

WiNLP 2026

The 10th Widening NLP Workshop

Proceedings of the Workshop

October 28, 2026

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

Platinum



©2026 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 979-8-89176-472-9

Introduction

Welcome to the 2026 Widening NLP (WiNLP) Workshop!

We are delighted to present this year’s proceedings and to welcome our community to another edition of Widening NLP. The workshop provides a space for researchers from underrepresented backgrounds to share their work, exchange ideas, and build connections across the Natural Language Processing (NLP) community.

Widening NLP grew out of a conversation at ACL 2016 in Berlin, where a small group of researchers came together to discuss the underrepresentation of women and other minorities in NLP. That conversation led to the inaugural Workshop for Women and Underrepresented Minorities in NLP at ACL 2017. Since then, the workshop has continued to evolve, guided by the conviction that a stronger NLP community depends on making room for a wider range of voices, experiences, and perspectives.

This commitment has shaped both the scope and the structure of the workshop. Following the inaugural event, we introduced two submission deadlines, including an early deadline to give authors additional time for visa applications. In 2019, we broadened our emphasis on diversity to encompass scientific and disciplinary backgrounds, educational pathways, training, and career stages. We also introduced a peer feedback system, allowing authors to receive constructive input from colleagues before formal review. Together, these efforts have sought to reduce barriers to participation and create a supportive environment for sharing research.

The 2026 program continues this tradition. This year, WiNLP places particular emphasis on accessibility and disability inclusion. We also continue to welcome researchers from diverse scientific and disciplinary backgrounds and training pathways, as well as research on underrepresented languages. We received 67 submissions and accepted 16 archival and 9 non-archival papers.

We thank our authors, reviewers, invited speakers, panelists, and participants for the time, care, and expertise they have contributed to this workshop. We hope that the research and conversations shared here will inspire new ideas, strengthen collaborations, and support continued efforts toward a more inclusive and welcoming NLP community.

-Chen, Emily, Meryem, Peerat, Richard, Santosh, Surendrabikram, Yinqiao, Organizing Co-Chairs

Organizing Committee

Organizing Chairs

Chen Zhang, NII
Emily Allaway, University of Edinburgh
Meryem M'hamdi, Meta
Peerat Limkonchotiwat, AI Singapore
Richard He Bai, Apple
Santosh T.Y.S.S., Amazon
Surendrabikram Thapa, Virginia Tech, USA
Yinqiao Li, City University of Hong Kong

Advisory Board

Alham Fikri Aji, MBZUAI & Google Research
Helena Gomez-Adorno, IIMAS, UNAM
Sarvnaz Karimi, CSIRO
Sunayana Sitaram, Microsoft Research India
Viviane Moreira, UFRGS - Brazil
Zeeraq Talat, University of Edinburgh

Program Committee

Program Chairs

Emily Allaway, University of Edinburgh
Richard He Bai, Apple
Yinqiao Li, City University of Hong Kong
Peerat Limkonchotiawat, AI Singapore
Meryem M'hamdi
Santosh T.Y.S.S, Amazon
Surendrabikram Thapa, Virginia Polytechnic Institute and State University
Chen Zhang, National Institute of Informatics

Area Chairs

Emily Allaway, University of Edinburgh
Peerat Limkonchotiawat, AI Singapore
Meryem M'hamdi
Pittawat Taveekitworachai, Nanyang Technological University
Can Udomcharoenchaikit, Vidyasirimedhi Institute of Science and Technology
Chen Zhang, National Institute of Informatics

Reviewers

Giuseppe Abrami, Arturo Argueta, Eleftherios Avramidis

JinYeong Bak, Leslie Barrett, Emanuela Boros

Sky CH-Wang, Rémi Cardon, Seungtaek Choi

Debarati Das, Ernest Davis, Wentao Ding, Saket Dingliwal, Nemanja Djuric

Micha Elsner

Neele Falk, Ge Fan, Alejandro Figueroa

Prakhar Ganesh, Sucheta Ghosh, Venkata S Govindarajan, Navita Goyal, Morgan Grobol, Zhen Guo

Kazuma Hashimoto, Zhe Hu, Minghui Huang

Hitoshi Isahara

Rishabh Jain

Teja Kanchinadam, Feiyang Kang, Surya Kanoria, Gyuwan Kim, Yeachan Kim, YoungBin Kim, Tracy Holloway King, Katsunori Kotani, Satyapriya Krishna

Stefan Larson, Changmao Li, Chong Li, Dongyuan Li, Ruosen Li, Siyan Li, Jasy Suet Yan Liew,

Gilbert Lim, Ke Lin, Lucy H. Lin, Xinyuan Lu, Jixiang Luo, Yiran Lawrence Luo, Pedro Henrique Luz de Araujo

Harsh Maheshwari, Magdalena Markowska, Marcos Martínez Galindo, John Philip McCrae, Guillaume Metzler, Hideya Mino, Taro Miyazaki, Masayasu Muraoka

Youyang Ng, Kiem-Hieu Nguyen, Dmitry Nikolaev, Sergiu Nisioi, Tadashi Nomoto

Naoki Otani

Aline Paes, Eunhwan Park, Kunwoo Park, Alicia Parrish, Van-Thuy Phi, Aidan Pine, Rajesh Piryani, Sukannya Purkayastha

Priya Rani, Matīss Rikters, Christophe Rodrigues, Angelika Romanou

Fatiha Sadat, Rita Sevastjanova, Quan Z. Sheng, Ning Shi, Kazutoshi Shinoda, Patrick Simianer, Hyun-Je Song, Katherine Stasaski

Khanh Quoc Tran, Dimitrios Tsarapatsanis

Can Udomcharoenchaikit

Sowmya Vajjala, Aditya Srikanth Veerubhotla, Thuy-Trang Vu

Ruibo Wang, Yaqiang Wang, Chen Wu

Kaige Xie, Jinan Xu

Changbing Yang, Mengxia Yu

Alessandra Zarcone, Runzhe Zhan, Chen Zhang, Ningyu Zhang, Zequn Zhang, Zhenjie Zhao, Junwei Zhou, Yi Zhou

Table of Contents

<i>Tug-of-War Tuning: Capacity and Cross-Lingual Transfer in Zero-Shot NER</i> Tanushree Ravindra Pratap Yadav	1
<i>Beyond Calibration: Temporal LLM Confidence Assessment Across Languages and Local Settings</i> Sahil Kale	10
<i>AmharicStoryQA: A Multicultural Story Question Answering Benchmark in Amharic</i> Israel Abebe Azime, Abenezer Angamo, Hana Mekonen Tamiru, Dagnachew Mekonnen Marilign, Philipp Slusallek, Seid Muhie Yimam and Dietrich Klakow	20
<i>What Improves Multimodal Misinformation Detection? Answers from a Large-Scale Empirical Study</i> Akshit Sharma and Prashant W Patil	37
<i>How Well Does It Generalize? A Robustness Study of Hausa Machine-Generated Text Detection</i> Abdulhamid Abubakar, Faisal Muhammad Adam, Shamsuddeen Hassan Muhammad, Abubakar Auwal Khalid and Babangida Sani	48
<i>The Energy-Accuracy Frontier: Benchmarking SNNs on Low-Resource Indian Linguistic Tasks</i> Priyam Saha, Debiprasanna Sahoo and Abhik Jana	61
<i>AmhTTS: A Low-Resource Amharic Text-to-Speech System</i> Walelign Tewabe Sewunetie, Rahel Mekonen Tamiru and Israel D. Gebru	68
<i>Marker Signal Void: Linguistic Deception Fingerprinting and Transformer Benchmarking on a Domain-Stratified Bangla Rumour Corpus</i> Farzana Zahid Manoshi, Sadman Bin Rashid, Israt Jahan Anika and Md. Rakibul Islam	77
<i>Does Live Browser Context Improve LLM-Assisted End-to-End Test Script Repair? An Empirical Study of Playwright MCP</i> Divi Galih Prasetyo Putri	87
<i>Assessing Large Language Models as Automated Writing Evaluation and Feedback Tools: A Human-AI Agreement Study on L2 English Writing</i> Kowshik Saha, Moyeen Uddin, Md Tanvirul Islam Turad and Mizanur Rahman	96
<i>Improved Personalized Typing Predictions via Long Left Context</i> Soufia Bahmani and Keith Vertanen	105
<i>Do Multilingual Encoders Produce Language-Consistent Semantic IDs?</i> Abhinav Bohra and Anuj Bohra	121
<i>When a Name Is Not a Name: A Benchmark Dataset and Distilled Reasoning for Culturally Entangled Bangla Homographs in Low-Resource LLMs</i> Md. Asaduzzaman Shuvo and Md. Adnanul Islam	128
<i>ASCRIBE: Atomic and Significance-Based Reasoning for Thai Clinical SOAP Note Generation</i> Tarm Kalavantavanich, Teerawut Ponarchar, Pattaramanee Arsomngern, Jenta Wonglertsakul, Watcharakorn Chuthong, Chiraphat Boonnag, Knot Pipatsrisawat and Titipat Achakulvisut	141
<i>KuMMLU: A Multitask Language Understanding Benchmark for Central Kurdish</i> Akam Nawzad	173
<i>A Theoretical Model of Hallucinations in NLP Large Language Models</i> Vicente Ivan Sanchez Carmona, Yuming Zhang, Junyu Zhou, Shanshan Jiang and Bin Dong	181

Tug-of-War Tuning: Capacity and Cross-Lingual Transfer in Zero-Shot NER

Tanushree Ravindra Pratap Yadav

IISER Bhopal

yadav23@iiserb.ac.in

Abstract

We investigate how the amount of fine-tuned capacity affects zero-shot cross-lingual transfer for named entity recognition (NER). Training exclusively on English WikiANN and evaluating on 14 typologically diverse target languages, we find a consistent *inverted-U* pattern: cross-lingual transfer is worst at both ends of the parameter-efficiency spectrum. Full fine-tuning, which updates all parameters, transfers worst overall and collapses on non-Latin-script languages, while the most parameter-efficient methods, (IA)³ and prefix tuning, also underperform. Methods with an intermediate number of trainable parameters (Low-Rank Adaptation (LoRA; Hu et al., 2022) and Housby adapters) transfer best, with LoRA outperforming full fine-tuning by roughly 7 F1 points on average and by over 10 points on languages such as Chinese and Japanese. We find the same inverted-U shape within LoRA alone: a sweep over ranks shows cross-lingual transfer peaking at an intermediate rank, distinct from the higher rank that maximises in-language English performance, revealing a rank-transfer tradeoff. We also examine, and do not find clean support for, the common assumption that better preservation of cross-lingual representational alignment explains PEFT’s transfer benefits: the method that best preserves a standard alignment measure transfers worst, while LoRA’s larger alignment shift coincides with its best transfer. Given known limitations of this alignment measure, we present this as a preliminary, metric-dependent observation that motivates further analysis with more robust alignment metrics, rather than a settled claim.

1 Introduction

Cross-lingual transfer from high-resource to low-resource languages is a central challenge in multilingual NLP (Pires et al., 2019; Wu and Dredze, 2019). A standard approach fine-tunes a multilingual pre-trained model on a source language and

evaluates zero-shot on target languages (Conneau et al., 2020). Full fine-tuning updates every weight in the model and is known to cause *catastrophic forgetting* (McCloskey and Cohen, 1989) of multilingual structure, degrading transfer especially to typologically and orthographically distant languages.

Parameter-efficient fine-tuning (PEFT) methods, including LoRA (Hu et al., 2022), Housby adapters (Housby et al., 2019), (IA)³ (Liu et al., 2022), and prefix tuning (Li and Liang, 2021), update only a small fraction of parameters and are often motivated as a way to preserve multilingual structure during fine-tuning (Ansell et al., 2022; Üstün et al., 2022). This suggests a natural hypothesis: the fewer parameters a method updates, the better it should preserve cross-lingual transfer. Our results complicate this picture in two ways.

First, transfer does not improve monotonically as fine-tuned capacity shrinks. Across five methods spanning nearly the full parameter-efficiency spectrum, we observe an inverted-U: full fine-tuning (100% of parameters) transfers worst, the most parameter-efficient methods, (IA)³ and prefix tuning, also underperform, and two methods with an intermediate number of trainable parameters, LoRA and Housby adapters, transfer best. We find the identical pattern within LoRA alone: sweeping the rank r , which controls how many parameters LoRA updates (a lower r means fewer trainable parameters and a more constrained, more parameter-efficient update, while a higher r allows a richer but less constrained one), shows cross-lingual transfer rising with r up to a point and then declining, at a lower rank than the one that maximises English-only performance.

Second, we ask whether a standard proxy for “preserving multilingual structure” (cosine similarity between source- and target-language representations) actually predicts this pattern. It does not: the method with the smallest average change in this

measure, prefix tuning, has the worst transfer, while LoRA, which shows the largest change, has the best. Taken at face value, this would suggest that methods which perturb representations more are, if anything, more transfer-preserving: the opposite of what the alignment-preservation motivation for PEFT would predict. However, mean-pooled cosine similarity is known to be a coarse, anisotropy-confounded proxy for cross-lingual structure (Cai et al., 2021; Vulić et al., 2020), so we treat this second finding as suggestive rather than conclusive, and offer a subspace-based hypothesis (§4.3, Appendix A) for why a method could shift this metric substantially while still preserving the narrower set of directions that matter for the NER task.

We study both patterns on WikiANN (Pan et al., 2017) (15 languages, 9 language families) using XLM-RoBERTa base (Conneau et al., 2020). Our contributions are:

- A controlled, three-seed comparison of five fine-tuning methods spanning the parameter-efficiency spectrum, showing a consistent inverted-U relationship with zero-shot cross-lingual NER transfer;
- A rank sweep over LoRA ($r \in \{1, \dots, 768\}$) showing the same inverted-U shape within a single method, with the optimal rank for cross-lingual transfer lower than the optimal rank for the English source language;
- A preliminary examination of whether representational alignment (via mean-pooled cosine similarity) explains these patterns, finding that it does not straightforwardly, along with a subspace-decomposition account for why a coarse alignment metric may be a poor predictor of task-relevant transfer.

2 Background and Related Work

PEFT for cross-lingual transfer. Prior work motivates PEFT as beneficial for cross-lingual transfer by limiting how much a fine-tuning step can disturb pretrained multilingual representations: Üstün et al. (2022) found Housby adapters outperform full fine-tuning for low-resource dependency parsing transfer, and Pfeiffer et al. (2020) proposed MAD-X, which composes language- and task-specific adapters to keep cross-lingual representations largely intact. Ansell et al. (2022) pursue a similar goal via composable sparse fine-tuning. These works establish that some PEFT methods

outperform full fine-tuning cross-lingually, but do not compare across a range of parameter budgets or examine whether increasingly parameter-efficient methods keep improving transfer. Lialin et al. (2023) and Chen et al. (2022) study how LoRA’s rank affects performance, but in monolingual settings; we are not aware of prior work characterising a rank-transfer tradeoff specifically for cross-lingual transfer. Yadav (2026) report related evidence of representational collapse in code-mixed generation, consistent with the idea that both too much and too little constraint on fine-tuning can distort multilingual representations. BitFit (Zaken et al., 2022), Compacter (Mahabadi et al., 2021), and AdaLoRA (Zhang et al., 2023) occupy additional points on the PEFT parameter-efficiency spectrum; AdaLoRA’s rank-pruning objective, which adaptively reallocates rank across layers, is a natural comparison against our fixed-rank sweep (§5).

Representational alignment and its limitations.

A common way to quantify how much fine-tuning disturbs cross-lingual structure is mean-pooled cosine similarity between source- and target-language sentence representations (Wu et al., 2020; Zhao et al., 2021). This measure has known weaknesses, however: retrieval-based metrics such as BLI precision@1 and CSLS (Conneau et al., 2018) correlate more reliably with downstream cross-lingual performance (Vulić et al., 2020; Ravishankar et al., 2019), and CKA (Kornblith et al., 2019) is more robust to isotropic scaling. Most importantly for our analysis, Cai et al. (2021) show that mean-pooled cosine similarity is confounded by representation anisotropy, so a high similarity score can reflect a general “cone collapse” of the representation space rather than genuine cross-lingual alignment. We use mean-pooled cosine similarity in §4.3 for direct comparability with this prior work, but given these confounds, we treat any conclusions drawn from it as preliminary (see Limitations).

3 Experimental Setup

Model and data. We fine-tune the pretrained XLM-R base checkpoint (Conneau et al., 2020) (125M parameters; this checkpoint has not been previously fine-tuned on any NER data) on the English portion of WikiANN (Pan et al., 2017) (entity types PER, ORG, LOC) and evaluate zero-shot, with no target-language supervision, on 14 additional languages: ES, FR, DE, ZH, AR, HI, RU,

JA, FI, SW, TR, KO, NL, VI, spanning 9 language families and 5 writing scripts.

Fine-tuning methods. We compare five methods that update different amounts of the underlying model, presented from most to least trainable parameters. **Full fine-tuning** updates every weight matrix in the model directly. **LoRA** (Hu et al., 2022) instead freezes a pretrained weight matrix $\mathbf{W}_0 \in \mathbb{R}^{d \times k}$ (with d and k the input and output dimensions of a given projection) and learns a low-rank update:

$$\begin{aligned} \mathbf{W} &= \mathbf{W}_0 + \mathbf{B}\mathbf{A}, \\ \mathbf{B} &\in \mathbb{R}^{d \times r}, \mathbf{A} \in \mathbb{R}^{r \times k}, r \ll \min(d, k) \end{aligned} \quad (1)$$

The rank r is a hyperparameter that controls how many parameters LoRA updates: a lower r means fewer trainable parameters (a more constrained, more parameter-efficient update), while a higher r allows a richer, less constrained update that approaches full fine-tuning as $r \rightarrow \min(d, k)$. We apply LoRA to the query and value projections of all 12 attention layers ($r=16$, $\alpha=32$, dropout 0.1; 0.60M trainable parameters, 0.47% of the model). Key and feed-forward projections are frozen; layer-wise ablations are underway (§5).

Houlsby adapters (Houlsby et al., 2019) insert bottleneck modules after *both* the self-attention and feed-forward sub-layers in all 12 layers, transforming a hidden state $\mathbf{h} \in \mathbb{R}^d$ as:

$$\mathbf{h}' = \mathbf{h} + \mathbf{W}_{\text{up}} \sigma(\mathbf{W}_{\text{dn}} \mathbf{h}), \quad (2)$$

with $\mathbf{W}_{\text{dn}} \in \mathbb{R}^{b \times d}$, $\mathbf{W}_{\text{up}} \in \mathbb{R}^{d \times b}$, bottleneck dimension $b=64$, and σ the GeLU nonlinearity. (IA)³ (Liu et al., 2022) rescales hidden states element-wise ($\mathbf{h}' = \mathbf{1} \odot \mathbf{h}$, $\mathbf{1} \in \mathbb{R}^d$) on keys, values, and feed-forward intermediates in all 12 layers (0.030M trainable parameters, 0.024%). **Prefix tuning** (Li and Liang, 2021) prepends $m=10$ trainable tokens to each layer’s key/value attention while keeping all original weights frozen.

Training details. All methods are trained with AdamW (Loshchilov and Hutter, 2019), a 10% warmup schedule, weight decay 0.01, for 3 epochs with batch size 32. We select the learning rate per method based on English validation F1: 2×10^{-4} for full fine-tuning, LoRA, and Houlsby adapters; 3×10^{-3} for (IA)³; 1×10^{-2} for prefix tuning. All results are reported as mean $\pm$ std over three random seeds ($\{42, 123, 456\}$); we use $n=3$ throughout to refer to this seed count. Significance is assessed

with paired t -tests ($df=2$) per language; given the exploratory scope and small n , we do not apply a multiple-comparison correction, and we treat significance at this sample size as suggestive rather than confirmatory.

Alignment measurement. Following Wu et al. (2020), for each target language $\mathcal{L}$ we compute alignment as the mean cosine similarity between ℓ_2 -normalised, mean-pooled representations of $N=1,000$ FLORES-200 (NLLB Team et al., 2022) devtest sentence pairs (the same sample is used for every method). For each sentence, we mean-pool the final transformer layer’s hidden states over non-padding tokens, then ℓ_2 -normalise; we use only this final layer throughout, so no per-layer index is needed. Writing $\mathbf{h}_i^{\text{EN}}$ and $\mathbf{h}_i^{\mathcal{L}}$ for the resulting representations of the i -th sentence pair ($i = 1, \dots, N$) in English and language $\mathcal{L}$:

$$\text{Align}(\mathcal{L}) = \frac{1}{N} \sum_{i=1}^N \frac{\mathbf{h}_i^{\text{EN}} \cdot \mathbf{h}_i^{\mathcal{L}}}{\|\mathbf{h}_i^{\text{EN}}\| \|\mathbf{h}_i^{\mathcal{L}}\|} \quad (3)$$

We define alignment degradation as $\Delta_{\mathcal{L}} = \text{Align}_{\text{base}}(\mathcal{L}) - \text{Align}_{\text{ft}}(\mathcal{L})$, where $\text{Align}_{\text{base}}(\mathcal{L})$ is computed on the pretrained model before any fine-tuning and $\text{Align}_{\text{ft}}(\mathcal{L})$ after fine-tuning with a given method. Under this definition, $\Delta_{\mathcal{L}} > 0$ indicates that fine-tuning *reduced* alignment relative to the pretrained model, so a larger $\Delta_{\mathcal{L}}$ means worse preservation, and a value near 0 means alignment was essentially unchanged. No anisotropy correction is applied; absolute alignment values near 0.99 are expected under this uncorrected measure (Wu et al., 2020), and we only interpret the relative $\Delta_{\mathcal{L}}$ across methods, not its absolute magnitude. Adapter alignment measurement is ongoing and excluded from Table 3.

4 Results

4.1 Main Transfer Results: A Parameter-Efficiency Sweet Spot

Table 1 reports span-level F1. Ordering the five methods by how many parameters they update (full fine-tuning (100%) > Houlsby adapters > LoRA (0.47%) > (IA)³ (0.024%) > prefix tuning), mean cross-lingual F1 traces an inverted U rather than a monotonic trend: full fine-tuning transfers worst (51.8%), performance rises through Houlsby adapters (58.6%) and peaks at LoRA (58.9%, a ~ 7.1 pp gap over full fine-tuning), then falls again

L	Fam.	FT	Adp	LoRA	IA ³	Pfx
en	Gmc	80.5	77.3	76.2	75.9	74.7
es	Rom	62.2	69.2	69.9	65.4	61.9
fr	Rom	71.1	72.7	73.0	70.4	65.6
de†	Gmc	67.3	71.3	71.2	70.1	66.0
nl†	Gmc	75.1	76.7	76.8	76.4	72.6
ru	Slv	55.4	58.6	58.0	57.0	46.3
fi†	Ura	70.1	72.0	71.4	71.9	66.9
sw	N-C	62.0	66.5	66.8	63.6	62.3
tr†	Trk	61.6	70.6	72.6	70.4	61.3
vi	A-A	60.9	63.0	65.5	60.2	60.6
zh†	S-T	13.1	26.1	26.9	21.5	19.6
ar	Sem	31.5	42.1	42.6	48.6	39.4
hi†	Ind	52.5	64.8	61.8	62.2	57.9
ja†	Jpn	10.2	20.2	21.4	16.4	14.6
ko†	Kor	32.7	46.3	46.1	46.4	41.0
Avg	—	51.8	58.6	58.9	57.2	52.6

Table 1: Span-level F1 (%) on WikiANN zero-shot NER, mean over 3 seeds (full std in Appendix C). † = LoRA significantly outperforms full FT ($p < 0.05$). The 0.3 pp LoRA–adapter gap is not statistically significant. Bold = best per row. Fam.: Gmc Germanic, Rom Romance, Slv Slavic, Ura Uralic, N-C Niger-Congo, Trk Turkic, A-A Austroasiatic, S-T Sino-Tibetan, Sem Semitic, Ind Indic, Jpn Japonic, Kor Koreanic.

through (IA)³ (57.2%) to prefix tuning, the second-worst (52.6%). LoRA significantly outperforms full fine-tuning ($p < 0.05$, paired t -test) in 8 of the 14 target languages, concentrated in typologically distant and non-Latin-script languages. The gap between LoRA and Housby adapters (0.3 pp) is small relative to seed variance and not significant at $n=3$.

Full fine-tuning collapses most severely on non-Latin-script languages: Chinese reaches only $13.1 \pm 3.6\%$ F1 under full fine-tuning versus $26.9 \pm 0.5\%$ under LoRA, and Japanese only $10.2 \pm 2.4\%$ versus $21.4 \pm 1.3\%$. Full fine-tuning’s high variance relative to LoRA’s stability on these languages suggests a high-variance, low-performance regime produced by unconstrained updates interacting with scripts that are less represented during pretraining.

This advantage is not limited to non-Latin scripts, however: Turkish, which uses the Latin script but is morphologically rich and typologically distant from English (agglutinative morphology, flexible word order), shows one of the largest gaps between LoRA and full fine-tuning (72.6% vs. 61.6%, +11.0 pp). This suggests the collapse under full parameter updates tracks typological distance more broadly, not simply script mismatch,

a distinction we did not draw clearly in an earlier draft, which grouped Turkish together with Chinese and Japanese under a single “script-distant” label.

At the other end of the spectrum, (IA)³, despite updating far fewer parameters than LoRA, outperforms LoRA on Arabic and Hindi, suggesting that for some language families, rescaling-only updates preserve relevant features about as well as, or better than, a low-rank update. Housby adapters are the most stable method overall (average std ± 0.74) and outperform LoRA on Hindi (64.8 vs. 61.8).

4.2 A Rank–Transfer Tradeoff Within LoRA

If the inverted-U pattern above reflects fine-tuned capacity broadly, rather than something specific to comparing different methods, the same shape should appear when varying capacity within a single method. Figure 1 shows English and cross-lingual F1 as a function of LoRA rank $r \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 768\}$, together with the corresponding trainable parameter count on a secondary axis. Both F1 curves rise together up to $r=64$; beyond that point they diverge. English F1 continues rising to its maximum at $r=768$ (81.3%), the highest-capacity setting we test, while cross-lingual F1 peaks earlier, at $r=256$ (61.2%), and then declines:

$$\begin{aligned} r_{\text{EN}}^* &= \arg \max_r \mathcal{F}_{\text{EN}}(r) = 768 \\ r_{\text{xling}}^* &= \arg \max_r \mathcal{F}_{\text{xling}}(r) = 256 \end{aligned} \quad (4)$$

This mirrors §4.1: just as the highest-capacity method (full fine-tuning) transfers worse than an intermediate-capacity one (LoRA), the highest-rank LoRA setting transfers worse than an intermediate rank. The decline beyond $r=256$ is concentrated in the same typologically distant languages implicated above: at $r=768$, Arabic falls from 50.6 (at $r=8$) to 39.4, and Chinese falls from 31.1 (at $r=256$) to 23.1, the same failure pattern seen in full fine-tuning. Full numeric results across all eleven ranks are provided in Appendix B (Table 4); Table 5 confirms the decline is consistent across seeds for four representative ranks, not a single-run artefact, with per-rank standard deviations (≤ 0.8 pp) well below the 2.5 pp gap between $r=256$ and $r=768$ (Fig. 1).

Practically, this means the rank that is best for the source language is not the rank that is best for cross-lingual transfer: a practitioner tuning LoRA rank

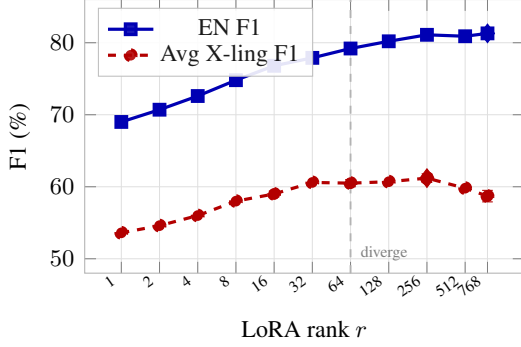


Figure 1: EN F1 and average cross-lingual F1 vs. LoRA rank r ($\log_2$ scale). Diamonds mark the optima ($r_{\text{xling}}^*=256$, $r_{\text{EN}}^*=768$). Error bars (± 1 std over 3 seeds; Table 5) are shown at the four seed-verified ranks; the $r=256$ -to- $r=768$ cross-lingual decline exceeds this seed variability. The trainable parameter count (Table 4) grows monotonically with rank, making clear that the cross-lingual optimum is not the highest-capacity setting available.

purely on English validation performance would select $r=768$ and lose 2.5 points of cross-lingual F1 relative to $r=256$.

4.3 Does Representational Alignment Explain This Pattern?

A natural explanation for the inverted-U in §4.1 would be that intermediate-capacity methods better preserve cross-lingual representational alignment, and that this preserved alignment is what drives better transfer. Table 2 tests this directly using the mean-pooled cosine alignment measure from §3 (we report full fine-tuning, LoRA, (IA)³, and prefix tuning; Houlsby adapter alignment is not yet computed, see Limitations).

Method	$\bar{\Delta} \downarrow$	Xling F1 $\uparrow$
Prefix	0.005	52.6
(IA) ³	0.038	57.2
Full FT	0.055	51.8
LoRA	0.061	58.9

$\downarrow$ =better preserved $\uparrow$ =better transfer

Table 2: Alignment degradation vs. cross-lingual transfer, averaged across languages.

This explanation does not hold cleanly. If anything, alignment preservation and transfer are inversely related across these four methods: prefix tuning preserves alignment almost perfectly ($\bar{\Delta}=0.005$) yet has the second-worst transfer, while LoRA shows the largest average shift ($\bar{\Delta}=0.061$) yet transfers best. Table 3 (per-language) shows this holds language by language, including for Chinese, where LoRA shifts alignment more than full

fine-tuning does (0.075 vs. 0.048) while still outperforming it by 13.7 pp in F1.

Lang	FT	LoRA	IA ³	Pfx
es	.038	.045	.046	.003
fr	.030	.050	.037	.003
de	.048	.050	.032	.004
zh	.048	.075	.042	.005
ar	.074	.064	.047	.007
hi	.075	.063	.037	.007
ru	.035	.063	.031	.005
ja	.065	.069	.035	.005
fi	.065	.067	.037	.006
sw	.074	.072	.046	.006
tr	.082	.064	.037	.006
ko	.082	.069	.037	.007
nl	.026	.046	.036	.004
vi	.035	.054	.028	.005
Avg $\bar{\Delta}$	.055	.061	.038	.005

Table 3: Alignment degradation $\Delta_{\mathcal{L}}$ (Eq. 3). Adapter scores not yet computed. Lower = better preserved. No anisotropy correction; see §2–3.

We do not read this as evidence that disturbing representational alignment *helps* transfer. Mean-pooled cosine similarity is a coarse, whole-representation measure known to be confounded by representation anisotropy: a change in this score can reflect a broad rotation or rescaling of the representation space that has little to do with the narrower set of directions a downstream task actually relies on (Cai et al., 2021). Appendix A sketches one way this could happen: if a method’s update is concentrated outside the task-relevant subspace, it can shift mean-pooled cosine substantially while leaving task-relevant directions largely intact: LoRA’s low-rank update is a natural candidate, since by construction it can only act within an r -dimensional subspace of the representation space. Conversely, a method that freezes all weights (prefix tuning) trivially preserves the whole-representation measure but also cannot re-route task-relevant directions if they need adjusting.

We want to be careful about how much weight this analysis can bear. Because our alignment measure has known limitations, we cannot rule out that a more targeted metric (CKA restricted to task-relevant directions, or retrieval-based measures like CSLS) would show a relationship more consistent with the standard alignment-preservation account. We therefore present this analysis as a preliminary, metric-dependent observation and a motivation for the subspace hypothesis in Appendix A, rather than as a robust finding that alignment preservation is unrelated to, or actively harmful for, transfer.

5 Discussion

Practical guidance. For cross-lingual NER with XLM-R base, our results suggest treating parameter efficiency as a dial with a sweet spot, rather than a scale where “more efficient is always safer”: both full fine-tuning and the most parameter-efficient methods we test underperform intermediate-capacity methods. Within LoRA specifically, rank $r \in [32, 256]$ gives the best transfer, with $r \leq 64$ preferable when the deployment targets typologically distant languages such as Chinese, Japanese, Korean, or Turkish, where the gap between LoRA and full fine-tuning is largest and most stable. Houlsby adapters are a reasonable alternative when rank tuning is impractical, given their stability across languages (58.6% average, std ± 0.74).

Limitations. Our central inverted-U finding rests on five methods and one encoder (XLM-R base) with English as the sole source language; whether it holds for larger encoders, decoder-only models, or other source languages is future work. Because LoRA, Houlsby adapters, (IA)³, prefix tuning, and full fine-tuning differ in *where* and *how* they modify the model (Eq. 1, Eq. 2), not only in how many parameters they update, our comparison confounds trainable-parameter count with adaptation architecture, and parameter count alone cannot be identified as the sole driver of the inverted-U pattern. WikiANN is silver-annotated, and we plan to validate our main comparison on MultiCoNER and UNER v2. Adapter alignment scores are not yet computed. Most importantly, our alignment analysis (§4.3) relies on mean-pooled cosine similarity, a measure we ourselves note is confounded by representation anisotropy and less predictive of downstream transfer than retrieval-based or CKA-based alternatives; we chose it for direct comparability with prior work using the same measure (Wu et al., 2020), but this means our alignment-related conclusions should be read as tentative and specific to this metric, not as a general claim about the relationship between representational alignment and cross-lingual transfer. We do not evaluate BitFit, Compacter, or AdaLoRA; AdaLoRA’s rank-pruning mechanism is a natural point of comparison against our fixed-rank sweep.

Future work. Layer-wise ablations (zeroing LoRA’s update per layer and measuring the resulting drop in cross-lingual F1) are underway; pre-

liminary results suggest layers 0–5 carry disproportionate transfer signal for typologically distant languages, which would directly test the subspace hypothesis in Appendix A. We plan to compute CKA restricted to task-relevant directions (Kornblith et al., 2019) and retrieval-based diagnostics (CSLS, BLI P@1; Vulić et al., 2020; Conneau et al., 2018) to assess whether a more targeted alignment measure better predicts the inverted-U pattern than mean-pooled cosine similarity, and we will extend all alignment measurements to adapters.

