chang et al. (2024) introduced Asymptotic Probability Decoding, which extrapolates output probabilities from LLMs of different sizes under Contrastive Decoding (CD) (Li et al., 2023). Another method, Sharma et al. (2024) showed that certain capabilities of Transformers are concentrated in the latter layers and achieved improvements in sentiment classification by applying linear extrapolation to a text classifier based on CD. These approaches estimate probabilities using linear regression or extrapolation, relying on only two data points (e.g., the outputs of two models or an intermediate layer and the final layer). In contrast, DeLTa predicts logits instead of probabilities and performs regression across the entire sequence of logits from intermediate layers to the final layer, then recalculates the probability values. Another non-training-based approach, DoLa (Chuang et al., 2024), uses the difference in log probabilities between a lower and higher layer of the model within CD to encourage factually based text generation.

3 Method

DeLTa aims to improve the probability of generating the correct token during decoding by focusing on changes in the logits across the Transformer's layers. Specifically, when decoding the next token in an N-layer Transformer, we regard the logits produced by each layer as a time series (§ 3.1). Based on the observation that higher layers generally assign higher probabilities to the correct token (Chuang et al., 2024), we employ a simple linear regression model (§ 3.2) to predict the logits of the higher layers. By leveraging the upward trend from lower to higher layers, this approach enhances the final prediction performance.

3.1 Token Probability at Arbitrary Layers

Let x_t denote the token at position t, and let the preceding token sequence be $x_{<t} = \{x_1, \ldots, x_{t-1}\}$. The application of the Logit Lens (nostalgebraist, 2020), enables the computation of the hypothetical probability distribution at any arbitrary layer denoted by the set of all possible values of the layer, i.e., ℓ $(1 \leq \ell \leq N)$.

$$P_\ell(x_t \mid x_{<t}) = \mathrm{softmax}\left(\mathbf{logit}^{(\ell)}\right)_{x_t} \quad x_t \in \mathcal{X}.$$

Here, softmax represents the softmax function, and $\mathcal{X}$ denotes the vocabulary set.

3.2 Decoding by Logit Trajectory

We employ a linear regression to model changes in logits across Transformer layers, thereby enabling the estimation of logits at virtual layers. Because the probability of the correct token generally increases in higher layers, we explicitly learn this upward trend to produce more reliable token probabilities. Concretely, we select an intermediate layer N_{mid} $(1 \leq N_{mid} \leq N-1)$ and use its logit vectors up to the final layer N to estimate logits. We then compute token probability $P_L(x_t \mid x_{<t})$ for a virtual layer L $(L \in \mathbb{R})$

Linear Regression. We define the explanatory variable $\boldsymbol{X}_{\text{reg}}$ as the Transformer layer indices and the response variable $\boldsymbol{Y}_{\text{reg}}$ as the logit vectors:

$$\boldsymbol{X}_{\text{reg}} = [N_{mid}, N_{mid}+1, \cdots, N],$$
$$\boldsymbol{Y}_{\text{reg}} = [\mathbf{logit}^{(N_{mid})}, \mathbf{logit}^{(N_{mid}+1)}, \cdots, \mathbf{logit}^{(N)}]$$

Based on the least squares method, the estimated logit at a virtual layer L is computed as follows:

$$\widehat{\mathbf{logit}}^{(L)} = \hat{\boldsymbol{\beta}}_0 + \hat{\boldsymbol{\beta}}_1 L,$$

where $\hat{\boldsymbol{\beta}}_0$ is the intercept and $\hat{\boldsymbol{\beta}}_1$ is the regression coefficient. These parameters are determined by:

$$\hat{\boldsymbol{\beta}}_0 = E(\boldsymbol{Y}_{\text{reg}}) - \hat{\boldsymbol{\beta}}_1 E(\boldsymbol{X}_{\text{reg}}), \hat{\boldsymbol{\beta}}_1 = \frac{C(\boldsymbol{X}_{\text{reg}}, \boldsymbol{Y}_{\text{reg}})}{V(\boldsymbol{X}_{\text{reg}})}$$

Here, E, V, and C represent the mean, variance, and covariance, respectively.

Token Probability Computation. The final token probability is computed from logit $\widehat{\mathbf{logit}}^{(L)}$, filtered by the candidate token set $\mathcal{V}_{\text{head}}$:

$$\hat{P}_L(x_t \mid x_{<t}) = \mathrm{softmax}(\widehat{\mathbf{logit}}^{(L)}_{\mathcal{V}_{\text{head}}})_{x_t}.$$

Here, the candidate set $\mathcal{V}_{\text{head}}$ is determined following Chuang et al. (2024) as:

$$\mathcal{V}_{\text{head}} = \{x_t \in \mathcal{X} : \hat{P}(x_t \mid x_{<t}) \geq \alpha \max_w P_N(w \mid x_{<t})\}. \quad (1)$$

Tokens that are not included in the candidate set are assigned a probability of 0.

Model	Factuality					CoT Reasoning	
	TruthfulQA (Open QA)			Closed QA			
	%True↑	%Info↑	%True*Info↑	**TriviaQA**	**NQ**	**StrQA**	**GSM8K**
Qwen2.5-7B	68.9	92.4	64.1	39.1	11.5	76.9	78.7
+ filter	67.4	93.3	60.7	**44.1**	**13.0**	78.1	**81.6**
+ DoLa (early-layer)	71.2	91.6	62.9	41.9	12.8	73.8	76.3
+ DoLa (late-layer)	79.6	75.0	55.0	33.6	10.2	67.7	67.9
+ DeLTa	66.8	98.0	**65.4**	**44.1**	**13.0**	**81.2**	**81.6**
Mistral-7B-v0.1	56.3	95.3	**53.8**	51.3	16.0	65.3	31.0
+ filter	59.4	81.1	40.9	**54.5**	**18.2**	69.7	35.8
+ DoLa (early-layer)	50.5	91.7	42.9	53.2	17.2	69.3	33.4
+ DoLa (late-layer)	51.2	91.2	42.9	53.3	17.0	71.3	33.7
+ DeLTa	54.3	92.1	47.0	54.1	17.9	**72.5**	**38.2**
Llama-3.1-8B	50.8	90.1	44.0	50.0	14.0	64.0	42.8
+ filter	50.7	95.2	46.9	**53.8**	**16.4**	66.0	47.8
+ DoLa (early-layer)	48.9	99.0	48.2	53.2	15.6	66.4	46.1
+ DoLa (late-layer)	49.2	99.3	48.5	53.1	15.3	64.9	45.9
+ DeLTa	51.5	97.1	**48.9**	**53.8**	**16.4**	**72.1**	**50.1**

Table 1: Experimental results on (1) factuality tasks, including TruthfulQA, TriviaQA, and Natural Questions (NQ) and (2) reasoning tasks involving Chain-of-Thought (CoT), including StrategyQA (StrQA) and GSM8K. Bold values represent the highest scores. DeLTa achieves a strong performance on the %True*Info metric for TruthfulQA and shows substantial improvements across multiple benchmarks, including TriviaQA and GSM8K. Importantly, in GSM8K, which requires not only factual knowledge but also arithmetic reasoning, DeLTa outperforms the baseline by more than 7 points. These results indicate that DeLTa enhances both knowledge-intensive tasks and complex reasoning capabilities.

4 Experiments

4.1 Setup

Models and Baselines. We use Qwen2.5-7B (Qwen Team, 2024), Mistral-7B-v0.1 (Jiang et al., 2023), and Llama-3.1-8B (Dubey et al., 2024), comparing them with four baselines. The first baseline is the raw model output. The second baseline (filter) applies $\mathcal{V}_{\text{head}}$ (Equation (1)) to the raw model output. This baseline is specifically introduced to determine whether the performance improvement of our method primarily results from the filtering mechanism rather than from DeLTa. The third and fourth baselines, DoLa (early-layer) and DoLa (late-layer), are derived from DoLa (Chuang et al., 2024), a state-of-the-art decoding method that significantly enhances generation quality by leveraging the difference in log probabilities between an intermediate layer and the final layer. DoLa dynamically selects the intermediate layer from predefined layer buckets, which are primarily partitioned into two groups: early layers (lower half of the model) and late layers (upper half of the model). We denote these two configurations as DoLa (early-layer) and DoLa (late-layer), respectively. Originally, DoLa determines the optimal bucket using a validation set. However, by comparing DeLTa with both DoLa (early-layer) and DoLa (late-layer), we assess whether DeLTa remains effective regardless of the specific intermediate layer bucket selection. This evaluation highlights the robustness and general applicability of DeLTa beyond DoLa's predefined selection strategy. We exclude methods such as Chang et al. (2024) and Sharma et al. (2024) as baselines. The former requires fine-tuning for optimal results despite being a non-training method, making fair comparison with our training-free approach difficult. The latter is designed for classification, not generation, and is thus unsuitable for our evaluation.

Tasks and Datasets. Following DoLa (Chuang et al., 2024), we evaluate open-ended generation tasks: TruthfulQA (Lin et al., 2022) (factual accuracy in open QA), StrategyQA (StrQA)(Geva et al., 2021), and GSM8K(Cobbe et al., 2021) (reasoning). To assess token-level accuracy across diverse tasks, we evaluate knowledge retrieval via closed QA tasks: TriviaQA (Joshi et al., 2017) and Natural Questions (NQ) (Kwiatkowski et al., 2019).

Evaluation metrics are in Appendix A, and Appendix B details the prompt structure and generation hyperparameters. These settings follow DoLa for fair comparation. Additionally, Appendix D explains the selection of N_{mid} and virtual layer L for DeLTa.

4.2 Results

Factuality. Table 1 summarizes the model performance across factuality benchmarks, demonstrating the effectiveness of DeLTa compared to strong baselines, including filtering and DoLa variants. On TruthfulQA, we focus on the %True*%Info metric, which better reflects factual and informative responses than %True or %Info alone. This metric avoids rewarding trivial but technically correct answers. Under this measure, DeLTa improves Llama-3.1-8B from 44.0% to 48.9%, surpassing the best baseline (46.9%) by 2 points. In contrast, existing methods like filtering and DoLa (early-layer) show limited and inconsistent gains. For Closed QA tasks such as TriviaQA and NQ, generated answers typically consist of just few tokens, limiting opportunities for adjustment based on logit trajectories. Additionally, during validation, DeLTa often selects the middle layer $N_{\text{mid}} = N-1$, causing logits from DeLTa and +Filter to coincide, thus leading to identical scores (e.g. Qwen2.5-7B achieves 44.1% on TriviaQA and 13.0% on NQ). Even in this constrained scenario, DeLTa consistently maintains or slightly exceeds baseline accuracy.

CoT Reasoning. DeLTa also substantially improves CoT reasoning accuracy, achieving up to a 7.3-point gain on GSM8K (e.g., Llama-3.1-8B: 42.8% to 50.1%), with similar improvements observed across other models. DoLa (early-layer) sometimes introduces minor improvements, while DoLa (late-layer) frequently fails to generalize, particularly on GSM8K. These results suggest that DeLTa enhances the accuracy of generated text, thereby leading to significant improvements in reasoning.

5 Analysis

In this section, we conduct a series of analyses to empirically validate the core hypotheses underpinning DeLTa. We first verify that deeper layers contribute more to task performance, then investigate the linearity of logit evolution across layers, and finally justify our choice of a linear regression model through an ablation study.

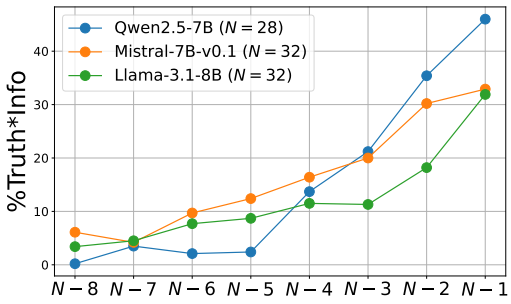


Figure 2: Direct decoding performance from intermediate layers on TruthfulQA (%Truth*Info↑). Performance consistently improves in deeper layers. The x-axis represents the layer depth from $N-8$ (left) to $N-1$ (right).

5.1 Information Salience in Deeper Layers

DeLTa builds upon the hypothesis, inspired by Chuang et al. (2024), that task-relevant information for generating the correct token becomes more salient in the upper (deeper) layers of a Transformer model. To empirically validate this hypothesis in our setting, we performed direct decoding from the hidden states of the final eight intermediate layers (from layer $N-8$ to $N-1$) of three language models: Qwen2.5-7B, Mistral-7B-v0.1, and Llama-3.1-8B. We then evaluated performance on the TruthfulQA dataset using the %Truth*Info score.