6 Conclusion

We present a controlled, five-way comparison of fine-tuning methods spanning the parameter-efficiency spectrum for zero-shot cross-lingual NER across 14 languages. Rather than transfer improving monotonically as fine-tuned capacity shrinks, we find an inverted-U: full fine-tuning and the most parameter-efficient methods, (IA)³ and prefix tuning, both underperform intermediate-capacity methods (LoRA, Houlsby adapters), with LoRA significantly outperforming full fine-tuning in 8 of 14 languages, especially typologically and orthographically distant ones. A rank sweep shows the same shape within LoRA alone, revealing a rank–transfer tradeoff in which the rank that maximises English performance is not the rank that maximises cross-lingual transfer. We additionally examine, using a standard but limited alignment metric, whether representational alignment preservation explains this pattern, and find that it does not straightforwardly, a preliminary result we discuss cautiously given the metric’s known confounds, alongside a subspace-based hypothesis for reconciling it with the main inverted-U finding.

References

- Alan Ansell, Edoardo M. Ponti, Anna Korhonen, and Ivan Vulić. 2022. Composable sparse fine-tuning for cross-lingual transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, pages 1778–1796. Association for Computational Linguistics.
- Xingyu Cai, Jiaji Huang, Yuchen Bian, and Kenneth Church. 2021. *Isotropy in the contextual embedding space: Clusters and manifolds*. In *Proceedings of ICLR*.
- Guanzheng Chen, Fangyu Liu, Zaiqiao Meng, and Shangsong Liang. 2022. Revisiting parameter-efficient tuning: Are we really there yet? In *Proceedings of the 2022 Conference on Empirical Methods*

- in *Natural Language Processing*, pages 2612–2626. Association for Computational Linguistics.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451. Association for Computational Linguistics.
- Alexis Conneau, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. [Word translation without parallel data](#). In *Proceedings of ICLR*.
- Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2790–2799. PMLR.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3519–3529. PMLR.
- Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 4582–4597. Association for Computational Linguistics.
- Vladislav Lialin, Vijeta Deshpande, and Anna Rumshisky. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.
- Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohata, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*, volume 35.
- Ilya Loshchilov and Frank Hutter. 2019. [Decoupled weight decay regularization](#). In *Proceedings of ICLR*.
- Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34.
- Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.
- NLLB Team and 1 others. 2022. [No language left behind: Scaling human-centered machine translation](#). *Preprint*, arXiv:2207.04672.
- Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1946–1958. Association for Computational Linguistics.
- Jonas Pfeiffer, Ivan Vulić, Iryna Gurevych, and Sebastian Ruder. 2020. MAD-X: An adapter-based framework for multi-task cross-lingual transfer. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 7654–7673. Association for Computational Linguistics.
- Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001. Association for Computational Linguistics.
- Vinit Ravishankar and 1 others. 2019. [Multilingual embeddings for low-resource transfer in ASR](#). In *Proceedings of NAACL-HLT*. Association for Computational Linguistics.
- Ahmet Üstün, Miryam de Lhoneux, Joakim Nivre, and Arzucan Özgür. 2022. UDapter: Language adaptation for truly Universal dependency parsing. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7302–7315. Association for Computational Linguistics.
- Ivan Vulić and 1 others. 2020. [Probing pretrained language models for lexical semantics](#). In *Proceedings of EMNLP*. Association for Computational Linguistics.
- Shijie Wu, Alexis Conneau, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034. Association for Computational Linguistics.
- Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 833–844. Association for Computational Linguistics.
- Tanushree Ravindra Pratap Yadav. 2026. [Competence collapse in code-mixed generation: Spectral evidence and mechanistic recovery via cross-lingual activation steering](#). In *Proceedings of the Second Workshop on Language Models for Low-Resource Languages (LoResLM 2026)*, pages 29–40, Rabat, Morocco. Association for Computational Linguistics.
- Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. 2022. [BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models](#). In *Proceedings of ACL*. Association for Computational Linguistics.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. 2023. [AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning](#). In *Proceedings of ICLR*.

Mengjie Zhao, Yi Zhu, Ehsan Shareghi, Ivan Vulić, Roi Reichart, Anna Korhonen, and Hinrich Schütze. 2021. A closer look at few-shot crosslingual transfer: The choice of shots matters. *arXiv preprint arXiv:2012.15682*.

Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, and Tom Goldstein. 2021. GradInit: Learning to initialize neural networks for stable and efficient training. In *Advances in Neural Information Processing Systems*, volume 34.

A A Subspace Account of the Alignment–Transfer Relationship

We sketch why a method’s effect on global cosine alignment need not track its effect on cross-lingual task performance, and how this could be consistent with the inverted-U pattern in §4.1–§4.2.

Setup. Let $f_\theta : \mathcal{X} \rightarrow \mathbb{R}^d$ be the encoder. Partition the representation space as $\mathbb{R}^d = \mathcal{S}_\tau \oplus \mathcal{S}_\tau^\perp$, where $\mathcal{S}_\tau$ is the *task-relevant subspace* (directions encoding entity-boundary features shared across languages) and $\mathcal{S}_\tau^\perp$ its complement. Any hidden state decomposes as $\mathbf{h} = \mathbf{h}_\tau + \mathbf{h}_\perp$.

Update decomposition. A weight update $\Delta\mathbf{W}$ acts as $\mathbf{h} \mapsto \mathbf{h} + \Delta\mathbf{W}\mathbf{h}$. Decomposing:

$$\Delta\mathbf{W}\mathbf{h} = \underbrace{\Delta\mathbf{W}\mathbf{h}_\tau}_{\text{task subsp.}} + \underbrace{\Delta\mathbf{W}\mathbf{h}_\perp}_{\text{orth. compl.}} \quad (5)$$

Too little capacity: prefix tuning. All original weights are frozen ($\Delta\mathbf{W}=\mathbf{0}$), so $\Delta_{\mathcal{L}}^{\text{pfx}} \approx 0$ (observed: $\hat{\Delta}=0.005$):

$$\Delta_{\mathcal{L}}^{\text{pfx}} = \text{Align}_{\text{base}}(\mathcal{L}) - \text{Align}_{\text{pfx}}(\mathcal{L}) \approx 0. \quad (6)$$

Soft prompts modulate attention but cannot directly re-route directions within $\mathcal{S}_\tau$ if the task requires it, consistent with prefix tuning’s weak transfer despite near-perfect alignment preservation.

Too much capacity: full fine-tuning. Under an isotropy assumption (Zhu et al., 2021), the expected interference of an unconstrained update with the task subspace scales as:

$$\mathbb{E}[\|\Pi_{\mathcal{S}_\tau}\Delta\mathbf{W}\|_F^2] \propto \frac{\dim(\mathcal{S}_\tau)}{d} \quad (7)$$

An unconstrained, full-rank update has no built-in mechanism preventing it from acting on $\mathcal{S}_\tau$, giving it more opportunity to disturb task-relevant directions than a rank-constrained update.

An intermediate regime: LoRA. From Eq. (1), $\Delta\mathbf{W} = \mathbf{B}\mathbf{A}$ has rank r , so $\Delta\mathbf{W}\mathbf{h}$ lies in a subspace of dimension at most $r \ll d$:

$$\Delta\mathbf{W}\mathbf{h} \in \text{col}(\mathbf{B}), \quad \dim(\text{col}(\mathbf{B})) \leq r \ll d. \quad (8)$$

When $\text{col}(\mathbf{B})$ is not aligned with $\mathcal{S}_\tau$, plausible when r is small relative to $\dim(\mathcal{S}_\tau)$, the task component $\mathbf{h}_\tau$ can be largely preserved even though the update induces a large rotation of the complement, which is exactly what a whole-representation measure like mean-pooled cosine similarity picks up:

$$\Delta_{\mathcal{L}}^{\text{LoRA}} \approx \frac{\|\text{proj}_{\text{col}(\mathbf{B})} \bar{\mathbf{h}}^{\text{EN}}\|}{\|\bar{\mathbf{h}}^{\text{EN}}\|} \cdot \alpha_{\mathbf{A}}, \quad (9)$$

where $\bar{\mathbf{h}}^{\text{EN}}$ is the mean English representation and $\alpha_{\mathbf{A}}$ scales with $\mathbf{A}$. If $\text{col}(\mathbf{B})$ aligns with high-variance directions outside $\mathcal{S}_\tau$, $\Delta_{\mathcal{L}}^{\text{LoRA}}$ can be large even when $\mathcal{S}_\tau$ is preserved, and representation anisotropy (Cai et al., 2021) amplifies this, since mean-pooled cosine is dominated by a small number of high-variance directions regardless of their task relevance. This offers one account of why LoRA can show a larger $\hat{\Delta}$ than full fine-tuning while still transferring better: the rank constraint concentrates the update’s effect into a subspace that need not overlap with $\mathcal{S}_\tau$, rather than acting on all of $\mathbb{R}^d$ indiscriminately.

This account also suggests why a sweet spot exists at all: at very low rank (or under prefix tuning/(IA)³’s limited parameterisations), $\text{col}(\mathbf{B})$ or its analogue may be too constrained to include the directions in $\mathcal{S}_\tau$ that genuinely need adjusting, hurting transfer through insufficient capacity; at very high rank (approaching full fine-tuning), the update is no longer confined to a small subspace and has more opportunity to interfere with $\mathcal{S}_\tau$ under the isotropy argument above, hurting transfer through excess capacity. An intermediate rank can thread this needle: expressive enough to adjust the necessary directions in $\mathcal{S}_\tau$, constrained enough to avoid broadly interfering with it.

Proposition 1 (Alignment–Transfer Decoupling). *There exist configurations $(\mathbf{B}, \mathbf{A})$ such that the LoRA update simultaneously (i) increases global cosine degradation beyond full fine-tuning’s ($\Delta_{\mathcal{L}}^{\text{LoRA}} > \Delta_{\mathcal{L}}^{\text{FT}}$) and (ii) better preserves $\mathcal{S}_\tau$, yielding higher cross-lingual F1.*

Remark 1. Proposition 1 formalises why global cosine similarity need not track cross-lingual task

performance, and offers one possible mechanism for the inverted-U pattern in §4.1–§4.2, though we stress (§4.3) that we have not directly measured S_τ or verified this mechanism empirically. CKA (Kornblith et al., 2019) restricted to S_τ would help disentangle global from task-subspace geometry, and retrieval-based metrics (CSLS, BLI P@1) are less susceptible to anisotropy confounds (Vulić et al., 2020), motivating both as planned analyses.

B Rank Sweep: Full Results and Seed Variance

Table 4 reports the full rank sweep (single run per rank) underlying Figure 1. Table 5 confirms the key qualitative pattern (the optimal rank diverging between English and cross-lingual performance) is stable across three seeds for four representative ranks.

Rank	Params (M)	EN F1	Xling F1
1	0.042	69.0	53.6
2	0.079	70.7	54.6
4	0.153	72.6	56.0
8	0.300	74.8	58.0
16	0.595	76.8	59.0
32	1.185	77.9	60.6
64	2.365	79.2	60.5
128	4.724	80.2	60.7
256	9.443	81.1	61.2
512	18.880	80.9	59.8
768	28.317	81.3	58.7

Table 4: LoRA rank sweep (single run; seed variance in Table 5 below). Bold = column best. $r_{\text{xling}}^* = 256 \neq r_{\text{EN}}^* = 768$ (Eq. 4).

Rank	EN F1		Avg X-ling F1	
	Mean	Std	Mean	Std
16	76.8	0.6	59.0	0.4
64	79.2	0.5	60.5	0.5
256	81.1	0.4	61.2	0.6
768	81.3	0.3	58.7	0.8

Table 5: Rank sweep F1, mean±std over 3 seeds. The cross-lingual decline at $r=768$ vs. $r=256$ (−2.5 pp) is consistent across seeds, confirming Eq. 4.

C Full Per-Language Results with Standard Deviations

Lang	Full FT	Adp	LoRA	IA ³	Pfx
en	80.5±.1	77.3±.2	76.2±.3	75.9±.1	74.7±.3
es	62.2±2.4	69.2±.4	69.9 ±2.8	65.4±.7	61.9±.4
fr	71.1±.9	72.7±.2	73.0 ±.8	70.4±.1	65.6±.3
de	67.3±.6	71.3 ±.3	71.2±.2	70.1±.2	66.0±.4
nl	75.1±.2	76.7±.2	76.8 ±.4	76.4±.3	72.6±.2
ru	55.4±1.0	58.6 ±.95	58.0±.8	57.0±.6	46.3±.5
fi	70.1±.4	72.0 ±.4	71.4±.2	71.9±.4	66.9±.3
sw	62.0±1.3	66.5±.3	66.8 ±1.1	63.6±.3	62.3±.4
tr	61.6±1.3	70.6±.8	72.6 ±2.7	70.4±.8	61.3±.7
vi	60.9±1.6	63.0±2.4	65.5 ±1.7	60.2±.0	60.6±.3
zh	13.1±3.6	26.1±.9	26.9 ±.5	21.5±.1	19.6±.8
ar	31.5±5.7	42.1±1.9	42.6±3.3	48.6 ±1.9	39.4±.6
hi	52.5±2.1	64.8 ±.5	61.8±1.1	62.2±.8	57.9±.4
ja	10.2±2.4	20.2±.2	21.4 ±1.3	16.4±1.0	14.6±.6
ko	32.7±1.4	46.3±.95	46.1±.7	46.4 ±.2	41.0±.5
Avg	51.8	58.6	58.9	57.2	52.6

Table 6: Full per-language F1 (%) with mean±std over 3 seeds. Bold = best cross-lingual result per row. Multi-seed adapter average: 58.59 ± 0.74 . The 0.3 pp LoRA–adapter gap is not statistically significant.

Beyond Calibration: Temporal LLM Confidence Assessment Across Languages and Local Settings

Sahil Kale¹

¹University of California, Los Angeles
CA, USA

Correspondence: sahilrkale@ucla.edu

Abstract

Large language models are expected to estimate their own uncertainty, with confidence signals used to guide retrieval, adaptive compute allocation, routing, and selective abstention. We investigate whether confidence signals measured at different temporal stages of problem solving encode distinct forms of epistemic information and remain reliable beyond English-centric settings. Inspired by classical meta-memory theory, we distinguish anticipatory pre-solve confidence (*Feeling-of-Knowing*, FOK) from retrospective post-solve confidence (*Judgement-of-Learning*, JOL). We evaluate these signals across two frontier LLMs, multiple reasoning domains, translated Marathi benchmarks, and a newly constructed benchmark of genuinely local Marathi factual questions. Across settings, the cheaper pre-solve signal remains weakly calibrated, while post-solve confidence generally provides stronger epistemic alignment despite its additional inference cost. Confidence behaviour transfers surprisingly well under direct language translation, with calibration trends largely preserved between English and Marathi, but degrades substantially on local Marathi knowledge, suggesting that linguistic transfer alone is insufficient for region-specific information needs. Finally, hallucinated responses maintain persistently elevated confidence after answer generation, highlighting important limitations of current LLM self-assessment mechanisms.

1 Introduction

Large language models (LLMs) are increasingly deployed across a wide range of real-world applications, including question answering, reasoning systems, retrieval-augmented generation (RAG), autonomous agents, and decision-support pipelines (Li et al., 2024; Cheng et al., 2025). Modern models contain substantial factual and procedural knowledge encoded within their parameters, while

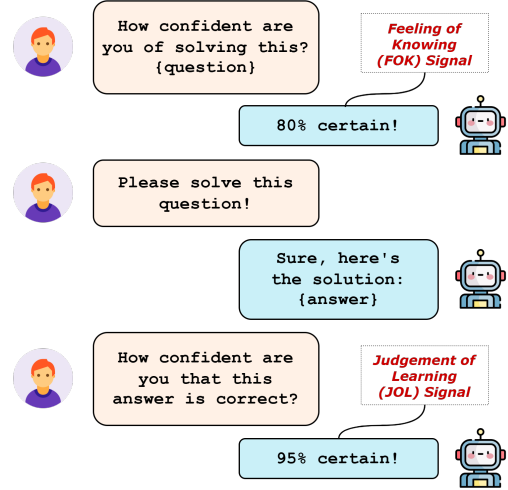


Figure 1: Illustration of pre-solve Feeling-of-Knowing (FOK) and post-solve Judgement-of-Learning (JOL) confidence signals in large language models.

also leveraging external context through tools, retrieval systems, and long-context prompting. In many such systems, models are further expected to make decisions about their own uncertainty and knowledge limitations, a capability often framed as *self-knowledge* (Kale and Nadadur, 2025) or *metacognition* (Ma et al., 2025). These internal confidence estimates are increasingly used to influence downstream behaviours such as deciding whether to invoke tools, request additional user clarification, defer generation, allocate additional compute, or trigger retrieval mechanisms (Li et al., 2025). However, recent work has shown that LLM confidence signals are often poorly calibrated and only weakly aligned with actual correctness. This limits the efficiency, reliability, and cost-effectiveness of such confidence-driven decision policies (Kale and Dhama, 2025).

While confidence calibration has been studied primarily in English-centric benchmarks, many practical deployments increasingly involve mul-

tilingual and underrepresented language settings where both model knowledge and evaluation resources are less abundant. Confidence estimates that appear reliable in high-resource settings may fail to transfer across languages or culturally specific knowledge domains, potentially leading to inappropriate retrieval decisions, overconfident hallucinations, or unsafe downstream behaviour. A particularly important open question is whether confidence calibration is primarily tied to language or to underlying knowledge availability. A model may retain similar confidence behaviour when questions are translated into another language, yet fail when confronted with genuinely local information that is sparsely represented in pre-training corpora and evaluation benchmarks. Distinguishing between these sources of distribution shift is crucial for deploying confidence-driven systems in multilingual settings.

In this work, we investigate whether confidence signals measured at different temporal stages of reasoning encode distinct forms of epistemic information, and also whether these carry across low-resource settings. Inspired by classical constructs from cognitive psychology and meta-memory research (Nelson, 1990; Koriati, 1997), we study two complementary signals: a pre-solve *Feeling-of-Knowing* (FOK) signal, and a post-solve confidence signal similar to *Judgement-of-Learning* (JOL). The FOK signal measures the model’s estimated likelihood of solving a problem correctly before attempting to start reasoning towards a solution, capturing a cheap yet possibly useful anticipatory signal. In contrast, the JOL-style signal measures confidence in a provided reasoning path and solution after answer generation, reflecting retrospective evaluation expected to be costlier but better calibrated. This framing introduces a temporal distinction that is largely absent in existing LLM calibration work, where confidence is typically treated as a static scalar attached only to the final answer. In this paper, we instead ask whether epistemic self-assessment changes between pre-solve and post-solve states, and whether this shift itself carries useful information about model behaviour, reliability, and failure modes.

In essence, our work is guided by the following research questions: *RQ1*: Is the pre-solve confidence signal (FOK) or the post-solve confidence signal (JOL) better calibrated to actual answer correctness, and can either signal support downstream decisions such as routing, retrieval triggering, or

adaptive compute allocation? We also explore the combined effect of these signals in error types by answering another important question, *RQ2*: Do strong values of FOK and JOL or delta trends correspond to distinct categories of model failure, such as knowledge gaps, reasoning failures, or hallucinated responses? In addition, we investigate a third research question motivated by multilingual deployment settings: *RQ3*: Do temporal confidence signals transfer across low-resource languages, and how are they affected by genuinely local knowledge distributions that differ from standard English-centric benchmarks?

We operationalise FOK and JOL confidence signals using prompting-based self-assessment before and after reasoning generation across multiple LLM families and domains. We evaluate their calibration, cost-effectiveness, and transition dynamics using correctness-conditioned analyses and confidence calibration metrics. The main contributions of our work can be highlighted as follows:

- We introduce a novel temporal framing of LLM self-confidence by separating anticipatory pre-solve confidence from retrospective post-solve confidence and show that the cheaper pre-solve signal is generally insufficiently calibrated for practical deployment, while post-solve confidence is often more informative but not uniformly more reliable
- We provide one of the first analyses of temporal confidence calibration in two frontier LLMs, demonstrating that the practical utility of confidence signals depends strongly on reasoning domain and model family
- We extend temporal confidence evaluation to low-resource settings in Marathi, showing that confidence behaviour transfers relatively well across language translation but degrades substantially for genuinely local knowledge

2 Related Work

2.1 Self-Knowledge and Confidence Estimation in LLMs

Recent work has increasingly framed large language models as systems capable of exhibiting forms of self-knowledge, self-monitoring, or metacognitive behaviour (Kale, 2025; Podolak and Verma, 2025). In this context, self-knowledge refers to a model’s ability to estimate whether its

own generated answer is likely to be correct or reliable (Li et al., 2025).

Prior work has shown that LLMs can often distinguish between questions they are likely to answer correctly and those where they are uncertain, particularly under calibrated confidence or selective abstention prompting (Kale and Dhimi, 2025). However, these signals remain imperfect, with models frequently exhibiting overconfidence, hallucination-prone certainty, and weak calibration under reasoning-heavy or distribution-shifted settings (Lin et al., 2022; Ren et al., 2026).

Despite these limitations, confidence-aware signals are increasingly important for applications such as retrieval triggering, adaptive compute allocation, selective generation, tool invocation, and human-in-the-loop systems (Wang et al., 2026; Zhang et al., 2026).

2.2 Measuring Self-Knowledge and Confidence Signals

Prior work has proposed a broad range of approaches for measuring confidence in language models. A major distinction concerns whether confidence is inferred from internal model representations (*white-box* methods) or elicited through prompting and behavioural outputs (*black-box* methods). White-box approaches typically rely on token probabilities, entropy, hidden-state representations, or calibration statistics derived directly from model distributions (Kadavath et al., 2022; Kissling et al., 2026). In contrast, black-box methods estimate self-knowledge through verbalised confidence ratings, self-evaluation prompts, selective prediction, answer consistency, critique-and-revision behaviours, and confidence-guided abstention mechanisms (Xu et al., 2024; Kale and Dhimi, 2025).

Most prior approaches measure confidence only after answer generation, implicitly treating self-knowledge as a static retrospective signal attached to a completed response. In contrast, our work introduces a temporal perspective on confidence estimation by distinguishing between anticipatory pre-solve confidence and retrospective post-solve confidence.

2.3 Metacognition, Distribution Shifts, and Underrepresented Languages

A growing body of work has examined the robustness of LLM capabilities under distribution shifts arising from changes in task domains, rea-

soning requirements, languages, or cultural contexts (Moskvoretskii et al., 2025; Wiemerslage and Wense, 2025). While multilingual evaluation has received increasing attention across generation, reasoning, and factual knowledge tasks, comparatively little work has investigated whether models can accurately assess their own uncertainty in these settings (Durrani et al., 2025). This distinction is important because strong task performance does not necessarily imply well-calibrated confidence estimates. Our work contributes to analysis beyond English by evaluating temporal confidence signals in Marathi using both translated factual QA tasks and locally grounded knowledge questions designed to probe underrepresented linguistic and cultural contexts.

3 Methodology

Our methodology is designed to measure two temporally distinct confidence signals associated with problem solving in large language models: a pre-solve anticipatory confidence signal and a post-solve retrospective confidence signal. Broadly inspired by classical meta-memory constructs, we refer to these signals as *Feeling-of-Knowing* (FOK) and *Judgement-of-Learning* (JOL), respectively. Because our evaluation includes closed-source models accessible only through API interfaces, we use prompting-based verbal confidence estimates as the primary confidence signal, despite known limitations. The overall pipeline consists of three sequential stages shown in Figure 1: (1) eliciting a pre-solve confidence estimate before reasoning begins, (2) generating an unconstrained answer to the task, and (3) eliciting a post-solve confidence estimate conditioned on the generated answer.

3.1 Pre-Solve Confidence Signal (FOK)

To obtain the pre-solve confidence signal, the model is first shown only the input question and is asked to estimate the likelihood that it would solve the problem correctly without actually attempting the solution. The prompt explicitly instructs the model not to reason about or solve the task, and to instead provide only a simple confidence estimate. The prompt used for pre-solve confidence elicitation is shown in Figure 2.

The scalar value returned by the model is treated as the FOK estimate. This signal captures the model’s anticipatory belief regarding whether its internal knowledge and reasoning capabilities are

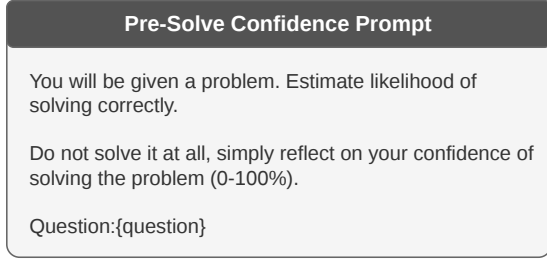


Figure 2: Prompt used for eliciting pre-solve confidence from LLMs

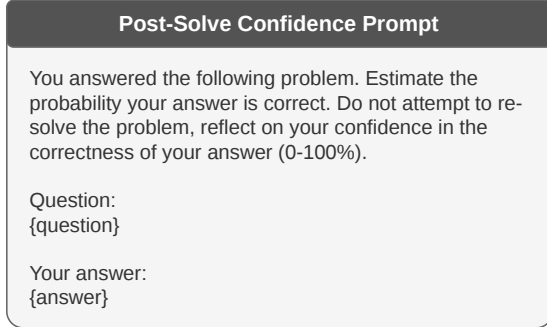


Figure 3: Prompt used for eliciting post-solve confidence from LLMs

sufficient to successfully solve the task prior to answer generation.

3.2 Answer Generation

After obtaining the pre-solve confidence estimate, the model is prompted again with the original question and asked to directly solve the task with chain-of-thought reasoning. No additional confidence instructions, calibration prompts, or answer formatting requirements are imposed during this stage. This design choice ensures that the generated reasoning trajectory more closely reflects the model’s natural inference behaviour under standard prompting conditions.

The generated answer is then stored and used as contextual input for retrospective confidence estimation in the next stage.

3.3 Post-Solve Confidence Signal (JOL)

To obtain the post-solve confidence signal, the model is shown both the original question and its previously generated answer. It is then asked to estimate the probability that its answer is correct. Unlike the pre-solve stage, this confidence estimate is conditioned on the model’s completed reasoning trajectory and final output. The post-solve confidence prompt is shown in Figure 3.

The scalar value returned by the model is then

used as a JOL-style retrospective confidence estimate. This signal reflects self-evaluation conditioned on the produced answer and reasoning path rather than on the question alone.

Following these steps, for each problem instance, we therefore obtain three core variables: a pre-solve confidence estimate (FOK), a generated answer, and a post-solve confidence estimate (JOL). The generated answer is independently evaluated against reference solutions to determine correctness labels. We then analyse confidence calibration estimates and cost-effectiveness of these signals ahead.

For multilingual experiments, the same three-stage procedure is retained without modification. Confidence elicitation, answer generation, and post-solve self-assessment prompts are translated into Marathi while preserving their original structure and intent. This design allows us to study whether temporal confidence signals remain stable under linguistic transfer without changing the underlying confidence estimation framework.

4 Experimental Setup

4.1 Models

We evaluate our framework using two frontier large language models: GPT-5.4 (gpt-5.4-2026-03-05) and Claude Sonnet 4.6. Models were accessed through the official OpenAI and Anthropic APIs. All experiments use deterministic decoding with temperature set to 0, while other generation parameters remain at default settings. Full details are provided in Section A in the Appendix.

4.2 Datasets

We evaluate across four domains probing mathematical reasoning, logical reasoning, factual recall, and low-resource multilingual factual recall.

4.2.1 Mathematical Reasoning

For mathematical reasoning, we use 500 randomly sampled questions from the Mathematics Stack Exchange split of RealMath (Zhang et al., 2025). The dataset contains non-trivial user-authored mathematical problems with verified solutions and diverse reasoning structures, making it suitable for studying confidence calibration under open-ended mathematical reasoning.

4.2.2 Logical Reasoning

For logical reasoning, we use 400 samples from RiddleBench (Halder et al., 2025), a benchmark of challenging natural-language puzzles involving deduction, constraint satisfaction, spatial reasoning, and multi-step inference. These tasks provide a cognitively demanding setting for evaluating confidence behaviour under complex reasoning.

4.2.3 Factual Recall

For factual recall, we use 750 randomly sampled questions from TriviaQA (Joshi et al., 2017), a large-scale benchmark of naturally authored trivia questions with verified answers. Unlike the reasoning datasets above, TriviaQA primarily evaluates factual retrieval and parametric knowledge recall.

4.2.4 Low-Resource Settings

To evaluate temporal confidence signals in an underrepresented language setting, we construct two Marathi factual-recall splits.

The first split (*Marathi-Translated*) is derived directly by translating TriviaQA, RealMath and RiddleBench to a low-resource language, Marathi with each split retaining the same number of examples as its English counterpart (750 TriviaQA, 500 RealMath, 400 RiddleBench). Questions and reference answers are translated from English to Marathi using the Google Translate API while preserving the original task structure and evaluation protocol. Two graduate-level native Marathi speakers reviewed translated prompts and dataset samples, performed spot-checks throughout the corpus, and verified semantic equivalence with the original English questions. This split isolates language transfer while keeping knowledge requirements unchanged. Additional details are in Section B in the Appendix.

The second split (*Marathi-Local*) evaluates genuinely local knowledge. Unlike translated benchmarks, these questions require region-specific information that is less frequently represented in English-language resources and often available primarily through Marathi-language sources. Three native Marathi speakers manually authored 201 factual questions based on local events and public information reported before model knowledge cut-off dates using trusted Marathi news outlets such as Zee News Marathi, Maharashtra Times, and ABP Majha. Each question was paired with supporting source URLs for verification, while reference answers and correctness assessments were indepen-

dently reviewed by two separate native speakers with agreement of Cohen’s $\kappa = 0.82$. Additional details and examples are in Section C in the Appendix.

4.3 Evaluation Protocol

Answers are evaluated using benchmark-specific matching procedures to obtain correctness labels. We report Expected Calibration Error (ECE) (Naeini et al., 2015), Brier Score (Naeini et al., 2015), and AUROC. ECE measures the gap between predicted confidence and empirical correctness, Brier Score evaluates the quality of probabilistic confidence estimates, and AUROC measures how effectively confidence scores distinguish correct from incorrect predictions. We note that AUROC evaluates discrimination independently of calibration and can remain high even when ECE or Brier Score indicate poor calibration, but we report all three because they capture complementary properties of confidence quality. Reliability curves are additionally used to visualise calibration behaviour. Uncertainty estimates are computed using 2000 bootstrap resamples and reported as 95% confidence intervals. We further report cost-benefit analyses and confidence trends across error categories.

For the Marathi-Translated split, evaluation was adapted to support valid multilingual answer variations. Each example contains both the original English reference answer and its Marathi translation. Predictions are considered correct if they match either reference set under the benchmark evaluation procedure, reducing false negatives caused by transliteration differences, bilingual generations, or alternative entity forms. Native-speaker spot checks were conducted to verify evaluation quality.

For the Marathi-Local split, no benchmark labels are available. Answer correctness is therefore assessed manually by native Marathi speakers using the associated source material. Following prior factuality evaluation work, correctness is scored on a continuous scale between 0 and 1, where 1 denotes a fully correct answer, 0 an incorrect answer, and intermediate values partially correct responses (AUROC computed with threshold of 0.5). These scores are subsequently used when computing calibration and confidence metrics.

Model	Domain	Pre-Solve (FOK)			Post-Solve (JOL)		
		ECE↓	Brier↓	AUROC↑	ECE↓	Brier↓	AUROC↑
GPT-5.4	TriviaQA	0.195 ± 0.027	0.209 ± 0.028	0.567 ± 0.053	0.194 ± 0.021	0.203 ± 0.025	0.606 ± 0.051
	Math	0.381 ± 0.041	0.366 ± 0.037	0.719 ± 0.047	0.315 ± 0.045	0.330 ± 0.037	0.751 ± 0.048
	Logic	0.483 ± 0.039	0.469 ± 0.035	0.679 ± 0.055	0.286 ± 0.041	0.313 ± 0.036	0.690 ± 0.049
Claude Sonnet 4.6	TriviaQA	0.071 ± 0.023	0.119 ± 0.019	0.656 ± 0.055	0.077 ± 0.024	0.114 ± 0.020	0.703 ± 0.046
	Math	0.188 ± 0.061	0.231 ± 0.051	0.500 ± 0.092	0.108 ± 0.064	0.140 ± 0.047	0.791 ± 0.085
	Logic	0.341 ± 0.051	0.316 ± 0.029	0.767 ± 0.057	0.472 ± 0.053	0.456 ± 0.041	0.695 ± 0.049

Table 1: Calibration and discrimination performance (95% confidence intervals) across domains and temporal stages

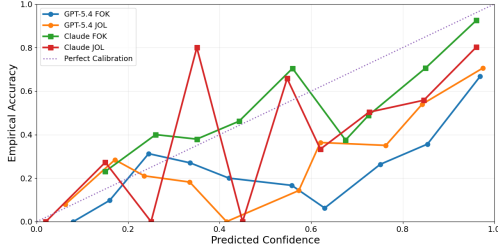


Figure 4: Aggregate reliability curves comparing pre-solve (FOK) and post-solve (JOL) confidence calibration across models and domains

5 Results

Table 1 reports calibration and discrimination performance for pre-solve (FOK) and post-solve (JOL) confidence signals across models and domains. Reported values are means with 95% bootstrap confidence intervals. Figure 4 shows aggregate reliability curves. We evaluate calibration using Expected Calibration Error (ECE) and Brier Score, where lower values indicate stronger alignment between confidence and correctness, and discrimination using AUROC.

5.1 Calibration Analysis

5.1.1 Post-solve confidence generally provides stronger epistemic alignment

Across most domains, JOL-style confidence estimates exhibit lower calibration error and stronger discrimination than FOK estimates, although gains are not uniform and remain modest for factual recall. This trend is particularly pronounced in mathematics, where Claude Sonnet 4.6 improves from near-random discrimination (0.500 ± 0.092) to strong discrimination (0.791 ± 0.085). The consistently weaker calibration of the cheaper pre-solve signal suggests that FOK alone is insufficiently reliable for downstream decisions such as retrieval triggering, model routing, or adaptive compute allocation. Instead, models appear to require answer generation before producing confidence estimates

that better reflect correctness.

5.1.2 Reasoning domains exhibit substantially higher calibration instability than factual recall

Both models remain notably miscalibrated on mathematics and logical reasoning despite moderate discrimination performance. Logic reasoning in particular exhibits consistently high ECE values, with GPT-5.4 reaching a pre-solve ECE of 0.483 ± 0.039 and Claude Sonnet 4.6 a post-solve ECE of 0.472 ± 0.053 . In contrast, TriviaQA shows lower calibration error and smaller temporal shifts between FOK and JOL. This suggests that retrospective confidence adjustment is more useful for multi-step reasoning tasks than for direct factual retrieval, where confidence remains relatively stable before and after answer generation.