Figure 2 shows a clear trend across all models where performance improves as layers get deeper. This provides a strong empirical support for our foundational hypothesis.

5.2 Linearity of Logit Evolution Across Layers

Given the increasing importance of upper layers, we now investigate the nature of their internal dynamics. Specifically, we evaluate the extent to which the logits retain a linear structure across layers using the coefficient of determination (R^2).

Experimental Procedure. First, a text is input into the LLM, and the top 50 tokens with the highest logits in the final layer are extracted. Next, following the procedure described in §3.2, the predicted and original logits of these tokens from layer N_{mid} to N are used to compute the R^2 for each token. This value is then averaged across all tokens and multiple input sentences from datasets in §4.1.

Results on Mean Linearity. As shown in Figure 3, all three LLMs exhibit a substantial increase in mean R^2 at higher layers, with Llama-3.1-8B

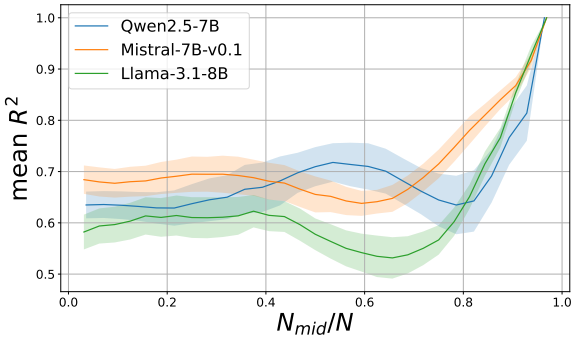


Figure 3: Mean coefficient of determination (mean R^2) and its standard deviation across input samples. The vertical axis represents the mean R^2, and the horizontal axis represents the ratio of layer indices (N_{mid}/N).

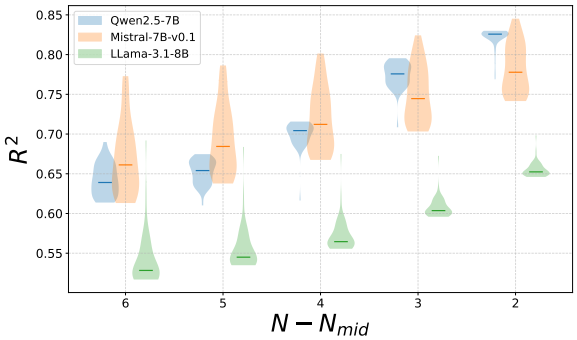


Figure 4: Violin plots of R^2 for Top-K tokens, grouped by model and layer difference ($N - N_{mid}$).

reaching approximately 0.9 near the final layer. These results indicate that a simple linear model can effectively capture logit relationships in higher layers. This finding is consistent with recent studies (Razzhigaev et al., 2024), which also demonstrate approximately linear behavior in later Transformer layers.

Distributional Analysis. To provide a more granular view, we also visualize the full distribution of R^2 for the Top-50 tokens using violin plots. As depicted in Figure 4, the R^2 distributions for all models shift toward higher values and become narrower as N_{mid} approaches the final layer. This trend reveals not only an increase in the average linearity but also a reduction in its variance, indicating more stable linear behavior. Mistral-7B-v0.1 consistently shows the highest median and the tightest distribution, while Llama-3.1-8B exhibits broader distributions, suggesting greater variability. These results reveal the stability and model-dependence of logit linearity across layers.

Model	%Truth↑	%Info↑	%Truth*Info↑
Qwen2.5-7B (+ DeLTa)	66.8	98.0	**65.4**
Qwen2.5-7B (+ DeLTa2)	64.7	94.3	45.2
Mistral-7B-v0.1 (+ DeLTa)	54.3	92.1	**47.0**
Mistral-7B-v0.1 (+ DeLTa2)	43.5	90.3	35.7
Llama-3.1-8B (+ DeLTa)	51.5	97.1	**48.9**
Llama-3.1-8B (+ DeLTa2)	39.4	98.4	38.1

Table 2: Performance comparison on TruthfulQA between linear regression (DeLTa) and quadratic regression (DeLTa2).

5.3 Ablation on Regression Model Choice

The observed linearity in upper layers motivates our choice of a linear regression model. To justify this design decision, we conducted an ablation study comparing DeLTa with a version using a more complex quadratic regression model, which we call DeLTa2.

The results in Table 2 show that the linear regression model (DeLTa) significantly outperforms the quadratic version (DeLTa2) across all models. This suggests that unnecessarily increasing the model's expressiveness harms generalization performance. We conclude that a simple and robust linear regression, which aligns with the observed linear dynamics of the upper layers, is a more effective and efficient approach.

6 Conclusion

This study aimed to enhance the factual accuracy and reasoning of text generated by LLMs. The proposed method, DeLTa, operates without additional training or data. By leveraging token probability distributions across Transformer layers and employing linear regression, we developed a framework that is both computationally efficient and easily integrable. Empirical evaluations across multiple benchmarks demonstrate that DeLTa significantly improves factual accuracy and exhibits effectiveness in reasoning tasks.

7 Limitation

The proposed method (DeLTa) in this study has limitation, as outlined below:

- Due to computational resource constraints, we could not conduct experiments on large-scale language models. Whether our approach maintains its effectiveness in larger models needs to be investigated in future studies.

Future research should focus on overcoming this limitation to establish a more generalizable and highly accurate factuality correction method applicable to a broader range of language models.

References

Haw-Shiuan Chang, Nanyun Peng, Mohit Bansal, Anil Ramakrishna, and Tagyoung Chung. 2024. Explaining and improving contrastive decoding by extrapolating the probabilities of a huge and hypothetical LM. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8503–8526, Miami, Florida, USA. Association for Computational Linguistics.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *The Twelfth International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.* Just Accepted.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12286–12312, Toronto, Canada. Association for Computational Linguistics.

Yuxin Liang, Zhuoyang Song, Hao Wang, and Jiaxing Zhang. 2024. Learning to trust your feelings: Leveraging self-awareness in llms for hallucination mitigation. *arXiv preprint arXiv:2401.15449*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, yelong shen, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, and Weizhu Chen. 2024. Not all tokens are what you need for pretraining. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yusuke Nakamura and Daisuke Kawahara. 2024. Construction of japanese truthfulqa. In *Proceedings of the 30th Annual Meeting of the Association for Natural Language Processing*, pages 1709–1714. In Japanese.

nostalgebraist. 2020. Interpreting gpt: the logit lens.

OpenAI. 2024. Hello gpt-4o.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning

library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. 2024. Your transformer is secretly linear. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5376–5384, Bangkok, Thailand. Association for Computational Linguistics.

Mayukh Sharma, Sean O'Brien, and Julian McAuley. 2024. Linear layer extrapolation for fine-grained emotion classification. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20880–20888, Miami, Florida, USA. Association for Computational Linguistics.

Fanqi Wan, Xinting Huang, Leyang Cui, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge verification to nip hallucination in the bud. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 2616–2633. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

A Evaluation Metrics

The evaluation metrics used for the datasets in our experiments are as follows:

- **TruthfulQA**: Following Lin et al. (2022), We use a fine-tuned GPT-4 model to compute the scores of truthfulness (%Truth) and informativeness (%Info) for responses generated by LLMs. %Truth measures the degree to which a response is factually accurate, while %Info quantifies the amount of useful information contained in the response. Additionally, %Truth*Info is the product of %Truth and %Info, representing the degree to which a response maintains truthfulness while being informative. Higher values indicate better performance.

Input Length	DoLa	DeLTa
128 tokens	0.0205	0.0013
256 token	0.0358	0.0014
512 token	0.0453	0.0015

Table 3: Comparison of inference time (seconds per sample) between DoLa and DeLTa across varying input lengths.

- **TriviaQA and NQ**: The accuracy is calculated based on the exact match between the responses generated by the LLM and the gold answers.

- **StrQA and GSM8K**: The accuracy is computed based on the exact match between the extracted final answer from the LLM-generated response and the gold answer.

B Generation Hyper-parameters

The hyper-parameters employed for generation were standardized across all experiments, with the temperature parameter fixed at 0.9 and the top-k and top-p sampling parameters set to 50 and 0.95, respectively. The repetition penalty was set to 1.0 for the raw model output and to 1.2 for other methods. Furthermore, the maximum token length was set to 50 for the TruthfulQA, TriviaQA, and Natural Questions datasets, and 256 for the StrategyQA and GSM8K datasets. About α in Equation (1), we set $\alpha = 0.1$. The above parameters are all derived from DoLa (Chuang et al., 2024).

For each task, the LLM was provided with prompts and questions, and the generated responses were evaluated. The prompt structure and their selection followed Chuang et al. (2024). Specifically, we adopted the same few-shot examples as in Chuang et al. (2024) to ensure a fair comparison. However, due to computational resource constraints, we set the number of few-shot examples to 6.

C Computational Cost

To evaluate the computational cost of the algorithm itself, we measured the inference time on a synthetic model with 32 layers, hidden size of 4096, and vocabulary size of 32,000. For each input length—128, 256, and 512 tokens—we used 100 randomly generated samples and reported the average inference time in Table 3. The reason DeLTa demonstrates superior algorithmic efficiency lies

in the fundamental differences between the underlying algorithms. DoLa requires sequential execution of multiple intermediate layers to dynamically determine the optimal layer using divergence-based criteria, including KL divergence and Jensen-Shannon divergence. This introduces significant computational overhead due to repeated forward passes and divergence evaluations.

In contrast, DeLTa employs a direct linear regression approach over precomputed hidden states, followed by normalization steps. Since it eliminates the need for iterative divergence computations and dynamic routing, DeLTa drastically reduces the overall computational cost.

D Configuration of DeLTa

DeLTa includes adjustable hyperparameters, N_{mid} and L. To select the optimal intermediate layer N_{mid} and the target virtual layer for estimation L for each model, we constructed validation and test datasets from each dataset. First, we determined the values of N_{mid} and L that maximize the accuracy of DeLTa for each model using the validation dataset. Then, using the selected N_{mid} and L, we conducted a comparison with the baseline on the test dataset.

For datasets without a validation set (TruthfulQA, StrQA), 10% of the test data was used as the validation data. On the other hand, for datasets with an existing validation set (TriviaQA, NQ, GSM8K), we extracted 10% of the existing validation dataset for use in our experiments.

In the experiments, we selected N_{mid} from $\{N-6, N-5, \ldots, N-1\}$ and L from $\{N, N+0.5\}$. The selected values of N_{mid} and L are presented in Table 4.

After experiments, as shown in Table 4, the values of N_{mid} and L selected based on validation exhibit different tendencies depending on the dataset. Notably, in TruthfulQA, selecting an outer layer contributed to performance improvement, whereas in other datasets, optimization through smoothing was found to be the most effective.

In conclusion, the range of selected N_{mid} and L values remains largely consistent across models, with no extreme differences observed between models. This suggests that DeLTa can be generally applied without dependence on specific datasets or models.

E Additional Experiments on Logit Linearity

E.1 Logit Linearity with Different Datasets

In this section, we examine the logit linearity across the intermediate layers of models for each dataset (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), following the experimental procedure described in §5.2. The results are presented in Figure 5. The horizontal axis, N_{mid}/N, represents the starting point of the explanatory variables, while the vertical axis, mean R^2, denotes the mean coefficient of determination.

When comparing the results across datasets, a general trend is observed: in the higher layers of the model (the last 4-5 layers), the meanR^2 values increase to around 0.8, indicating relatively high logit linearity. Notably, Mistral-v0.1-7B and Qwen2.5-7B consistently exhibit higher mean R^2 values than Llama-3.1-8B, suggesting that logit linearity is more pronounced in these models.

Conversely, in the lower layers, the mean R^2 values are relatively low, with significant variability across datasets and models. In particular, Llama-3.1-8B tends to have determination coefficients below 0.6 in the lower layers, suggesting lower linearity compared to other models.

Additionally, in the middle layers around $N_{mid} = 20$, a decline in mean R^2 is observed in some models. This phenomenon suggests that logits undergo nonlinear transformations in the intermediate layers. However, as the model approaches the final layers, mean R^2 increases again, indicating that logit representations become more linear.

Overall, consistent with the experimental results described in §5.2, these findings suggest that while logit linearity across layers is dataset-dependent, it generally stabilizes and improves as the model approaches the final layers.