5.1.3 Temporal confidence transitions reveal domain-dependent failure behaviour

The transition from FOK to JOL is not uniformly beneficial. While post-solve confidence often improves calibration and ranking quality, certain domains exhibit confidence inflation after reasoning. Most notably, Claude Sonnet 4.6 on logic reasoning degrades from an ECE of 0.341 to 0.472, while AUROC falls from 0.767 to 0.695. The corresponding reliability curves are also less stable after reasoning. These results suggest that generating extended reasoning traces can sometimes induce false certainty rather than improved self-knowledge.

5.2 Cost-Benefit Analysis

We next examine the trade-off between the additional inference cost required to obtain JOL confidence and the calibration gains achieved. FOK cost includes only the confidence estimation prompt, whereas JOL additionally requires answer generation. We note this cost comparison assumes an application where answer generation is not otherwise required (e.g., pre-generation retrieval routing). In settings where an answer must be produced

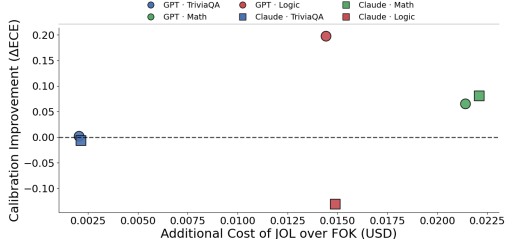


Figure 5: Calibration improvement versus additional inference cost for post-solve confidence (JOL) relative to pre-solve confidence (FOK) across models and domains.

regardless (e.g., post-hoc abstention or answer verification), the marginal cost of JOL is limited to the additional confidence-elicitation query. Costs were estimated using average dataset lengths and provider pricing rates as of May 2026. Positive ΔECE values indicate improved calibration from JOL relative to FOK, as shown in Figure 5.

5.2.1 JOL is primarily cost-effective in structured reasoning domains

For GPT-5.4, JOL consistently improves calibration, with the largest gains occurring in Logic ($\Delta ECE = 0.198$). Mathematics also shows meaningful improvements for both models, whereas TriviaQA exhibits only marginal gains. These results suggest that post-solve confidence becomes most useful as reasoning complexity increases and can justify additional inference cost in structured problem-solving settings.

5.2.2 Additional reasoning does not always improve self-assessment

TriviaQA shows minimal calibration improvement from JOL and slight degradation for Claude Sonnet 4.6, suggesting that the cheaper FOK signal may already be sufficient for factual recall. More notably, Claude on Logic becomes substantially less calibrated despite higher inference cost, indicating that additional reasoning can sometimes amplify miscalibration rather than improve confidence reliability.

5.3 Error Type Analysis

To investigate whether confidence patterns correspond to distinct failure mechanisms, we manually analysed 200 incorrect responses sampled across models and domains. Two graduate-level annotators independently reviewed all samples, achieving a Cohen’s κ of 0.92. Disagreements were subse-

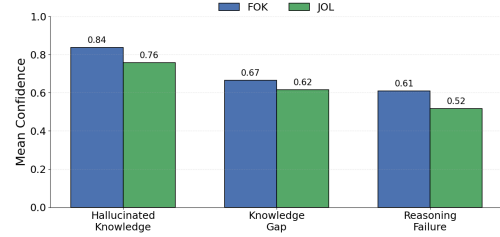


Figure 6: Mean pre-solve (FOK) and post-solve (JOL) confidence across annotated error categories.

quently resolved through discussion. Errors were assigned to one of three categories:

- *Knowledge gap*: The model lacked the required information or explicitly hedged between alternatives.
- *Hallucinated knowledge*: The model confidently generated fabricated facts, entities, or incorrect theorem usage.
- *Reasoning failure*: The answer was incorrect due to flawed reasoning despite the absence of fabricated knowledge.

Absolute confidence levels proved more informative than confidence transitions for distinguishing failure categories. Confidence deltas exhibited high variance and did not significantly separate error types ($p = 0.366$, Kruskal-Wallis test). Nevertheless, clear differences emerge in Figure 6, where confidence generally decreases from FOK to JOL, consistent with the calibration improvements observed earlier.

5.3.1 Hallucinations exhibit persistently elevated confidence

Hallucinated responses showed the highest confidence both before and after reasoning (FOK: 0.837, JOL: 0.758) despite being incorrect. This suggests that hallucinations are associated less with confidence increases during reasoning and more with overconfident internal representations that remain stable throughout answer generation, consistent with prior hallucination research (Simhi et al., 2025; Kalai and Vempala, 2024).

5.3.2 Reasoning failures show lower and more unstable confidence patterns

Reasoning failures exhibited the lowest confidence levels alongside substantial variance in confidence

transitions, indicating that models partially recognise reasoning difficulty even when failing. Knowledge gaps occupied an intermediate regime, where models often began uncertain but also decreased confidence after producing an answer.

5.4 Multilingual Transfer and Local Knowledge

5.4.1 Temporal confidence signals largely transfer across language translation

Direct translation of benchmark questions into Marathi (Marathi-Translated split) produces only modest changes in calibration and discrimination performance relative to the original English benchmarks, as seen in Table 2. Across models and domains, most ECE differences remain small and within bootstrap uncertainty intervals. GPT-5.4 exhibits only minor calibration changes after translation, and temporal confidence dynamics remain largely unchanged across mathematical, logical, and factual tasks. Claude Sonnet 4.6 shows somewhat larger degradation for factual recall, but the overall pattern of stronger post-solve confidence discrimination persists.

These findings suggest that the temporal confidence mechanisms captured by FOK and JOL are relatively robust to language transfer. Translating tasks into an underrepresented language does not substantially alter how these frontier models assess their own uncertainty.

5.4.2 Locally grounded knowledge produces substantially larger calibration failures

A markedly different picture emerges on the Marathi-Local split. Both models exhibit substantially higher calibration error and weaker discrimination than on either English or translated Marathi benchmarks (Table 3).

For GPT-5.4, pre-solve ECE increases from 0.195 on English TriviaQA and 0.205 on translated Marathi TriviaQA to 0.421 on Marathi-Local questions. Claude Sonnet 4.6 exhibits a similar increase from 0.071 and 0.122 to 0.378. AUROC values simultaneously approach near-random ranking behaviour, falling to 0.521 and 0.558 respectively.

These results indicate that calibration degrades far more strongly under local knowledge shifts than under language shifts alone. While models largely preserve self-assessment behaviour across languages, they remain substantially less reliable when confronted with region-specific information

Model	Domain	FOK Δ ECE	JOL Δ ECE
GPT-5.4	TriviaQA	+0.010	-0.005
	Math	+0.030	-0.016
	Logic	+0.028	+0.019
Claude	TriviaQA	+0.051	+0.045
	Math	+0.004	+0.016
	Logic	-0.009	+0.040

Table 2: Change in calibration error (Δ ECE) after translation from English to Marathi. Positive values indicate worse calibration in Marathi.

Model	Signal	ECE	AUROC
GPT-5.4	FOK	0.421	0.521
GPT-5.4	JOL	0.287	0.579
Claude	FOK	0.378	0.558
Claude	JOL	0.255	0.616

Table 3: Confidence calibration on the Marathi-Local benchmark.

that is sparsely represented in training data and retrieval resources.

6 Conclusion and Future Work

In this work, we introduced a temporal framing of confidence estimation in large language models by separating pre-solve (FOK) and post-solve (JOL) confidence signals. Across models and reasoning domains, post-solve confidence generally provides stronger calibration and discrimination than the cheaper pre-solve signal, although gains depend strongly on the model and task. In Marathi, confidence behaviour transfers well under direct language translation, but calibration degrades substantially on genuinely local knowledge questions. Hallucinated responses also maintain persistently elevated confidence, highlighting limitations of current self-assessment mechanisms. Overall, confidence-aware systems should account for both temporal stage and knowledge domain when using confidence for retrieval, routing, or adaptive compute decisions.

Future work can extend this framework to additional low-resource languages, multimodal settings, agentic systems, and model families beyond the two evaluated here. Because Marathi-Local and Marathi-Translated differ along several axes besides locality, the calibration gap cannot be attributed to locality alone; a matched benchmark controlling for these factors would help isolate its effect. It would also be valuable to examine whether white-box uncertainty signals, such as to-

ken probabilities and representation-based confidence, exhibit similar temporal dynamics. More broadly, our findings motivate studying calibration not only across languages but also across local knowledge distributions, where linguistic transfer alone may be insufficient for reliable self-assessment.

References

- Jingwen Cheng, Kshitish Ghate, Wenye Hua, William Yang Wang, Hong Shen, and Fei Fang. 2025. [REALM: A dataset of real-world LLM use cases](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8331–8341, Vienna, Austria. Association for Computational Linguistics.
- Nadir Durrani, Basel Mousi, and Fahim Dalvi. 2025. [Editing across languages: A survey of multilingual knowledge editing](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 15906–15918, Suzhou, China. Association for Computational Linguistics.
- Deepon Halder, Alan Saji, Thanmay Jayakumar, Ratish Puduppully, Anoop Kunchukuttan, and Raj Dabre. 2025. [Riddlebench: A new generative reasoning benchmark for llms](#). *Preprint*, arXiv:2510.24932.
- Mandar Joshi, Eunsol Choi, Daniel S. 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*, Vancouver, Canada. Association for Computational Linguistics.
- 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, and 17 others. 2022. [Language models \(mostly\) know what they know](#). *Preprint*, arXiv:2207.05221.
- Adam Tauman Kalai and Santosh S. Vempala. 2024. [Calibrated language models must hallucinate](#). In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, STOC 2024, page 160–171, New York, NY, USA. Association for Computing Machinery.
- Sahil Kale. 2025. [Mirage of mastery: Memorization tricks llms into artificially inflated self-knowledge](#). *arXiv preprint arXiv:2506.18998*.
- Sahil Kale and Devendra Singh Dhami. 2025. [Knowrl: Teaching language models to know what they know](#). *Preprint*, arXiv:2510.11407.
- Sahil Kale and Vijaykant Nadadur. 2025. [Line of duty: Evaluating LLM self-knowledge via consistency in feasibility boundaries](#). In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 127–140, Albuquerque, New Mexico. Association for Computational Linguistics.
- Christopher Kissling, Elena Merdjanovska, and Alan Akbik. 2026. [Self-aware knowledge probing: Evaluating language models’ relational knowledge through confidence calibration](#). *Preprint*, arXiv:2601.18901.
- Asher Koriath. 1997. [Monitoring one’s own knowledge during study: A cue-utilization approach to judgments of learning](#). *Journal of Experimental Psychology: General*, 126(4):349–370.
- Jiawei Li, Yizhe Yang, Yu Bai, Xiaofeng Zhou, Yinghao Li, Huashan Sun, Yuhang Liu, Xingpeng Si, Yuhao Ye, Yixiao Wu, Yiguan Lin, Bin Xu, Bowen Ren, Chong Feng, Yang Gao, and Heyan Huang. 2024. [Fundamental capabilities of large language models and their applications in domain scenarios: A survey](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11116–11141, Bangkok, Thailand. Association for Computational Linguistics.
- Wenjun Li, Dexun Li, Kuicai Dong, Cong Zhang, Hao Zhang, Weiwen Liu, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025. [Adaptive tool use in large language models with meta-cognition trigger](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13346–13370, Vienna, Austria. Association for Computational Linguistics.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [Teaching models to express their uncertainty in words](#). *Preprint*, arXiv:2205.14334.
- Ziyang Ma, Qingyue Yuan, Zhenglin Wang, and Deyu Zhou. 2025. [Large language models have intrinsic meta-cognition, but need a good lens](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3460–3477, Suzhou, China. Association for Computational Linguistics.
- Viktor Moskvoretskii, Maria Marina, Mikhail Salnikov, Nikolay Ivanov, Sergey Pletenev, Daria Galimzianova, Nikita Krayko, Vasily Kononov, Irina Nikishina, and Alexander Panchenko. 2025. [Adaptive retrieval without self-knowledge? bringing uncertainty back home](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6355–6384, Vienna, Austria. Association for Computational Linguistics.
- 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*, AAAI’15, page 2901–2907. AAAI Press.

Thomas O. Nelson. 1990. [Metamemory: A theoretical framework and new findings](#). volume 26 of *Psychology of Learning and Motivation*, pages 125–173. Academic Press.

Jakub Podolak and Rajeev Verma. 2025. [Read your own mind: Reasoning helps surface self-confidence signals in LLMs](#). In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertainNLP 2025)*, pages 247–258, Suzhou, China. Association for Computational Linguistics.

Jingyi Ren, Ante Wang, Yunghwei Lai, Xiaolong Wang, Linlu Gong, Weitao Li, Weizhi Ma, and Yang Liu. 2026. [Beyond "i don't know": Evaluating llm self-awareness in discriminating data and model uncertainty](#). *Preprint*, arXiv:2604.17293.

Adi Simhi, Itay Itzhak, Fazl Barez, Gabriel Stanovsky, and Yonatan Belinkov. 2025. [Trust me, I'm wrong: LLMs hallucinate with certainty despite knowing the answer](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 14665–14688, Suzhou, China. Association for Computational Linguistics.

Mengyu Wang, Xiaoying Zhi, Zhiyi Li, Robin Schmucker, Shay B. Cohen, Tiejun Ma, and Fran Silavong. 2026. [Self-knowledge re-expression: A fully local method for adapting llms to tasks using intrinsic knowledge](#). *Preprint*, arXiv:2604.22939.

Adam Wiemerslage and Katharina Von Der Wense. 2025. [Improving low-resource morphological inflection via self-supervised objectives](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 24494–24510, Vienna, Austria. Association for Computational Linguistics.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. 2024. [SaySelf: Teaching LLMs to express confidence with self-reflective rationales](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5985–5998, Miami, Florida, USA. Association for Computational Linguistics.

Jie Zhang, Cezara Petrucci, Kristina Nikolić, and Florian Tramèr. 2025. [Realmath: A continuous benchmark for evaluating language models on research-level mathematics](#). *Preprint*, arXiv:2505.12575.

Qingjie Zhang, Yujia Fu, Yang Wang, Liu Yan, Tao Wei, Ke Xu, Minlie Huang, and Han Qiu. 2026. [Stop before you fail: Operational capability boundaries for mitigating unproductive reasoning in large reasoning models](#). *Preprint*, arXiv:2509.24711.

A Model and Inference Details

Models were accessed through the official OpenAI and Anthropic Python SDKs using their public APIs. All experiments used deterministic decoding with temperature set to 0. Maximum output length

was capped at 450 tokens. Remaining provider-specific generation parameters were left at default values. Table 4 summarises the inference settings used throughout all experiments.

Setting	GPT-5.4	Claude Sonnet 4.6
Provider	OpenAI	Anthropic
Access Method	Python SDK/API	Python SDK/API
Temperature	0.0	0.0
Decoding	Deterministic	Deterministic
Other Parameters	Default	Default

Table 4: Model access and inference settings.

B Marathi Dataset Translation

TriviaQA, RiddleBench and RealMath datasets with questions and reference answers were translated from English to Marathi using the Google Translate API. Two independent native Marathi speakers reviewed translated samples and verified semantic equivalence through spot-checking. Prompt templates were independently translated and reviewed by the same annotators. Proper nouns were transliterated into Marathi where appropriate.

C Marathi Local Benchmark

Figure 7 presents representative examples from the Marathi-Local benchmark. Questions were manually authored by native Marathi speakers based on region-specific events reported in Marathi-language news sources prior to model knowledge cut-off dates.

Question	Reference Answer	Source
नाशिकमधील कोणत्या परिसरातील वृक्षतोडीला राष्ट्रीय हरित लवादने 2022 मध्ये स्थगिती दिली आहे?	तपोवन परिसरातील वृक्षतोडीला	ABP Majha
प्रहार जनशक्ती पक्षाचे प्रमुख बच्चू कडू 2023 मध्ये कोणत्या शहराच्या दौऱ्यावर होते?	सोलापूर	Maharashtra Times

Figure 7: Examples from the Marathi-Local benchmark

***AmharicStoryQA*: A Multicultural Story Question Answering Benchmark in Amharic**

**Israel Abebe Azime¹, Abenezer Kebede Angamo², Hana Mekonen Tamiru³,
Dagnachew Mekonnen Marilign⁴, Philipp Slusallek¹, Seid Muhie Yimam⁵,
Dietrich Klakow¹**

¹ Saarland University, ² AIMS AMMI, ³ Resonance AI4D Lab, Addis Ababa University,

⁴ HILCOE School of Computer Science and Technology, ⁵University of Hamburg

Abstract

With the growing emphasis on multilingual and cultural evaluation benchmarks for large language models, language and culture are often treated as synonymous, and performance is commonly used as a proxy for a model’s understanding of a given language. In this work, we argue that such evaluations overlook meaningful cultural variation that exists within a single language. We address this gap by focusing on narratives from different regions of Ethiopia and demonstrate that, despite shared linguistic characteristics, region-specific and domain-specific content substantially influences language evaluation outcomes. To this end, we introduce *AmharicStoryQA*, a long-form story question answering benchmark grounded in culturally diverse narratives from Amharic-speaking regions. Using this benchmark, we reveal a significant narrative understanding gap in existing LLMs, highlight pronounced regional differences in evaluation results, and show that supervised fine-tuning yields uneven improvements across regions and evaluation settings. Our findings emphasize the need for culturally grounded benchmarks that go beyond language-level evaluation to more accurately assess and improve narrative understanding in low-resource languages.

1 Introduction

Evaluating Large Language Models (LLMs) on low-resource languages is essential for uncovering their true language understanding capabilities in specific linguistic and cultural contexts, especially given that these models are predominantly trained on large-scale English corpora.

Despite increasing efforts to include more languages in LLM evaluations, particularly African-centric languages, there remains a notable gap in the creation of multicultural evaluation datasets that capture the diversity of linguistic and cultural contexts (Alabi et al., 2025b). Another important

limitation in low-resource LLM evaluation is the scarcity of generation-focused datasets designed to simulate real-life model usage, which are essential for ensuring that benchmarking accurately reflects how LLMs perform in practical, everyday scenarios.

To capture the diversity of linguistic and cultural contexts, most recent works (Havaladar et al., 2023; Azime et al., 2025b) expand evaluations by adding more languages and incorporating culturally grounded information drawn from the regions they represent. This idea, *which treats each language as being associated with a single cultural identity, has a solid foundation but still fails to capture the cultural variations that exist within the same language community*. These variations arise from differences in governmental structures, regional histories, social norms, migration patterns, and intra-country ethnic diversity. As a result, evaluations risk oversimplifying cultural nuance and underrepresenting the lived experiences of multilingual and multicultural societies

In this work, we introduce *AmharicStoryQA*, a multicultural story-based question answering benchmark available in Amharic and English. The benchmark is constructed from pre-translated stories and folktales collected from nine different regions of Ethiopia. Our goal is to provide an evaluation resource that captures Ethiopia’s rich regional diversity while enabling the study of long-form question answering. *AmharicStoryQA* focuses on culturally grounded narrative understanding and offers one of the first long-context QA benchmarks centered on Ethiopian stories across nine regions.

Based on the benchmark we introduce, we aim to answer the following research questions: **(RQ1)** *How well do Large Language Models comprehend Amharic narratives?* This is measured through performance on a story-based question answering datasets. **(RQ2)** *How cultural differences influence LLM performance in a single language?* and

(RQ3) *Can we improve LLM performance using the culturally grounded training dataset we introduce, demonstrating the importance of such data for addressing the limitations observed?* To support these questions, we further conduct human error analysis to better identify the linguistic and cultural factors contributing to model failures.

To address this question, our work makes the following contributions.

- We introduce *AmharicStoryQA*, a multicultural story-based question answering benchmark in Amharic and English, consisting of 571 training and 649 test examples derived from 244 stories collected across nine Ethiopian regions. The training and test sets are constructed with disjoint stories, ensuring that no story appears in both splits and thereby minimizing the risk of story-level data leakage. The benchmark includes both multiple-choice question answering (MCQA) and open-ended generation questions, enabling comprehensive evaluation of model understanding across formats.¹
- We explore the zero-shot performance of seven opensource LLMs and analyze how their results vary across regions, languages, prompt languages, and task types with human error analysis.
- We explore the effect of supervised fine-tuning on improving the performance of selected LLMs.

2 Related Work

Low Resource LLM Evaluations. Low-resource conditions, characterized by the lack of datasets for specific tasks and domains, limited digitization of languages, and insufficient data to effectively train and improve models, are common across many African languages (Nigatu et al., 2024). Evaluating Large Language Models (LLMs) in such settings presents unique challenges, primarily due to data scarcity and the dominance of English-centric benchmarks (Singh et al., 2025; Anugraha et al., 2026). Recent efforts have begun to address this gap by introducing multilingual evaluation benchmarks that include African languages (Adelani et al., 2025; Ojo et al.,

2025). Despite the great deal of work done to include number of tasks and languages and cultural values (Romero et al., 2024; Myung et al., 2024) there is still a need to deeper exploration in single language multiple cultural evaluations.

Long Sequence LLM Evaluations. While the capability of LLMs to process long contexts is a rapidly evolving area of research, benchmarks evaluating this capability are predominantly available only for high-resource languages. Recent works such as Alabi et al. (2025a) evaluate long-context document translation across five low-resource languages, including Amharic. Despite these efforts, there remains a critical absence of long-form benchmarks for Amharic that evaluate an LLM’s ability to maintain coherence and track narrative arcs over extended contexts, leaving a significant blind spot in understanding how modern LLMs handle extensive texts in Ethiopian languages.

Story Question Answering. Story-based question answering assesses the ability of the model to comprehend narrative structures, character motivations, and causal chains. AfriQA (Ogundepo et al., 2023) introduces a cross-lingual open-retrieval QA dataset covering seven African languages, evaluating models on their ability to retrieve and answer questions using evidence that is often available only in English or French. Similarly, Belebele (Bansarkar et al., 2024) offers a parallel reading comprehension benchmark across 122 language variants, including Amharic. However, these evaluations typically rely on short passages or translated content, which may not fully capture the linguistic depth required for native-level understanding. Furthermore, multilingual benchmarks often treat languages such as Amharic as monolithic, overlooking the need for region-specific evaluations that account for local variations in dialect and usage.

Cultural Evaluations Truly robust LLMs must capture not only the syntactic structure of a language but also its cultural context and dialectal variation. Recent work has examined the fairness and robustness of LLMs across different dialects of English, even when a single language is used as the communication medium (Srirag et al., 2025). Other studies leverage culturally grounded proverbs and sayings across multiple languages to more rigorously evaluate language understanding capabilities (Cecilia Liu et al., 2024; Azime

¹<https://huggingface.co/collections/israel/amharicstoryqa>

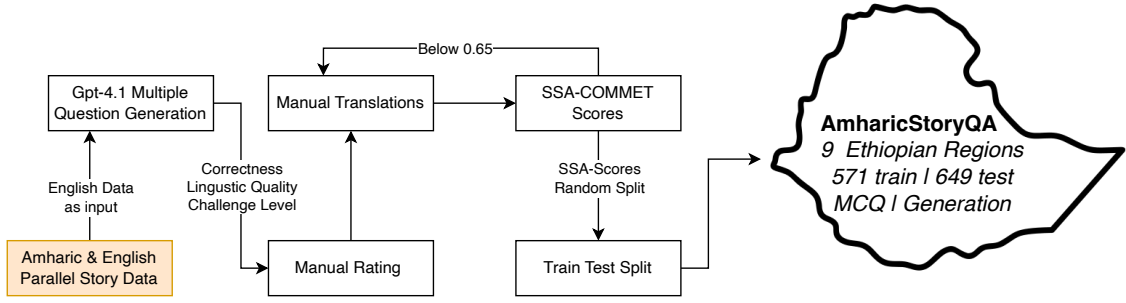


Figure 1: **Overview of the *AmharicStoryQA* dataset** creation process, from Amharic and English parallel stories through question generation, manual translation rating, quality control, and train–test split verification.

et al., 2025b). In this work, we adopt a single-language, multi-cultural evaluation framework and extend prior efforts by introducing long-form, culturally grounded, story-based question answering, enabling a more comprehensive assessment of language understanding across cultural variations.

Amharic Question Answering Recent work such as (Taffa et al., 2024; Belay et al., 2022) introduces an extensive Amharic question answering benchmark used to evaluate Amharic QA capabilities; however, it focuses primarily on open-ended question answering, which is more challenging to evaluate reliably. In this work, we contribute a multi-task story-based question answering dataset that spans multiple cultures in same language setting, resulting in broader cultural diversity and improved regional coverage additional to narrative style long sequence source documents that was not included in previous works.

3 Methodology

3.1 Dataset

The stories used in this dataset are sourced from Ethiopian folktales² webpages which was previously included for story generation tasks in low resource LLM work Azime et al. (2024).

As shown in Figure 1, we followed the steps described below, building on prior human-in-the-loop dataset creation approaches (Yuen et al., 2025; Berkane et al., 2025).

We worked with 244 stories collected from nine distinct regions, with the SNNP region further subdivided into multiple zonal classifications, as detailed in Appendix B. We will release the dataset

strictly for research purposes only. Figure 3 illustrates the *Llama 3.1 8B Instruct* token counts (sequence lengths) for each story across regions showing coverage of large sequence stories for this work. Average sequence length of stories by region is shown in Table 3.

3.1.1 Question Generation

Given that we have verified English question answer pairs for all Amharic stories collected and presented in Ethiopian folktales webpages, we leveraged Gpt-4.1 to generate batches of five questions per story. This approach enabled us to produce five distinct questions for each story without repeating ideas across them. For each generated question, we also produced one correct answer and three distractor choices: (choice B) a choice biased by world knowledge, (Choice C) an unrelated statement, and (Choice D) a factually false option. We generated 5 questions from each story in parallel to avoid repetitive questions. All content was generated in a single step using structured generation³, which helped prevent parsing inconsistencies that can arise when handling multiple separate generations.

3.1.2 Manual Question Rating

After obtaining the English stories, Amharic stories, four distractor choices, and the correct answer, we developed a question-rating guideline to ensure that the generated English questions met standards of correctness and faithfulness, linguistic quality and clarity, and comprehension depth and challenge level. The evaluation was conducted by three human annotators. **Correctness and Faithfulness** evaluates whether the question is grounded in the

²<https://www.ethiopianfolktales.com/am>

³<https://platform.openai.com/docs/guides/structured-outputs>

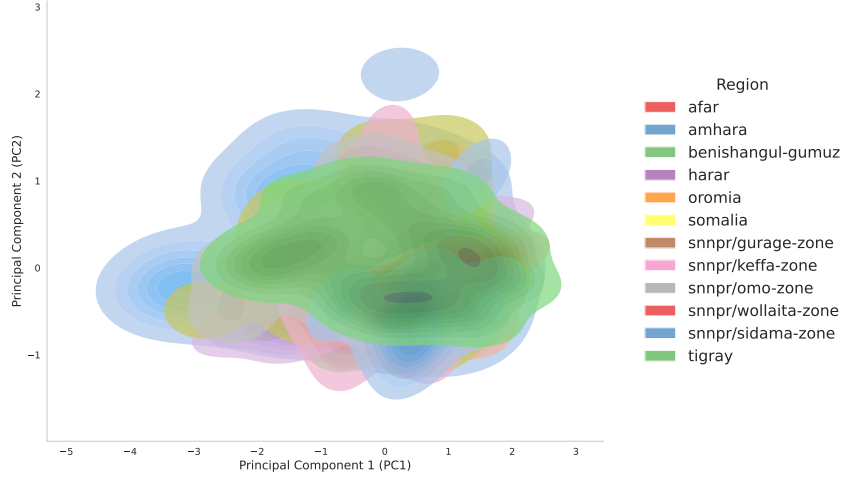


Figure 2: **Regional semantic density of Amharic stories across regions** in the *AmharicStoryQA* corpus. Each contour shows a kernel density estimate (KDE) over 2D PCA projections of multilingual MiniLM sentence embeddings, coloured by region. The plot shows that regional stories largely share a common semantic space, with subtle outer-region contours revealing culturally specific variations within the same language.

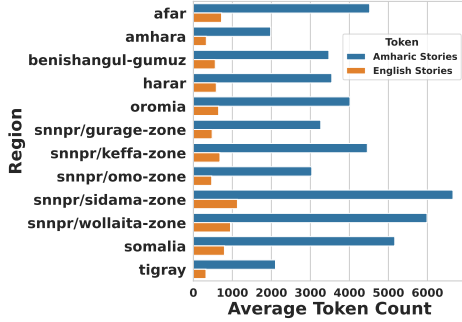


Figure 3: Average Stories Sequence Length by Region

story, whether the correct answer is fully supported by the narrative, and whether the distractors follow the intended structure (plausible, world-knowledge-biased, irrelevant, or factually false). **Linguistic Quality and Clarity** assesses whether the question is grammatically correct, fluent, and natural, particularly for Amharic, and whether it is easy to understand and translate. **Comprehension Depth and Challenge Level** examines whether the question tests meaningful understanding of the story, including inferential or reasoning-based comprehension rather than simple surface-level recall. We used Gwet’s AC1 to measure agreement between raters beyond chance, as it is more robust to class imbalance and prevalence effects. Exact annotation guideline can be found in Appendix F.

3.1.3 Manual Translations

We translated each question and its answer choices into Amharic using experienced annotators who had previously worked on translation tasks. The

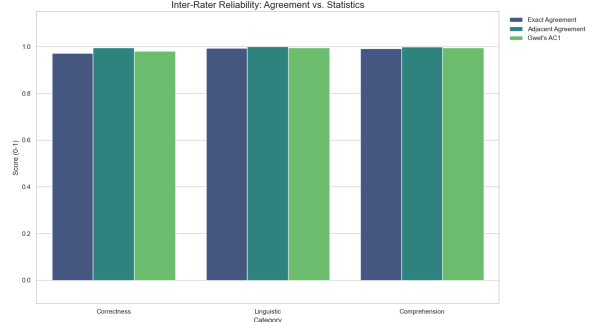


Figure 4: Inter-Rater Reliability agreement in Correctness, Linguistic, and Comprehension categories assessed with three human evaluators

translators were familiar with the narrative content, enabling them to preserve meaning, style, and cultural relevance across all generated questions.

3.1.4 SSA-COMET scoring

To evaluate machine translation (MT) quality for our low resource African language setting, we adopt both reference-based and reference-free evaluation metrics introduced in Li et al. (2025). These metrics allow us to systematically assess the quality of the human-translated Amharic outputs. Following prior work (Alabi et al., 2025a; Azime et al., 2025a), we flagged all translations with a score below 0.65 for manual inspection. While we found that many low-scoring cases were in fact false negatives produced by SSA-COMET (Li et al., 2025), reflecting known limitations of automatic evaluation in low-resource languages, we nonetheless identified and corrected a small number of gen-

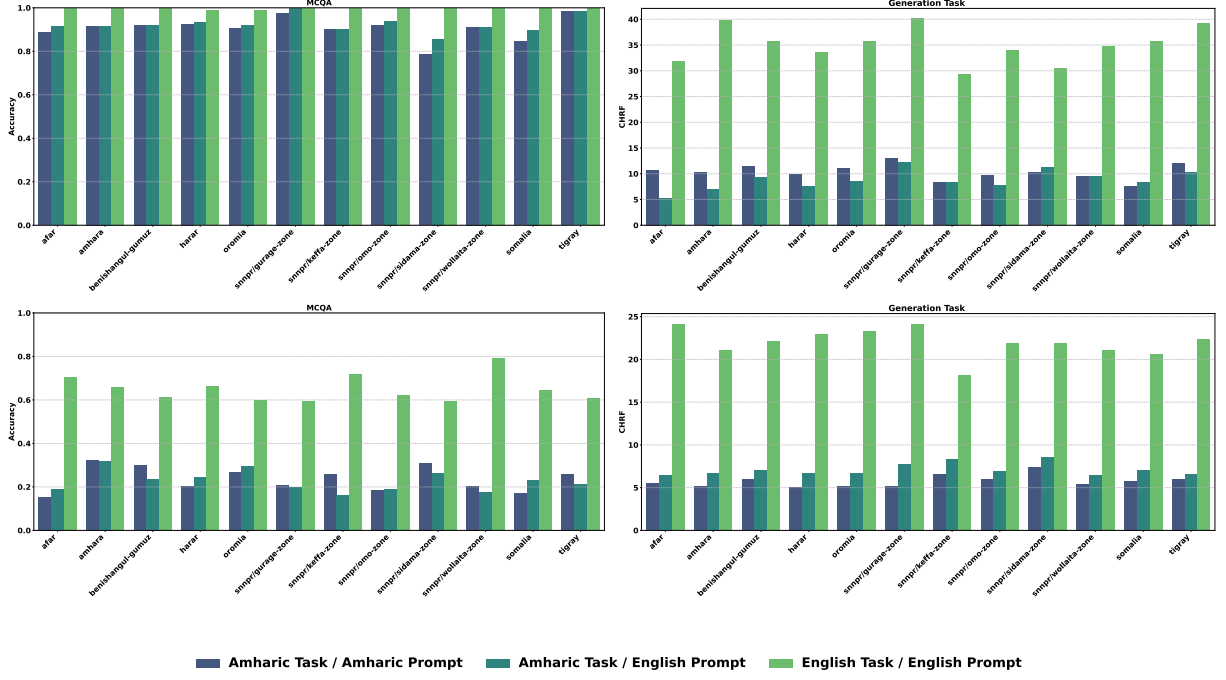


Figure 5: **Regional Performance.** Performance of the two differently performing models (top) (*Gemma-3-27b-it*) and (*Llama 3.2 1B Instruct*), showing performance variation across narratives from different regions in our dataset.

uine translation errors attributable to the translators. This combined automatic-plus-manual verification ensured that our final Amharic translations maintained high linguistic quality and fidelity to the original stories.

3.1.5 Train-Test Split

We sorted all story and question-answer pairs by their average SSA-COMET scores and used this ordering to create the train-test split. This approach ensures that translation inconsistencies do not disproportionately influence model evaluation and failures due to translation problems does not result evaluation problem.

After randomizing the answer choices from the structured-generation labels (where the correct answer was always originally assigned to option A) to simplify downstream quality control, we created a balanced 50/50 split for the benchmark. This split was lightly informed by the overall SSA-COMET score distribution to avoid translation-quality imbalance, resulting in 571 training instances and 649 test instances. This include instances where ssa-comet fails to capture translation quality according to human raters.

4 Experimental Setup

4.1 Evaluation

We selected 7 models that support a maximum context length of 128k tokens to align with the long-form nature of our tasks.

We conducted all evaluations using EleutherAI’s open-source Language Model Evaluation Harness (lm-eval) (Gao et al., 2024). The framework supports multiple evaluation strategies, including log-likelihood, text generation, and perplexity, with experiments configured and managed via configuration files. For open-source models, we employed log-likelihood and generation-based evaluation for multiple-choice tasks. In multiple-choice each candidate option is appended to the corresponding question prompt, after which the log-likelihood is computed. Model accuracy is then reported based on the option with the highest log-likelihood score.