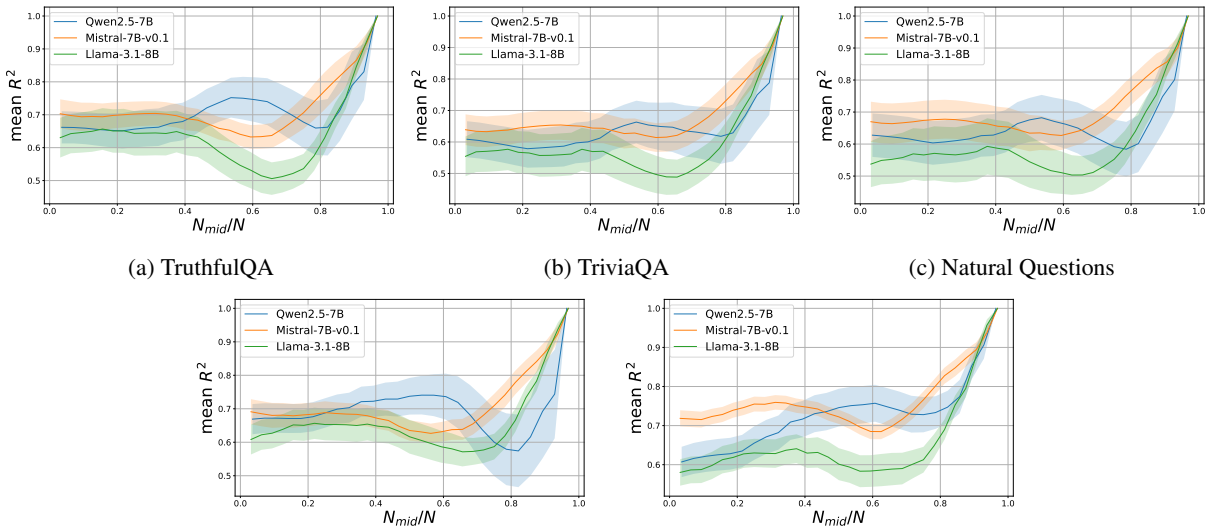
E.2 Distributional Analysis of Logit Linearity with Different Datasets

Following §5.2, we analyze the distribution of logit linearity for the Top-K $(= 50)$ tokens across intermediate layers, separately for each dataset (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), following the experimental procedure described in § 5.2. The results are presented in Figure 6.

Comparing the distributions across datasets in Figure 6, several consistent trends and dataset-specific differences can be observed. In the upper

Dataset	Qwen2.5-7B	Mistral-7B-v0.1	Llama-3.1-8B
TruthfulQA	26, 28.5	31, 32.5	30, 32
TriviaQA	27, 28	28, 32	31, 32
Natural Questions	27, 28	27, 32	31, 32
Strategy QA	25, 28	29, 32	28, 32
GSM8K	27, 28	26, 32	28, 32

Table 4: Results of the selected M and L. The left and right numbers in each cell represent M and L, respectively.



(a) TruthfulQA (b) TriviaQA (c) Natural Questions

(d) StrategyQA (e) GSM8K

Figure 5: Logit linearity of different models (Qwen2.5-7B, Mistral-v0.1-7B, Llama-3.1-8B) on various datasets (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), as calculated in § 5.2. The horizontal axis represents the layer ratio, while the vertical axis shows the mean R^2, which denotes the average coefficient of determination.

layers (small $N - N_{mid}$), the R^2 distributions for the Top-K tokens shift towards higher values and become significantly narrower, indicating that the linearity between predicted and original logits becomes both stronger and more stable as the model approaches the final layers.

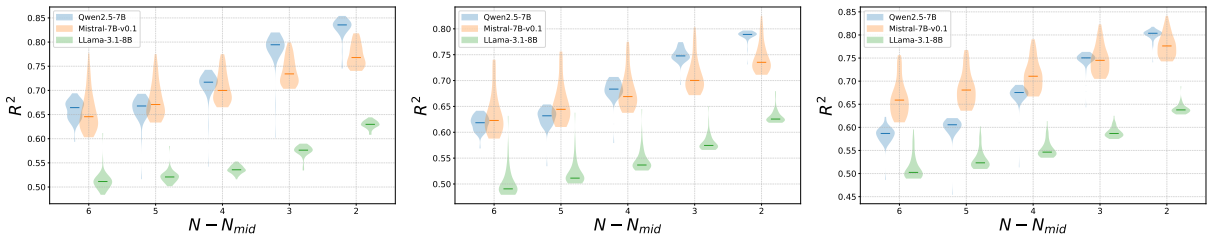
In contrast, Llama-3.1-8B shows markedly lower median R^2 values and broader distributions across all datasets and layers, suggesting that its logit linearity is both weaker and less stable, especially for high-probability tokens. This model-dependent difference is especially notable in more challenging datasets such as Natural Questions and GSM8K, where the separation between models becomes even more apparent in the upper layers.

Across all models and datasets, the lower layers (large $N - N_{mid}$) display lower R^2 values and broader distributions, indicating that the predictive power of the linear model is limited in the earlier stages of computation. In several datasets, such as TriviaQA and StrategyQA, a gradual and monotonic improvement in R^2 is observed as the model moves toward the output layer, while for others, such as TruthfulQA, some non-monotonicity and broadening of distributions in the intermediate layers can be seen, reflecting possible nonlinear transformations at these stages.

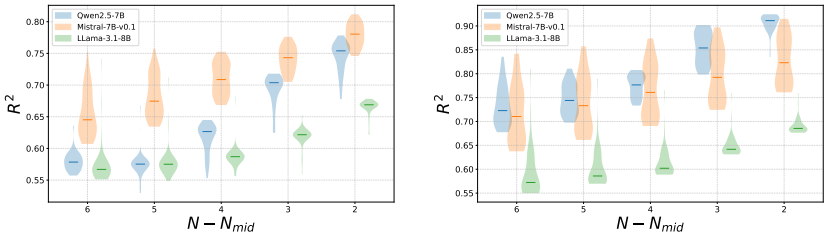
Overall, these results demonstrate that the distributional characteristics of logit linearity for Top-K tokens are jointly influenced by both model architecture and dataset properties. Nevertheless, the general tendency across all settings is that logit linearity is strengthened and stabilized in the upper layers, consistent with findings in § 5.2 and previous sections.

E.3 Qualitative Evaluation of DeLTa

In this section, we qualitatively evaluate DeLTa using yes/no question datasets. Specifically, we input particular questions into Qwen2.5-7B and constrain their outputs to either **yes** or **no**, allowing



(a) TruthfulQA (b) TriviaQA (c) Natural Questions



(d) StrategyQA (e) GSM8K

Figure 6: Distribution of logit linearity (R^2) for different models (Qwen2.5-7B, Mistral-v0.1-7B, Llama-3.1-8B) across various datasets (TruthfulQA, TriviaQA, Natural Questions, StrategyQA, GSM8K), as calculated in §E.2. The horizontal axis denotes the difference between the final layer and the intermediate layer ($N - N_{mid}$), while the vertical axis shows the coefficient of determination (R^2) for Top-50 tokens. Each violin plot visualizes the distribution of R^2 values across input samples, allowing for a comparison of both the central tendency and variability of logit linearity among models, layers, and datasets.

for a detailed analysis of changes in logit scores.

First, in E.3.1, we investigate the impact of different settings of M and L on logit scores, and clarify under what conditions incorrect answers are corrected. Next, in E.3.1, we compare the logit scores with and without the application of DeLTa, verifying the effectiveness of the correction. In particular, we focus on cases where the correct answer is **no** but the model originally outputs the incorrect answer **yes**, and analyze how the logit score of **no** changes after applying DeLTa.

Through this analysis, we qualitatively evaluate how the appropriate selection of M and L influences the correction of incorrect answers, thereby demonstrating the effectiveness of DeLTa.

E.3.1 Logit Changes for Different M and L

As an example using Qwen2.5-7B, we consider the input question: `What is the population of the country?\nA: The population is about 320 million.\nTrue:`. In this case, the correct answer is **no**; however, the model's original output is **yes**, resulting in an incorrect answer. The results of applying DeLTa under different values of M and L are shown in Figure 7.

Figure 7a and Figure 7b display the changes in the logits for "yes" and "no" over $M \in \{0, \ldots, 27\}$ for $L = 28$ and $L = 29$, respectively. In the case of $L = 28$ (Figure 7a), the logit for the incorrect answer **yes** remains higher than that for the correct answer **no** across different M.

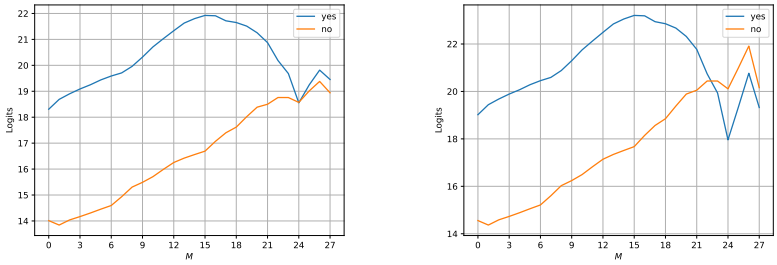
On the other hand, for $L = 29$ (Figure 7b), the logit for **no** begins to exceed that of **yes** around $M = 23$, indicating that DeLTa has successfully corrected the model's error. These results demonstrate that selecting an appropriate L is crucial for effective correction of incorrect answers.

E.3.2 Detailed Analysis of the Effect of Regression-Based Correction

The effect of regression-based correction using Qwen2.5-7B is shown in Figure 8. Figure 8a ($M = 24$) and Figure 8b ($M = 27$) indicate the original logit scores with dashed lines and the corrected logit scores after applying DeLTa with solid lines.

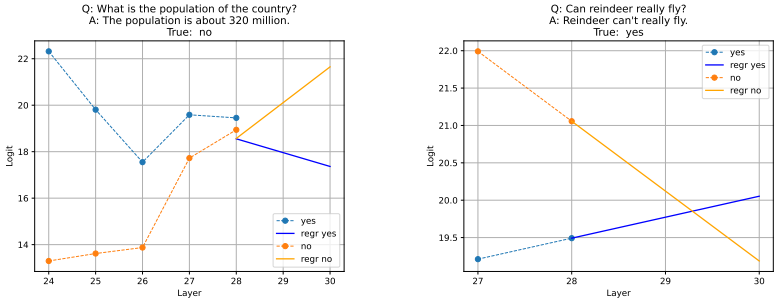
For $M = 24$ (Figure 8a), the logit for **yes** significantly exceeds that for **no** in the original scores, but after regression-based correction, the score for **no** increases and the gap between **yes** and **no** narrows. However, this correction is not complete, and **yes** still remains dominant.

Conversely, for $M = 27$ (Figure 8b), the corrected logit for **no** surpasses that for **yes**, leading to the correct answer. These results indicate that as M increases, the correction effect becomes more pro-



(a) $L = 28$ (b) $L = 29$

Figure 7: Results of applying DeLTa to the Qwen2.5-7B model. The plots show the trajectories of the logit scores for tokens **yes** and **no** as M varies under different values of L ($L = 28, 29$). The proposed method tends to reduce the gap between the logit scores of **yes** and **no** for certain M.



(a) $M = 24$ (b) $M = 27$

Figure 8: Examples of DeLTa applied to Qwen2.5-7B. The dashed lines show the original logit scores, while the solid lines show the logit scores after regression-based correction by DeLTa (e.g. label "regr yes" stands for regression line for logit "yes".)

nounced. Thus, by appropriately setting M, DeLTa can suppress incorrect answers and induce correct ones.

F Cross-Lingual Evaluation on Japanese

To examine whether the proposed decoding strategy generalizes beyond English, we evaluate its performance on a Japanese benchmark, **JTruthfulQA** (Nakamura and Kawahara, 2024), a Japanese counterpart of the TruthfulQA dataset. The factual accuracy of generated responses is automatically assessed using DeepSeek-V3 (DeepSeek-AI, 2024), which assigns a score between 0 (incorrect) and 1 (correct) to each prediction, with the final accuracy computed as the average across all instances.

DeLTa achieves consistent performance gains across all three models. On Qwen2.5-7B-Instruct, it improves accuracy to 68.6, outperforming both the base model (61.9) and the filter baseline (63.7). On Mistral-7B-Instruct-v0.1, it yields the highest improvement of 7.7 points over the base model. Similarly, on Llama-3.1-8B-Instruct, it attains 64.1, surpassing all other variants. These results demonstrate that the benefits of DeLTa extend to Japanese, highlighting its cross-lingual effectiveness in enhancing factuality.

Model	**Accuracy**	**Gain**
Qwen2.5-7B-Instruct	61.9	–
+filter	63.7	(+1.8)
+DoLa (early-layer)	63.2	(+1.3)
+DoLa (late-layer)	62.7	(+0.8)
+DeLTa	**68.6**	**(+6.7)**
Mistral-7B-Instruct-v0.1	28.2	–
+filter	34.7	(+6.5)
+DoLa (early-layer)	27.9	(–0.3)
+DoLa (late-layer)	28.0	(–0.2)
+DeLTa	**35.9**	**(+7.7)**
Llama-3.1-8B-Instruct	60.7	–
+filter	62.7	(+2.0)
+DoLa (early-layer)	56.2	(–4.5)
+DoLa (late-layer)	56.5	(–4.2)
+DeLTa	**64.1**	**(+3.4)**

Table 5: Factual accuracy on JTruthfulQA (Japanese). Evaluation is based on DeepSeek-V3.

G Implementation library

We used Pytorch (Paszke et al., 2019) and huggingface transformers (Wolf et al., 2020) for all experiments.

H Computational Resources

In all experiments in this study, the GPUs used were NVIDIA RTX 6000 Ada Generation or NVIDIA RTX A6000. All experiments were performed using torch.float16, and the VRAM used was approximately 15GB to 40GB.

I Generated Examples

In this section, we present response examples from Qwen2.5-7B and DeLTa for questions in the GSM8K dataset. The example sentences included in the few-shot prompt are omitted, and only the pure model responses are compared. These examples illustrate cases where DeLTa functioned effectively. Among the responses to each question shown in Table 6, the output of DeLTa is listed in the right column.

Method	Qwen2.5-7B	Qwen2.5-7B + DeLTa
Question	Cecilia just bought a new puppy. According to her veterinarian, she has to feed the puppy 1 cup of dog food every day for the first 180 days. Then she has to feed the puppy 2 cups of dog food every day for the rest of its life. If one bag of dog food contains 110 cups, how many bags of dog food will Cecilia use in the first year?	
Answer	The puppy needs to be fed 1 cup of dog food per day for 180 days, so the total amount of food needed for the first 180 days is 1 cup/day x 180 days = 180 cups of dog food. If one bag of dog food contains 110 cups, then Cecilia will need 180 cups / 110 cups/bag = 1.64 bags of dog food. Since she can't buy a fraction of a bag, she will need to buy 2 bags of dog food for the first year. The answer is 2.	In the first 180 days, the puppy will consume 180 * 1 = 180 cups of dog food. In the first year, there are 365 days. So after the first 180 days, there are 365 - 180 = 185 days remaining. In those remaining days, the puppy will consume 185 * 2 = 370 cups of dog food. In total, the puppy will consume 180 + 370 = 550 cups of dog food in the first year. Since one bag contains 110 cups, Cecilia will need 550 / 110 = 5 bags of dog food in the first year. The answer is 5.
Question	Sadie slept 8 hours on Monday. For the next two days, she slept 2 hours less, each, because she had to complete some assignments. If the rest of the week she slept 1 hour more than those two days, how many hours did she sleep in total throughout the week?	
Answer	Sadie slept 8 hours on Monday. On the next two days, she slept 2 hours less each day, so 8 - 2 = 6 hours each day. For those two days, she slept 6 * 2 = 12 hours in total. For the rest of the week, she slept 1 hour more than those two days, so 6 + 1 = 7 hours each day. There are 5 days in the week, so for the rest of the week, she slept 7 * 5 = 35 hours. In total, she slept 8 (Monday) + 12 (next two days) + 35 (rest of the week) = 55 hours throughout the week. The answer is 55.	Sadie slept 8 hours on Monday. On Tuesday and Wednesday, she slept 2 hours less than on Monday, so she slept 8 - 2 = 6 hours on each of those days. In total for those three days, she slept 8 + 6 + 6 = 20 hours. For the rest of the week, she slept 1 hour more than on those two days, so she slept 6 + 1 = 7 hours on each of those days. She has four more days in the week, so she slept 7 * 4 = 28 hours on those days. In total, she slept 20 + 28 = 48 hours throughout the week. The answer is 48.
Question	There are 700 bees in a hive. There are twice as many worker bees as baby bees, and there are twice as many babies as queens. How many worker bees are there?	
Answer	The total ratio is 2 (workers) + 1 (baby) + 1 (queen) = 4 parts. Each part has 700 / 4 = 175 bees. Therefore, there are 2 x 175 = 350 worker bees. The answer is 350.	Let's represent the number of worker bees as W, the number of baby bees as B, and the number of queens as Q. We know that there are 700 bees in total, so W + B + Q = 700. We also know that there are twice as many worker bees as baby bees, so W = 2B, and there are twice as many babies as queens, so B = 2Q. We can substitute these equations into the first equation to get 2B + B + Q = 700. Combining like terms, we get 3B + Q = 700. Now we can substitute B = 2Q into this equation to get 3(2Q) + Q = 700. Simplifying, we get 6Q + Q = 700. Combining like terms, we get 7Q = 700. Dividing both sides by 7, we get Q = 100. Now we can substitute Q = 100 back into B = 2Q to get B = 2(100) = 200. Finally, we can substitute B = 200 back into W = 2B to get W = 2(200) = 400. The answer is 400.
Question	Billy is peeling and cutting potatoes in preparation for making potato salad for his big family reunion barbecue. It's a big event, so he has 60 potatoes to get through. The potatoes are roughly the same size, so it takes about the same amount of time to peel and cut each one. It takes Billy about a minute and a half to peel a potato, but only about 5 seconds to cut it up. How long will it take Billy to finish prepping the potatoes?	
Answer	It takes Billy about 1.5 minutes to peel a potato, which is 1.5 x 60 = 90 seconds. It takes him about 5 seconds to cut each potato. So it takes him about 90 + 5 x 60 = 90 + 300 = 390 seconds. To convert this into minutes, we divide by 60, so it takes Billy about 390 / 60 = 6.5 minutes to finish prepping the potatoes. The answer is 6.5.	It takes 1.5 minutes to peel each potato, so for 60 potatoes, it will take 1.5 * 60 = 90 minutes to peel them all. It takes 5 seconds to cut each potato, so for 60 potatoes, it will take 5 * 60 = 300 seconds to cut them all. To convert seconds to minutes, we divide by 60, so it will take 300 / 60 = 5 minutes to cut the potatoes. In total, it will take 90 minutes + 5 minutes = 95 minutes to prep the potatoes. The answer is 95.
Question	Juan and his brother Carlos are selling lemonade. For each gallon they make it costs $3 for lemons and $2 for sugar. They sell each glass for $0.50 and get 20 glasses per gallon. If they made $25 in profit, how much did they spend on lemons?	
Answer	They spent $5 per gallon because 3 + 2 = 5. They sold 8 gallons because 25 / 3.125 = 8. They spent $40 on lemons because 8 x 5 = 40. The answer is 40	They spent $3 per gallon on lemons and $2 per gallon on sugar, so a total of $3 + $2 = $5 per gallon. They sell 20 glasses per gallon at $0.50 each, so they make 20 * $0.50 = $10 per gallon. They made $25 in profit, so they must have made $25 + $5 = $30 in total. Since they make $10 per gallon, they must have made $30 / $10 = 3 gallons. So they spent 3 * $5 = $15 on lemons. The answer is 15.

Table 6: Examples of responses to the GSM8K dataset by Qwen2.5-7B and DeLTa. The baseline exhibits numerical calculation errors and reasoning mistakes, whereas DeLTa yields more accurate computation results.

Investigating Factuality in Long-Form Text Generation: The Roles of Self-Known and Self-Unknown

Lifu Tu, Rui Meng*, Shafiq Joty, Yingbo Zhou, Semih Yavuz
Salesforce AI Research

Abstract

Large language models (LLMs) have demonstrated strong capabilities in text understanding and generation. However, they often lack factuality, producing a mixture of true and false information, especially in long-form generation. In this work, we investigates the factuality of long-form text generation across various large language models (LLMs), including GPT-4, Gemini-1.5-Pro, Claude-3-Opus, Llama-3-70B, and Mistral. Our analysis reveals that factuality tend to decline in later sentences of the generated text, accompanied by a rise in the number of unsupported claims. Furthermore, we explore the effectiveness of different evaluation settings to assess whether LLMs can accurately judge the correctness of their own outputs: Self-Known (the percentage of supported atomic claims, decomposed from LLM outputs, that the corresponding LLMs judge as correct) and Self-Unknown (the percentage of unsupported atomic claims that the corresponding LLMs judge as incorrect). The results indicate that even advanced models fail to achieve perfect Self-Known scores, while their Self-Unknown scores remain notably above zero, reflecting ongoing uncertainty in their self-assessments. Moreover, we find a correlation between higher Self-Known scores and improved factuality, while higher Self-Unknown scores are associated with lower factuality. Even without significant changes in the models' self-judgment (Self-Known and Self-Unknown), the number of unsupported claims can increases, likely as an artifact of long-form generation. Additional Retrieval-Augmented Generation (RAG) experiments also show the limitations of current LLMs in long-form generation, and provide the more research is needed to improve factuality in long-form text generation.

1 Introduction

The long-context capabilities of large language models (LLMs) (OpenAI, 2023b; AI@Meta, 2024; Jiang et al., 2024; GeminiTeam, 2024; Anthropic, 2024) have seen significant advancements in recent years. Lots of work (Shaham et al., 2023; Bai et al., 2024; An et al., 2024; Zhang et al., 2024; Kuratov et al., 2024) have explored the ability of LLMs to handle long contexts, however, relatively few have examined their ability for long-form text generation.

Despite LLMs have the impressive generative abilities, these models are prone to producing hallucinations (Li et al., 2023; Min et al., 2023) where the generated content often blends factual and fabricated information. This tendency not only undermines performance but also poses substantial risks in practical applications. To assess the factuality of responses from LLMs, recent research (Fan et al., 2020; Wright et al., 2022; Min et al., 2023; Manakul et al., 2023) has introduced a method that breaks down generations into atomic claims – short statements each containing a single piece of information. These atomic claims are then individually evaluated to determine whether they are supported by evidence or unsupported.

To ensure the reliable use of LLMs, it is also crucial that they possess the ability to recognize not only "what they know" but also "what they don't know." Recent studies, such as those by Kadavath et al. (2022); Liu et al. (2022); Guerreiro et al. (2023), have shown that language models can assess the validity of their own claims. However, Srivastava et (2023); Yin et al. (2023) have pointed out the limitations of LLMs in recognizing their own knowledge gaps.

In this work, we investigate the factuality of long-form text generation across various LLMs. We first check the factuality of long-form generation at different relative positions using two annotated datasets and two models: ChatGPT and PerplexityAI (which integrates a search engine). Our findings verify that sentences generated earlier in the sequence generally demonstrate higher fac-

*Now at Google.

Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025), pages 322–336
November 9, 2025 ©2025 Association for Computational Linguistics

tuality. However, these later-generated sentences contain more unsupported claims and fewer supported claims.

To explain this phenomenon, we try to examine whether LLMs exhibit reduced knowledge in later generations with wo metrics: the **Self-Known score** (the percentage of supported atomic claims judged as correct by the LLMs) and the **Self-Unknown score** (the percentage of unsupported atomic claims judged as incorrect by the LLMs). These two metrics are used to quantify the corresponding models' ability to judge the correctness of atomic claims. In order to accurately compute the two metrics, we have tried three different approaches, one of which is a novel approach that adds an answer option: 'None of the above'. This modification appears to provide a more accurate measure of the LLMs' abilities, as evidenced by a higher flip rate for supported claims and an increasing flip rate at later relative positions. This suggests that the model reassesses its confidence when faced with an option signaling uncertainty. In contrast, the low flip rate for unsupported claims indicates a consistent judgment of their incorrectness. These results suggest a nuanced understanding by LLMs of supported versus unsupported claims and underscore the importance of specific evaluation settings to accurately gauge model performance. Our findings align with human annotations for two LLMs, although some discrepancies, particularly with the PerplexityAI model, suggest gaps in estimation.

Later, we apply this modified approach to compute the Self-Known and Self-Unknown scores across various LLMs, including GPT-4, Gemini-1.5-Pro, Claude-3-Opus, Llama-3-70B, and Mistral. We also develop a mathematical framework that links these scores to factuality. Overall, both empirical and theoretical results demonstrate a strong relationship between factuality and the Self-Known and Self-Unknown scores. We observe that these scores can vary significantly across different models. However, even when the Self-Known and Self-Unknown scores remain relatively stable, the number of unsupported claims tends to increase in later parts of the generated text. This suggests that lower factuality in later sentences is not solely due to score changes, but also influenced by error propagation and diminished model knowledge over time.