4.2 Finetuning

To fine-tune LLLMs, we leveraged **LLaMA-Factory** (Zheng et al., 2024), an open-source framework that supports efficient fine-tuning through parameter-efficient methods and scalable training pipelines. We performed LoRA-based supervised fine-tuning using two A100 GPUs on English-only, Amharic-only, and combined multi-lingual datasets. All datasets were converted to the

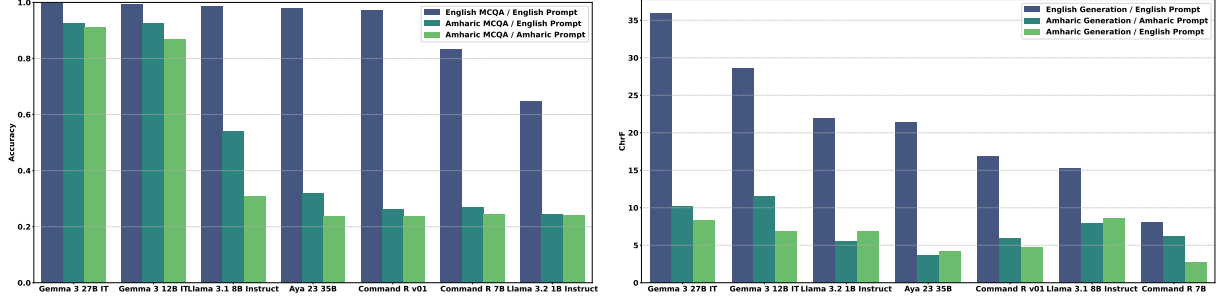


Figure 6: **Results on Amharic vs. English tasks.** Left: Accuracy on multiple-choice question answering (MCQA) tasks under different task–prompt language settings. Right: Generative performance measured using chrF under the same conditions.

Alpaca instruction format and will be shared as part of our released resources. Training was conducted for three epochs using a LoRA rank of 8 and an effective batch size of 8 via gradient accumulation. A systematic exploration and optimization of hyperparameters, including LoRA configuration and training dynamics, is left for future work.

4.3 Task choice

Previous work has shown that large language models’ performance on multiple-choice question answering (MCQA) tasks is heavily influenced by factors such as option ordering and other structural biases (Loginova et al., 2025). These limitations are further amplified in low-resource language settings, as demonstrated by recent analyses in (Azime et al., 2025b). Despite these known issues, MCQA remains one of the most widely used evaluation paradigms due to its ease of benchmarking, reproducibility, and straightforward metric computation. For this reason, we include MCQA tasks in our evaluation while complementing them with generative tasks, allowing for a more balanced assessment across different task formulations and reducing reliance on a single evaluation signal.

5 Result and Analysis

Cultural Differences Following Cecilia Liu et al. (2024), we visualize regional semantic variation using kernel density estimates (KDE) over 2D PCA projections of multilingual MiniLM sentence embeddings. As shown in Figure 2, the substantial overlap among regions suggests that stories written in Amharic largely occupy a shared semantic space, reflecting common narrative structures, themes, and stylistic conventions across Ethiopia. At the same time, the outer contour differences observed for specific regions reveal localized seman-

tic shifts, indicating that certain regions contribute stories with distinctive cultural motifs or domain-specific content. These patterns demonstrate that, although the language is the same, regional storytelling traditions introduce measurable semantic variation.

(RQ1) How well do Large Language Models comprehend Amharic narratives?

Multiple-choice question answering (MCQA).

As shown in the Figure 6 (Left), The MCQA results reveal a bimodal pattern in Amharic narrative comprehension. While Gemma models retain high accuracy on Amharic tasks (above 0.80%), all other models exhibit a sharp performance collapse to the 0.20–0.35 range, despite achieving great accuracy on English tasks. This separation suggests that cross-lingual narrative transfer is not gradual but model-dependent, likely reflecting differences in pretraining data or representational alignment rather than model size alone. Notably, English and Amharic prompting yield nearly identical performance for low-performing models, indicating a hard comprehension bottleneck where prompt language has diminishing returns. Moreover, strong English performance does not reliably predict Amharic comprehension: large multilingual models such as Aya 23 35B perform comparably to much smaller models on Amharic MCQA. Together, these findings indicate a structural limitation in Amharic narrative understanding rather than a failure of instruction following. Notably, Llama 3.1 8B Instruct exhibits the largest performance gap between English and Amharic prompting, suggesting that its narrative comprehension abilities are disproportionately stronger in English. This gap points to insufficient Amharic narrative exposure during training and motivates RQ3, where we in-

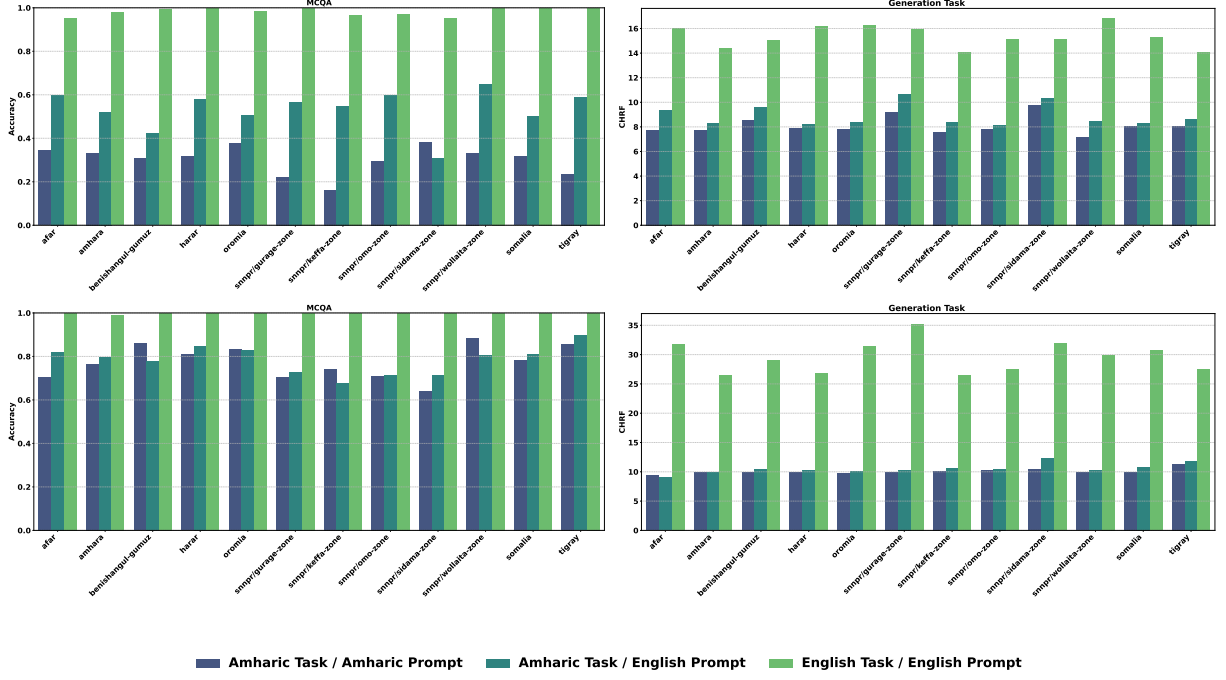


Figure 7: **Effect of SFT on AmharicStoryQA across regions.** Performance of *Llama 3.1 8B Instruct*(top) and its SFT variant trained on *AmharicStoryQA* (*SFT Llama 3.1 8B (All)*, bottom) on MCQA and generative evaluation.

investigate whether increased and targeted Amharic narrative data can mitigate this disparity.

Generative comprehension. As shown in Figure 6 (right), in contrast to the sharp performance collapse observed in MCQA, generative evaluation using chrF exhibits a more gradual degradation when narratives are presented in Amharic. This pattern indicates a consistent under-performance across all Amharic results, rather than abrupt failure on specific instances. While chrF scores drop substantially from English to Amharic tasks, models retain partial generation capability, suggesting preserved lexical and syntactic knowledge despite limited narrative understanding. Providing English prompts, in most cases, for Amharic narratives yields higher chrF scores than Amharic prompts, with gains of roughly 1–3 points, indicating improved surface-level fluency. However, these improvements do not correspond to higher MCQA accuracy, highlighting a dissociation between generative fluency and true comprehension.

Across both **MCQA and generative benchmarks**, models exhibit a consistent and systematic performance degradation when operating on Amharic narratives, preserving relative model rankings and prompt-language effects, which jointly indicate a structural limitation in Amharic narrative understanding rather than benchmark-specific

failure.

(RQ2) How cultural differences influence LLM performance in a single language? In Figure 3, we observe variability in sequence length across stories from different regions, highlighting region specific narrative characteristics beyond the semantic based analysis presented in Figure 2. In this research question, we aim to break down Amharic narrative understanding capabilities and examine how they are influenced by regional differences. Through manual exploration, we observe that stories often focus on region specific topics shaped by local traditions; for example, narratives from the Afar region where pastoralism is prevalent frequently involve camel herding and cattle theft.

To answer this question, Figure 5 examines two models with contrasting characteristics, *Gemma-3-27b-it* and *Llama 3.2 1B Instruct*, and analyzes their performance across regions. For English narrative understanding, we observe noticeable regional variation in most model–task settings, with differences of up to ± 10 chrF points or nearly 20% accuracy. An exception is the MCQA performance of *Gemma-3-27b-it*, which remains relatively stable across regions. In contrast, Amharic narrative understanding exhibits reduced regional variance, likely due to the uniformly low baseline performance of the models. Similarly, differences arising

Model Name (Trained Split)	language	prompt_language	MCQA (accuracy)	Generation(chrF)
<i>SFT Llama 3.1 8B (All)</i>	English	English	0.998	28.931
<i>SFT Llama 3.1 8B (Eng)</i>		English	0.992	23.901
<i>SFT Llama 3.1 8B (Amh)</i>		English	0.988	18.023
<i>Llama 3.1 8B Instruct</i>		English	0.986	15.283
<i>SFT Llama 3.1 8B (All)</i>	Amharic	English	0.802	10.410
<i>SFT Llama 3.1 8B (All)</i>		Amharic	0.787	10.086
<i>SFT Llama 3.1 8B (Amh)</i>		English	0.758	10.056
<i>SFT Llama 3.1 8B (Amh)</i>		Amharic	0.748	9.758
<i>SFT Llama 3.1 8B (Eng)</i>		Amharic	0.540	8.259
<i>SFT Llama 3.1 8B (Eng)</i>		English	0.309	7.661
<i>Llama 3.1 8B Instruct</i>		Amharic	0.383	7.944
<i>Llama 3.1 8B Instruct</i>		English	0.701	8.603

Table 1: Effect of training *Llama 3.1 8B Instruct* on different *AmharicStoryQA* subsets (English-only, Amharic-only, and combined) for narrative understanding.

from prompt language choice in Amharic are attenuated in regional evaluations, suggesting that regional effects are overshadowed by broader limitations in Amharic narrative comprehension.

(RQ3) Can we improve LLM performance using the culturally grounded training dataset we introduce, demonstrating the importance of such data for addressing the limitations observed?

Using both the English and Amharic training splits of the *AmharicStoryQA* dataset, we aim to improve the narrative understanding capabilities of existing LLMs and to demonstrate that our curated dataset can mitigate the narrative comprehension limitations observed in prior evaluations. We select *Llama 3.1 8B Instruct* as our primary model due to its comparatively low and variable performance in the benchmarking results shown in Figure 6. This choice allows us to more clearly assess the impact of our training data. In contrast to models such as the Gemma family, which exhibited stronger baseline performance, *Llama 3.1 8B Instruct* provides a suitable testbed for evaluating whether targeted supervision can substantially improve Amharic narrative understanding.

As shown in Table 1, LoRA fine-tuning on the combined and English-only splits leads to clear gains in English narrative comprehension, with the largest improvements observed in generative evaluation, reaching up to +13 chrF. In contrast, MCQA performance shows limited gains, likely due to near-saturated accuracy in English settings.

For Amharic narrative understanding, we observe substantially larger benefits, with MCQA accuracy improving by nearly 40% compared to the base instruction model. Notably, the performance gap between English and Amharic prompting narrows when training on the combined dataset, indicating reduced prompt-language sensitivity. More-

over, cross-lingual training, using English stories to improve Amharic evaluation and vice versa, consistently improves both MCQA and generative scores, providing evidence of effective cross-lingual narrative transfer.

Fine-tuned models and regional improvements.

Figure 7 shows that fine-tuning on *AmharicStoryQA* both improves MCQA performance and stabilizes results across regions, substantially reducing the regional disparities observed in the base model. This stabilization effect is particularly evident for Amharic narrative understanding, where fine-tuning leads to more consistent performance across diverse regional story distributions.

In contrast, regional differences are more pronounced for English narratives, even after fine-tuning. This suggests that fine-tuning primarily benefits Amharic narrative representations, while also revealing underlying regional characteristics that remain less influential for English benchmarks. As a result, fine-tuned models exhibit clearer regional variation in English tasks, indicating that regional narrative properties are more visible once the model’s overall comprehension improves.

6 Conclusion and Future Work

In this work, we investigated multicultural variation within a single language and its impact on long-form Amharic story question answering. We introduced *AmharicStoryQA*, a regionally grounded benchmark designed to capture cultural and narrative diversity among Amharic speakers, and used it to systematically evaluate the narrative understanding capabilities of large language models across regions. Our results show that cultural variation within the same language meaningfully affects story comprehension, revealing limitations in current LLMs that are not captured by standard,

language-level evaluations. In particular, we find that regional narrative characteristics influence both selection-based and generative understanding, highlighting the need for culturally aware evaluation and training data. For future work, we plan to extend this single-language, multi-culture analysis to a broader multilingual and multicultural setting.

Limitations

Due to the scarcity of volunteers and the high level of manual effort required, this study does not extend the analysis to multiple languages. As a result, we are unable to determine whether the observations reported in this work generalize across languages or are specific to Amharic.

References

- David Ifeoluwa Adelani, Jessica Ojo, Israel Abebe Azime, Jian Yun Zhuang, Jesujoba Oluwadara Alabi, Xuanli He, Millicent Ochieng, Sara Hooker, Andiswa Bukula, En-Shiun Annie Lee, Chiamaka Ijeoma Chukwuneke, Happy Buzaaba, Blessing Kudzaishie Sibanda, Godson Koffi Kalipe, Jonathan Mukiibi, Salomon Kabongo Kabenamualu, Foutse Yuehgoh, Mmasibidi Setaka, Lolwethu Ndolela, and 8 others. 2025. [IrokoBench: A new benchmark for African languages in the age of large language models](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2732–2757, Albuquerque, NM, USA. Association for Computational Linguistics.
- Jesujoba Oluwadara Alabi, Israel Abebe Azime, Miaoran Zhang, Cristina España-Bonet, Rachel Bawden, Dawei Zhu, David Ifeoluwa Adelani, Clement Oyeleke Odoje, Idris Akinade, Iffat Maab, Davis David, Shamsuddeen Hassan Muhammad, Neo Putini, David O. Ademuyiwa, Andrew Caines, and Dietrich Klakow. 2025a. [AFRIDOC-MT: Document-level MT corpus for African languages](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27758–27794, Suzhou, China. Association for Computational Linguistics.
- Jesujoba Oluwadara Alabi, Michael A. Hedderich, David Ifeoluwa Adelani, and Dietrich Klakow. 2025b. [Charting the landscape of African NLP: Mapping progress and shaping the road ahead](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 27795–27829, Suzhou, China. Association for Computational Linguistics.
- David Anugraha, Shou-Yi Hung, Zilu Tang, En-Shiun Annie Lee, Derry Wijaya, and Genta Winata. 2026. [mr3: Multilingual rubric-agnostic reward reasoning models](#). In *International Conference on Learning Representations*, volume 2026, pages 149218–149272.
- Israel Abebe Azime, Tadesse Destaw Belay, Dietrich Klakow, Philipp Slusallek, and Anshuman Chhabra. 2025a. [Bridging the Culture Gap: A Framework for LLM-Driven Socio-Cultural Localization of Math Word Problems in Low-Resource Languages](#). *arXiv preprint arXiv:2508.14913*.
- Israel Abebe Azime, Atnafu Lambebo Tonja, Tadesse Destaw Belay, Yonas Chanie, Bontu Fufa Balcha, Negasi Haile Abadi, Henok Biadgign Ademteu, Mulubrhan Abebe Nerea, Debela Desalegn Yadeta, Derartu Dagne Geremew, Assefa Atsbiha Tesfu, Philipp Slusallek, Thamar Solorio, and Dietrich Klakow. 2025b. [ProverbEval: Exploring LLM evaluation challenges for low-resource language understanding](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 6250–6266, Albuquerque, NM, USA. Association for Computational Linguistics.
- Israel Abebe Azime, Atnafu Lambebo Tonja, Tadesse Destaw Belay, Mitiku Yohannes Fuge, Aman Kassahun Wassie, Eyasu Shiferaw Jada, Yonas Chanie, Walegn Tewabe Sewunetie, and Seid Muhie Yimam. 2024. [Walia-LLM: Enhancing Amharic-LLaMA by integrating task-specific and generative datasets](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 432–444, Miami, FL, USA. Association for Computational Linguistics.
- Lucas Bandarkar, Davis Liang, Benjamin Muller, Mikel Artetxe, Satya Narayan Shukla, Donald Husa, Naman Goyal, Abhinandan Krishnan, Luke Zettlemoyer, and Madian Khabsa. 2024. [The Belebele Benchmark: a Parallel Reading Comprehension Dataset in 122 Language Variants](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 749–775, Bangkok, Thailand. Association for Computational Linguistics.
- Tadesse Destaw Belay, Seid Muhie Yimam, Abinew Ayele, and Chris Biemann. 2022. [Question answering classification for Amharic social media community based questions](#). In *Proceedings of the 1st Annual Meeting of the ELRA/ISCA Special Interest Group on Under-Resourced Languages*, pages 137–145, Marseille, France. European Language Resources Association.
- Thomas Berkane, Marie-Laure Charpignon, and Maimuna S. Majumder. 2025. [LLM-based web data collection for research dataset creation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12610–12622, Suzhou, China. Association for Computational Linguistics.
- Chen Cecilia Liu, Fajri Koto, Timothy Baldwin, and Iryna Gurevych. 2024. [Are multilingual LLMs culturally-diverse reasoners? an investigation into](#)

- [multicultural proverbs and sayings](#). 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 2016–2039, Mexico City, Mexico. Association for Computational Linguistics.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [A framework for few-shot language model evaluation](#).
- Shreya Havaldar, Sunny Rai, Bhumika Singhal, Langchen Liu, Sharath Chandra Guntuku, and Lyle Ungar. 2023. [Multilingual language models are not multicultural: A case study in emotion](#). In *Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis*, pages 202–214, Toronto, Canada. Association for Computational Linguistics.
- Senyu Li, Jiayi Wang, Felermio D. M. A. Ali, Colin Cherry, Daniel Deutsch, Eleftheria Briakou, Rui Sousa-Silva, Henrique Lopes Cardoso, Pontus Stenetorp, and David Ifeoluwa Adelani. 2025. [SSA-COMET: Do LLMs outperform learned metrics in evaluating MT for under-resourced African languages?](#) In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12979–12998, Suzhou, China. Association for Computational Linguistics.
- Olga Loginova, Oleksandr Bezrukov, Ravi Shekhar, and Alexey Kravets. 2025. [Addressing Blind Guessing: Calibration of Selection Bias in Multiple-Choice Question Answering by Video Language Models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3216–3246, Vienna, Austria. Association for Computational Linguistics.
- Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, and 3 others. 2024. [Blend: a benchmark for llms on everyday knowledge in diverse cultures and languages](#). In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS ’24*, Red Hook, NY, USA. Curran Associates Inc.
- Hellina Hailu Nigatu, Atnafu Lambebo Tonja, Benjamin Rosman, Thamar Solorio, and Monojit Choudhury. 2024. [The Zeno’s Paradox of ‘Low-Resource’ Languages](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17753–17774, Miami, Florida, USA. Association for Computational Linguistics.
- Odunayo Ogundepo, Tajuddeen R. Gwadabe, Clara E. Rivera, Jonathan H. Clark, Sebastian Ruder, David Ifeoluwa Adelani, Bonaventure F. P. Dos-sou, Abdou Aziz Diop, Claytone Sikasote, Gilles Hacheme, Happy Buzaaba, Ignatius Ezeani, Roowei-ther Mabuya, Salomey Osei, Chris Emezue, Albert Njoroge Kahira, Shamsuddeen Hassan Muhammad, Akintunde Oladipo, Abraham Toluwase Owodunni, and 33 others. 2023. [AfriQA: Cross-lingual open-retrieval question answering for African languages](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14957–14972, Singapore. Association for Computational Linguistics.
- Jessica Ojo, Odunayo Ogundepo, Akintunde Oladipo, Kelechi Ogueji, Jimmy Lin, Pontus Stenetorp, and David Ifeoluwa Adelani. 2025. [AfroBench: How good are large language models on African languages?](#) In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19048–19095, Vienna, Austria. Association for Computational Linguistics.
- David Romero, Chenyang Lyu, Haryo Akbarianto Wibowo, Teresa Lynn, Injy Hamed, Aditya Nanda Kishore, Aishik Mandal, Alina Dragonetti, Artem Abzaliev, Atnafu Lambebo Tonja, Bontu Fufa Balcha, Chenxi Whitehouse, Christian Salamea, Dan John Velasco, David Ifeoluwa Adelani, David Le Meur, Emilio Villa-Cueva, Fajri Koto, Fauzan Farooqui, and 56 others. 2024. [CVQA: Culturally-diverse Multilingual Visual Question Answering Benchmark](#). Preprint, arXiv:2406.05967.
- Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiawat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. [Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.
- Dipankar Srirag, Nihar Ranjan Sahoo, and Aditya Joshi. 2025. [Evaluating Dialect Robustness of Language Models via Conversation Understanding](#). In *Proceedings of the Second Workshop on Scaling Up Multilingual & Multi-Cultural Evaluation*, pages 24–38, Abu Dhabi, UAE. Association for Computational Linguistics.
- Tilahun Abedissa Taffa, Ricardo Usbeck, and Yaregal Assabie. 2024. [Low Resource Question Answering: An Amharic Benchmarking Dataset](#). In *Proceedings of the Fifth Workshop on Resources for African Indigenous Languages @ LREC-COLING 2024*, pages 124–132, Torino, Italia. ELRA and ICCL.

Sizhe Yuen, Ting Su, Ziyang Wang, Yali Du, and Adam J Sobey. 2025. [Automatic Dataset Generation for Knowledge Intensive Question Answering Tasks](#). *arXiv preprint arXiv:2505.14212*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. [Llamafactory: Unified efficient fine-tuning of 100+ language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.

Appendix

A Dataset Distribution in Regional

Stories depicting cultural practices, daily life, and narratives from the Amhara region constitute approximately 20% of the total *AmharicStoryQA* dataset. Stories referring specifically to the Tigray and Oromia regions account for 12% and 11% of the dataset, respectively. As shown in Figure 8, the remaining regions of Ethiopia collectively represent more than 50% of the dataset.

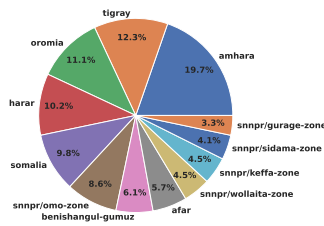


Figure 8: Dataset distribution in Region

B Ethiopian Regions

Tigray: is the northernmost regional state, bordering Eritrea to the north. It is the homeland of the Tigrayan people, who speak Tigrinya. The region is historically significant as the center of the ancient Aksumite Empire.

Oromia: is the largest regional state in Ethiopia by both population and land area, stretching across the central, western, and southern parts of the country. It is the homeland of the Oromo people, the largest ethnic group in Ethiopia, and the administrative center is Addis Ababa (Finfinne), which also serves as the national capital.

Amhara: is a regional state located in the north-western and central highlands of Ethiopia. It is the historic homeland of the Amhara people and is known for its agricultural potential and major tourist attractions like Lalibela and Gondar. The primary language spoken is Amharic.

Somali: is the second-largest region by area, located in the eastern part of Ethiopia bordering Somalia and Djibouti. It is predominantly inhabited by the Somali people who share a common language, culture, and Islamic heritage with Somalis in neighboring countries. The region is largely arid and known for pastoralism.

Sidama: is a regional state in southern Ethiopia formed in 2020 after separating from the former SNNPR. It is the homeland of the Sidama people, who are famous for their high-quality coffee pro-

duction, a major export for the country. The capital is Hawassa.

Afar: is located in the northeastern part of Ethiopia, home to the Afar people. The region is known for the Danakil Depression, one of the hottest and lowest places on Earth, and is geologically active. The inhabitants are traditionally pastoralists.

Benishangul-Gumuz: is located in western Ethiopia along the border with Sudan. It is home to diverse ethnic groups including the Berta and Gumuz.

Gambela: is situated in the western tip of Ethiopia, bordering South Sudan. It has a tropical climate and is inhabited primarily by the Nuer and Anuak peoples. The region is rich in wildlife and water resources, including the Baro River.

Harari: is the smallest regional state in Ethiopia, located in the east. It is the homeland of the Harari people and centers around the ancient walled city of Harar (Jugol), a UNESCO World Heritage site and a major center of Islamic culture in the Horn of Africa.

South West Ethiopia Peoples’ (SWEP): is a regional state in southwestern Ethiopia formed in 2021. It consists of the Kaffa (the birthplace of coffee), Sheka, Bench Sheko, Dawro, and West Omo zones, representing a diverse mix of ethnic groups and languages.

Central Ethiopia: is a newly formed regional state (established in August 2023) created from the northern part of the former Southern Nations, Nationalities, and Peoples’ Region (SNNPR). It is home to the Gurage, Hadiya, and Silte people, among others.

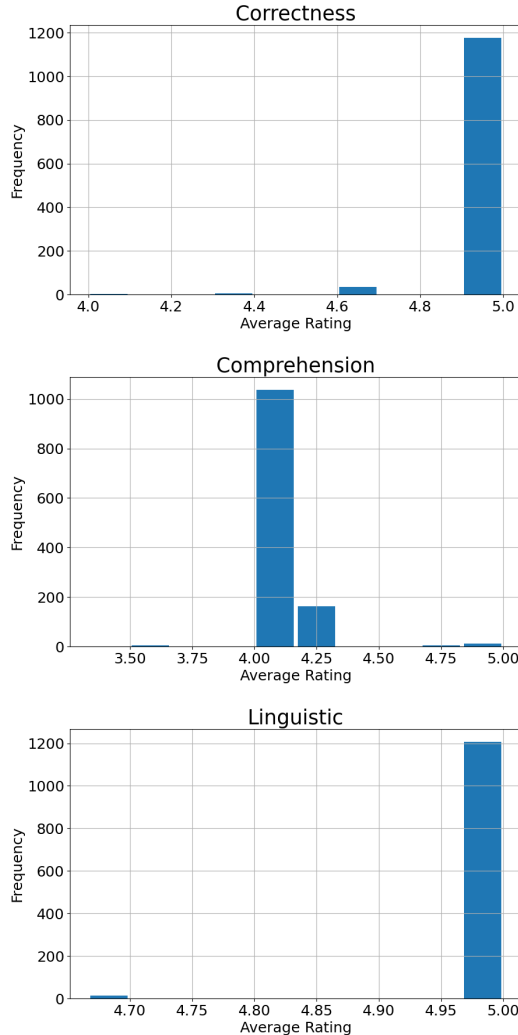
South Ethiopia: is a newly formed regional state (established in August 2023) created from the southern part of the former SNNPR. It includes zones such as Wolayta, Gamo, Gofa, and South Omo, which is famous for its diverse indigenous cultures.

Addis Ababa: is the capital city of Ethiopia and a chartered city (city administration). It serves as the headquarters for the African Union and is a melting pot of all Ethiopian ethnic groups.

Dire Dawa: is the second chartered city in Ethiopia, located in the eastern part of the country. It is a major commercial and industrial hub, historically developed around the Ethiopia-Djibouti Railway.

C Average Agreement Rating

Figure 9: Average rating of correctness and faithfulness, linguistic quality and clarity across three human evaluators



D Training Parameters

Fine-tuning Configuration. We fine-tune Meta-Llama-3-8B-Instruct using a supervised fine-tuning (SFT) stage with LoRA adapters. The training uses `lora_rank=8` with `lora_target=all` parameters while enabling `trust_remote_code`. Data preprocessing uses 16 workers and the dataloader uses 4 workers, with caching enabled and no limit on the number of samples. Training is performed with a per-device batch size of 1 and gradient accumulation of 8 steps, resulting in an effective batch size of 8. We train the model for 10 epochs using a learning rate of 1×10^{-4} with a cosine learning rate scheduler and a warmup ratio of 0.1. Mixed

precision training with bf16 is enabled. Model checkpoints are saved every 500 steps, training logs are recorded every 10 steps, and loss curves are plotted during training. All outputs are stored in the specified output directory with overwriting enabled when necessary.

E Question Generation

Question Generation

You are an expert educational content creator and an analytical text comprehension specialist. Your task is to generate challenging multiple-choice questions from the provided stories. These questions are designed to assess and diagnose language model reading comprehension.

Question Design Guidelines

- Focus on key story elements such as characters, events, causes, motivations, and outcomes.
- Ensure the question cannot be answered without understanding important parts of the story.
- Avoid vague or superficial questions.

Answer Choice Requirements

Each question must contain **four answer choices (A–D)** with the following roles:

A. Correct Answer

- The only correct option.
- Explicitly stated or directly derivable from the story.
- Requires no external knowledge.

B. Common-Sense Distractor

- Sounds plausible using real-world knowledge.
- Not supported by the story.

C. Unrelated Hallucination

- Completely unrelated to the story.
- Tests rejection of irrelevant information.

D. Factual Contradiction

- Directly contradicts a fact stated in the story.
- Must be clearly false given the text.

Input: Story: {story}

Output: Generate {num_questions} multiple-choice questions and return the result as a JSON list.

F Annotation Guidelines

Annotation Guidelines

1. Correctness and Faithfulness

Guiding Questions

- Is the question grounded in the story?
- Is the correct answer fully supported by the story?
- Do distractors follow the intended roles (plausible, opposite, irrelevant)?
- **5** Perfectly grounded in the story. Correct answer fully supported and all distractors follow their roles.
- **4** Mostly accurate with minor issues. One distractor may be weak or ambiguous.
- **3** Some factual mismatch or unclear reference. Distractors may be too similar.
- **2** Loosely based on the story. Correct answer weakly supported.
- **1** Major flaws. Distractors poorly constructed.
- **0** Completely invalid or unrelated to the story.

2. Linguistic Quality and Clarity

Guiding Questions

- Is the question grammatically correct and natural in Amharic?
- Is it fluent and easy to understand?
- **5** Excellent grammar and fluency.
- **4** Minor grammatical issues.
- **3** Understandable but awkward phrasing.
- **2** Difficult to understand.
- **1** Poor grammar and confusing phrasing.
- **0** Incomprehensible.

3. Comprehension Depth and Challenge Level

Guiding Questions

- Does the question test meaningful comprehension?
- Does it require reasoning beyond surface-level recall?
- **5** Deep reasoning or inference required.
- **4** Solid comprehension question.
- **3** Basic understanding required.
- **2** Too simple or shallow.
- **1** Trivial or irrelevant.
- **0** Completely unrelated to the task.

G Evaluation Prompt

Prompt 1 -- English:

Read the following story carefully and answer the question that follows.

Story:

"{story}"

Question:

"{question}"

Choices:

A) {choice1}

B) {choice2}

C) {choice3}

D) {choice4}

Your answer (A, B, C, or D):

Prompt 1 -- Amharic:

የሚከተለውን ታሪክ በጥንቃቄ ያንብቡ እና ከዚያ በኋላ የሚከተለውን ጥያቄ ይመልሱ።

ታሪክ:

"{story}"

ጥያቄ:

"{question}"

አማራጮች:

A) {choice1}

B) {choice2}

C) {choice3}

D) {choice4}

መልስዎ (A, B, C, ወይም D):

Prompt 2 -- English:

Read the story below carefully, paying attention to the details of what happens, and then answer the question that tests your understanding of it.

Story:

"{story}"

Question:

"{question}"

Choices:

A) {choice1}

B) {choice2}

C) {choice3}

D) {choice4}

Your answer (A, B, C, or D):

Prompt 2 -- Amharic:

ከታች ያለውን ታሪክ በጥንቃቄ ያንብቡ፤ የተከሰቱትን ነገሮች ዝርዝሮች በትክክል ይጠብቁ፤
ከዚያም የገባዎትን ለመፈተሽ የሚከተለውን ጥያቄ ይመልሱ።

ታሪክ:

"{story}"

ጥያቄ:

"{question}"

አማራጮች:

A) {choice1}

B) {choice2}

C) {choice3}

D) {choice4}

መልስዎ (A, B, C, ወይም D):

Prompt 3 -- English:

Carefully read the following passage to understand the events, emotions, and main ideas it conveys, then choose the most accurate answer to the question that follows.

Story:

"{story}"

Question:

"{question}"

Choices:

- A) {choice1}
- B) {choice2}
- C) {choice3}
- D) {choice4}

Your answer (A, B, C, or D):

Prompt 3 -- Amharic:

የሚከተለውን ጽሑፍ በጥንቃቄ ያንብቡ እና ውስጡ የተገለጹትን ክስተቶች፣ ስሜቶች እና ዋና ሀሳቦች በትክክል ለመረዳት ይሞክሩ። ከዚያም የሚከተለውን ጥያቄ ለመመለስ ትክክለኛውን አማራጭ ይምረጡ።

ታሪክ:

"{story}"

ጥያቄ:

"{question}"

አማራጮች:

- A) {choice1}
- B) {choice2}
- C) {choice3}
- D) {choice4}

መልስዎ (A, B, C, ወይም D):

What Improves Multimodal Misinformation Detection? Answers from a Large-Scale Empirical Study

Akshit Sharma and Prashant W. Patil

CVPR Lab, MFSDSAI

Indian Institute of Technology Guwahati

akshitsharma.rs@gmail.com pwpatil@iitg.ac.in

Abstract

Multimodal misinformation is increasingly crafted to look convincing by pairing a textual claim with an image that appears to “prove” it. Yet in practice, building effective detectors often hinges on a small set of design choices that are rarely examined in a controlled way. In this paper, we conduct a large-scale study of multimodal design choices for misinformation detection with over 3,375 experiments spanning three benchmark datasets and a broad range of pre-trained vision and language backbones. Through systematic comparisons and targeted robustness analyses, we distill practical guidance on which design choices help, when do they fail silently, and what aspects of the pipeline most strongly shape model behavior, answering 4 key Research Questions (RQs). We aim to provide a reliable foundation for designing stronger and more dependable multimodal misinformation detection systems, thus contributing to the broader research community.