The main contributions of our work are as follows:

1. We explore the factuality patterns of long-form text generation across various model families (GPT-4, Gemini-1.5-Pro, Claude-3-Opus, Llama-3-70B, and Mistral). We find that even the most advanced LLMs typically exhibit lower factuality scores in the later segments of long-form text. Retrieval-Augmented Generation (RAG) systems show a similar trend, although they tend to maintain higher factuality overall.

2. We analyze Self-Known and Self-Unknown ratios across different segments of generated text. While Self-Known scores are relatively high, even the strongest LLMs (e.g., GPT-4, Gemini-1.5-Pro, Claude-3-Opus) average only around 50%, with Self-Unknown scores remaining well above zero. This suggests that even advanced models struggle to recognize the limits of their own knowledge.

3. We develop a mathematical framework linking Self-Known and Self-Unknown scores to factuality. Empirical and theoretical results show higher Self-Known scores improve factuality, while higher Self-Unknown scores reduce it. Notably, unsupported claims can increase even without major changes in self-judgment, highlighting challenges in long-form generation.

4. We find that Retrieval-Augmented Generation (RAG), which supplies needed knowledge, can improve factuality. However, it fails to fully address the issue of lower factuality at a later position. This highlights the need for alternative framework specifically designed for long-form generation tasks.

2 Long-Form Text Generation

To evaluate the factuality of LLM responses, recent work (Liu et al., 2023; Chen et al., 2022; Min et al., 2023) breaks a generation into a series of atomic claims—short statements that each contain one piece of information. Each atomic claim is then individually evaluated to determine whether it is supported or unsupported. In this section, we first explore the factuality patterns of these atomic claims in long-form text generation.

2.1 Observations

In order to explore the factuality of long-form generation at different relative positions, we use the human annotated data from Min et al. (2023) to compute the macro-average percentage of three different claims (supported, unsupported, and irrelevant) across five different relative positions. In their human-annotated data, each long LLM generation is decomposed into atomic claims and each

atomic claim is assigned with one of the three labels ("supported", "not-supported", "irrelevant").

The detailed procedures for computing fractions of different type claims at different relative positions are as following: 1) Calculate the fraction of supported, unsupported, and irrelevant claims for each sentence; 2) Determine each sentence's relative position in the generation , e.g., if it is the third sentence out of six, its relative position would be 3/6 = 50%; 3) Group sentences into relative position ranges: 0-20%, 20%-40%,, etc.; 4) Compute the macro-average claim percentages within each group

Figure 1 (a) shows ChatGPT results (PerplexityAI results are in the Appendix). Unsupported claims increase in later-generated sentences. Figure 1 (b) further shows that as generation continues, LLMs produce more unreliable and unsupported claims.

Open Questions. Is the phenomenon above due to LLMs having less knowledge about later generations? Can LLMs recognize when claims are supported and when they are not? Do LLMs identify unsupported claims more effectively when they appear later in the text compared to earlier ones?

3 Self-Known and Self-Unknown

To investigate these questions, we examine whether the corresponding LLMs recognize their atomic claims by computing two metrics: **Self-Known** (the percentage of supported atomic claims that the corresponding LLMs judge as correct) and **Self-Unknown** (the percentage of unsupported atomic claims that the corresponding LLMs judge as incorrect). While there is related work, such as Rajpurkar et al. (2018); Xiong et al. (2024), our approach differs in two key ways: (1) Evaluation is conducted on atomic claims, which are derived from sentences in long-form generation, rather than assigning a score to the entire model output; (2) Our focus is on factuality (whether an atomic claim is true or false), rather than on uncertainty scores (i.e., "How likely is the above answer to be correct?").

We explore the computation of **Self-Known** and **Self-Unknown** using the following three approaches (with the corresponding prompt templates provided in Appendix Section B):

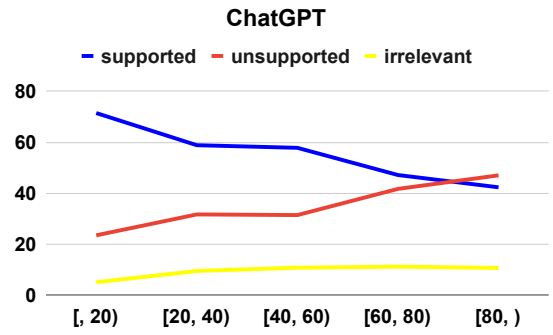
- **Direct-Asking**: In this approach (Rajpurkar et al., 2018), the atomic claim is directly given to the corresponding LLMs and be asked whether the statement is true or false.
- **Question-Answering**: Given an atomic claim, a question-answer pair can be derived (Trischler et al., 2017; Rajpurkar et al., 2018; Hu et al., 2024) with GPT-4 Turbo. For example, "Lanny Flaherty is an American." can be used to derived a question-answer pair ("What nationality is Lanny Flaherty?", "American"). Then, given the question and answer, we ask the corresponding LLMs whether the answer is true or false.
- **Question-Answering w/NOA**: Similar to the above approach, a question-answer pair is derived according to each atomic claim. One big different is: given question and answer, one more addition choice ("None of the above") (Rajpurkar et al., 2018) is given to the corresponding LLMs. This is a well-defined evaluation because it can check whether the model actually knows the answer of the question, especially if the question is vague or context-information is missing.

We compute the Self-Known score and the Self-Unknown score using these prompt templates. The human annotated data on ChatGPT[1] are used in this experiments. Figure 2 presents the results on ChatGPT.

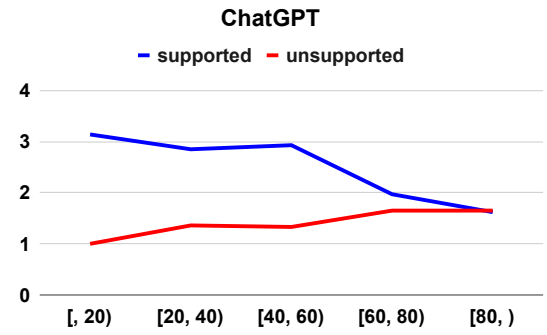
Comparison on the above three evaluation settings With the first two settings, the results of **Self-Known** score and **Self-Unknown** score are similar. However, the results of the third setting differ from the other two. We hypothesize that the reason is that the added choice, "None of the above" which allows the LLM to determine whether it knows the answer to the question.

To examine the effect of this setting, we plot the flip rate (claims judged as correct by the LLM in setting (b) but judged as incorrect in setting (c)) for supported and unsupported claims. As shown in Figure 2d, there is a high flip rate for supported claims, and this rate increases with higher relative positions. In contrast, there is almost no flipping for unsupported claims. Therefore, setting (c) is more suitable for checking whether the LLM knows a atomic claim. The high flip rate observed for supported claims suggests that the model is reconsidering its initial judgments when presented with the

[1]The labeled ChatGPT data is also from Min et al. (2023) as above. There are 183 long generations of ChatGPT.

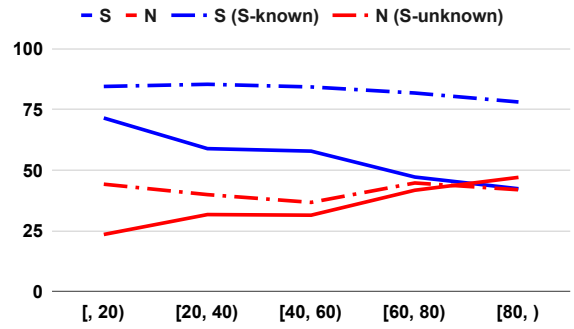


(a) Percentage (%) of supported, unsupported and irrelevant atomic claims.

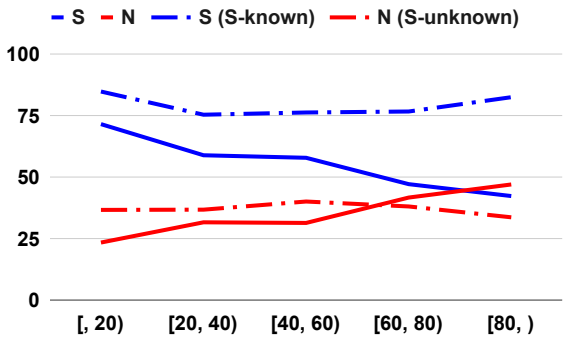


(b) Number of supported and unsupported atomic claims.

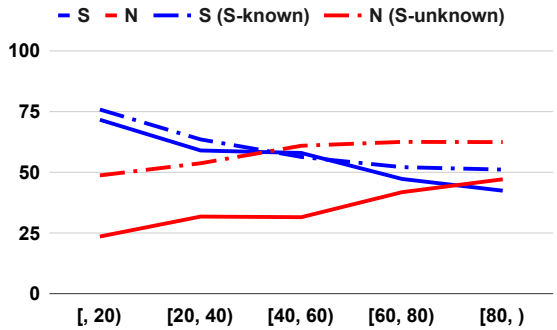
Figure 1: Long-form generation across different relative positions (%) for ChatGPT.



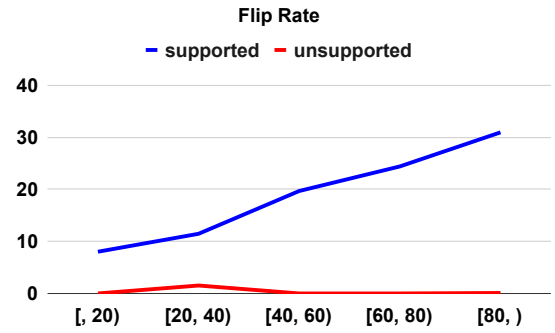
(a) Direct-Asking



(b) Question-Answering



(c) Question-Answering W/ NOA



(d) Flip rate (%) from setting (b) to setting (c) across different relative positions (%) for both supported claims and unsupported claims. The high flip rate of supported claims indicates that prompting with setting (c) better recognizes whether LLMs accurately assess their knowledge of atomic claims.

Figure 2: Self-Know and Self-Unknown results of ChatGPT across different relative positions (%). **S**: **factuality** (percentage of supported atomic claims); **N**: percentage of unsupported atomic claims; **S (S-known)**: **Self-Known** score; **N (S-unknown)**: **Self-Unknown** score

option "None of the above". This indicates that the model may not be entirely confident in its original answers and is more likely to recognize uncertainty. The increasing flip rate for higher relative positions further supports this, implying that the model's confidence decreases as the position of the claim within the context changes.

In summary, we observed similar results between setting (a) (Direct-Asking) and setting (b) (Question-Answering), and a significant difference between setting (b) (Question-Answering) and setting c (Question-Answering W/ NOA). **The deeper**

analysis between setting (b) and setting (c) revealed that setting (c) recognizes atomic claims more confidently and treats atomic claims that flip as unknown. This is why we chose to use setting (c) in the subsequent experiments.

4 Analysis

We denote the prompt input of LLMs as x and long output of LLMs as y. The binary auxiliary label $d = 1$ indicates the LLM output is factual correct and $d = 0$ indicates LLM output is wrong.

We assume that $P(d = 1 \mid y, x)$ is equal to **factuality score**[2] σ of LLM output y. Given x, the joint distribution of between the auxiliary label and model output (d, y) is

$$\begin{aligned} & \sigma * P(y \mid x) && (1) \\ =& P(d=1 \mid y,x) * P(y \mid x) = P(d=1, y \mid x) \\ =& P(d=1, \mathrm{y}_{\text{correct}} \mid x) * \sigma + \\ & P(d=1, \mathrm{y}_{\text{wrong}} \mid x) * (1-\sigma) \\ =& P(d=1 \mid \mathrm{y}_{\text{correct}}) * P(\mathrm{y}_{\text{correct}} \mid x) * \sigma + \\ & P(d=1 \mid \mathrm{y}_{\text{wrong}}) * P(\mathrm{y}_{\text{wrong}} \mid x) * (1-\sigma) && (2) \end{aligned}$$

$\mathrm{y}_{\text{correct}}$ refers to model outputs aligned with the ground truth and $\mathrm{y}_{\text{wrong}}$ refers to outputs that are wrong. Because y is the generated output according to the log-likelihood, the correct part and incorrect part have similar log-likelihood. Then, it is reasonable to have this following assumption:

$$P(y \mid x) \approx P(\mathrm{y}_{\text{correct}} \mid x) \approx P(\mathrm{y}_{\text{wrong}} \mid x)$$

Then, after cancel the above three terms in Equation 1 and Equation 2 ,

$$\sigma = P(d=1 \mid \mathrm{y}_{\text{correct}})\sigma + P(d=1 \mid \mathrm{y}_{\text{wrong}})(1-\sigma)$$

We denote $P(d = 1 \mid \mathrm{y}_{\text{correct}})$ and $P(d = 0 \mid \mathrm{y}_{\text{wrong}})$ as **Self-Known** score (percentage of supported atomic claims judged as correct by LLMs) and **Self-Unknown** score (percentage of unsupported atomic claims judged as incorrect by LLMs) respectively. Once the above formula is solved, we can determine the relationship among the factuality score, Self-Known score, and Self-Unknown score:

$$\sigma = \frac{1 - \text{Self-Unknown}}{2 - \text{Self-Unknown} - \text{Self-Known}} \quad (3)$$

Where σ is the factuality score.