1 Introduction

Misinformation on social media and online news is increasingly multimodal: the core claim is expressed through text, while an accompanying image is often interpreted as corroborating evidence. In practice, however, the image may be irrelevant, out-of-context, weakly related, or intentionally manipulated, yet still persuasive enough to mislead readers. This makes credibility assessment difficult for unimodal systems, since text-only models cannot verify visual evidence, while image-only models cannot interpret what is being claimed. As a result, multimodal misinformation classification has become a standard framing for fake-news detection, rumor detection, and image–text inconsistency analysis (Abdali et al., 2024; Comito et al., 2023; Jin et al., 2017; Luo et al., 2021). A large body of work addresses this setting, but the design space is now broader than fusion alone. Att-

RNN (Jin et al., 2017) integrates textual, visual, and social-context signals; EANN (Wang et al., 2018) targets event-specific bias; SpotFake (Singhal et al., 2019) and SpotFake+ (Singhal et al., 2020) show the strength of transfer-learning pipelines built from pre-trained encoders; MVAE (Khattar et al., 2019) studies shared latent spaces; and models such as SAFE (Zhou et al., 2020), MCNN (Xue et al., 2021), and recent out-of-context image–text verification work such as NewsCLIPPings (Luo et al., 2021) and SNIFFER (Qi et al., 2024) explicitly target cross-modal mismatch. Adjacent image–text classification tasks expose similar design tensions. In multimodal meme classification, CLIP-based methods investigate explicit cross-modal interaction and lightweight adaptation of pretrained representations (Kumar and Nandakumar, 2022; Shah et al., 2024), while more recent work explores keyword-guided representation learning (Sharma and Patil, 2026a) and local–global pivotal-cue grounding with frozen CLIP features (Sharma and Patil, 2026b). Although these tasks target harm, hate, or sarcasm rather than veracity, they motivate the same broader question of how encoder choice, fusion, and modality reliance shape robust image–text classification.

Across this literature, however, several recurring design factors are typically entangled: the visual backbone, the text backbone, the level of multimodal interaction, the downstream head, and the model’s dependence on each modality. Yet a basic question remains unresolved: *when it comes to multimodal misinformation classification, which of these design factors actually contribute to model performance and which don’t?* Prior work often varies several factors at once, or reports only the best benchmark-specific configuration, making it difficult to isolate what actually drives performance (Abdali et al., 2024; Comito et al., 2023; Chen et al., 2023). This matters because many practical pipelines rely on frozen pre-trained en-

coders and lightweight heads, where small design decisions can dominate the final result. In this paper, we conduct a controlled study of multimodal misinformation detection that isolates each design component and evaluate the resulting benefits, trade-offs at scale. Concretely, we define three fusion strategies based on the abstraction level being combined—*early* fusion (feature-level), *mid* fusion (score-level), and *late* fusion (decision-level)—and evaluate them as part of the design factors under a unified protocol. We then address four key research questions:

- **RQ1. Generalized Performance Trends:** Which end-to-end performance patterns are seen to remain stable across different large-scale multimodal misinformation benchmarks?
- **RQ2. Modality Complementarity:** When do text and image contribute complementary evidence, and when does one modality dominate when it comes to multimodal misinformation detection?
- **RQ3. Impact of Each Component on Overall Performance:** Which pipeline components—visual encoder, text encoder, fusion strategy, or classifier, exert the most reliable effect on the classification performance?
- **RQ4. Benchmark-Shift Generalization:** To what extent do strong in-benchmark design choices and observed performance trends generalize under transfer to unseen misinformation benchmarks?

Across the three benchmark datasets, we conduct a total of 3,375 experiments—consisting of 675 in-domain model evaluations, 1,350 modality-drop evaluations, and 1,350 cross-dataset transfer evaluations. **Contributions.**

1. **RQ-based Analysis:** We provide a unified empirical analysis of performance trends, modality dependence, component robustness, and transfer under benchmark shift, providing answers to the 4 key research questions.
2. **Evidence at scale:** 675 + 1,350 + 1,350 = 3,375 controlled experiments across three benchmark datasets a unified protocol and diverse backbone combinations, supporting reproducible and broadly applicable conclusions.

3. **Statistically grounded conclusions:** We report confidence intervals, paired hypothesis tests, and transfer analyses that separate benchmark-stable findings from dataset-contingent effects.

2 Methodology

2.1 Problem Formulation

Let $\mathcal{D} = \{(x_i^{\text{img}}, x_i^{\text{txt}}, y_i)\}_{i=1}^N$ be a multimodal misinformation dataset, where x_i^{img} is an image, x_i^{txt} is the associated text, and $y_i \in \{1, \dots, C\}$ is the class label. We learn a classifier

$$f : \mathcal{X}_{\text{img}} \times \mathcal{X}_{\text{txt}} \rightarrow \{1, \dots, C\}$$

using a modular pipeline with a vision encoder $\phi_v : \mathcal{X}_{\text{img}} \rightarrow \mathbb{R}^{d_v}$ and a text encoder $\phi_t : \mathcal{X}_{\text{txt}} \rightarrow \mathbb{R}^{d_t}$:

$$\begin{aligned} z^{\text{img}} &= \phi_v(x^{\text{img}}) \in \mathbb{R}^{d_v}, \\ z^{\text{txt}} &= \phi_t(x^{\text{txt}}) \in \mathbb{R}^{d_t}. \end{aligned}$$

Fusion combines $(z^{\text{img}}, z^{\text{txt}})$ into a joint representation, which is then passed to the final classifier.

2.2 Encoders and Classifiers Used

2.2.1 Vision Encoders

We instantiate ϕ_v using five pre-trained vision backbones that span both convolutional and transformer-based families: ResNet-50 (He et al., 2016), ViT-B/16 (Dosovitskiy et al., 2021), CLIP ViT-B/32 (Radford et al., 2021), SigLIP (Zhai et al., 2023), and VGG-16 (Simonyan and Zisserman, 2015). This set covers standard CNNs, vanilla vision transformers, and image encoders trained with image–text pretraining objectives.

This selection is included to make the visual encoder a controlled experimental factor rather than a single default backbone. VGG-16 and ResNet-50 represent canonical convolutional architectures, with VGG-16 providing a plain deep CNN baseline and ResNet-50 representing residual CNNs. ViT-B/16 introduces a transformer-based visual encoder that operates over image patches rather than convolutional feature maps. CLIP ViT-B/32 and SigLIP add vision towers trained through paired image–text supervision, which is especially relevant for misinformation detection because the visual evidence must be interpreted relative to a textual claim. Holding the downstream pipeline fixed while varying these encoders allows us to test whether performance is driven mainly by convolutional versus transformer architecture, by unimodal versus

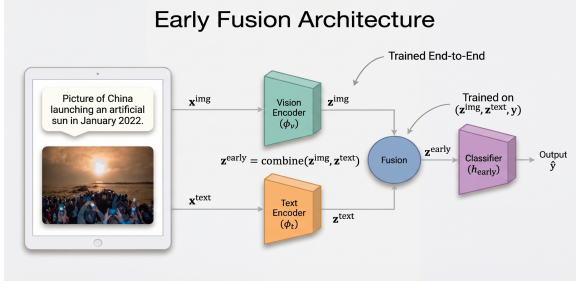


Figure 1: Figure showcasing the early fusion architecture used as part of our experiments

multimodal pretraining, or by the specific visual representation learned by each backbone.

2.2.2 Text Encoders

We instantiate ϕ_t using five pre-trained language backbones: BERT-base (Devlin et al., 2019), DistilBERT (Sanh et al., 2019), SBERT (Reimers and Gurevych, 2019), CLIP-Text (Radford et al., 2021), and SigLIP-Text (Zhai et al., 2023). Together, these models cover standard bidirectional transformers, distilled encoders, sentence-oriented encoders, and text towers trained jointly with images.

This set is included to make the text encoder a separate controlled axis of the study. BERT-base provides a standard bidirectional Transformer baseline for contextual token-level language representations, while DistilBERT tests whether a smaller distilled encoder preserves enough semantic signal for the same downstream task. SBERT represents sentence-level representation learning, which is useful because misinformation examples are classified from complete claims or captions rather than isolated tokens. CLIP-Text and SigLIP-Text represent text towers trained under image–text alignment objectives, making them natural counterparts to multimodal vision encoders. Comparing these choices allows us to distinguish gains from general language understanding, sentence-level semantic embedding quality, and multimodal alignment.

2.2.3 Classifiers

We use three lightweight heads on top of frozen multimodal representations: logistic regression (Cox, 1958), random forest (Breiman, 2001), and XGBoost (Chen and Guestrin, 2016). This set provides a clean comparison between a standardized linear classifier, a bagging-based ensemble, and a boosting-based ensemble. Including logistic regression is important in our setting because frozen pre-trained representations can often be evaluated effectively with simple linear probes (Kornblith et al., 2019), making a strong linear baseline

difficult to beat.

These classifiers are included because the goal is to evaluate the representations and fusion strategies rather than to hide their effects behind a large trainable prediction head. Logistic regression provides the lowest-capacity linear probe: if it performs well, the frozen multimodal representation already exposes task-relevant information in a nearly linearly separable form. Random forest adds non-linear decision boundaries through bagged decision trees, while XGBoost provides a stronger boosting-based alternative that can capture feature interactions more aggressively. Using these three lightweight heads therefore gives an interpretable range of classifier capacity while keeping the main comparison focused on the encoders, fusion level, and modality dependence.

2.2.4 Fusion Architectures

Early fusion performs **feature-level** fusion by concatenating the image and text embeddings:

$$z^{\text{early}} = [z^{\text{img}}, z^{\text{text}}] \in \mathbb{R}^{d_v + d_t},$$

$$\hat{y} = h_{\text{early}}(z^{\text{early}}).$$

This is a single-stage scheme with no intermediate unimodal decision layer.

Mid fusion is a **score-level** two-stage scheme. We first train unimodal classifiers h^{img} and h^{text} and extract unimodal score representations $\ell^{\text{img}}, \ell^{\text{text}} \in \mathbb{R}^C$. These are concatenated and passed to a final fusion classifier:

$$z^{\text{mid}} = [\ell^{\text{img}}, \ell^{\text{text}}] \in \mathbb{R}^{2C}, \quad \hat{y} = h_{\text{mid}}(z^{\text{mid}}).$$

Late fusion is a **decision-level** two-stage scheme. We first obtain unimodal probabilities $p^{\text{img}}, p^{\text{text}} \in [0, 1]^C$, concatenate them, and train a final classifier on the resulting vector:

$$z^{\text{late}} = [p^{\text{img}}, p^{\text{text}}] \in \mathbb{R}^{2C}, \quad \hat{y} = h_{\text{late}}(z^{\text{late}}).$$

2.3 Benchmark Setup

We evaluate on three large-scale multimodal misinformation benchmarks: MMFakeBench, MiRA-GeNews, and VLDBench (Liu et al., 2025; Huang et al., 2024; Raza et al., 2026). MMFakeBench contains 11,000 image–text pairs, MiRA-GeNews contains 15,000 pairs across its training, validation, and test partitions, and VLDBench contains 31,339 article–image pairs. Together, they cover mixed-source content, AI-generated multimodal news, and vision–language disinformation.

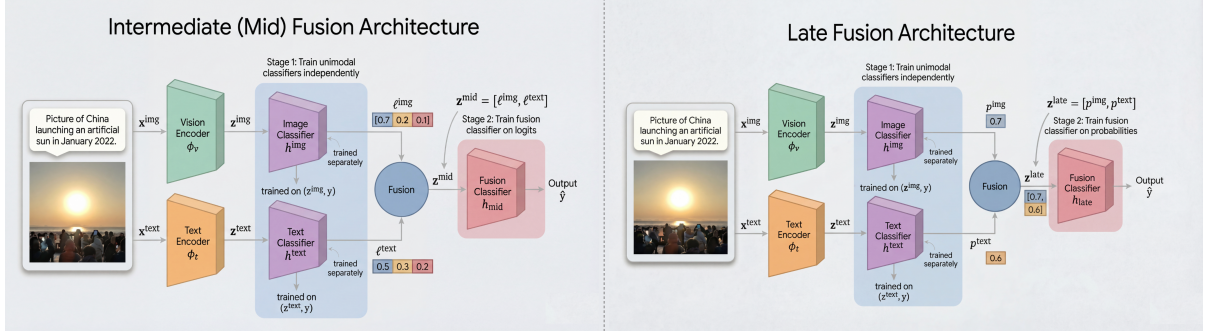


Figure 2: Figure showcasing the mid and late fusion architecture used as part of our experiments

For consistent supervised evaluation, we construct 80/20 train–test partitions for all three datasets, resulting in 8,800/2,200 examples for MMFakeBench, 12,000/3,000 for MiRAGeNews, and 25,071/6,268 for VLDBench. For MiRAGeNews, we remove 841 records with missing images—673 from the training partition and 168 from the test partition—leaving 14,159 instances (11,327/2,832). The final study comprises 56,498 examples: 11,000 from MMFakeBench, 14,159 from MiRAGeNews, and 31,339 from VLDBench. For cross-dataset transfer, we train on the constructed source training partition and evaluate on the target test partition.

2.4 Implementation Details

All encoders are frozen, and only the downstream classifier heads are trained. Logistic regression uses standardized inputs, the lbfgs solver, and $\text{max_iter}=2000$; random forest uses 300 trees; and XGBoost uses 300 trees, a 0.05 learning rate, maximum depth 6, row and column subsampling of 0.8, histogram-based tree construction, and a binary logistic objective. Feature extraction uses batch sizes of 16 and 32 for vision and text, respectively, with a maximum text length of 64. All experiments run on an NVIDIA RTX A4000 GPU with 16 GB VRAM.

Mid and late fusion use leakage-free stacking. For each outer training split, the unimodal branches generate out-of-fold predictions or features using shuffled 5-fold StratifiedKFold with a fixed random seed. The fusion classifier is trained only on these outputs, ensuring that no example both trains a unimodal model and contributes its in-sample prediction to the fusion model. The unimodal branches are then retrained on the full outer training split to generate outputs for the held-out test split, which is never used for training, calibration, or model selection. Cross-dataset transfer follows the same protocol: out-of-fold features and the fusion model are learned only from the source training set, and

the frozen source-trained pipeline is evaluated directly on the target test set.

Statistical analysis. For all four research questions, we report Macro-Precision, AUROC, and Accuracy with 95% confidence intervals. Macro-Precision is the primary metric because false-positive credibility flags can be costly, while AUROC and Accuracy assess ranking quality and operating-point behavior. Aggregate comparisons over matched configurations use bootstrap confidence intervals. Paired comparisons use Wilcoxon signed-rank tests, with Holm correction for multiple pairwise tests; multi-level component comparisons use Friedman tests followed by Holm-corrected Wilcoxon post-hoc tests. We report Wilson score intervals for proportions, such as the frequency with which early fusion wins, and use exact McNemar tests for selected instance-level comparisons.

3 Results and Discussion

We organize the results by the four research questions. Unless otherwise noted, all table entries are mean percentages over matched configurations, and comparative claims such as “best,” “strongest,” or “default” refer to Macro-Precision, with AUROC and Accuracy reported as secondary diagnostics.

3.1 RQ1: Generalized Performance Trends

Based on Table 2, one in-domain trend is exceptionally stable: **early fusion is the only strategy that is best on average across all three datasets and metrics**. Over 225 matched in-domain configurations, it achieves 76.02% Macro-Precision (95% CI [74.58, 77.50]), 82.92% AUROC ([81.46, 84.39]), and 76.87% Accuracy ([75.51, 78.26]), compared with 74.23 / 81.64 / 76.04 for mid fusion and 74.09 / 81.37 / 75.91 for late fusion. Its Macro-Precision advantage is significant over both mid and late fusion ($p = 1.00 \times 10^{-8}$ and 5.45×10^{-11}), and

it wins 136/225 matched blocks, or 60.4% with Wilson interval [53.9, 66.6].

Classifier rankings support exclusion rather than a single universal winner. **Random forest is the weakest head across all three datasets and metrics**, while logistic regression and XGBoost form the competitive pair in Table 1. Logistic regression is nominally best in Macro-Precision at 76.42%, but is tied with XGBoost at 76.34% ($p = 0.160$). XGBoost achieves the highest AUROC at 84.07%, significantly above logistic regression at 83.33% ($p = 0.0022$), whereas logistic regression has the highest Accuracy at 78.08%, with no significant difference from XGBoost’s 77.93% ($p = 0.175$). XGBoost leads on MMFakeBench and VLDBench, while logistic regression leads on MiRAGeNews. Thus, random forest should be avoided, and the final choice should be between logistic regression and XGBoost depending on whether Macro-Precision or AUROC is prioritized.

The single-dataset evidence is strongest on VLDBench and MMFakeBench, where early fusion improves Macro-Precision by roughly two to three points over the best alternative, while preserving the same ordering on MiRAGeNews. This benchmark-consistent ranking shows that the result is not merely a small average effect across runs. When the evaluation target is unknown, early fusion is therefore the safest in-domain default for Macro-Precision, with AUROC and Accuracy providing consistent support.

3.2 RQ2: Modality Complementarity

The modality-drop analysis still supports complementarity, but the dominant pattern is not symmetric synergy; it is **image-heavy reliance**. Table 3 shows the full multimodal result together with both single-modality drop tests for every dataset–fusion–classifier block. Across all 675 matched comparisons, removing text reduces Macro-Precision by 4.83 points (95% CI [4.18, 5.53]), AUROC by 3.93 ([3.65, 4.23]), and Accuracy by 5.78 ([5.28, 6.31]), whereas removing the image stream reduces them by 13.42 [12.20, 14.69], 12.08 [11.48, 12.66], and 16.93 [16.10, 17.79] points, respectively. Full multimodal models significantly outperform both text-drop and image-drop variants overall on all three metrics (all $p < 10^{-78}$), and the difference between image-drop and text-drop degradation is also highly significant (Macro-Precision: $p = 8.99 \times 10^{-46}$; AUROC: $p = 2.26 \times 10^{-66}$; Accuracy: $p = 1.31 \times 10^{-68}$). The asymmetry is

strongly dataset dependent, but the direction is now uniform across all three benchmarks. MiRAGeNews is by far the most image-dependent benchmark: removing images causes mean losses of 21.68 points in Macro-Precision, 17.05 in AUROC, and 26.62 in Accuracy, far larger than the corresponding text-removal losses of 6.48, 3.27, and 8.52 points. MMFakeBench is the most balanced benchmark, although image removal is still worse on average (9.62 / 10.60 / 10.01 versus 6.61 / 6.42 / 5.33). VLDBench shows the smallest text losses and remains image-dominated as well (8.96 / 8.60 / 14.17 versus 1.39 / 2.11 / 3.49). Over all configurations, image removal hurts more than text removal in 76.3% [72.9, 79.4] of Macro-Precision comparisons, 80.0% [76.8, 82.8] of AUROC comparisons, and 82.4% [79.3, 85.1] of Accuracy comparisons. The practical conclusion is therefore not merely that multimodal models help, but that their gains are real, asymmetric, and usually driven by harder-to-replace visual evidence.

3.3 RQ3: Component Impact Analysis

The component-level picture remains stable and practically actionable under the revised metric protocol. All four component families exhibit significant overall effects under Friedman testing on Macro-Precision, AUROC, and Accuracy, but their magnitudes differ substantially. Vision encoder choice is dominant on all three metrics (all $p < 8 \times 10^{-94}$), with average per-dataset spreads of 7.46 points on Macro-Precision, 7.34 on AUROC, and 6.95 on Accuracy. Classifier choice is the next largest source of stable variation (5.32 / 5.75 / 5.68 points), whereas text encoder choice (1.99 / 2.33 / 1.70) and fusion strategy (1.93 / 1.55 / 0.97) are materially smaller. Put differently, once a reasonable text encoder is in place, most of the robust variance comes from the image tower and the downstream head.

Two trends are especially stable. First, **SigLIP is the strongest vision encoder on all three reported metrics**, reaching 79.19% Macro-Precision, 86.01% AUROC, and 80.37% Accuracy overall, with CLIP ViT-B/32 ranking second at 77.43 / 84.93 / 78.67. Post-hoc testing confirms that SigLIP significantly exceeds CLIP ViT-B/32 after Holm correction on all three metrics ($p_{\text{Holm}} = 7.40 \times 10^{-15}$, 2.53×10^{-20} , and 5.67×10^{-15}). Second, **text encoders matter less and separate much less cleanly**. DistilBERT has the strongest overall in-domain text means at 75.17 / 82.57 /

Classifier	Metric	MMFakeBench	MiRAGeNews	VLDBench
Logistic Regression	Prec	73.93 (± 1.07)	91.65 (± 0.95)	63.68 (± 0.48)
	AUROC	83.55 (± 1.18)	96.80 (± 0.54)	69.63 (± 0.67)
	Acc	77.87 (± 0.87)	91.65 (± 0.95)	64.73 (± 0.45)
XGBoost	Prec	74.51 (± 0.74)	90.00 (± 0.83)	64.52 (± 0.32)
	AUROC	85.45 (± 0.82)	96.18 (± 0.48)	70.60 (± 0.42)
	Acc	78.29 (± 0.59)	89.99 (± 0.83)	65.53 (± 0.30)
Random Forest	Prec	71.06 (± 1.31)	83.34 (± 1.29)	60.32 (± 0.68)
	AUROC	79.79 (± 1.27)	90.94 (± 1.02)	64.86 (± 0.92)
	Acc	73.74 (± 0.65)	83.32 (± 1.29)	61.37 (± 0.67)

Table 1: Classifier-wise in-domain averages (%) over all encoder pairs and fusion strategies. Values are mean $\pm$ 95% bootstrap CI half-width across matched configurations.

Fusion	Metric	MMFakeBench	MiRAGeNews	VLDBench
Early	Prec	75.20 (± 1.07)	88.80 (± 1.15)	64.06 (± 0.41)
	AUROC	83.73 (± 1.20)	95.10 (± 0.76)	69.93 (± 0.59)
	Acc	76.79 (± 0.85)	88.77 (± 1.16)	65.06 (± 0.40)
Mid	Prec	72.22 (± 1.07)	88.20 (± 1.41)	62.27 (± 0.71)
	AUROC	82.78 (± 1.26)	94.51 (± 1.02)	67.63 (± 0.98)
	Acc	76.62 (± 0.86)	88.19 (± 1.41)	63.31 (± 0.70)
Late	Prec	72.09 (± 1.07)	88.00 (± 1.39)	62.20 (± 0.71)
	AUROC	82.28 (± 1.22)	94.31 (± 0.99)	67.52 (± 0.96)
	Acc	76.48 (± 0.85)	87.99 (± 1.38)	63.25 (± 0.69)

Table 2: Fusion-wise in-domain averages (%) over all encoder pairs and classifier heads. Values are mean $\pm$ 95% bootstrap CI half-width across matched configurations.

76.67, but the top four text encoders are all within 0.32 Macro-Precision points of one another, and DistilBERT does not significantly exceed BERT-base or CLIP-Text after Holm correction. This is exactly the pattern one would expect from a component family that matters, but matters less than the vision side.

Classifier conclusions mirror RQ1. Logistic regression and XGBoost again form the leading pair: logistic regression has the highest overall Macro-Precision and Accuracy means (76.42 and 78.08), but both are statistically tied with XGBoost ($p = 0.160$ and 0.175), whereas XGBoost has the strongest AUROC at 84.07%, significantly above logistic regression ($p = 0.0022$). Random forest is substantially behind on every metric. Figure 3 reinforces the same conclusion visually: variation across vision rows is larger and more systematic than variation across most text columns. Taken together, the component analysis suggests a clear priority order for future system design: get the vision encoder right first, choose between logistic regression and XGBoost as the serious downstream heads, then refine the fusion strategy, and only then expect modest gains from swapping among strong text encoders such as DistilBERT, BERT-base, CLIP-Text, and SBERT.

3.4 RQ4: Benchmark-Shift Generalization

Benchmark shift is the most challenging setting. As Table 4 shows, all six source–target transfers degrade substantially relative to their target in-domain baselines. Macro-Precision gaps range from 8.33 points for MMFakeBench→VLDBench to 36.30 for VLDBench→MiRAGeNews; the corresponding AUROC and Accuracy gaps range from 11.76 to 41.57 and from 6.45 to 36.93 points, respectively. The highest transfer Macro-Precision is achieved by MMFakeBench→MiRAGeNews at 63.25% (± 0.59), followed by MiRAGeNews→MMFakeBench at 62.56% (± 0.53), whereas VLDBench→MiRAGeNews exhibits the largest overall collapse. Every transfer direction is significantly worse than its target in-domain baseline on all three metrics (paired Wilcoxon tests, all $p \leq 1.15 \times 10^{-38}$).

Although transfer rankings are flatter than in-domain rankings, early fusion remains the strongest precision-first default. Across 450 matched transfer blocks, it achieves 56.16% Macro-Precision, significantly exceeding mid fusion (55.51%, $p = 1.73 \times 10^{-5}$) and late fusion (55.45%, $p = 1.47 \times 10^{-5}$). It also leads in AUROC at 58.78%, while Accuracy is nearly tied across fusion strategies. Direction-specific reversals occur—MiRAGeNews→MMFakeBench slightly favors

Dataset	Fusion	Classifier	Full	Text drop	Δ_{text}	Image drop	Δ_{img}
MMFakeBench	Early	Logistic Reg.	72.2/82.0/76.6	64.4/75.2/68.2	7.8/6.8/8.5	62.8/71.0/65.7	9.4/11.0/11.0
		XGBoost	76.2/86.2/79.0	71.0/79.2/72.5	5.2/7.1/6.5	63.4/72.9/65.7	12.9/13.3/13.3
		Random Forest	77.2/83.0/74.7	74.2/76.8/72.8	3.0/6.1/1.9	58.4/72.3/71.0	18.8/10.7/3.8
	Mid	Logistic Reg.	74.9/84.9/78.6	66.4/78.0/72.3	8.4/6.9/6.3	69.2/77.7/73.2	5.7/7.2/5.4
		XGBoost	73.8/85.2/78.0	69.5/81.0/73.6	4.3/4.2/4.4	67.2/76.2/61.9	6.6/9.0/16.2
		Random Forest	68.0/78.2/73.2	62.3/70.4/68.5	5.7/7.8/4.8	58.0/65.1/64.2	10.0/13.1/9.0
	Late	Logistic Reg.	74.7/83.7/78.4	59.2/78.0/72.4	15.5/5.7/6.0	68.4/77.7/72.9	6.3/6.0/5.5
		XGBoost	73.5/84.9/77.8	69.5/79.6/72.9	4.0/5.3/4.9	66.4/73.0/60.7	7.1/12.0/17.1
		Random Forest	68.0/78.2/73.2	62.4/70.4/68.5	5.6/7.8/4.7	58.1/65.1/64.3	9.8/13.1/8.9
MiRAGeNews	Early	Logistic Reg.	91.6/96.9/91.6	86.0/93.8/83.4	5.6/3.1/8.3	77.1/82.7/61.9	14.6/14.2/29.7
		XGBoost	89.8/96.1/89.7	84.5/92.4/81.9	5.3/3.7/7.9	67.4/79.1/64.5	22.3/17.0/25.2
		Random Forest	85.0/92.3/84.9	79.9/87.8/73.4	5.1/4.5/11.5	67.4/76.2/62.4	17.6/16.1/22.5
	Mid	Logistic Reg.	91.9/97.0/91.9	82.9/93.5/75.0	9.0/3.5/16.8	78.8/87.3/64.6	13.1/9.7/27.3
		XGBoost	90.2/96.3/90.2	86.5/93.9/84.8	3.7/2.4/5.4	66.6/80.2/65.0	23.6/16.1/25.1
		Random Forest	82.5/90.3/82.5	79.2/86.9/78.1	3.4/3.4/4.5	64.1/64.0/55.2	18.4/26.3/27.3
	Late	Logistic Reg.	91.4/96.5/91.4	72.1/93.5/78.6	19.4/3.0/12.8	47.9/87.3/60.7	43.6/9.2/30.8
		XGBoost	90.0/96.2/90.0	86.5/93.7/85.0	3.6/2.5/5.1	66.5/77.7/65.6	23.6/18.5/24.4
		Random Forest	82.5/90.3/82.5	79.1/86.9/78.0	3.4/3.4/4.5	64.1/63.9/55.3	18.4/26.3/27.2
VLDBench	Early	Logistic Reg.	62.9/68.4/63.9	62.8/67.3/58.6	0.1/1.1/5.4	55.8/58.5/48.7	7.1/9.9/15.3
		XGBoost	65.0/71.3/66.0	63.7/69.3/63.3	1.4/2.0/2.7	59.8/59.4/46.3	5.2/11.9/19.7
		Random Forest	64.2/70.1/65.2	62.6/67.7/62.7	1.6/2.4/2.5	45.3/59.2/46.6	18.9/10.9/18.6
	Mid	Logistic Reg.	64.1/70.2/65.1	63.1/68.3/59.6	1.0/1.9/5.5	51.1/64.5/52.0	13.0/5.7/13.1
		XGBoost	64.4/70.4/65.4	64.1/69.6/63.9	0.3/0.9/1.5	59.4/61.7/52.3	4.9/8.8/13.1
		Random Forest	58.4/62.2/59.4	56.3/59.3/56.8	2.1/2.9/2.7	54.9/56.2/49.8	3.5/6.1/9.6
	Late	Logistic Reg.	64.1/70.3/65.1	61.6/68.3/60.1	2.5/2.0/5.0	46.8/64.5/51.4	17.3/5.8/13.7
		XGBoost	64.2/70.0/65.2	62.7/67.4/61.7	1.5/2.7/3.4	57.0/57.9/50.3	7.1/12.2/14.8
		Random Forest	58.3/62.2/59.4	56.3/59.3/56.8	2.0/3.0/2.6	54.8/56.1/49.8	3.5/6.1/9.7

Table 3: Modality-drop study. Each cell reports Macro-Precision/AUROC/Accuracy (%). “Full” is the complete multimodal model, and Δ_{text} and Δ_{img} are performance drops in percentage points when one modality is removed at test time. Detailed entries are rounded to one decimal for readability.

Train → Test	Macro-Precision	AUROC	Accuracy
MMFakeBench → MiRAGeNews	63.25 (± 0.59)	67.70 (± 0.70)	57.84 (± 0.57)
MMFakeBench → VLDBench	54.51 (± 0.33)	56.60 (± 0.25)	57.42 (± 0.14)
MiRAGeNews → MMFakeBench	62.56 (± 0.53)	69.26 (± 0.70)	55.44 (± 0.70)
MiRAGeNews → VLDBench	49.25 (± 0.15)	48.88 (± 0.22)	44.85 (± 0.19)
VLDBench → MMFakeBench	52.64 (± 0.26)	54.69 (± 0.46)	48.68 (± 0.39)
VLDBench → MiRAGeNews	52.03 (± 0.35)	53.07 (± 0.52)	51.39 (± 0.27)

Table 4: Cross-dataset transfer by source–target direction. Values are mean ± 95% bootstrap CI half-width (%).

mid fusion, and VLDBench→MiRAGeNews favors late fusion—but do not alter the aggregate ranking. Table 5 shows that other component rankings also flatten under transfer. XGBoost has the highest Macro-Precision at 56.31% but is statistically tied with logistic regression at 56.21% ($p = 0.264$). SigLIP leads the vision encoders at 57.73% but is tied with CLIP ViT-B/32 at 57.49% ($p = 0.819$), while BERT-base leads the less-stable text encoders at 56.00%, only 0.19 points above SBERT.

Overall, the safest precision-first transfer configuration uses early fusion, SigLIP or CLIP ViT-B/32, BERT-base or SBERT, and either XGBoost or logistic regression. Source-specific ef-

fects may change the finer ranking, but strong visual representations and early fusion remain the most dependable choices. Cross-dataset evaluation should therefore be treated as a first-class requirement, as even strong in-domain models can lose more than 30 points under benchmark shift.

3.5 SHAP-Based Interpretation

We also use SHAP to test whether the modality conclusions are visible in feature attributions rather than only in modality-drop ablations. Across 675 in-domain SHAP analyses, image features account for a mean 61.57% of total attribution mass (95% bootstrap CI [60.52, 62.61]). Image attribution exceeds text attribution in 81.6% of configurations,

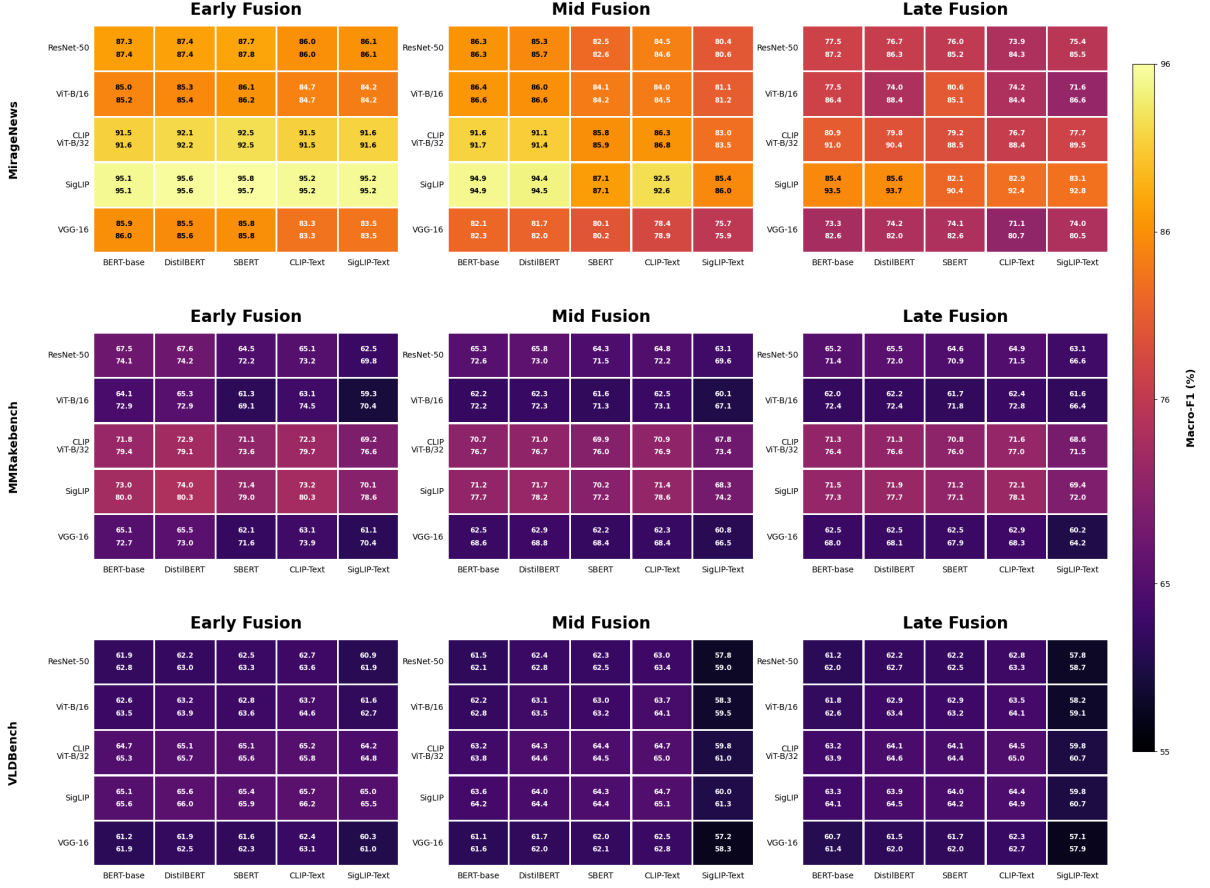


Figure 3: Main encoder-pair heatmap across the three datasets. Each cell reports mean Macro-Precision / Accuracy (%) for a fixed vision–text encoder pair, averaged over the three classifier heads under the given fusion strategy; color encodes AUROC.

and the paired image-vs-text difference is highly significant ($p = 7.24 \times 10^{-50}$). This pattern is not a fusion-specific artifact: the mean image share is essentially identical for early, mid, and late fusion (all about 61.6%).