[2]This is an assumption we are making: that there is no overconfidence, and the confidence score is approximately equal to the factuality score.

Factuality Vs. Self-Known Vs. Self-Unknown Given Self-Unknown $\in [0, 1]$ and Self-Known $\in [0, 1]$, the factuality score increases when the **Self-Known** score is increased or the **Self-Unknown** score is decreased. This matches our observations in Section 3 and Figure 2 (c).

Estimation of factuality Score In Equation 3, we present a method for estimating the factuality score. We use the Self-Known and Self-Unknown results of the corresponding model (ChatGPT) with configuration (c) to estimate the factuality score across different relative positions. As shown in Figure 3, our estimation closely matches the human-annotation results.

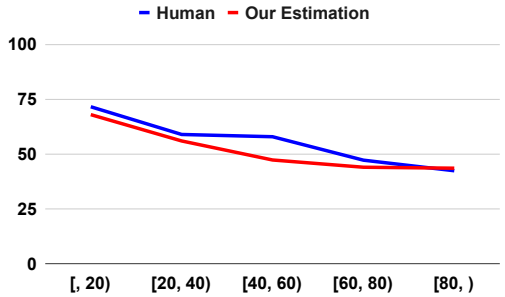


Figure 3: Human-annotation factuality score (%) and our estimation with Equation 3 across different relative positions (%).

5 Automatic Results on Additional LLMs

In the previous section, our experiments were conducted using human-annotated factuality data. In this section, we first introduce an automated tool for factuality evaluation. Then, using the proposed approach from Section 3 to compute Self-Known and Self-Unknown scores, we analyze the trends in factuality, Self-Known, and Self-Unknown scores across other advanced LLMs

5.1 Automatic Tool Setting

In Section 2, we used the human annotated data (atomic claims are short statements that are decomposed from the model's generation, and each atomic claim is labeled as either supported or unsupported based on its factual correctness.).

Configuration We use the tool FActScore (Min et al., 2023) for factuality evaluation with the following configuration: the latest version of GPT-3.5 (gpt-3.5-turbo-0125) is used to break a generated text into a series of atomic claims and evaluate each

atomic claim against a retrieved knowledge (model name "retrieval+llama+npm" is used during the evaluation)[3].

Results Figure 7 in the Appendix shows the comparison between the tool's evaluation and human annotation results. We notice the tool's estimation is highly correlate well with human annotations. For number of atomic claims, the absolute difference is not bigger than 1. And the trend of tool's estimation is almost the same as human annotation. For factuality estimation, the tool's results are well-aligned with human annotations for two OpenAI models. Although there is an estimation gap for the PerplexityAI model, the trend of the estimation remains consistent with human annotations.

Takeaway. The tool with above configurations can well capture the trend of number of atomic claim and factuality.

5.2 Additional LLMs

In this section, we explore the factuality of long-form text generation across different relative positions using automatic tools.

5.2.1 Experimental Setup

For each LLM, we follow four key steps to obtain experimental results: (1) generating text outputs; (2) filtering the generated content; (3): evaluating factuality; and (4): estimating **Self-Known** and **Self-Unknown** scores with the corresponding LLM. For more details on each step, please refer to Appendix Section E.

5.2.2 Results

Figure 4 show results of several powerful LLMs (Gemini-1.5-pro, Claude-3-opus, and two Mistral AI models). Two additional LLMs (GPT-4, and Llama-3-70B-Instruct) results are provided in Figure 9 in the Appendix.

Decreasing Factuality: Strong Start, Later Decline According to the bold blue lines in Figure 4, we observe the highest factuality scores are observed at the beginning of the generated text across all relative positions.

Factuality Vs. Self-Known Vs. Self-Unknown Overall, we observe that the Self-Known score is positively correlated with factuality, as indicated by the **two blue lines**, and the Self-Unknown score is positively correlated with the percentage of unsupported atomic claims, as shown by the **two red lines** in each figure. For these advanced LLMs, the trend of these three scores across different positions shows smaller variation.

Clear Difference in the Number of Unsupported Claims Across Positions In Figure 4 (e) and (f), observed minimal differences in factuality for the two models (Mixtral-8x7b and Mistral-Large). However, as depicted in Figure 5, the number of unsupported claims increases significantly from the beginning to the end of the generated text. It indicates the challenges of long-form generation. This also highlights a limitation in relying solely on factuality scores for evaluation.

No Significant Changes in Self-Judgment for Some Advanced LLMs We can observe that there is no big change according to dashed lines (Self-Known and Self-Unknown) in Figure 4. However, the number of unsupported claims are increasing as shown in Figure 5.

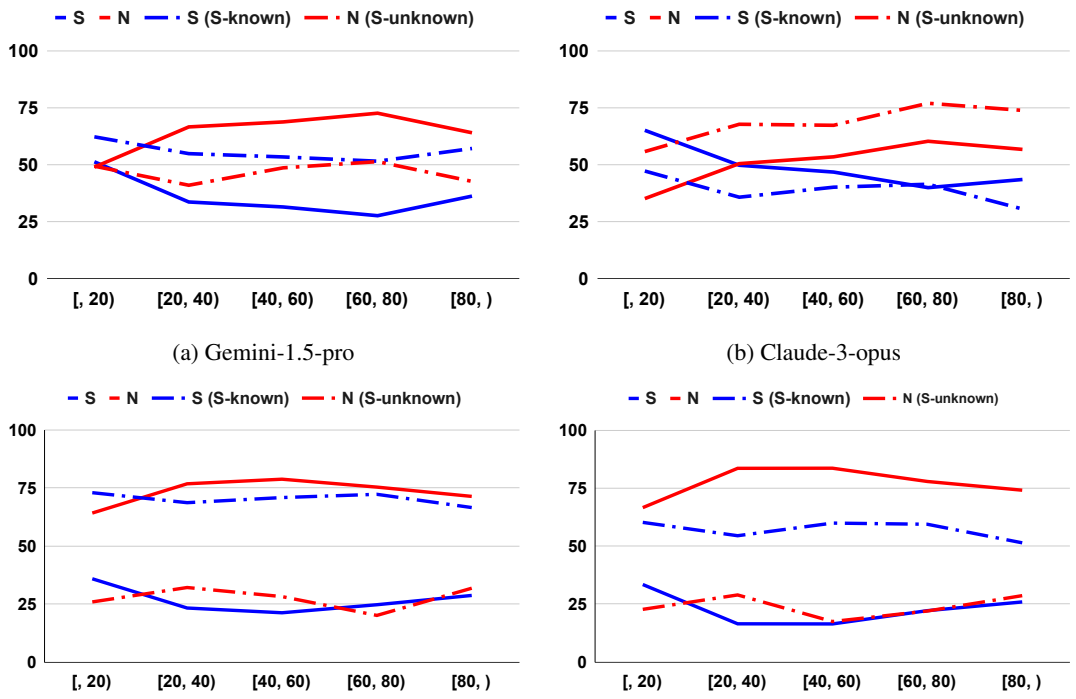
How to Improve Factuality Score? In Equation 3, we propose estimating the factuality of a LLM using Self-Known and Self-Unknown scores. A higher Self-Known score typically corresponds to higher factuality. However, does this mean LLMs would achieve 100% factuality if they had a 100% Self-Known score and 0 Self-Unknown score on their own generation? The answer is no. It is a necessary condition, not a sufficient one for achieving 100% factuality. In the derivation of Equation 3, several additional assumptions are made[4].

According to our results, a higher Self-Known score is usually associated with higher factuality, while a higher Self-Unknown score is associated with lower factuality for LLMs. This indicates that it is challenging for LLMs to recognize unsupported claims on their own. Therefore, a judgment model that incorporates an external knowledge source is necessary for this recognition.

Some reasonable questions arise: Are decoding errors in LLMs caused by a lack of relevant knowledge? Can Retrieval-Augmented Generation (RAG), which supplies additional context, address

[3]In the original work, text-davinci-003 was used to get atomic claims and ChatGPT is used to evaluate whether each atomic is supported or unsupported.

[4]For instance, one key assumption is that the probability of correctness given the model output and input $P(d = 1 \mid y,x)$, equals the factuality score σ of output y, However, if a LLM becomes overconfident in generating answers, the term $P(d = 1 \mid y,x)$ may significantly exceed the actual factuality score.



(a) Gemini-1.5-pro

(b) Claude-3-opus

(c) Mixtral-8x7b

(d) Mistral-Large

Figure 4: Self-Know and Self-Unknown results of different LLMs across different relative positions (%). **S**: **factuality** (percentage of supported atomic claims); **N**: percentage of unsupported atomic claims; **S (S-known)**: **Self-Known** score; **N (S-unknown)**: **Self-Unknown** score.

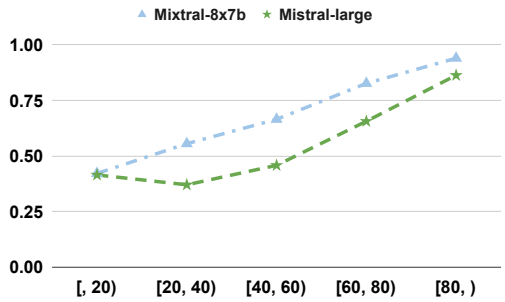


Figure 5: There may be minimal change in the factuality score, but a significant increase in the number of unsupported claims across different relative positions(%).

the decline in factuality during later stages of generation? In the next section, we present our exploration of RAG-based experiments across different LLMs.

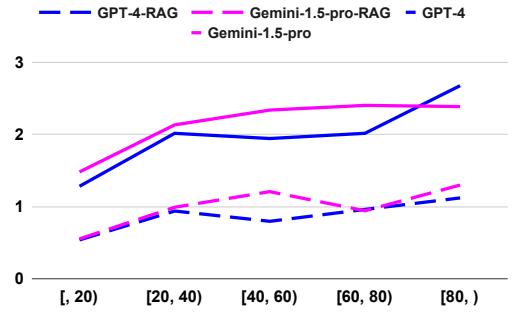
5.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a widely used approach for enhancing language model performance in various applications. In RAG, relevant text segments are retrieved from an external knowledge source and integrated into the model's responses. For our retrieval corpus, we utilized the English Wikipedia as of April 1, 2023, with each page divided into chunks of up to 256 tokens. These retrieved passages, containing facts relevant to the entity, were incorporated into the LLMs' context to improve the factual accuracy of the generated content.[5].

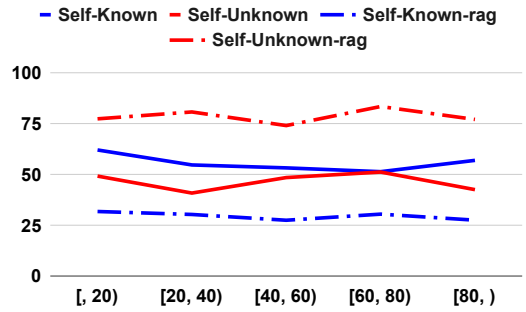
According to Figure 6 (a), in the RAG setting, although there are significantly fewer unsupported atomic claims overall, a notable increase in the number of unsupported claims is observed in later stages of generation. As shown in Table F, LLMs can still response with lots of unsupported claim even given context knowledge. This increase is likely due to error propagation within the LLMs, highlighting the challenges of long-form generation even when relevant parts are provided.

Figure 6 (b) demonstrates that the RAG system exhibits significantly lower Self-Known scores and higher Self-Unknown scores. This discrepancy may stem from the corresponding LLM's lack of prior knowledge regarding the retrieved content in the RAG system, causing it to mistakenly assess accurate information as incorrect.

[5]One example is shown in Table F.



(a) Number of unsupported claims for two LLMs and their RAG models across different relative positions(%).