The dataset breakdown mirrors RQ2 almost exactly. MMFakeBench is the most balanced benchmark at 53.8% mean image share, MiRAGENews is the most visually dominated at 67.1%, and VLD-Bench remains clearly image-leaning at 63.8%. In the best-performing SHAP-analyzed configurations for each dataset, the top individual contributors are overwhelmingly SigLIP dimensions, which independently reinforces the RQ3 conclusion that visual encoder quality—and especially strong contrastive vision backbones—explains a large share of stable multimodal performance.

3.6 Practical Recommendations

The aggregate results point to two concrete default approaches for future multimodal misinformation studies. If the goal is strong in-domain performance under frozen representations, a robust Macro-Precision-oriented starting point is

SigLIP on the visual side, with CLIP ViT-B/32 as the main alternative, DistilBERT on the text side, early fusion, and either logistic regression or XGBoost as the downstream head. This recommendation is supported by the in-domain averages: early fusion is the strongest fusion strategy overall on Macro-Precision (76.02%), SigLIP is the strongest vision encoder (79.19%), DistilBERT has the strongest text-encoder mean (75.17%), and logistic regression and XGBoost form the leading classifier pair, with logistic regression nominally highest on Macro-Precision (76.42%) and XGBoost essentially tied (76.34%).

If the focus is stronger OOD performance, the default should be chosen for transfer stability rather than in-domain dominance. Our results suggest early fusion with a strong contrastive vision encoder, especially SigLIP or CLIP ViT-B/32, together with a strong generic text encoder such as BERT-base or SBERT, and either XGBoost or logistic regression as the final head. This recommendation follows the transfer-wide Macro-Precision ranking: early fusion is best overall (56.16%), SigLIP is the strongest vision encoder (57.73%)

(a) Classifier-wise transfer averages by target dataset				
Classifier	Metric	MMFakeBench	MiRAGeNews	VLDBench
Logistic Regression	Prec	58.22 ($\pm$ 0.97)	58.77 ($\pm$ 1.02)	51.64 ($\pm$ 0.52)
	AUROC	63.10 ($\pm$ 1.46)	62.02 ($\pm$ 1.32)	52.80 ($\pm$ 0.59)
	Acc	52.38 ($\pm$ 0.85)	55.21 ($\pm$ 0.73)	51.00 ($\pm$ 1.05)
XGBoost	Prec	58.22 ($\pm$ 1.01)	58.45 ($\pm$ 1.12)	52.26 ($\pm$ 0.55)
	AUROC	62.62 ($\pm$ 1.39)	61.89 ($\pm$ 1.53)	52.82 ($\pm$ 0.74)
	Acc	51.89 ($\pm$ 1.02)	55.99 ($\pm$ 0.84)	51.24 ($\pm$ 1.07)
Random Forest	Prec	56.36 ($\pm$ 0.87)	55.70 ($\pm$ 1.05)	51.73 ($\pm$ 0.51)
	AUROC	60.19 ($\pm$ 1.28)	57.24 ($\pm$ 1.22)	52.60 ($\pm$ 0.69)
	Acc	51.92 ($\pm$ 0.79)	52.64 ($\pm$ 0.58)	51.17 ($\pm$ 0.93)

(b) Fusion-wise transfer averages by target dataset				
Fusion	Metric	MMFakeBench	MiRAGeNews	VLDBench
Early	Prec	57.69 ($\pm$ 0.92)	58.74 ($\pm$ 1.21)	52.04 ($\pm$ 0.57)
	AUROC	62.28 ($\pm$ 1.33)	60.96 ($\pm$ 1.49)	53.09 ($\pm$ 0.70)
	Acc	52.55 ($\pm$ 0.92)	53.92 ($\pm$ 0.73)	51.50 ($\pm$ 0.99)
Mid	Prec	57.59 ($\pm$ 0.96)	57.11 ($\pm$ 1.00)	51.82 ($\pm$ 0.49)
	AUROC	61.90 ($\pm$ 1.38)	60.19 ($\pm$ 1.35)	52.58 ($\pm$ 0.66)
	Acc	51.74 ($\pm$ 0.86)	54.98 ($\pm$ 0.77)	50.91 ($\pm$ 1.02)
Late	Prec	57.51 ($\pm$ 0.94)	57.07 ($\pm$ 0.99)	51.78 ($\pm$ 0.52)
	AUROC	61.74 ($\pm$ 1.35)	60.01 ($\pm$ 1.33)	52.55 ($\pm$ 0.68)
	Acc	51.90 ($\pm$ 0.84)	54.95 ($\pm$ 0.77)	50.99 ($\pm$ 1.02)

Table 5: Cross-dataset transfer averages by component family and target dataset. Values are mean $\pm$ 95% bootstrap CI half-width (%).

but statistically tied with CLIP ViT-B/32 (57.49%), BERT-base is the strongest text encoder (56.00%) with SBERT close behind (55.81%), and XGBoost is nominally best among classifiers (56.31%) but statistically tied with logistic regression (56.21%). The main shift from the in-domain recommendation is therefore on the text side: DistilBERT is the strongest in-domain text choice, whereas under benchmark shift the more reliable text defaults become BERT-base and SBERT.

4 Conclusion

In this paper, we presented a large-scale empirical study of multimodal misinformation detection, covering 3,375 controlled experiments across three benchmarks, multiple vision and text encoders, fusion strategies, and classifiers. Our findings show that early fusion is relatively the most reliable in-domain strategy, visual evidence often contributes more strongly than text, and vision encoder and classifier choices have the largest impact on performance. We also find that cross-dataset generalization remains challenging, highlighting the need for stronger transfer evaluation. We hope that these findings and protocols will serve as a useful reference for the broader research community.

Limitations

Our study intentionally focuses on frozen encoders and simple downstream heads. This choice improves comparability and isolates component effects, but it does not cover fully trainable cross-attention architectures or parameter-efficient adaptation methods. Finally, statistical significance should not be confused with practical magnitude: because we aggregate many matched configurations, some small gaps are estimated precisely without necessarily justifying additional system complexity. Future work should therefore extend the benchmark scope and pair significance testing with stronger effect-size, calibration, and more in-depth analyses.

References

- Sara Abdali, Sina Shaham, and Bhaskar Krishnamachari. 2024. [Multi-modal misinformation detection: Approaches, challenges and opportunities](#). *ACM Computing Surveys*, 57(3):76:1–76:29.
- Leo Breiman. 2001. [Random forests](#). *Machine Learning*, 45:5–32.
- Jinyin Chen, Chengyu Jia, Haibin Zheng, Ruoxi Chen, and Chenbo Fu. 2023. [Is multi-modal necessarily better? robustness evaluation of multi-modal fake news](#)

- detection. *IEEE Transactions on Network Science and Engineering*, 10:3144–3158.
- Tianqi Chen and Carlos Guestrin. 2016. **XGBoost: A scalable tree boosting system**. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.
- Carmela Comito, Luciano Caroprese, and Ester Zumpano. 2023. **Multimodal fake news detection on social media: A survey of deep learning techniques**. *Social Network Analysis and Mining*, 13(1):101.
- D. R. Cox. 1958. **The regression analysis of binary sequences**. *Journal of the Royal Statistical Society: Series B (Methodological)*, 20(2):215–232.
- 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*, pages 4171–4186.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. **Deep residual learning for image recognition**. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.
- Runsheng Huang, Liam Dugan, Yue Yang, and Chris Callison-Burch. 2024. **MIRAGeNews: Multimodal realistic AI-generated news detection**. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16436–16448, Miami, Florida, USA. Association for Computational Linguistics.
- Zhiwei Jin, Juan Cao, Han Guo, Yongdong Zhang, and Jiebo Luo. 2017. **Multimodal fusion with recurrent neural networks for rumor detection on microblogs**. In *Proceedings of the 25th ACM International Conference on Multimedia*, pages 795–816.
- Dhruv Khattar, Jaipal Singh Goud, Manish Gupta, and Vasudeva Varma. 2019. **MVAE: Multimodal variational autoencoder for fake news detection**. In *The World Wide Web Conference*, pages 2915–2921.
- Simon Kornblith, Jonathon Shlens, and Quoc V. Le. 2019. **Do better ImageNet models transfer better?** In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2661–2671.
- Gokul Karthik Kumar and Karthik Nandakumar. 2022. **Hate-CLIPper: Multimodal hateful meme classification based on cross-modal interaction of CLIP features**. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 171–183, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.
- Xuannan Liu, Zekun Li, Pei Li, Huaibo Huang, Shuhan Xia, Xing Cui, Linzhi Huang, Weihong Deng, and Zhaofeng He. 2025. **MMFakeBench: A mixed-source multimodal misinformation detection benchmark for LVLMS**. In *International Conference on Learning Representations*.
- Grace Luo, Trevor Darrell, and Anna Rohrbach. 2021. **NewsCLIPPings: Automatic generation of out-of-context multimodal media**. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6801–6817.
- Peng Qi, Zehong Yan, Wynne Hsu, and Mong Li Lee. 2024. **SNIFFER: Multimodal large language model for explainable out-of-context misinformation detection**. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13052–13062.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.
- Shaina Raza, Ashmal Vayani, Aditya Jain, Aravind Narayanan, Vahid Reza Khazaie, Syed Raza Bashir, Elham Dolatabadi, Gias Uddin, Christos Emmanouilidis, Rizwan Qureshi, and Mubarak Shah. 2026. **VLD Bench evaluating multimodal disinformation with regulatory alignment**. *Information Fusion*, 130:104092.
- 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*, pages 3982–3992.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. **DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter**. Preprint, arXiv:1910.01108.
- Siddhant Bikram Shah, Shuvam Shiwakoti, Maheep Chaudhary, and Haohan Wang. 2024. **MemeCLIP: Leveraging CLIP representations for multimodal meme classification**. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17320–17332, Miami, Florida, USA. Association for Computational Linguistics.

- Akshit Sharma and Prashant W. Patil. 2026a. [MemeTAG: Keyword-driven meme classification through tag embedding reconstruction](#). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 7679–7688.
- Akshit Sharma and Prashant W. Patil. 2026b. [Small cues, big consequences: Learning pivotal cues for multimodal meme classification](#). *Preprint*, arXiv:2609.26907. Accepted to Findings of the Association for Computational Linguistics: EMNLP 2026.
- Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.
- Shivangi Singhal, Anubha Kabra, Mohit Sharma, Rajiv Ratn Shah, Tanmoy Chakraborty, and Ponnurangam Kumaraguru. 2020. [SpotFake+: A multi-modal framework for fake news detection via transfer learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13915–13916.
- Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin’ichi Satoh. 2019. [SpotFake: A multi-modal framework for fake news detection](#). In *2019 IEEE Fifth International Conference on Multimedia Big Data (BigMM)*, pages 39–47.
- Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. [EANN: Event adversarial neural networks for multi-modal fake news detection](#). In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 849–857.
- Junxiao Xue, Yabo Wang, Yichen Tian, Yafei Li, Lei Shi, and Lin Wei. 2021. [Detecting fake news by exploring the consistency of multimodal data](#). *Information Processing & Management*, 58(5):102610.
- Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. [Sigmoid loss for language image pre-training](#). In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11975–11986.
- Xinyi Zhou, Jindi Wu, and Reza Zafarani. 2020. [SAFE: Similarity-aware multi-modal fake news detection](#). In *Advances in Knowledge Discovery and Data Mining*, volume 12085 of *Lecture Notes in Computer Science*, pages 354–367. Springer, Cham.

How Well Does It Generalize?

A Robustness Study of Hausa Machine-Generated Text Detection

Abubakar Abdulhamid^{1,6}, Faisal Muhammad Adam^{2,6},
Abubakar Auwal Khalid^{3,6}, Babangida Sani^{4,6}, Shamsuddeen Hassan Muhammad^{5,6}

¹Nasarawa State University, Keffi ²Maryam Abacha American University of Nigeria

³Usman Danfodiyo University, Sokoto ⁴Bayero University, Kano

⁵Imperial College London ⁶HausaNLP

abdulhamid@ab-bkr.com

Abstract

Sani et al. (2025) introduced the first dedicated Hausa machine-generated text (MGT) detector, an AfroXLMR-large classifier trained on 2,586 paired articles from seven Hausa news outlets and Gemini 2.0 Flash, reporting F1 of 0.9921 on a headline-grouped split. They identified three limitations of their work: a single news domain, a small batch and short training schedule, and a single generator (Gemini 2.0 Flash) for the machine-generated data. We extend their work along the third axis. First, we replicate their reported result across four random seeds, obtaining F1 of 0.992 ± 0.005 . Test recall on the AI class is 1.000 at every seed. Second, we analyse the detector’s false positives. All errors fall on Human articles, and two native-speaker reviewers confirm these are genuine, natural Hausa journalism. The clearest correlate of misclassification is the absence of numerical content: the median digit fraction is 0.0% in misclassified articles vs. 0.6% in correctly classified ones. Third, we test the detector on six non-training generators spanning four model families. Detection rates span from 0% (Aya-Expanse 8B) to nearly 100% (Gemma 4). All three Google-family models in our experiment (training-distribution Gemini 2.0 Flash and the two non-training Gemma 4 variants) are detected at rates above 98% with mean $P(\text{machine})$ above 0.98. No non-Google generator exceeds 21%. Three contamination checks fail to explain this away: generated articles overlap very little with their human sources, detection barely moves when we remove every article that shares a 13-gram with its source, and the Google output that gets detected is at least as diverse as everyone else’s, so it is not memorised or templated text. What the detector has learned looks less like a general Hausa MGT signal and more like a Gemini-family stylistic fingerprint. We discuss what this means for deployment, including a 3.1% false-positive rate on real Hausa journalists.

1 Introduction

Machine-generated text detection has become a pressing concern for content moderation, academic integrity, and combating misinformation. Most published detectors and benchmarks focus on English. Hausa, a Chadic language with over 100 million speakers across West Africa (Inuwa-Dutse, 2021) and a major media language in Nigeria, has only recently received attention.

The imbalance matters beyond coverage. Detection tools shape who gets believed. A detector that misfires on genuine human writing casts suspicion on real journalists; one that misses machine text lets fabricated news circulate. Hausa is widely spoken but thinly resourced, and independent fact-checking infrastructure is scarcer than for English, so the reliability of a single published detector carries unusual weight. Knowing when that detector can be trusted, and when it cannot, is part of what equitable access to trustworthy information means for the language.

Sani et al. (2025) released the first dedicated Hausa MGT detection benchmark at AfricaNLP 2025. The dataset pairs 2,586 articles scraped from seven Hausa news outlets with 2,586 articles generated by Gemini 2.0 Flash from the same headlines. They fine-tuned four Afri-centric pre-trained models on the resulting dataset, with AfroXLMR-large achieving the highest F1 at 0.9921. They split the data with no headline overlap across train/validation/test to avoid leakage, and acknowledged three limitations: single domain (news), short training schedule (three epochs, batch 8), and a single generator (Gemini 2.0 Flash) for the AI class.

This paper takes up the third limitation directly. The single-generator concern is the most consequential for deployment: a detector trained on one model’s output may fail on output from other generators in unpredictable ways. We also examine two

related questions that the original paper did not. We re-train the AfroXLMR-large detector across four random seeds to quantify run-to-run variance, and analyse the detector’s false positives to characterise what the model has learned.

We organise the study around three research questions:

RQ1: Does the reported AfroXLMR-large result replicate across random seeds?

RQ2: What kinds of Human articles does the detector falsely flag as machine-generated?

RQ3: Does a detector trained on Gemini 2.0 Flash transfer to non-training generators?

Our contributions are:

1. A multi-seed replication of [Sani et al. \(2025\)](#)’s baseline across four random seeds (mean F1 = 0.992 ± 0.005). The reported number sits at our mean, confirming the result is robust to initialisation. Recall on the AI class is 1.000 at all four seeds.
2. An error analysis showing that all detector errors are false positives on genuine human-written news (confirmed by two native-speaker reviewers). The strongest correlate of misclassification is missing numerical content. We expected the article’s register to matter, but native-speaker review did not bear that out.
3. A cross-generator evaluation across six non-training generators (Cohere Aya, Meta Llama 3.1, Alibaba Qwen 3, Google Gemma 4 E4B, Google Gemma 4 E2B, Meta NLLB-200 round-trip). Detection rates span from 0% to 100%. Both non-training Google-family models are detected at rates above 98%, matching the training generator’s 100% rate; no non-Google generator exceeds 21%. Our reading is that the detector has learned a Gemini-family stylistic signature, not a general MGT signal.

2 Related Work

2.1 MGT detection

Machine-generated text detection has matured since the release of GPT-3-class models. Zero-shot statistical approaches ([Gehrmann et al., 2019](#);

[Mitchell et al., 2023](#)) and supervised classifiers fine-tuned on paired human/AI corpora are the dominant families. Cross-generator transfer has been studied repeatedly in English. The RAID benchmark ([Dugan et al., 2024](#)) and the M4 multilingual benchmark ([Wang et al., 2024](#)) both find that supervised detectors trained on output from one generator fail to transfer to output from other generators. We extend this line of work to Hausa, where we find the failure is not uniform: transfer ranges from total to nonexistent depending on the generator’s family.

2.2 African-language MGT detection and resources

[Sani et al. \(2025\)](#) is the first dedicated Hausa MGT detection study to our knowledge. Related work on African-language NLP has focused on translation ([Team et al., 2022](#); [Adelani et al., 2022](#)), language identification, and general-purpose model development ([Üstün et al., 2024](#); [Dang et al., 2024](#)). The AfroXLMR family of models that powers the detector ([Alabi et al., 2022](#); [Adelani et al., 2024](#)) is adapted from XLM-R via multilingual adaptive fine-tuning on African languages, including Hausa. Robustness studies for African-language NLP systems remain rare.

2.3 Hausa orthography

Standard written Hausa uses four hooked consonants (ḃ, ḉ, ɓ, y’), which fall outside the ASCII range. These characters are commonly mishandled in NLP pipelines. We apply Unicode repair ([Speer, 2019](#)) to generator inputs in our cross-generator experiments to ensure non-Hausa-trained models receive valid orthography.

3 Methodology

3.1 Data

We use the [Sani et al. \(2025\)](#) release without modification. The dataset contains 5,172 articles across two label classes: 2,586 Human articles scraped from seven Hausa media outlets and 2,586 AI articles produced by Gemini 2.0 Flash from the corresponding headlines. We follow their headline-grouped 80/10/10 split: each headline is assigned to one of three splits, and both the Human and AI article for that headline go together.

The released dataset contains a class-correlated asymmetry in Hausa orthography. Human articles average 29.3 hooked-consonant occurrences per article after Unicode normalisation; Gemini articles

average 14.4 (a $2.0\times$ ratio). 72.7% of Human articles contain at least one hooked letter, vs. 37.9% of Gemini articles. The class composition is also subdomain-skewed: 53.5% politics, 10.4% health, 9.7% sports, with smaller shares for business, entertainment, religion, and technology (Sani et al., 2025).

3.2 Detector training

We follow Sani et al. (2025)’s hyperparameters: AfroXLMR-large with learning rate 1×10^{-5} , weight decay 0.01, maximum sequence length 512, and effective batch size 32 (per-device batch 8, gradient accumulation 4). We train for up to 8 epochs with early stopping on validation F1 (patience 2), an extension of their 3 epochs. The longer schedule is motivated by inspection of validation curves on seed 42: validation F1 was 0.961 at epoch 3 (matching the paper’s reported number) but peaked at 0.990 at epoch 6 before declining slightly. The original 3-epoch schedule appears to have stopped training short of the achievable peak, though the paper’s hyperparameters reach the reported test F1 within run-to-run variance regardless.

We train at fp32 due to a classifier-head initialisation issue in transformers 5.8.1 that produces NaN weights when the classification head is newly added. We re-initialise the head manually with $\mathcal{N}(0, 0.02)$ weights and zero biases. We run four random seeds (42, 1, 7, 123) and report mean and standard deviation across runs. The seed-42 checkpoint is used for all subsequent experiments unless noted.

3.3 Cross-generator data

For each of six non-training generators, we produce 260 Hausa articles paired with test-split Human articles. Each generator is given a chat-template prompt asking it to write a Hausa news article matching the topic of an input headline. We test:

- **Aya-Expanse 8B** (Dang et al., 2024): Cohere; explicit African-language training.
- **Llama 3.1 8B Instruct** (Grattafiori et al., 2024): Meta; English-centric.
- **Qwen 3 8B** (Yang et al., 2025): Alibaba; 119-language multilingual pre-training.
- **Gemma 4 E4B** (DeepMind, 2026): Google; 140-language multilingual.

- **Gemma 4 E2B** (DeepMind, 2026): smaller variant of the same family.
- **NLLB-200 distilled 600M** (Team et al., 2022): Meta; translation paradigm, Hausa $\rightarrow$ English $\rightarrow$ Hausa round-trip.

All generators run via vLLM with bf16, temperature 0.8, top-p 0.95. Headline inputs are passed through ftfy (Speer, 2019) to repair mojibake (Appendix B). For thinking-capable models (Qwen 3, Gemma 4), thinking mode is explicitly disabled via the chat template so the full output-token budget is available for visible output.

The NLLB round-trip serves a second purpose. Because it preserves meaning while scrambling surface stylistic features through a translation pivot, it functions as a paraphrase-attack probe. We report the same detection number under both interpretations.

3.4 Evaluation

The trained detector is applied to each generator’s 260-article set and to test-split Human articles. We report the percentage classified as AI, mean and median P(machine), hooked-consonant density per 1,000 characters, and digit density. For Aya specifically, we additionally compare detection on output produced from mojibake-corrupted headlines (our first generation run) vs. output produced from Unicode-repaired headlines, as an ablation of input quality.

To test whether our cross-generator finding could be an artifact of data contamination (generators reproducing memorised human source text), we run three further analyses, all reported in Section 4.4. We measure 13-gram token overlap and longest common token substring between each generated article and its paired human source, with Unicode normalisation applied to both sides; we re-evaluate detection on a decontaminated subset that drops every generated article sharing a 13-gram with its source; and we measure output-distribution peakedness by generating multiple samples per headline and computing their pairwise edit-distance diversity, following Dong et al. (2024).

3.5 Native-speaker validation

Two Hausa-speaking colleagues reviewed two artefacts: the eight false-positive Human articles from Section 4.2 and a 30-article random sample of NLLB outputs from Section 4.3. They classified

Seed	Acc	F1	Prec	Rec
42	0.985	0.985	0.970	1.000
1	0.996	0.996	0.992	1.000
7	0.994	0.994	0.989	1.000
123	0.994	0.994	0.989	1.000
Mean $\pm$ std	0.992 $\pm$ 0.005	0.992 $\pm$ 0.005	0.985 $\pm$ 0.010	1.000 $\pm$ 0.000
Sani et al. (2025)	0.9923	0.9921	–	–

Table 1: Replication of Sani et al. (2025) for the AfroXLMR-large baseline across four seeds.

the first set by register without access to the detector’s predictions. For the second set, they rated fluency on a 1–5 scale and judged whether content was preserved. The two reviewers were given identical spreadsheets and worked independently.

4 Results

4.1 Replication

Table 1 reports per-seed and aggregate test metrics on the headline-grouped split.

Our four-seed mean F1 of 0.9924 matches the paper’s reported 0.9921 within run-to-run variance.

Two things about Table 1 are easy to miss. Run-to-run variance is small but real: seeds 42 and 1 differ by 1.1 F1 points, and seed 42 happens to be the worst draw. And recall on the AI class is 1.000 at every seed. The detector never misses a Gemini 2.0 Flash article. Whatever varies between runs, it is entirely on the precision side, in how many Human articles get falsely flagged (between 3 and 8 per run, out of 260).

4.2 What does the detector use? Error analysis on false positives

We examine the eight false-positive Human articles from the seed-42 run. Two Hausa-speaking colleagues independently reviewed each article without access to the detector’s predictions, and both judged all eight to be natural Hausa journalism. So the errors are not scraped boilerplate or degraded text slipping through; the detector is misclassifying genuine human-written news.

The clearest correlate of misclassification is the absence of numerical content. Six of the eight false positives contain zero digit characters, and the median digit-character fraction is 0.000 in the false positives vs. 0.006 in correctly classified Humans (Table 2). Hard-news writing routinely carries dates, ages, casualty counts, and percentages;

the digit-sparse Human articles evidently look more machine-like to the detector. This does not explain everything, though. Two of the flagged articles are statistic-dense and were misclassified anyway.

We had also expected the false positives to concentrate in non-news registers (opinion, feature, cultural commentary), on the intuition that the training data is overwhelmingly hard news (53.5% politics). The native-speaker review did not support this cleanly. While a bylined opinion essay, a celebrity-feud feature, and a current-affairs programme summary were judged opinion or feature, both reviewers classified other false positives (a desertification-policy report, a diabetes and sugar-tax report, and a security-chiefs meeting) as straight news, and the two reviewers did not fully agree on the borderline cases. We therefore do not claim a register effect: the misclassified set spans both non-news and unambiguous hard-news writing, so register does not characterise the errors. Per-article register labels from both reviewers are in Appendix E.

We caution that this analysis is on the seed-42 checkpoint, which had the largest false-positive set (8) among our four seeds. Seeds 1, 7, and 123 each produced only 3 false positives. Confirming the digit-sparsity pattern on those checkpoints is left for future work.

We also asked whether orthographic density explains the false positives. The intuitive story would be that the detector flags Human articles with few hooked consonants as machine-like. Our data say otherwise. Median hooked-consonant density per 1,000 characters is 14.87 in the false positives vs. 8.09 in correctly classified Humans, so the misclassified articles actually have *more* Hausa-specific orthography than the ones the detector gets right. A one-sided Mann-Whitney U test ($FP < TN$) gives $p = 0.96$. At least where its errors are concerned, the detector is not applying a “low Hausa orthography means machine” rule.

Orthographic density does play a role in the AI class, however. On the test-split AI articles, median hooked-consonant density is 0.0 per 1,000 characters: more than half of Gemini-generated articles contain zero hooked consonants. The signal is about presence, not density, and on the training distribution it is strong enough to explain why test recall never drops below 1.000. We return to this in Section 4.3, where the cross-generator data narrow down what the detector can actually be using.

Feature	FP (n=8)	TN (n=252)
Median digit fraction	0.000	0.006
Zero-digit articles	6 (75%)	21 (8%)
Median hooked/1k	14.87	8.09
Natural Hausa (both reviewers)	8 (100%)	not anno.

Table 2: Feature comparison of false-positive vs. correctly classified Human articles (seed 42). The false positives have *higher* hooked-consonant density than correctly classified Humans, refuting an orthographic-sparsity explanation.

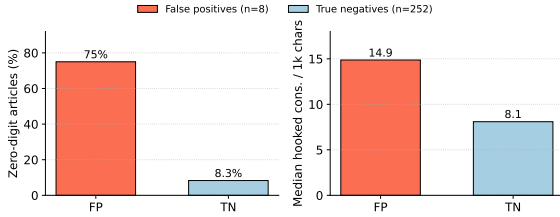


Figure 1: False-positive vs. true-negative Human articles (seed 42): share of zero-digit articles and median hooked-consonant density. The misclassified articles are digit-sparse but orthographically *richer* than correctly classified Humans.

4.3 Cross-generator evaluation

Table 3 and Figure 2 report detector behaviour on six non-training generators, plus the in-distribution training generator and the Human reference. Generators are sorted by detection rate.

Three patterns stand out.

Detection rates span the full range. Across the six non-training generators we see everything from 0% to 100%. The detector generalises to some out-of-distribution generators and fails on others. The English-language cross-generator literature has mostly reported a binary picture, “single-generator detectors do not transfer” (Dugan et al., 2024; Wang et al., 2024); in Hausa, at least for this detector, the reality is messier.

Google-family models are detected at near-perfect rates. Both Google-family models in our test set, Gemma 4 E4B and Gemma 4 E2B, are detected at 100.0% and 98.9% respectively, with mean $P(\text{machine})$ of 1.000 and 0.989. No other tested non-training generator exceeds 21%. The simplest hypothesis is that the detector has learned features shared across Google’s LLM family, plausibly arising from common training data, tokenizer family, instruction-tuning style, or RLHF templates, rather than features general to machine-written Hausa. Models from other families (Co-

here/Aya, Meta/Llama, Alibaba/Qwen, Meta translation/NLLB) lack these features and are missed at rates from 79% to 100%. Across all three Google models in our experiment (the training-distribution Gemini 2.0 Flash plus two non-training Gemma variants of different sizes), detection is consistently above 98% with mean $P(\text{machine})$ above 0.98. Across all four non-Google generators, detection is at or below 20.8% with mean $P(\text{machine})$ at or below 0.21. The gap is large enough that we doubt the pattern is incidental.

Orthographic density does not explain the detection signal. Hausa-specific hooked-consonant density varies substantially across our generators (full per-generator statistics in Appendix D). Aya produces almost no hooked consonants (0.5 per 1,000 characters under mojibake input, 1.5 under repaired input). Gemma 4 produces more than Humans (10.3 to 12.1 per 1,000 characters across the two Gemma variants vs. 7.9 for Humans). The three Google models in our experiment span hooked-consonant densities from 5.2 (Gemini 2.0 Flash) to 12.1 (Gemma 4 E2B), a $2.3\times$ range, yet all three are detected at rates above 98% with maximum confidence. That spread leaves no room for a “machine-generated Hausa lacks hooked consonants” explanation. Hooked-consonant density was a correlate within the training distribution; it is not what the detector learned.

The Aya input-quality ablation makes the same point from a different angle. With mojibake-corrupted headlines, Aya output averages 0.5 hooked consonants per 1,000 characters, with 75% of articles containing zero. With Unicode-repaired headlines, output averages 1.5 hooked consonants per 1,000 characters with 58% zero-hook articles. Cleaning the input triples Aya’s orthographic richness, and the detector does not react at all: 0% called AI in both conditions, mean $P(\text{machine})$ below 0.001. Whatever the detector reads in a generator’s output, it is not the surface orthography.

Aya’s output also deserves a qualitative note. Its Hausa is often partially code-switched into Somali, which a Hausa-speaking reviewer flagged as unfluent. The Aya generation with the highest $P(\text{machine})$ in our run (article #49, $P = 5.3 \times 10^{-5}$) contains explicit Somali function words. The detector still rates it 99.99% Human. The detector is willing to classify text that is not even fluent Hausa as Human, provided it lacks the Gemini-family signature.

Conversely, the Gemma 4 output we self-

Generator	Family	% AI	95% CI	Mean P(machine)	Hooked/1k
Aya-Expanse 8B (mojibake input)	Cohere	0.0	[0.0, 1.5]	< 0.001	0.50
Aya-Expanse 8B (repaired input)	Cohere	0.0	[0.0, 1.5]	< 0.001	1.53
Llama 3.1 8B	Meta	1.2	[0.4, 3.3]	0.013	5.18
Qwen 3 8B	Alibaba	10.0	[6.9, 14.2]	0.108	4.79
NLLB-200 round-trip	Meta (translation)	20.8	[16.3, 26.1]	0.210	3.82
Gemma 4 E2B	Google	98.9	[96.7, 99.6]	0.989	12.09
Gemma 4 E4B	Google	100.0	[98.5, 100.0]	1.000	10.33
Gemini 2.0 Flash	Google (training)	100.0	[98.5, 100.0]	1.000	5.23
Human (test split)	—	3.1	[1.6, 6.0]	0.031	7.94

Table 3: Cross-generator detector behaviour (n=260 each, seed-42 checkpoint). Intervals are Wilson 95% confidence intervals for the percentage classified as AI.

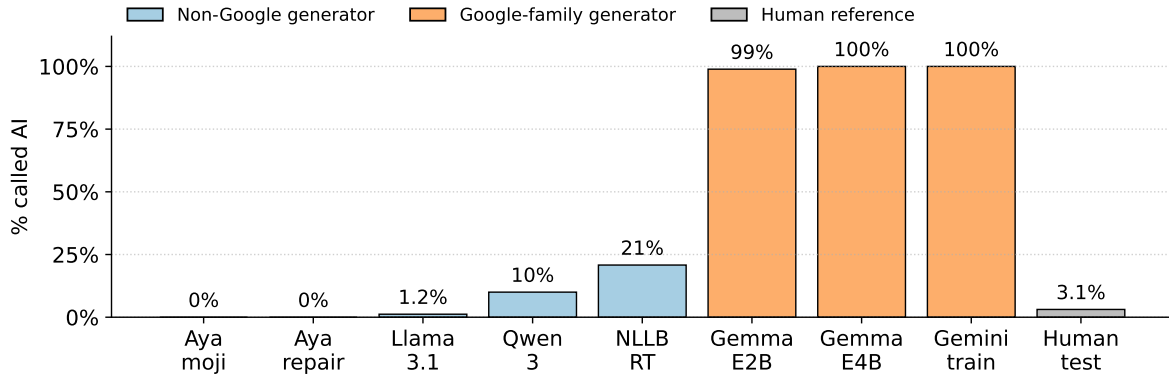


Figure 2: Detection rate by generator (n=260 each, seed-42 checkpoint). All three Google models are detected at or above 99%; no non-Google generator exceeds 21%.

inspected (10 random samples from the 260-article set) is fluent, on-topic Hausa news prose with proper hooked-consonant orthography throughout (mean 299 words per article, hooked-consonant density of 10.3 per 1,000 characters, higher than the Human reference at 7.9). We did notice recurring article-opening templates, “*An tattara labarin*” and “*An bayar da labari*” among them, the kind of stylistic uniformity a detector could plausibly latch onto. We saw no language drift, repetition loops, or refusals. So the 100% detection of Gemma 4 cannot be blamed on degraded output; the detector is confidently flagging genuine, fluent Hausa that happens to carry Google-family stylistic features. Appendix C gives qualitative examples from each generator.

4.4 Is the family signature a contamination artifact?

The human articles in the Sani et al. (2025) corpus were scraped from public Hausa news sites, so they may appear in the pretraining data of the generators we test. If a generator had memorised a human source article, its “machine-generated” out-

put could carry verbatim spans of that human text, and the detector might then be responding to regurgitated human writing rather than to a genuine generator signature. We run three analyses to test this.

First, we measure surface overlap between each generated article and its paired human source, using 13-gram token overlap (the convention of Brown et al., 2020) and longest common token substring, with both sides Unicode-normalised so that mojibake does not artificially depress the counts. As a control, we also measure each generated article’s maximum overlap against a random sample of *other* human articles, to separate topical overlap (shared headline, shared topic words) from source-specific copying. Table 4 reports the result. Source overlap is low in absolute terms for every generator, and, more importantly for the family-signature claim, it is not higher for Google models. Gemma 4 E4B has the *lowest* source overlap of any generator (mean 13-gram Jaccard 0.0002, 4.6% of articles sharing any 13-gram) yet is detected at 100%. Qwen 3 has by far the *highest* overlap

Generator	src Jac.	ctrl Jac.	% w/ 13-gram
Aya (mojibake)	0.0027	0.0000	23.1
Aya (repaired)	0.0025	0.0000	23.5
Llama 3.1 8B	0.0017	0.0000	15.0
Qwen 3 8B	0.0278	0.0001	76.2
NLLB round-trip	0.0010	0.0002	23.1
Gemma 4 E2B	0.0087	0.0000	51.2
Gemma 4 E4B	0.0002	0.0000	4.6

Table 4: Surface overlap between generated articles and human sources. **src Jac.** is mean 13-gram Jaccard against the paired human source; **ctrl Jac.** is mean maximum 13-gram Jaccard against random other human articles; **% w/ 13-gram** is the fraction of articles sharing at least one 13-gram with their source. Google’s Gemma 4 E4B has the lowest source overlap yet the highest detection; Qwen has the highest overlap yet near-floor detection.

(0.028, 76.2% sharing a 13-gram) yet is detected at only 10%. If anything, detection runs in the opposite direction from overlap: the generator that copies least is caught most.

Second, we re-run the full cross-generator evaluation on a decontaminated subset, dropping every generated article that shares at least one 13-gram with its paired human source and recomputing the detection rate on what remains (Table 5). This removes 51% of Gemma 4 E2B’s articles and 76% of Qwen’s, a severe filter. The headline finding is unchanged. Both Google models remain at or above 99% on the clean subset (Gemma 4 E2B in fact rises slightly, from 98.85% to 99.21%, as its most source-similar articles are removed), while no non-Google generator approaches Google territory. Qwen’s detection *falls* from 10.0% to 4.8% under decontamination, so what little detection Qwen attracted came partly from its most headline-echoing articles. Where overlap mattered at all, it was proping up the score of a weakly detected model; it cannot be the source of the Google result.

The one non-trivial upward shift is NLLB (20.8% to 24.0%). This is expected and mild: NLLB output is a back-translation of the human source, so its highest-overlap articles are the ones that stayed closest to the human original, and removing them leaves a more drifted, more machine-like remainder. That fits the paraphrase-probe reading of the NLLB condition (Section 4.5).

Third, we test whether the Google models might be detected because their output is memorisation-like in a way the surface-overlap analyses would miss, namely low-diversity or templated generation.

Generator	full %	clean %	Δ
Aya (repaired)	0.0	0.0	0.00
Aya (mojibake)	0.0	0.0	0.00
Llama 3.1 8B	1.2	1.4	+0.20
Qwen 3 8B	10.0	4.8	−5.16
NLLB round-trip	20.8	24.0	+3.23
Gemma 4 E2B	98.9	99.2	+0.37
Gemma 4 E4B	100.0	100.0	0.00

Table 5: Detection rate before (**full**) and after (**clean**) removing every generated article that shares a 13-gram with its human source. The family signature survives decontamination; the two Google models stay at $\geq 99\%$ while non-Google models stay low.

Generator	peaked frac.	mean pair ED
Qwen 3 8B	0.0098	0.780
Aya-Expanse 8B	0.0000	0.844
Gemma 4 E2B	0.0000	0.824
Gemma 4 E4B	0.0000	0.892
Llama 3.1 8B	0.0000	0.905

Table 6: Output-distribution peakedness (15 samples $\times$ 40 headlines per generator). **peaked frac.** is the fraction of sample pairs with normalised edit distance ≤ 0.05 ; **mean pair ED** is mean pairwise normalised edit distance (higher = more diverse). Both Google models produce zero near-duplicate pairs and highly diverse output, ruling out a memorisation explanation for their detection.

Following the output-distribution intuition of Dong et al. (2024), we generate 15 independent samples per headline for 40 headlines from each local generator and measure how self-similar those samples are (sampling protocol in Appendix F). We report two quantities per generator: the fraction of sample pairs whose normalised token edit distance is at most 0.05 (a peakedness measure; higher means more near-duplicate generation), and the mean pairwise normalised edit distance (higher means more diverse). Table 6 shows the result. The two Google models are not peaked at all: both Gemma 4 variants produce zero near-duplicate sample pairs, and their samples are as diverse as, or more diverse than, the non-Google generators (mean pairwise edit distance 0.82 and 0.89, against 0.78 for Qwen and 0.84 for Aya). If the detector were keying on memorisation-like, low-diversity Google output, we would expect the Google models to be the most peaked; they are among the least. Whatever the detector reads in Google output, it persists across samples that differ almost completely from one another.

None of the three checks leaves room for a contamination account of the Google-family gradient. The generators do not reproduce their human sources (Tables 4 and 5), and the Google output the detector flags is diverse, with no sign of memorisation (Table 6).

We cannot rule out a distinct contamination pathway on the *detector* side: AfroXLMR-large was itself pretrained on Hausa web text and may have seen the test-set human articles. However, this cannot explain the central finding, because detector-side memorisation of *human* articles offers no mechanism for the detector to fire on Gemma output while ignoring Aya, Llama, and Qwen output, all of which are machine-generated articles the detector never saw during its own fine-tuning. We discuss this limitation further in Section 5.1.

4.5 Paraphrase robustness via NLLB round-trip

The NLLB round-trip also serves as a paraphrase-attack probe. Translation through English preserves meaning while scrambling surface stylistic features. On NLLB-paraphrased Hausa articles, the detector classifies 20.8% as AI, compared to 100% on the original Gemini output. The detector retains partial signal after this transformation but loses approximately four-fifths of it. In a deployment context, an adversary can evade detection on roughly 80% of articles by running LLM output through a translation round-trip.

To check that the NLLB output is fluent enough to count as a meaningful paraphrase rather than degraded noise, two Hausa-speaking annotators rated a 30-article sample of round-trip outputs for fluency (1–5) and content preservation. The two reviewers diverged in strictness. A lenient reviewer rated most outputs 4–5 and judged nearly all as natural Hausa with preserved content; a stricter reviewer rated roughly a third of the sample at 3 or below and flagged hallucinated named entities and topic drift on the harder articles (for example, a desertification agency rendered as a “cooking agency”, and a post-partum health column drifting into unrelated material). Both reviewers converged on the same qualitative picture: the round-trip output reads as surface-fluent Hausa, but harder source articles accumulate content drift and occasional invented entities. That is close to the profile a paraphrase probe needs, though plainly an imperfect one. Per-article ratings are in Appendix E.

4.6 Real-world false-positive rate

On the test-split Human articles, the detector classifies 3.1% as AI. For deployment this is the number that matters: in a moderation pipeline, roughly 3 of every 100 real Hausa articles would be falsely flagged, and the error analysis in Section 4.2 suggests digit-sparse articles are the most exposed.

5 Discussion

Three implications follow from our results.

First, **generator-family specificity is the central deployment risk**. Our cross-generator gradient shows the detector reliably catches text from one model family (Google) and largely misses text from other families. Practitioners deploying detectors trained on a single generator’s output should expect family-specific behaviour and either retrain when the underlying generator changes or train on a broader generator mix from the outset. This empirically confirms the concern Sani et al. (2025) raised in their own limitations section.

Second, **the adversary’s evasion path is trivial**. Anyone wanting to publish AI-written Hausa text without detection can use Aya-Expanse, Llama, Qwen, or an NLLB-translated rewrite of an English source. All evade the detector at rates above 79%.

Third, **the false positives matter operationally**. A 3.1% Human FP rate sounds tolerable in isolation, but the flagged articles are real journalism, and digit-sparse pieces (no dates, ages, or statistics) are hit hardest. A journalist wrongly flagged as a machine has little practical recourse, so false-positive behaviour deserves monitoring in its own right, separately from aggregate accuracy.

5.1 Limitations

Our cross-generator evaluation tests six generators. Three of them are Google models (Gemini 2.0 Flash, the training generator, plus Gemma 4 E4B and E2B). While the Google-family detection pattern across these three is sharp and consistent (all detected at $\geq 98\%$, with maximum confidence on two of three), we acknowledge the family-signature claim would be further strengthened by an additional non-Google MoE model or a separate Google model from a different architectural lineage. The current evidence is consistent with the family-signature hypothesis but does not exclude alternative explanations such as multilingual-training-data overlap or shared instruction-tuning corpora.

We do not perform probing experiments to identify the specific features the detector keys on. Approaches such as LIME, SHAP, or attention-based attribution would help pinpoint whether the Gemini-family signal arises from specific tokens, n-grams, sentence patterns, or higher-level stylistic features. We leave this for future work.

We evaluate on a single language. The orthographic-density finding raises questions about whether similar patterns arise for Yoruba (tone marks), Igbo (subdot diacritics), and other African languages with non-ASCII orthography. Multi-language replication is an obvious next step.

Cross-generator evaluation is based on a single seed-42 checkpoint. Multi-seed evaluation across the cross-generator data would confirm whether 0% on Aya and 100% on Gemma 4 hold across initialisations. Given that AI-class recall was 1.000 at every seed on the training distribution, we expect the cross-generator pattern to replicate, but we have not verified this.

The training-distribution model (Gemini 2.0 Flash) was deprecated by Google in March 2026, which means within-family transfer between Gemini 2.0 and Gemini 2.5 Flash cannot be cleanly tested from a research-budget free tier. Future work with paid API access could close this gap.

On contamination, our analyses in Section 4.4 address the generator-side pathway (generators regurgitating memorised human source text) and show the family signature survives it. We cannot directly test the detector-side pathway: AfroXLMR-large was pretrained on Hausa web text and may have encountered the test-set human articles before our fine-tuning. We argue this does not threaten the central finding, since memorisation of human articles gives the detector no mechanism to separate Gemma output from Aya, Llama, or Qwen output, which are all machine articles unseen at fine-tuning time. A full test would require retraining the encoder on a corpus with the evaluation articles held out, which is beyond our compute budget.

5.2 Future work

- Probing the learned features through attribution analysis on the trained checkpoint to identify the actual Google-family signature.
- Multi-language replication on Yoruba and Igbo to test whether orthographic and family-signature findings generalise.
- A direct LLM-paraphrase attack with quality

controls, complementing the NLLB round-trip evidence.

- Additional Google-family generators (Gemma 4 26B, paid-tier Gemini 2.5 runs) to strengthen the family-signature claim.
- Re-running the error analysis on seeds 1/7/123 to test whether the digit-sparsity pattern replicates across initialisations.
- Extending the dataset to non-news domains and additional generators, consistent with the future work [Sani et al. \(2025\)](#) outlined in their original paper.

6 Conclusion

We confirm [Sani et al. \(2025\)](#)’s AfroXLMR-large detector replicates cleanly across four random seeds, achieving F1 of 0.992 ± 0.005 . Recall on the AI class is perfect at every seed. All errors are false positives on genuine human-written news, with the absence of numerical content the clearest correlate of misclassification. Across six non-training generators, detection rates span from 0% to 100%; the two non-training Google-family models in our test set are both detected at rates above 98% with maximum confidence, matching the training-distribution rate, while no non-Google generator exceeds 21%. On our evidence the detector is better described as a Gemini-family-output classifier than as a general Hausa MGT detector, and practitioners should expect single-generator detectors to behave in this family-specific way. Our work addresses the single-generator limitation [Sani et al. \(2025\)](#) acknowledged in their original paper, and we hope our cross-generator results inform the design of future Hausa MGT detection benchmarks.

Acknowledgments

We thank our two Hausa-speaking colleagues for native-speaker validation. We thank [Sani et al. \(2025\)](#) for releasing the dataset and detector that made this study possible. We thank the maintainers of AfroXLMR-large, vLLM, and the open generators we used. We also thank the HausaNLP community for the open scholarly environment in which this work was possible.

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. 2022. 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 (NAACL)*.
- David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and Annie En-Shiun Lee. 2024. [SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245. Association for Computational Linguistics.
- Jesujoba O. Alabi, David Ifeoluwa Adelani, Marius Mosbach, and Dietrich Klakow. 2022. Adapting pre-trained language models to African languages via multilingual adaptive fine-tuning. In *Proceedings of the 29th International Conference on Computational Linguistics (COLING)*, pages 4336–4349.
- Tom B. 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 M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- John Dang, Shivalika Singh, Daniel D’souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, Sandra Kublik, Meor Amer, Viraat Aryabumi, Jon Ander Campos, Yi-Chern Tan, Tom Koemi, Florian Strub, Nathan Grinsztajn, Yannis Flet-Berliac, and 26 others. 2024. Aya Expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.
- Google DeepMind. 2026. [Gemma 4 model card](#).
- Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv:2402.15938*.
- Liam Dugan, Alyssa Hwang, Filip Trhlík, Andrew Zhu, Josh Magnus Ludan, Hainiu Xu, Daphne Ippolito, and Chris Callison-Burch. 2024. RAID: A shared benchmark for robust evaluation of machine-generated text detectors. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: Statistical detection and visualization of generated text. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 111–116.
- 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 541 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Isa Inuwa-Dutse. 2021. The first large scale collection of diverse Hausa language datasets. *arXiv preprint arXiv:2102.06991*.
- Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. 2023. DetectGPT: Zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.
- Babangida Sani, Aakansha Soy, Sukairaj Hafiz Imam, Ahmad Mustapha, Lukman Jibril Aliyu, Idris Abdulmumin, Ibrahim Said Ahmad, and Shamsuddeen Hassan Muhammad. 2025. [Who wrote this? Identifying machine vs human-generated text in Hausa](#). In *Proceedings of the Sixth Workshop on African Natural Language Processing (AfricaNLP 2025)*, pages 82–88. Association for Computational Linguistics.
- Robyn Speer. 2019. [ftfy](#). Version 5.5.
- NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Hefernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. Technical Report arXiv:2207.04672, Meta AI.
- Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D’souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. Aya model: An instruction finetuned open-access multilingual language model. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Yuxia Wang, Jonibek Mansurov, Petar Ivanov, Jinyan Su, Artem Shelmanov, Akim Tsvigun, Chenxi Whitehouse, Osama Mohammed Afzal, Tarek Mahmoud, Toru Sasaki, Thomas Arnold, Alham Fikri Aji, Nizar Habash, Iryna Gurevych, and Preslav Nakov. 2024. [M4: Multi-generator, multi-domain, and multilingual black-box machine-generated text detection](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational*

Linguistics (Volume 1: Long Papers), pages 1369–1407, St. Julian’s, Malta. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2505.09388*.

A Generator prompts

All open generators received the same chat-template prompt structure.

System message:

You are a professional Hausa journalist writing for a major Nigerian newspaper. You write in fluent, natural Hausa using standard orthography including the hooked consonants *ɓ*, *ɗ*, *ɓ̌*, *y̌* where appropriate. You write only in Hausa, never mixing other languages.

User message template:

Write a complete Hausa news article based on the following headline. The article should be between 300 and 500 words. Write only the article, no preamble, no translation, no explanation.

Headline: {headline}

The {headline} placeholder was filled with the headline of a Human article from the test split, after passing it through ftfy to repair Unicode mojibake. For thinking-capable models (Qwen 3, Gemma 4), the thinking budget was set to zero via the chat-template enable_thinking=False flag.

For the NLLB round-trip generator, no chat template was used. The Human article was translated Hausa → English with source tag hau_Latn and target tag eng_Latn, then back English → Hausa with the tags reversed. Long articles were chunked at 800-character boundaries.

Sampling parameters for all open generators: temperature 0.8, top-p 0.95, max output tokens 600 (1,500 for thinking-capable models, with thinking disabled).

B Mojibake in the released dataset

The released CSV contains Unicode corruption of Hausa hooked consonants in approximately 9% of articles. Examples (raw → repaired): karÃfÃ,Ã,Ã’uwa → karɓuwa; ÃfÃ,Ã,Ã’oÃfÃ,Ã,Ã’ari → ɓɔɓari; ÃfÃ,Ã,Ã’arshen → ɓarshen. The pattern is consistent with UTF-8 bytes being misinterpreted

as Latin-1 and re-encoded, a common scraping pipeline error.

We did not modify the dataset for our training runs; we trained on the data as released. For cross-generator experiments, we passed headlines through ftfy before sending them to the generators, so the generators received valid Hausa orthography.

C Examples of cross-generator output

All generators below were given the same Hausa input headline about a wedding-day attack in Myanmar (headline_id 1383 in our test split):

Students’ Revolutionary Force — Ranar daurin auren Hnin Si a farkon watan Yuni, wanda ya kamata ya zama ranar farin ciki a ɓauyen Mataw da ke yankin Sagaing na Myanmar, ya rikiɗe zuwa bala’i.

C.1 Aya-Expanse 8B

Aya routinely produced output containing material in non-Hausa languages, most often Somali. The Hausa-speaking reviewer characterised it as code-switched and not fluent. From an Adam Zango profile headline:

*Adam Zango Fitaccen, kuna da kalla da Kannywood, **ayaa ka soo baxay** da amma n’ita aikinsa alhalarin kallu da abokon hira. Wanaa 20 sanno da soo baxay da aikinsa, ta’an da aikinsa akochan hira da kallu, **waa inuu** abokon hira da Adam Zango Fitaccen aikinsa ka soo baxay.*

The bolded function words (*ayaa ka soo baxay*, *waa inuu*) are Somali, not Hausa. The detector classified this article as Human with P(machine) = 1.4×10^{-5} , near the median for the full Aya generation set.

C.2 Llama 3.1 8B

Llama produced grammatically reasonable Hausa with attempted journalistic detail. On the wedding-bombing headline:

Ranar daurin auren Hnin Si, wanda aka fi sani da ‘Hnin Si’s Wedding Day’, ya bayyana ne a farkon watan Yuni, wanda ya kai laushi a ɓauyen Mataw, cikin yankin Sagaing na Myanmar. Hnin Si, ‘yar shekara 25, ta auri ɗan’uwan auren ta, Linn Htwe, a ranar 11 ga watan Yuni...

Names and ages are partly invented (the real headline does not name the groom), but the prose is on-topic and reads as Hausa news. The detector classified this article as Human (P(machine) = 0.013, near the Llama mean).

C.3 Qwen 3 8B

Qwen 3 frequently entered repetition loops, repeating a phrase or sentence with minor variations until the token budget was exhausted. On the wedding-bombing headline:

*Ranar daurin auren Hnin Si, wanda ya kamata ya zama ranar farin ciki a wauyen Mataw, da ke yankin Sagaing na Myanmar, ya rikiɗe zuwa bala'i. **Daga kuma, ranar daurin auren Hnin Si, wanda ya kamata ya zama ranar farin ciki a wauyen Mataw, da ke yankin Sagaing na Myanmar, ya rikiɗe zuwa bala'i. Kuma, ranar daurin auren Hnin Si...***

The repeated clauses are bolded. Despite the repetition, the detector classified 10% of Qwen 3 outputs as AI, the second-highest non-Google rate after NLLB.

C.4 NLLB-200 round-trip (paraphrase)

The NLLB round-trip preserves the topic but produces characteristic translation artefacts. On the wedding-bombing headline:

Hsin Si ta ce yayin da kusan abokai da dangi 100 ke taruwa don bikin bikin aurenta, jirgin sama ya jefa bama-bamai ba zato ba tsammani, ya kashe mutane 33, ciki har da mata da yara. “Bayan da muka yi aure, mun fara daukar hotuna tare da iyalai da abokai, kawai don jin bam sun fara tsakaninmu yayin da Eaya ta fashe kusa da gidan amarya”. Yankin Sagaing, tare da Mandalay da Magway...

This is a back-translation of the source Human article. The named entity “Hnin Si” became “Hsin Si” through the pivot. The detector classified this article as AI ($P(\text{machine}) = 0.45$). 21% of NLLB outputs were classified as AI overall.

C.5 Gemma 4 E4B

Gemma 4 produced fluent, on-topic Hausa with consistent stylistic patterns. On the wedding-bombing headline:

An tattara labarin daga yankin Sagaing a Myanmar inda an samu wani abin mamaki da kuma walubale da ya faru a kan wani taron tarihi mai muhimmanci. wungiya ta kan malamai da dalibai, da aka fi sani da sunan Hukumar Daidaito ta dalibai (Students' Revolutionary Force), ta shirya wani taro don fara bikin aure na Hnin Si a farkon watan Yuni...

The bolded opening (“An tattara labarin” — “the news has been gathered”) recurs across many Gemma 4 generations and is one of the recurring stylistic templates we noted in Section 4.3. The detector classified this article as AI with $P(\text{machine}) \approx 1.000$. 100% of Gemma 4 E4B outputs were classified as AI.

C.6 Gemma 4 E2B

The smaller Gemma 4 variant produced similar Hausa news prose with some content drift. On the wedding-bombing headline:

Wata wungiya ta dalibai ta yi tashin hankali a wauyen Mataw na yankin Sagaing na Myanmar a farkon watan Yuni, wanda ya yi tsanani da gaba daya da ya haifar da wani tsanani mai tsanani...

Note the awkward phrase repetition (“tsanani da gaba daya da ya haifar da wani tsanani mai tsanani”). The article preserves the topic but loses the wedding framing entirely. The detector classified this article as AI with $P(\text{machine}) \approx 1.000$. 98.9% of Gemma 4 E2B outputs were classified as AI.

D Hooked-consonant density distributions

Generator	Mean	Med.	% zero
Human (test split)	7.94	8.58	27.7
Gemini 2.0 Flash	5.23	0.00	58.9
Aya 8B (mojibake)	0.50	0.00	75.0
Aya 8B (repaired)	1.53	0.00	58.5
Llama 3.1 8B	5.18	2.60	26.2
Qwen 3 8B	4.79	0.25	48.9
Gemma 4 E2B	12.09	10.83	0.8
Gemma 4 E4B	10.33	9.82	0.0
NLLB-200 round-trip	3.82	3.38	18.5

Table 7: Hooked-consonant statistics per generator (per 1,000 characters).

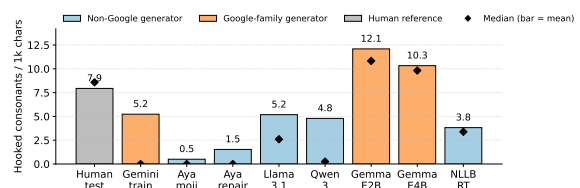


Figure 3: Mean and median hooked-consonant density by generator. The Google-family generators vary widely in orthographic density while remaining highly detectable, whereas some non-Google generators have similar densities to Gemini but much lower detection rates.

Both Gemma 4 variants have non-zero hooked-consonant density in virtually all articles, exceeding the Human reference. Figure 3 shows that the three Google models span 5.2 to 12.1 hooks per 1,000 characters but are all detected at $\geq 98\%$, refuting the hypothesis that orthographic density is the detection signal.

E Native-speaker validation protocol

Two Hausa-speaking colleagues each received an identical Excel spreadsheet with two sheets and worked independently. The first sheet contained the eight false-positive Human articles with columns for register classification (news, opinion, feature, cultural, other), a yes/no judgement on whether the article reads as a straight news report rather than opinion or feature, a yes/no judgement on whether the language is natural Hausa journalism, and free-text comments. Reviewers were not told which articles the detector had misclassified or what hypothesis was under test.

Table 8 summarises the register classifications. Both reviewers judged all eight articles to be natural Hausa journalism (8/8 “natural Hausa = yes”), confirming the false positives are genuine human writing rather than scraped noise. On register, the reviewers diverged: one assigned single-register labels (four opinion/feature/cultural, three news, one mixed), while the other often gave compound labels such as “news, feature, as well as cultural”. Because the two agreed that several false positives (desertification policy, diabetes/sugar-tax report, security-chiefs meeting) are straight news, we do not treat register as characterising the errors; the digit-sparsity correlate in Section 4.2 is the finding the data supports.

FP	Topic	R1	R2
1	Tailoring/fashion	news/feat/cult	cultural
2	Desertification	news	news
3	Diabetes report	news	news
4	SA election	news	opinion
5	Musician feud	news/feat/opin	feature
6	Tax-bill TV prog.	news	opinion
7	Security meeting	news	news
8	Africa-image essay	news/opinion	opinion

Table 8: Native-speaker register classifications of the eight false positives. R1 and R2 are the two independent reviewers. Both rated all eight as natural Hausa journalism.

The second sheet contained 30 randomly sampled NLLB round-trip outputs paired with their source Human articles, with columns for fluency (1–5), a yes/no judgement on whether a Hausa reader would accept the text as written by a Hausa journalist, a yes/no judgement on whether the translation preserved the source article’s key facts, and free-text comments. The two reviewers diverged in strictness. The lenient reviewer gave a mean fluency near 4.6, with natural-Hausa = yes and

content-preserved = yes on almost all 30 items. The strict reviewer gave a mean fluency near 3.5 and flagged content problems on roughly a third of the sample, including hallucinated named entities and topic drift (for example, NLLB_12, where the “Great Green Wall” agency became a “cooking agency”, and NLLB_15, where a post-partum health column drifted into unrelated material). A website-boilerplate item (NLLB_13: “Wikki Times Media... All Rights Reserved”) made the difference plain: the strict reviewer scored it 1 (“this has no meaning”), the lenient reviewer 5. Both reviewers’ free-text comments recurred on the same theme, summarised by one as “a little hallucination as well as grammatical errors”. Both ended up in the same place: NLLB round-trip output is surface-fluent Hausa with measurable content drift on harder source articles, which is what makes it usable as a paraphrase probe.

F Software and reproducibility

All experiments were run on a single Ubuntu 24.04 workstation with an NVIDIA RTX 3090 (24 GB VRAM) and 64 GB RAM. Python 3.11, PyTorch 2.11.0+cu130, transformers 5.8.1, vLLM 0.21.0, ftfy 6.1.

Random seeds for the four-seed training runs: 42, 1, 7, 123. The vLLM sampler was set to seed 42 for all generator runs to make the generations reproducible. Hugging Face model commits at time of access can be reconstructed from the dates of our runs (May 2026).

For the peakedness analysis (Table 6), we generated 15 samples per headline for 40 randomly selected test headlines from each local generator, at the same temperature 0.8 and top-p 0.95 used for the main generations, with the vLLM sampler seeded at 42. Peakedness is the fraction of within-headline sample pairs whose token-level normalised edit distance is at most 0.05; diversity is the mean pairwise normalised edit distance.

All scripts used to produce the results in this paper, along with the trained-checkpoint hashes and the cross-generator output CSVs, will be released on publication.

The Energy-Accuracy Frontier: Benchmarking SNNs on Low-Resource Indian Linguistic Tasks

Priyam Saha¹, Debiprasanna Sahoo¹, Abhik Jana¹

¹IIT Bhubaneswar, India

{21cs01076, dpsahoo, abhikjana}@iitbbs.ac.in

Abstract

The increasing complexity of state-of-the-art neural networks creates significant computational energy overhead in Natural Language Processing (NLP) applications. A few studies have explored opportunities to reduce overhead by using a Spiking Neural Network (SNN) instead of a mainstream artificial neural network (ANN) in computer vision and sensory domains. A comprehensive analysis of SNN’s usability for NLP tasks in low-resource languages has yet to be conducted. Hence, in this paper, we present the first-ever comparative energy-overhead analysis of ANNs vs. SNNs for text classification tasks in low-resource Indian languages such as Hindi, Bengali, and Telugu. Our experiments show that although we lose classification accuracy by an acceptable margin when replacing an ANN with an SNN, we save significant energy, which highlights the potential for efficient utilization of edge computing devices to run NLP tasks for low-resource languages. This work shows a direction for advancing energy-efficient NLP applications, especially for low-resource languages. We make code and dataset publicly available ¹.

1 Introduction

With the emergence of Transformers (Vaswani et al., 2017), the domain of Natural Language Processing has been revolutionized. This has led to a substantial requirement for computational resources and hence, energy for training and deployment. The rise in model complexity poses challenges for the practical use of NLP tasks in resource-constrained environments, such as mobile devices and edge sensors (De Vries, 2023). Hence, efforts are underway to incorporate energy efficiency into Artificial Intelligence (AI) models.

Such efforts are primarily focused on quantization, pruning (Liang et al., 2021), and model distillation (Poddar et al., 2025). These techniques reduce computational overhead by a significant margin, while preserving dense, real-valued operations, as seen in traditional artificial neural networks (ANNs) (Lv et al., 2024).

In contrast, Spiking Neural Networks (SNNs) (Maass, 1997) present a fundamentally different, event-driven paradigm inspired by biological neurons, where computation occurs only in response to discrete spikes. This leads to considerable reductions in computational overhead and, hence, energy demands (Lv et al., 2024; Knipper et al., 2024). Although SNNs have demonstrated strong results with competitive accuracy and significant energy savings in vision and sensory domains (Yamazaki et al., 2022), the applicability of SNNs to NLP tasks remains relatively underexplored. In one recent work, Lv et al. (2024) attempt to compare computational efficiency through both floating-point operations (FLOPs) for ANNs and synaptic operations (SOPs) for SNNs, providing a clear, dual-metric framework. This work demonstrates that SNN-based text classifiers can also achieve accuracy close to that of their ANN (i.e., Convolution Neural Network) counterparts while exhibiting enhanced robustness to adversarial attacks. In another direction, Knipper et al. (2024) investigate methods for encoding text into spikes and found that deterministic rate-coding techniques boost SNN text classification accuracy significantly compared to stochastic Poisson rate-coding, narrowing the performance gap between SNNs and ANNs. Together, these two studies establish the techniques for applying SNNs to efficient and robust natural language processing settings.

In this paper, we extend these state-of-the-art approaches and investigate the accuracy-energy trade-offs in a Spiking Neural Network for text classifica-

¹<https://github.com/PS1076/Energy-Accuracy-Frontier>

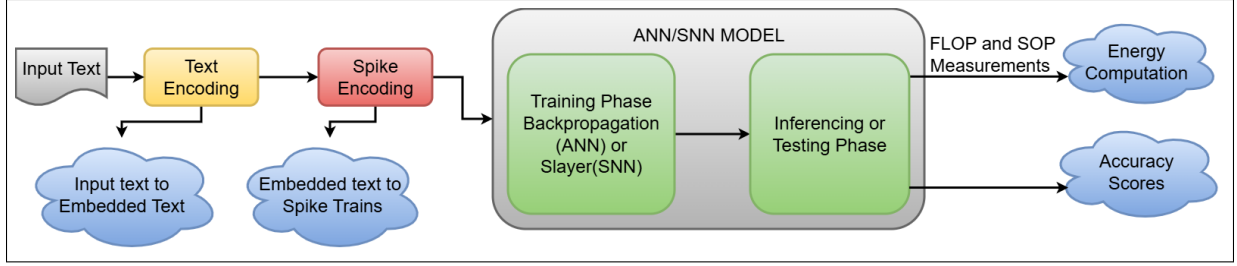


Figure 1: The flowchart of the pipeline used for our analysis.

tion tasks while dealing with low-resource Indian Languages like Hindi, Bengali, and Telugu. To ensure a fair comparison between ANN and SNN, we deploy a unified pipeline and measure the compute load using FLOPs (Floating-Point Operations) and SOPs (Synaptic Operations), respectively.

Our experiments show that, for the target low-resource languages, SNN reduces energy consumption by an average of 41.91x and 44.3x maximum compared to the ANN counterpart, while losing a minimal 0.895% in classification accuracy on average and a max of 3.99%. To the best of our knowledge, this is the first attempt to fairly compare the accuracy-energy trade-off of ANN and SNN for low-resource Indian languages. These findings will act as the stepping stone towards sustainable NLP, specifically in resource-constrained settings. We will make the code and data publicly available upon acceptance.

2 Proposed Pipeline and Workflow

The pipeline, as shown in Figure 1, consists of five key stages: Text embedding, Spike Train Encoding, Neural Model design (SNN/ANN), Training via backpropagation (for ANN) or surrogate gradients (for SNN), and Inference evaluation.

Text Embedding Input text (e.g., sentences, documents) is first converted into continuous-valued embeddings. For word-level tasks, pre-trained embeddings such as FastText (Bojanowski et al., 2017) are typically used, yielding 300-dimensional vectors. For sentence-level tasks, Transformer-based encoders such as Universal Sentence Encoder (USE) are employed to generate 512-dimensional vectors per sentence (Cer et al., 2018).

Spike Train Encoding A spike train is a temporal sequence of "spikes" (represented as 1s) and "no-spike" periods (represented as 0s). Unlike tra-

ditional artificial neural networks (ANNs) that communicate via continuous values, SNNs use these binary, event-driven signals to process data (Tavanaei et al., 2019).

Binarization (SNN-bin): Binarization reduces embeddings to binary (0/1) vectors using autoencoders or hash-based clustering. Here, each vector directly maps to either a spike or a non-spike event.

Poisson Rate Coding (SNN-rate-rand): In this method, we convert Floating-point values stochastically to spikes, simulating typical neural firing rates (Kalra and Sadanand, 2025).