(b) Self-Known and Self-Unknown scores for Gemini-1.5-pro and the RAG model across different relative positions(%).

Figure 6: RAG experiments on two strong LLMs (Gemini-1.5-pro and GPT-4).

In these RAG experiments, incorporating all relevant knowledge leads to improved factuality in LLMs. However, they still exhibit a decline in factuality during later stages of generation. This underscores the need for alternative frameworks specifically tailored to long-form generation tasks. For instance, employing more sophisticated decoding strategies may help mitigate the challenges associated with long-form generation.

6 Related Work

Factuality Evaluation Recent advancements have seen significant efforts in quantifying the factuality of LLM generations. For short answers, factuality often correlates with fact verification, which directly assesses whether the generation aligns with extensive knowledge sources and references (Thorne et al., 2018; Honovich et al., 2022) or utilizes language models (Lin et al., 2022). However, evaluating factuality in long-form content poses greater challenges due to the complexity of the generation process. Recent studies (Fan et al., 2020; Wright et al., 2022; Min et al., 2023) have approached this challenge by breaking down long generations into atomic claims. While these approaches predominantly focus on factual precision, some studies (Wei et al., 2024) also consider evaluating factual recall. In our work, we concentrate on factual precision akin to Min et al. (2023). Moving forward, the development of more robust automatic tools will be crucial for advancing factuality exploration in long-form generation tasks.

Self-Know and Self-Unknown Recent studies have extensively explored the concepts of Self-Known and Self-Unknown in language models. For instance, Kadavath et al. (2022); Liu et al. (2022); Guerreiro et al. (2023) demonstrated that language models are capable of assessing the validity of their own claims and predicting their ability with answering true/false questions accurately. Meanwhile, Srivastava et (2023); Yin et al. (2023) highlighted the limitations of LLMs in acknowledging their unknowns, focusing on their ability to recognize unknown knowledge. In our work, we specifically investigate whether LLMs can identify and reconsider unsupported claims generated from their own outputs. Our results indicate that LLMs struggle to accurately judge unsupported atomic claims from their own generations. We also find that a lower Self-Unknown score or a higher Self-Known score corresponds to higher factuality.

7 Conclusion

n this study, we investigate the factuality of long-form text generation across different language model families and at various stages of generation. We observe a consistent decline in factuality in sentences generated later in the sequence. To understand the underlying causes, we explore two possible factors: diminished self-knowledge in later generations and the accumulation of earlier generation errors (i.e., error propagation). To analyze this, we introduce the concepts of Self-Known and Self-Unknown scores, which measure a model's confidence in its own knowledge. We find that current LLMs struggle to maintain factual accuracy over extended generations, partly due to limitations in their internal knowledge representation and propagation mechanisms. Addressing these challenges requires further research. Promising directions include the development of external

factuality evaluation modules (e.g., dedicated judge models) and the design of more robust decoding strategies tailored to long-form generation

8 Limitations

Following are limitations in our work.

Evaluation of Self-Know and Self-Unknown In this work, we design three different methods for estimating Self-Known and Self-Unknown scores on LLMs' own generation. We find that the third setting (c), which includes the option "None of the above," is effective in determining whether LLMs can accurately judge the correctness of claims generated from their own outputs. Although our results show that these scores are well aligned with the estimation of factuality scores using Equation 3, exploring better methods for evaluating the correctness of claims with LLMs would still be beneficial for future study.

Factuality Evaluation In this work, we limit the domain of long-form generation to ensure accurate factuality evaluation. The concern is that broadening the topic range might compromise the accuracy of our factuality assessments, rendering our study less effective. Recently, evaluation tools (Guan et al., 2024; Es et al., 2023; Tang et al., 2024; Wei et al., 2024) have been explored. With stronger tools, it is possible to explore a wider range of domains beyond Wikipedia.

Moreover, in this work, we primarily focus on factuality precision. However, considering factuality recall is also important, as it ensures that the omission of significant pieces of information is penalized during evaluation. By incorporating both precision and recall, we can achieve a more comprehensive and accurate assessment of factuality in long-form generation.

References

AI@Meta. 2024. Llama 3 model card.

Chenxin An, Shansan Gong, Ming Zhong, Xingjian Zhao, Mukai Li, Jun Zhang, Lingpeng Kong, and Xipeng Qiu. 2024. L-eval: Instituting standardized evaluation for long context language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14388–14411, Bangkok, Thailand. Association for Computational Linguistics.

Anthropic. 2024. Introducing the next generation of claude.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Jifan Chen, Aniruddh Sriram, Eunsol Choi, and Greg Durrett. 2022. Generating literal and implied subquestions to fact-check complex claims. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3495–3516, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. 2023. Ragas: Automated evaluation of retrieval augmented generation. *Preprint*, arXiv:2309.15217.

Angela Fan, Aleksandra Piktus, Fabio Petroni, Guillaume Wenzek, Marzieh Saeidi, Andreas Vlachos, Antoine Bordes, and Sebastian Riedel. 2020. Generating fact checking briefs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7147–7161, Online. Association for Computational Linguistics.

GeminiTeam. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Jian Guan, Jesse Dodge, David Wadden, Minlie Huang, and Hao Peng. 2024. Language models hallucinate, but may excel at fact verification. *Preprint*, arXiv:2310.14564.

Nuno M. Guerreiro, Elena Voita, and André Martins. 2023. Looking for a needle in a haystack: A comprehensive study of hallucinations in neural machine translation. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1059–1075, Dubrovnik, Croatia. Association for Computational Linguistics.

Or Honovich, Roee Aharoni, Jonathan Herzig, Hagai Taitelbaum, Doron Kukliansy, Vered Cohen, Thomas Scialom, Idan Szpektor, Avinatan Hassidim, and Yossi Matias. 2022. TRUE: Re-evaluating factual consistency evaluation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3905–3920, Seattle, United States. Association for Computational Linguistics.

Xuming Hu, Junzhe Chen, Xiaochuan Li, Yufei Guo, Lijie Wen, Philip S. Yu, and Zhijiang Guo. 2024. Towards understanding factual knowledge of large language models. In *The Twelfth International Conference on Learning Representations*.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Yuri Kuratov, Aydar Bulatov, Petr Anokhin, Ivan Rodkin, Dmitry Sorokin, Artyom Sorokin, and Mikhail Burtsev. 2024. Babilong: Testing the limits of llms with long context reasoning-in-a-haystack. *Preprint*, arXiv:2406.10149.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. A token-level reference-free hallucination detection benchmark for free-form text generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.

Yixin Liu, Alex Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, and Dragomir Radev. 2023. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4140–4170, Toronto, Canada. Association for Computational Linguistics.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.

OpenAI. 2023a. Chatgpt. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2023b. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Uri Shaham, Maor Ivgi, Avia Efrat, Jonathan Berant, and Omer Levy. 2023. ZeroSCROLLS: A zero-shot benchmark for long text understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7977–7989, Singapore. Association for Computational Linguistics.

al Srivastava et. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Preprint*, arXiv:2206.04615.

Liyan Tang, Philippe Laban, and Greg Durrett. 2024. Minicheck: Efficient fact-checking of llms on grounding documents. *Preprint*, arXiv:2404.10774.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. 2017. NewsQA: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 191–200, Vancouver, Canada. Association for Computational Linguistics.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models.

Dustin Wright, David Wadden, Kyle Lo, Bailey Kuehl, Arman Cohan, Isabelle Augenstein, and Lucy Lu Wang. 2022. Generating scientific claims for zero-shot scientific fact checking. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2448–2460, Dublin, Ireland. Association for Computational Linguistics.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

A LLMs

Following are the models that are used in our experiments.

- InstructGPT (text-davinci-003) (Ouyang et al., 2022):
- ChatGPT (OpenAI, 2023a)
- PerplexityAI[6]
- GPT-4, GPT-4-Turbo (OpenAI, 2023b)
- Claude-3-Opus (Anthropic, 2024)
- Gemini-1.5-Pro (GeminiTeam, 2024)
- Mixtral-8x7b (Jiang et al., 2024)
- Mistral-Large[7]
- Llama-3-70B-Instruct (AI@Meta, 2024)

[6]https://www.perplexity.ai/

[7]https://mistral.ai/technology/#models

B Prompt Templates

B.1 Prompts Self-known and Self-unknown

Direct-Asking Given an atomic claim `{claim}` and person `{person}`, we use the following template:

```
Following is a statement from a
bio of {person}. Please check
whether the statement is correct
or wrong according to your
knowledge.

{claim}
Is this statement true or false?
```

Question-Answering Give a question answer pair `<{q}, {a}>` that is derived from an atomic claim, the following template is used to determine whether LLMs consider the proposed answer is correct:

```
Question: {q}
Proposed Answer: {a}
Is the proposed answer:
 (A) True
  (B) False
The proposed answer is:
```

Question-Answering w/ None of the above Given the question answer pair `<{q}, {a}>` derived from an atomic claim, the following template is used:

```
Question: {q}
Proposed Answer: {a}
Is the proposed answer:
 (A) True
  (B) False
  (C) None of the above
The proposed answer is:
```

B.2 Prompts for Creating the Question-answer Pair

Given an atomic claim `{claim}` of a bio and the person `{person}`, a question-answer pair can be derived with gpt-4 with the following template:

```
Following is a fact from a bio of
{person}. Please ask a question
and provide the answer. The
answer is as concise as you
can, using a single phrase if
possible. The answer is also
part of the provided fact. The
question and answer is separetd
with #.

{claim}
```

C Rules for Filtering Generations

Following are the rules we find that are useful to filter out unresponsive generation.

```
I don't have ...
I do not have ...
I need more information ...
Please provide me ...
Please clarify
I apologize ...
there isn't enough information
Unfortunately, there is no ...
If you can provide more
information ...
you could provide more ...
It seems you might ...
```

D Automatic Tool Results

See Figure 7

E Details on Computing Experimental Result For each LLM

Step 1: Obtaining generations We feed a prompt "Tell me a bio of <entity>" to the LLM and take the generation. 500 human entities (Min et al., 2023) are used to generate these biographies.

Step 2: Filtering generations For lots of LLMs, a biography is not provided if they think they do not have enough detailed information to provide a biography. We implement rules to filter out these generations[8].

Step 3: Evaluation factuality We use the tool for breaking generations into atomic claims and evaluate each claim whether it is supported or not. In order to save cost, we randomly sampled 100 samples among the filtered generations. During factuality evaluation, Wikipedia's knowledge source is used in the automatic tool.

Step 4: Estimation of Self-Known and Self-Unknown With above decomposed atomic claims, we use GPT-4 Turbo to get question-answer pairs. For each question-answer pair, a prompt template (see 3) is used to determine whether LLMs consider the proposed answer to be correct. The ratios of supported claims judged as correct, and unsupported claims judged as incorrect are then obtained.

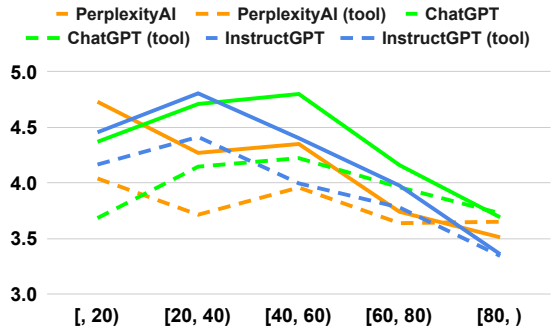
F More results

	#Claims / Gen	Filtered Rate (%)
GPT-4	60.8	12.0
Gemini-1.5-pro	67.5	30.0
Claude-3-opus	41.0	42.0
Llama-3-70B-Instruct	45.9	17.2
Mixtral-8x7b	44.8	0.4
Mistral-Large	48.3	5.0

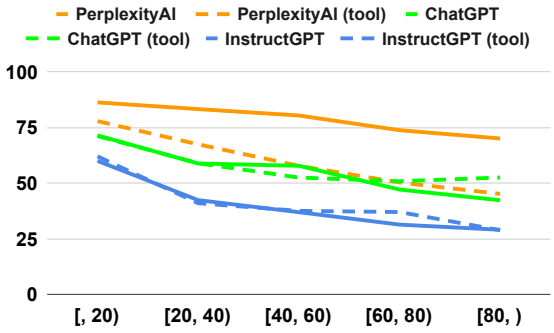
Table 1: Statistics for various LLMs when generating biographical paragraphs.

Table 1 in the Appendix presents two results for various LLMs: the average number of atomic claims per generation and the filtered rate. The filtered rate represents the percentage of instances where the LLMs do not provide valuable responses, often due to perceiving insufficient information to generate a meaningful answer. We notice that the behavior of Claude-3-opus and Gemini-1.5-pro is more conservative. These models frequently decide not to provide a valuable response, instead stating something like "I do not have enough verified information".

[8]The useful rules are shown in Section C.

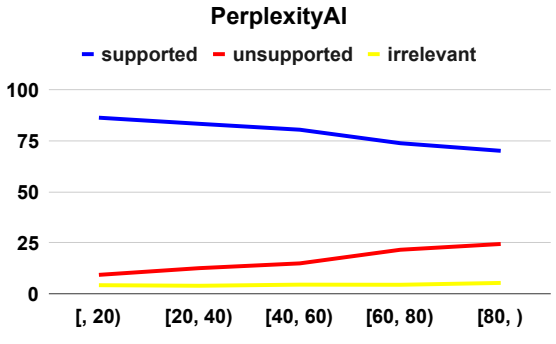


(a) Number of atomic claims

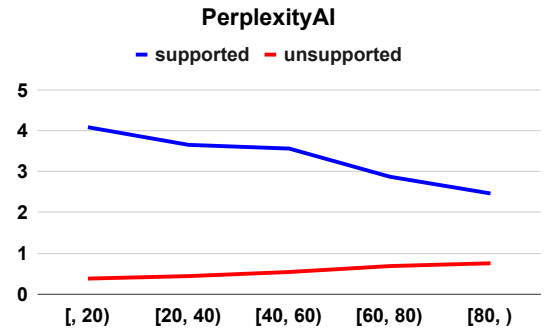


(b) Factuality

Figure 7: Comparison between our used tool and human annotation. The number of atomic claims and factuality (%) across different relative positions (%) are shown for three LLMs: InstructGPT (text-davinci-003), ChatGPT and PerplexityAI.

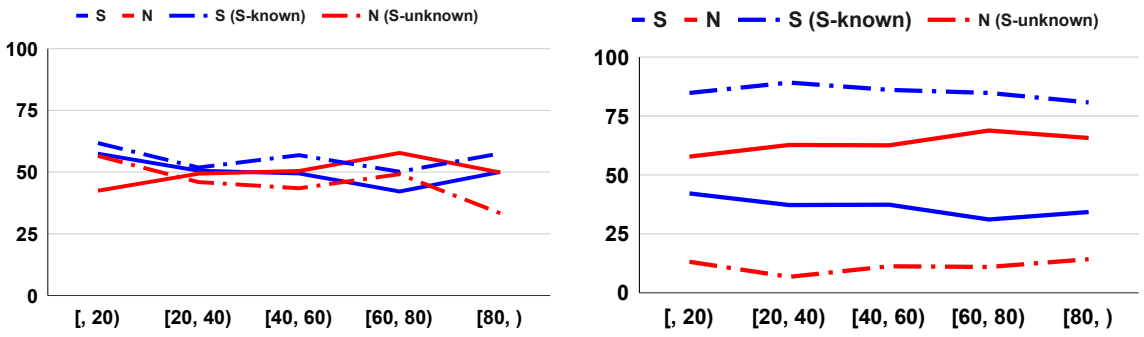


(a) Percentage (%) of supported, unsupported and irrelevant atomic claims.



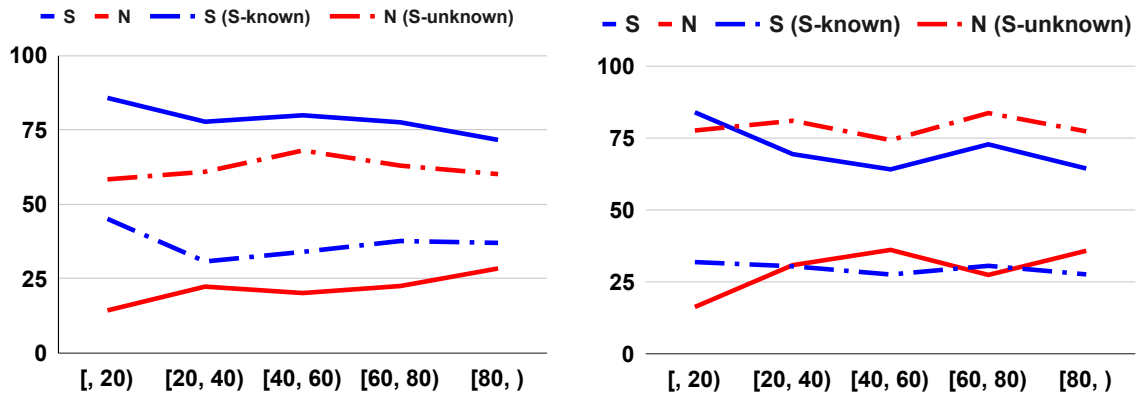
(b) Number of supported and unsupported atomic claims.

Figure 8: Long-form generation across different relative positions (%) for PerplexityAI.



(a) GPT-4 (b) Llama-3-70B-Instruct

Figure 9: Self-Know and Self-Unknown results of different LLMs across different relative positions (%). **S**: **factuality** (percentage of supported atomic claims); **N**: percentage of unsupported atomic claims; **S (S-known)**: percentage of supported atomic claims judged as correct by LLMs; **N (S-unknown)**: percentage of unsupported atomic claims judged as incorrect by LLMs.



(a) GPT-4-RAG (b) Gemino-1.5-pro-RAG

Figure 10: Self-Know and Self-Unknown results of different **RAG** models across different relative positions (%). **S**: **factuality** (percentage of supported atomic claims); **N**: percentage of unsupported atomic claims; **S (S-known)**: percentage of supported atomic claims judged as correct by LLMs; **N (S-unknown)**: percentage of unsupported atomic claims judged as incorrect by LLMs.

Document [0] Jessie Mae Brown Beavers Jessie Mae Brown Beavers (March 18, 1923 – September 6, 1989) was an American journalist based in Los Angeles, California. She was an editor at the "Los Angeles Sentinel" from 1949 to 1989, and served sixteen years on the city's Human Relations Commission, beginning with her 1973 appointment by mayor Tom Bradley.Early life. Jessie Mae Brown was born in Los Angeles, the daughter of Arnetta Hoyt Brown, a Baptist deaconess. She attended the University of California, Los Angeles, where she earned a bachelor's degree in sociology.Career. Brown was editor of the family section of the "California Eagle" from 1944 to 1949, when she joined the staff of the "Los Angeles Sentinel" as an editor. In 1966 she was one of the organizers and leaders of the Los Angeles chapter of the National Association of Media Women. In 1969, she was given the Outstanding Woman in Journalism Award by the University of Southern California chapter of Theta Sigma Phi. Beavers served on the Los Angeles Human Relations Commission for sixteen years, after she was appointed by mayor Tom Bradley in 1973. On the commission, she worked closely with fellow commissioner Toshiko S. Yoshida during the American bicentennial
Document [1] , chaired the affirmative action subcommittee, and in 1982 organized hearings on racial bias in the entertainment industry. She also served one the Los Angeles County Music and Performing Arts Commission. She ran for a seat on the Los Angeles City Council in 1987. Beavers was a member of Jack and Jill and the Lullaby Guild. In 1968 she was cited for her work by the President's Youth Council. In 1972 she was honored alongside Quincy Jones, editor Ruth Washington, and student leader Willis Edwards by the Committee of Women for Good Government. In 1973 she was honored by colleagues and admirers (including Mayor Bradley, judge Vaino Spencer, columnist Gertrude Gipson, and clubwoman Leontyne Butler King) at a tribute event at the Beverly Wilshire Hotel, hosted by singer Thelma Houston.Personal life. In 1947 Brown married insurance executive Leroy A. Beavers, Jr., nephew of Los Angeles businessman George A. Beavers Jr. Actress Dorothy Dandridge was one of her bridesmaids in the wedding. They had three children. In 1979, she was assaulted and robbed in the driveway of her home. She died in 1989, aged 66 years, in Los Angeles. Her grave is at Forest Lawn Cemetery in Gl
Document [2] endale, California.

Question: Tell me a bio of Jessie Mae Brown Beavers.

Response: Jessie Mae Brown Beavers (March 18, 1923 – September 6, 1989) was a prominent American journalist and community leader based in Los Angeles, California. Born in Los Angeles, she obtained her bachelor's degree in sociology from the University of California, Los Angeles.
Beavers's career was deeply intertwined with journalism and activism. She served as an editor for the *California Eagle* and later the *Los Angeles Sentinel*, a prominent African American newspaper, for over four decades. Her commitment to social justice led her to co-found the Los Angeles chapter of the National Association of Media Women and actively advocate for fair representation in the media.
Appointed by Mayor Tom Bradley, Beavers dedicated sixteen years to the Los Angeles Human Relations Commission, tackling issues of racial bias and promoting affirmative action. Her influence extended to her service on the Los Angeles County Music and Performing Arts Commission.
Throughout her life, Beavers received numerous accolades, including recognition from the President's Youth Council and a tribute event graced by notable figures like Mayor Bradley and singer Thelma Houston. Her enduring legacy continues to inspire aspiring journalists and advocates alike.

Table 2: Generated outputs of Gemini-1.5-pro: each document represents a retrieval chunk, with text highlighted in red and yellow indicating unsupported claims. Red text represents claims identified as unsupported by the factuality tool. Yellow text represents unsupported claims that are not identified by the factuality tool.

Author Index

Agarwal, Chirag, 89
Aldarmaki, Hanan, 1
Ali Alblooshi, Humaid, 1
Amar, Shmuel, 138
Ananiadou, Sophia, 200
Antonucci, Alessandro, 208
Aïdasso, Henri, 287

Brown, Jason, 61

C. Snell, Jake, 194
Cardenas-Cartagena, Juan, 120
Cattan, Arie, 138
Cho, Ye-eun, 298

Dagan, Ido, 138
Datla, Vivek, 184
Do, Timothy, 273
Durai, Aneesh, 11

Ellinger, Lukas, 229

F. P. Dossou, Bonaventure, 287
Fokkens, Antske, 66
Fu, Yicheng, 273
Fukuda, Ken, 179

Garces Arias, Esteban, 36
Groh, Georg, 229
Groot, Tobias, 73

He, Yunzhen, 309
Heumann, Christian, 36
Huang, Zhiqi, 184

Ilia, Evgenia, 73

J. Witbrock, Michael, 61
Jeyaganthan, Athikash, 273
Joty, Shafiq, 322

Kamsteeg, Iris, 120
Kanjirangat, Vani, 208
Kumar, Anoop, 184
Kunitomo-Jacquin, Lucie, 179

L. Griffiths, Thomas, 194
Lacunes, Salo, 73
Landowska, Alina, 17
Liu, Daben, 184

Machcha, Sravanthi, 27
Maeng, Yunho, 298
Manocha, Dinesh, 89
Marrese-Taylor, Edison, 179
Meng, Rui, 322
Misikir Tashu, Tsegaye, 120
Mohammad Khalid, Haziq, 273

Nachshoni, Eviatar, 138
Ngonga Ngomo, Axel-Cyrille, 259
Niculae, Vlad, 66

O'Brien, Sean, 273

Padriac Amato Tahua O'Leary, Dr, 61
Partalidou, Eleni, 200
Passali, Tatiana, 200
Pistotti, Timothy, 61
Podolak, Jakub, 247

Radu, Stefania, 103
Rodemann, Julian, 36

Samuel, Alfy, 184
Saparina, Irina, 66
Seth, Ashish, 89
Shapira, Ori, 138
Sharma, Arnab, 259
Sharma, Vasu, 273
Shelmanov, Artem, 1
Shimodaira, Hidetoshi, 309
Skorski, Maciej, 17
Snigdha Sarathi Das, Sarkar, 160
Soni, Ritesh, 184
Sultana, Sharmin, 27

Takase, Yusuke, 309
Toney, Autumn, 51
Troshin, Sergey, 66
Tsoumakas, Grigorios, 200
Tu, Lifu, 322

Valdenegro-Toro, Matias, 103, 120
van Beers, Floris, 120
Verma, Rajeev, 247

Vollmers, Daniel, 259

Wails, Ryan, 51
William Lithgow-Serrano, Oscar, 208

Yao, Zonghai, 27
Yavuz, Semih, 322
Yerra, Sushrita, 27
yu, hong, 27

Zerva, Chrysoula, 200
Zhang, Liyi, 194
Zhang, Rui, 160
Zhang, Yusen, 160
Zhou, Yingbo, 322
Zhu, Chenyang, 184
Zhu, Kevin, 273
Zullich, Marco, 103