Deterministic Rate Coding (SNN-rate): A self-accumulate-and-fire (SAF) neuron deterministically integrates an embedding component over time and generates spikes when a membrane threshold is crossed. This method provides a higher-fidelity representation of the input compared to stochastic Poisson coding, yielding notable accuracy improvements (Knipper et al., 2024). **Temporal Coding (e.g., Time-to-First-Spike):** The information is encoded in the precise timing of the first spike. Earlier spikes often represent stronger stimuli (Roy et al., 2019). In temporal coding, the network can often decide as soon as the first few spikes propagate through the layers, without waiting for the entire simulation window.

Neural Model Architecture We enforce strict architectural symmetry between the Artificial Neural Networks (ANNs) and Spiking Neural Networks (SNNs). Following the foundational framework proposed by Knipper et al. (Knipper et al., 2024), both paradigms are implemented as fully-connected, feedforward multilayer perceptrons. The dimensionality of the input layer is dynamically governed by the selected embedding strategy to accommodate various scales of linguistic representation: 512 neurons for continuous sentence-level embeddings, 300 for continuous word-level vectors, and 64 for binarized encoding

Language	Name	Class	Task	Source
Hindi	Sentiment Data(SDH)	7	Fine-grained emotion classification	SD source ¹
	IITP Product Review (PR)	3	Product review sentiment classification	Akhtar et al.(Akhtar et al., 2016)
	IITP Movie Review (MR)	3	Movie review sentiment classification	Akhtar et al.(Akhtar et al., 2016)
	BBC News Articles (BNA)	14	BBC Hindi news topic classification	BNA source ²
Bengali	Social Media Data (SMD)	2	Social media text classification	Hasan et al.(Hasan et al., 2023)
	Sentiment Data (SDB)	3	Standard sentiment analysis	Islam et al.(Islam et al., 2020)
	SentNoB (SNB)	3	Bengali text sentiment analysis	Islam et al.(Islam et al., 2021)
Telugu	Telugu sentiment analysis (TSA)	3	Sentiment analysis for Telugu corpus	Mukku et al.(Mukku and Mamidi, 2017)
	IndicNLP News (INN)	3	Telugu news classification	INN source ³
	INLTK Headlines (IH)	3	Telugu news headline classification	IN source ^{4,5}

Table 1: Data Set Description

schemes. These inputs propagate through a hierarchical feature-extraction pipeline with two hidden layers, empirically fixed at 256 and 128 neurons, respectively. The output layer size is strictly tailored to map to the number of target classes K dictated by the specific downstream classification task. Crucially, while the macroscopic topologies of the networks remain isomorphic, their underlying computational mechanisms diverge significantly. The baseline ANN utilizes standard continuous-valued, floating-point dense layers paired with Rectified Linear Unit (ReLU) (Nair and Hinton, 2010) activation functions. In contrast, the SNN operates fundamentally in the temporal domain. All continuous activations are replaced with event-driven, discrete-spiking nodes. Specifically, the SNN hidden layers employ Current-Based Leaky Integrate-and-Fire (CUBA LIF) (Shrestha and Orchard, 2018) neuron dynamics, characterized by distinct voltage and current decay parameters or the proposed Spiking Activation Function (SAF) (Ahmed and Orrje, 2023) neurons.

Training and Optimization For ANN Training, conventional backpropagation is applied using cross-entropy loss whereas for SNN Training, surrogate gradient methods (Neftci et al., 2019) such as SLAYER (Shrestha and Orchard, 2018) are used. SLAYER is a training algorithm designed for Spiking Neural Networks (SNNs) that addresses the challenge of non-differentiable spike activations by using surrogate gradients. It treats spikes probabilistically and approximates the gradient of the spiking function to enable backpropagation through time (BPTT). (Lv et al., 2024). Both model types are trained for the same number of epochs (typically 50) with analogous hyperparameters. We use Adam optimizer with a learning rate of 1×10^{-5} along with 15 samples/batch and 50 epochs of training. For

backpropagation, a spike-rate loss function with SLAYER’s surrogate gradient method is deployed.

Inference A major component of the inference time is the Inference Window. It is a hyperparameter that must be set to allow sufficient time for spike integration (e.g., a 50ms window). This must be tuned appropriately to achieve competitive accuracy while significantly reducing the latency penalty. The output neuron with the highest spike rate over the inference window determines the final classification.

3 Details of Core Components

Leaky Integrate-and-Fire (LIF) Neuron Dynamics. The core computational element in the SNN model is the Leaky Integrate-and-Fire (LIF) spiking neuron, modeled with discrete-time dynamics (Lv et al., 2024)

$$U_t = I_t + \beta \cdot U_{t-1} - S_{t-1} \cdot U_{thr} \quad (1)$$

$$I_t = W \cdot X_t \quad (2)$$

$$S_t = \begin{cases} 1, & \text{if } U_t \geq U_{thr} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Where U_t is the membrane potential at timestep t ; I_t is the input current, calculated by the weighted sum of inputs X_t with synaptic weights W ; β is the voltage decay factor (corresponding to membrane leak); S_t is the spike output, a binary signal indicating neuron firing when membrane potential crosses the threshold U_{thr} .

¹<https://www.kaggle.com/datasets/prath71018/hindi-sentiment-dataset/data>

²<https://github.com/NirantK/hindi2vec/releases/tag/bbc-hindi-v0.1>

³https://github.com/AI4Bharat/indicnlp_catalog

⁴<https://github.com/goru001/inltk>

⁵https://github.com/AI4Bharat/indicnlp_catalog

Language	Hindi (Hi)				Bengali (Bn)			Telugu (TI)		
Dataset	SDH	PR	MR	BNA	SMD	SDB	SNB	TSA	INN	IH
ANN Accuracy (%) (↑)	46.33	56.00	42.22	39.69	88.93	62.32	61.95	47.03	88.04	46.13
SNN Accuracy (%) (↑)	46.83	54.29	41.91	47.42	85.47	58.33	58.99	44.32	86.00	46.13
ANN FLOPs ($\times 10^8$)	1.33	0.57	0.34	0.72	2.56	1.51	1.74	0.59	2.64	0.59
ANN Energy ($\downarrow$) ($\times 10^7$ pJ)	165.69	71.88	42.61	90.48	319.73	188.71	217.98	74.36	329.86	74.36
SNN SOPs ($\times 10^8$)	4.96	2.25	1.32	2.65	10.90	5.89	6.83	2.35	9.76	2.31
SNN Energy ($\downarrow$) ($\times 10^7$ pJ)	3.82	1.74	1.02	2.04	8.36	4.53	5.26	1.81	7.51	1.78
Accuracy Drop (%) ANN $\rightarrow$ SNN	-0.5	1.71	0.31	-7.73	3.46	3.99	2.96	2.71	2.04	0
Energy Eff. Ratio ($\times$)	43.40	41.41	41.95	44.30	38.26	41.63	41.42	41.06	43.90	41.78

Table 2: Performance of ANN and SNN in terms of Accuracy and Energy Comparisons for text classification task across target languages. Hi $\rightarrow$ Hindi; Bn $\rightarrow$ Bengali; TI $\rightarrow$ Telugu.

Language	Hindi (Hi)				Bengali (Bn)			Telugu (TI)			
Dataset	SDH	PR	MR	BNA	SMD	SDB	SNB	TSA	INN	IH	Mean
$\bar{\gamma}$	0.373	0.395	0.388	0.368	0.426	0.390	0.393	0.398	0.370	0.392	≈ 0.389

Table 3: Firing rate ($\bar{\gamma}$) per dataset

Upon spiking, the membrane potential is reset by subtracting the threshold U_{thr} (soft reset). The LIF neuron acts as a stateful unit accumulating inputs over discrete simulation timesteps and emitting spikes based on internal dynamics.

Rate Coding Method. Processing the continuous-valued input representations like word or sentence embeddings within the spiking domain requires employing a deterministic rate-coding technique known as the Self-Accumulate-and-Fire (SAF) method (Ahmed and Orrje, 2023). Unlike stochastic encoding schemes, the SAF neuron deterministically accumulates the normalized input value at each discrete time step and triggers a spike exactly when the neuron’s membrane threshold is exceeded. The temporal dynamics of this encoding mechanism are mathematically defined as follows:

$$V_t = V_{t-1} + x \quad (4)$$

$$Y_t = \begin{cases} 1, & \text{if } V_t \geq \alpha \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

To preserve the residual potential and maintain a high-fidelity representation of the input value, the membrane potential undergoes a subtractive reset immediately following a spike emission (when $Y_t = 1$):

$$V_t \leftarrow V_t - \alpha \quad (6)$$

In this formulation: x represents the static input value from the specific embedding dimension being encoded, strictly normalized to the range $[0, 1]$. V_t denotes the internal membrane potential

of the encoding neuron, which steadily integrates the input x over successive time steps. α is the predefined neuron firing threshold that dictates the exact condition for encoding a spike. Y_t is the generated output spike train, forming a temporal binary sequence where 1 indicates an emitted spike and 0 indicates a resting state.

Energy Estimation Process. *Floating Point Operations (FLOPs)* (Lv et al., 2024) refer to the total number of arithmetic operations involving floating-point numbers performed by a model, including multiplications and additions, typically used to measure computational complexity and cost. On the other hand, *Synaptic Operations (SOPs)* (Hu et al., 2021; Indiveri et al., 2015) are the count of operations in a spiking neural network where energy is consumed only when spikes occur; The number of SOPs in an SNN layer is estimated by:

$$\text{SOPs}_{\text{layer}} = T \times \text{firing_rate}_{\text{layer}} \times \text{FLOPs}_{\text{layer}} \quad (7)$$

where, T is the number of discrete simulation timesteps ($T = 10$ timesteps for all datasets); $\text{firing_rate}_{\text{layer}}$ is the average neuron firing rate in a layer, defined as the fraction of active spikes in the input tensor to that layer, measured directly during inference via forward hooks. For each dataset, we also compute overall *firing rate* $\bar{\gamma}$ (averaged across the 3 layers), which summarizes how active the spiking network’s neurons are on average. The details are presented in Table 3. $\text{FLOPs}_{\text{layer}}$ is the number of floating-point operations required for the equivalent ANN layer. The SNN energy

consumption per layer:

$$E_{\text{SNN,layer}} = \text{SOP}_{\text{SNN,layer}} \times E_{\text{SOP}} \quad (8)$$

where $E_{\text{SOP}} = 77$ femtojoules is the theoretical energy cost per synaptic operation, as reported in the literature (Lv et al., 2024). On the other hand, the ANN energy consumption per layer:

$$E_{\text{ANN,layer}} = \text{FLOP}_{\text{ANN,layer}} \times E_{\text{FLOP}} \quad (9)$$

where $E_{\text{FLOP}} = 12.5$ picojoules is the theoretical energy cost per floating-point operation (Lv et al., 2024).

4 Experimental Results and Analysis

For our experiments, as shown in Figure 1, we deploy the pipeline in word-level, rate-coded, and deterministic SNN mode for various classification tasks. This choice is grounded in recent studies (Tavanai et al., 2019; Kim et al., 2022) that highlight the significant energy advantages of spiking models and the robust performance of rate-coded inputs in text-based tasks. Specifically, we utilize the fully-connected (MLP) SNN configuration, with deterministic rate-coding of word embeddings.

Dataset Description. Table 1 shows 10 datasets used in our experiments.

Data Preprocessing and Embedding. We preprocess the input text segments to lowercase, remove the punctuations, and normalize them to preserve script-specific characters for potential multilingual use. Each input token is mapped to a pre-trained 300-dimensional FastText word embedding (Bojanowski et al., 2017), enabling the model to capture both semantic and subword information inherent in the text. For SNN input, we normalize these embeddings to the range $[0, 1]$ and convert them to spike trains via the SAF deterministic coding scheme. The spike trains accumulate the normalized value at each time step and trigger spikes when the neuron membrane threshold is exceeded.

Classification Accuracy. We report classification accuracies (for ANN and SNN) for all three target low-resource languages in Table 2. We observe that for Hindi, Bengali, and Telugu, the drop in accuracy is marginal (at most 3.99%). Note that for two Hindi datasets (SD and BNA), SNN even outperforms ANN. The SNN performs the same as the ANN on one of the Telugu datasets (IH).

Energy Profiling. We conduct our experiments using two NVIDIA 48 GB L40 GPUs. During

inference, we compute the number of Synaptic Operations (SOPs) for each layer, enabling us to calculate the total SOPs across all layers. Next, we utilize the aggregated SOP count to estimate the approximate computational energy consumption of the SNN. Similarly, for the ANN, we compute the layer-wise count of Floating Point Operations (FLOPs), then aggregate them across all layers to determine the total FLOP count. We use the total FLOP count to approximate the ANN model’s overall computational energy expenditure. Table 2 presents the computational energy results derived from these measurements. Energy consumption results show consistent savings of 38.26x to 44.30x using SNN compared to ANN baselines, due to sparse firing ($\text{firing_rate} \ll 1$) and lower energy per synaptic operation. This shows the direction towards sustainable NLP for low-resource Indian languages.

5 Conclusion and Future Work

In this paper, we investigated using SNNs (instead of ANNs) for sustainable, energy-efficient NLP in low-resource languages. We focused on text classification, with Hindi, Bengali, and Telugu as our target low-resource languages. We conducted a comprehensive study to examine the accuracy-energy trade-off for ANNs and SNNs. The experiments showed that SNN saves a significant amount of energy (up to 44.30x) while incurring no more than 4% accuracy loss across a relatively large number of target languages. Note that, our experiments use identical configurations to isolate architectural effects, but each model family may benefit from its own hyperparameters and training duration.

Hence, going forward we plan to tune ANNs and SNNs independently under an equal tuning budget, including SNN-specific parameters such as timesteps, decay factor, and firing threshold, and train each to convergence. We will also account for training cost alongside inference energy, compare models at matched accuracy and matched energy, and explore ANN-to-SNN conversion and hybrid training to further close the accuracy gap. We would also like to extend SNN methods to transformer-like architectures, hardware-aware energy modeling, including memory operations, and to investigate different genres of NLP tasks in the near future.

References

- Hisham Ahmed and Martin Orrje. 2023. Comparison of output decoding techniques for spiking neural networks.
- Md Shad Akhtar, Asif Ekbal, and Pushpak Bhat-tacharyya. 2016. Aspect based sentiment analysis in hindi: resource creation and evaluation. In *Proceedings of the tenth international conference on language resources and evaluation (LREC’16)*, pages 2703–2709.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.
- Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope, and Ray Kurzweil. 2018. Universal sentence encoder for english. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 169–174.
- Alex De Vries. 2023. The growing energy footprint of artificial intelligence. *Joule*, 7(10):2191–2194.
- Md Arid Hasan, Firoj Alam, Anika Anjum, Shudipta Das, and Afiyat Anjum. 2023. Blp-2023 task 2: Sentiment analysis. *arXiv preprint arXiv:2310.16183*.
- Yangfan Hu, Huajin Tang, and Gang Pan. 2021. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*, 34(8):5200–5205.
- Giacomo Indiveri, Federico Corradi, and Ning Qiao. 2015. Neuromorphic architectures for spiking deep neural networks. In *2015 IEEE International Electron Devices Meeting (IEDM)*, pages 4–2. IEEE.
- Khondoker Ittehadul Islam, Md Saiful Islam, and Md Ruhul Amin. 2020. Sentiment analysis in bengali via transfer learning using multi-lingual bert. In *2020 23rd International Conference on Computer and Information Technology (ICCIT)*, pages 1–5. IEEE.
- Khondoker Ittehadul Islam, Sudipta Kar, Md Saiful Islam, and Mohammad Ruhul Amin. 2021. Sentnob: A dataset for analysing sentiment on noisy bangla texts. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3265–3271.
- Archit Kalra and Midhun Sadanand. 2025. Word2spike: Poisson rate coding for associative memories and neuromorphic algorithms. *arXiv preprint arXiv:2509.07361*.
- Youngeun Kim, Hyoungseob Park, Abhishek Moitra, Abhiroop Bhattacharjee, Yeshwanth Venkatesha, and Priyadarshini Panda. 2022. Rate coding or direct coding: Which one is better for accurate, robust, and energy-efficient spiking neural networks? In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 71–75. IEEE.
- R Alexander Knipper, Kaniz Mishty, Mehdi Sadi, and Shubhra Kanti Karmaker Santu. 2024. Snnlp: energy-efficient natural language processing using spiking neural networks. *arXiv preprint arXiv:2401.17911*.
- Tailin Liang, John Glossner, Lei Wang, Shaobo Shi, and Xiaotong Zhang. 2021. Pruning and quantization for deep neural network acceleration: A survey. *Neurocomputing*, 461:370–403.
- Changze Lv, Jianhan Xu, and Xiaoqing Zheng. 2024. [Spiking convolutional neural networks for text classification](#). *Preprint*, arXiv:2406.19230.
- Wolfgang Maass. 1997. Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9):1659–1671.
- Sandeep Sricharan Mukku and Radhika Mamidi. 2017. Actsa: Annotated corpus for telugu sentiment analysis. In *Proceedings of the first workshop on building linguistically generalizable NLP systems*, pages 54–58.
- Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML’10*, page 807–814, Madison, WI, USA. Omnipress.
- Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. 2019. Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks. *IEEE Signal Processing Magazine*, 36(6):51–63.
- Soham Poddar, Paramita Koley, Janardan Misra, Niloy Ganguly, and Saptarshi Ghosh. 2025. Towards sustainable nlp: Insights from benchmarking inference energy in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 12688–12704.
- Deboleena Roy, Priyadarshini Panda, and Kaushik Roy. 2019. [Synthesizing images from spatio-temporal representations using spike-based backpropagation](#). *Frontiers in Neuroscience*, Volume 13 - 2019.
- Sumit B Shrestha and Garrick Orchard. 2018. Slayer: Spike layer error reassignment in time. *Advances in neural information processing systems*, 31.
- Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. 2019. Deep learning in spiking neural networks. *Neural networks*, 111:47–63.

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.

Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. 2022. Spiking neural networks and their applications: A review. *Brain sciences*, 12(7):863.

AmhTTS: A Low-Resource Amharic Text-to-Speech System

Walegn Tewabe Sewunetie¹, Rahel Mekonen Tamiru², Israel D. Gebru²

¹AIMS Research & Innovation Centre, Kigali, Rwanda

²Ethiopian Artificial Intelligence Institute (EAIL), Addis Abeba, Ethiopia

³Meta Codec Avatars Lab in Pittsburgh, USA

Correspondence: wsewunetie@aimsric.org

Abstract

The scarcity of high-quality, phonetically diverse datasets for low-resource languages like Amharic remains a significant barrier to high-fidelity speech synthesis. We present **AmhTTS**, the first open, natively evaluated Amharic text-to-speech system, developed by fine-tuning SpeechT5 on a curated 25-hour, single-speaker Amharic corpus paired with a HiFi-GAN vocoder. Unlike Meta MMS, the only other publicly available Amharic TTS system, which requires Latin script transliteration and reports no Amharic-specific evaluation, AmhTTS operates natively on Ethiopic script and is the first system to report human MOS scores on Amharic speech synthesis. To address the orthographic complexities of the Ethiopic script, we implement a three-stage text normalisation pipeline covering homophone unification, *Ge'ez* numeral expansion, and Amharic punctuation handling. System quality is assessed through a rigorous human evaluation via a purpose-built web platform, involving **28 native Amharic-speaking voluntary evaluators** who rated 35 linguistically motivated samples across four domains: phonetic coverage (ejectives and labialized consonants), gemination and prosody, numeral and date reading, and conversational fluency. Results yield an overall naturalness MOS of **3.71** and intelligibility MOS of **4.02** on a 5-point scale, demonstrating that the rich pre-trained representations of SpeechT5 transfer effectively to a low-resource language with only 25 hours of data a fraction of the 460 hours used in the original English fine-tuning. While the system achieves high phonetic accuracy (MOS: 4.22/4.36), numeral synthesis and gemination modeling remain areas for future improvement. Given the demonstrated community demand for accessibility tools, we publicly release our fine-tuned model weights, the AmhTTS corpus, and evaluation data to advance low-resource TTS research. Our web application is currently active, serving

users at multiple university disability centers in Ethiopia.

<https://huggingface.co/amhtts>.

1 Introduction

Amharic is the official working language of Ethiopia, spoken natively by approximately 57 million people and used as a lingua franca by over 25 million additional speakers (Ethnologue, 2023). Despite its scale, Amharic remains severely under-resourced in speech technology, with no standardized TTS evaluation benchmark and very few publicly available systems Sebsibe et al. (2004).

Amharic presents several unique challenges for TTS synthesis. The Ge'ez/Ethiopic script encodes consonants and vowels in syllabic units (*fidel*), where multiple characters share identical pronunciations for example, **ሀ**, **ሐ**, and **ኸ** all realise /ha/ creating ambiguity during training. **Gemination** is phonemically contrastive yet difficult for neural models to learn reliably, while **ejective** and **labialized consonants** are typologically absent from English, limiting naive transfer learning. Additionally, Ethiopic numerals (**፩**, **፪**, ..., **፳**) and punctuation (**፡**, **፣**, **፥**) require dedicated normalisation pipelines absent from general-purpose pre-trained models.

Recent neural TTS architectures from the end-to-end Tacotron (Wang et al., 2017) and Tacotron 2 (Shen et al., 2018), to the non-autoregressive FastSpeech 2 (Ren et al., 2021) and the vocoderless VITS (Kim et al., 2021) have advanced synthesis quality significantly, but remain focused on high-resource languages. Meta MMS (Pratap et al., 2023), the only other publicly available Amharic TTS, provides no Amharic-specific evaluation and requires Latin transliteration via uroman, limiting practical usability. SpeechT5 (Ao et al., 2022), pre-trained exclusively on English, offers rich cross-modal representations

that we show transfer effectively to Amharic with only 25 hours of data.

In this work, we introduce **AmhTTS**, the first open, natively evaluated Amharic TTS system, fine-tuned from SpeechT5 (Ao et al., 2022) on a curated 25-hour single-speaker corpus with a HiFi-GAN vocoder (Kong et al., 2020). A three-stage text normalisation pipeline handles homophone unification, numeral expansion, and punctuation filtering. Human evaluation by 28 native Amharic-speaking evaluators yields a naturalness MOS of **3.71** and intelligibility MOS of **4.02**. Our main contributions are:

- The first open, fully reproducible Amharic TTS system operating natively on Ethiopic script without transliteration, with human MOS evaluation by native speakers.
- A three-stage text normalisation pipeline addressing homophone unification, *Ge'ez* numeral expansion, and Amharic punctuation handling.
- A linguistically motivated human evaluation framework combining ITU-T P.800 MOS (ITU-T, 1996) with forced transcription (Wester et al., 2015) involving 28 native Amharic-speaking evaluators across four phonetic domains (ejectives, gemination, numerals, and conversational fluency).
- Public release of model weights, the 25-hour AmhTTS corpus, evaluation data, and a live demo system to support reproducibility and Afro-Asiatic NLP research.

The remainder of this paper is organised as follows. Section 2 reviews related work. Section 3 describes the dataset and normalisation pipeline. Section 4 presents the model and fine-tuning. Section 5 details evaluation and results. Section 6 discusses findings and community impact. Section 7 concludes the paper, and Section 8 outlines the limitations of this work.

2 Related Work

2.1 Amharic Speech Synthesis

Early Amharic TTS relied on rule-based, cepstral, and concatenative synthesis with limited prosodic control (Sebsibe et al., 2004). More recent efforts have adapted neural sequence-to-sequence architectures, including Tacotron 2 (Shen et al., 2018)

and FastSpeech 2 (Ren et al., 2021) variants trained on modest corpora, with reporting naturalness scores between 3.2 and 3.8 MOS. The only other publicly available Amharic TTS system, Meta MMS (Pratap et al., 2023), covers 1,107 languages using a VITS-based (Kim et al., 2021) architecture but provides no Amharic-specific MOS evaluation and requires Latin transliteration via uroman. Table 1 summarises the landscape of relevant TTS systems; AmhTTS is the only system to report human MOS scores natively evaluated on Amharic.

2.2 Neural TTS Architectures

End-to-end TTS has progressed from the seq2seq Tacotron (Wang et al., 2017) and Tacotron 2 (Shen et al., 2018), to non-autoregressive models such as FastSpeech 2 (Ren et al., 2021) with explicit pitch, energy, and duration predictors. VITS (Kim et al., 2021) eliminated the need for a separate vocoder via VAE and GAN-based training, and serves as the backbone of Meta MMS (Pratap et al., 2023). Zero-shot multilingual models such as YourTTS (Casanova et al., 2022) and XTTS-v2 (Casanova et al., 2024) support voice cloning across languages with minimal data, though neither supports Amharic. SpeechT5 (Ao et al., 2022) unified speech and text in a shared encoder-decoder framework pre-trained on large-scale English data. AmhTTS adopts SpeechT5 as its backbone, demonstrating for the first time its cross-lingual transferability to a low-resource Afro-Asiatic language.

2.3 TTS Evaluation and African Language NLP

The ITU-T P.800 MOS framework is the standard for TTS naturalness assessment (ITU-T, 1996), while forced transcription provides a more reliable intelligibility measure for low-resource languages (Wester et al., 2015). AmhTTS combines both. The Mozilla Common Voice project has begun collecting Amharic speech (Ardila et al., 2020), and Masakhane-Speech (Olatunji et al., 2023) has produced benchmarks for several African languages. However, community-scale Amharic TTS evaluation with native speakers remains absent from the literature, a gap this work addresses.

Table 1: Comparative metrics for TTS systems relevant to Amharic (2026). MOS scores marked with † are reported on English benchmarks in the original papers, included for architectural reference only. Meta MMS Amharic checkpoint (facebook/mms-tts-amh) is released as part of the MMS project (Pratap et al., 2023) but Amharic is not explicitly evaluated in the paper. N/A = not reported. N/I = Naturalness / Intelligibility.

Model	Year	Key Feature	MOS (N/I)	Data Size	Language(s)	Data	Model
Tacotron (Wang et al., 2017)	2017	First end-to-end seq2seq TTS from characters	3.82† / N/A	24.6 hrs	English	Closed	Open
Tacotron 2 (Shen et al., 2018)	2018	Mel-spectrogram prediction conditioned WaveNet vocoder	4.53† / N/A	~24 hrs	English	Closed	Open
FastSpeech 2 (Ren et al., 2021)	2021	Non-autoregressive; explicit pitch, energy & duration	3.83† / N/A	~24 hrs	English	Open	Open
VITS (Kim et al., 2021)	2021	End-to-end VAE+GAN; no separate vocoder needed	4.43† / N/A	~24 hrs	English	Open	Open
YourTTS (Casanova et al., 2022)	2022	Zero-shot multi-speaker multilingual; ~1 min audio	3.98† / N/A	~1k spkrs	Multilingual (3)	Open	Open
SpeechT5 (Ao et al., 2022)	2022	Unified encoder-decoder pre-training for speech & text	3.65† / N/A	460 hrs	English	Open	Open
AmhTTS (Ours)	2025	SpeechT5 fine-tuned on Amharic; first open natively evaluated system	3.71 / 4.02	25 hrs	Amharic	Open	Open

† Amharic checkpoint at facebook/mms-tts-amh; not evaluated in Pratap et al. (2023).

3 Dataset and Text Normalisation

3.1 Training Corpus

AmhTTS is built on a high-fidelity, single-speaker corpus of **25 hours** of speech recorded by a female native Amharic speaker, comprising approximately **11,000 sentences**. The corpus contains **254 unique Ethiopic characters** with **747,546** total occurrences, ranging from a minimum count of 1 (፩, ፪, ፫) to a maximum of 43,622 (፳፻፲፱). Table 2 summarises the corpus statistics. Figure 1 visualises the character frequency distribution.

Table 2: AmhTTS training corpus statistics.

Property	Value
No. of sentences	11,000
Max sentence length	23 sec
Min sentence length	5 sec
Avg sentence length	≈16 sec
Total duration	25 hours
Sampling rate	16,000 Hz
Speaker	Female, native Amharic
Unique characters	254
Total characters	747,546

3.2 Text Normalisation Pipeline

Amharic orthography (Ethiopic Abugida, Unicode U+1200–U+137F) presents pre-processing challenges absent from general-purpose pre-trained models. We implement a three-stage pipeline.

Stage 1: Homophone Unification. Several Ge’ez characters share identical phonetic realisations, creating training noise. We map variants to

canonical forms: {ሀ, ሐ, ኀ, ኸ} → ሀ; {ሰ, ሠ} → ሰ; {ጸ, ፀ} → ጸ. Table 3 summarises key script properties.

Table 3: Amharic script properties affecting TTS.

Property	Details
Base consonants	33, each with 7 vowel orders (231 base forms)
Homophone groups	ሀ/ሐ/ኀ; ሰ/ሠ; ጸ/ፀ
Ge’ez numerals	፩=1, ፪=2, ፫=100, ፬=10,000
Key punctuation	፥ stop, ፣ comma, ፤ semi, ፡ question, ፡ word sep.

Stage 2: Numeral Expansion. Both Arabic (e.g., 2024) and Ge’ez numerals (e.g., ፳፻፲፱) are expanded to full Amharic written-out equivalents. Without this step the model skips or mispronounces numeric tokens (G3: MOS 2.92/3.16).

Stage 3: Punctuation Filtering. Non-standard symbols are stripped, while essential markers ፥, ፣, ፤, ፡ are preserved and mapped to prosodic pause durations.

4 Model Architecture and Fine-Tuning

4.1 Base Model: SpeechT5

AmhTTS fine-tunes **SpeechT5** (Ao et al., 2022), a unified-modal encoder-decoder pre-trained on large-scale unlabeled English speech and text. The framework consists of a shared Transformer encoder-decoder backbone and six modal-specific pre/post-nets (Ao et al., 2022). For TTS, text tokens are mapped via a text pre-net; the encoder-

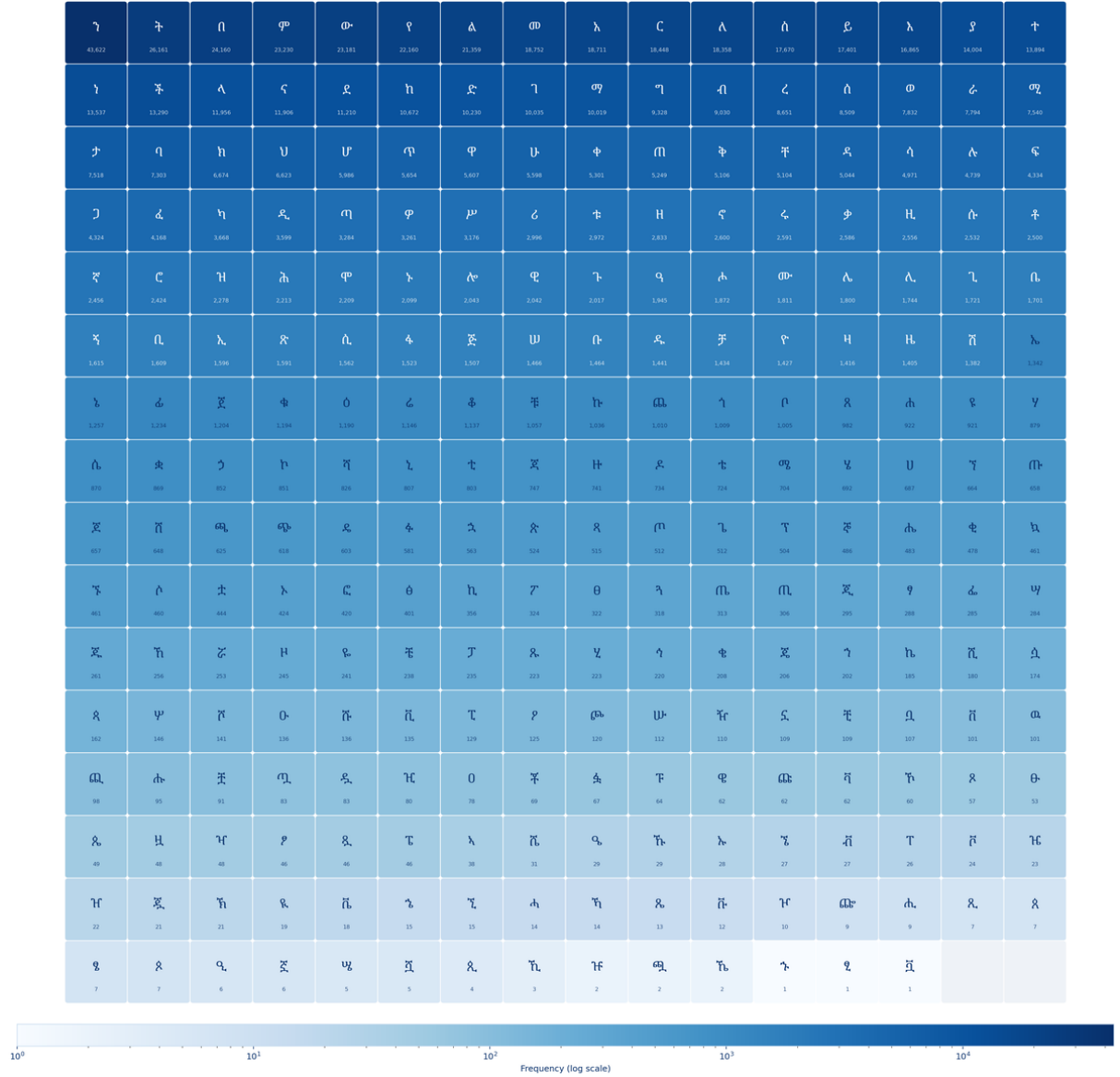


Figure 1: Frequency heatmap of the 254 Ge’ez characters in the training corpus. Each cell shows the character glyph and its raw count. Colour intensity is logarithmic (light blue = rare, deep navy = frequent). Characters are ordered left-to-right, top-to-bottom by decreasing frequency.

decoder models the sequence-to-sequence transformation; and the speech post-net refines predicted log Mel-filterbank features. **HiFi-GAN** (Kong et al., 2020) converts mel-spectrograms to waveforms. Speaker identity is conditioned via x-vector embeddings. The original SpeechT5 was pre-trained and evaluated exclusively on English; AmhTTS is among the first efforts to adapt it to a low-resource morphologically rich language.

4.2 Fine-Tuning Procedure

Fine-tuning adapts the SpeechT5 attention mechanism to Ethiopic syllabic units using the normalised corpus (Section 3.2). Training uses L1

loss on log Mel-filterbank features and binary cross-entropy (BCE) loss on the stop token, following the original SpeechT5 procedure (Ao et al., 2022). All experiments used a single **NVIDIA Tesla V100-SXM2** (32 GB GPU), fully utilised over **≈30.5 hours** of training via Transformers 4.38.1 (Wolf et al., 2020) and PyTorch 2.1.0. Table 4 summarises all hyperparameters.

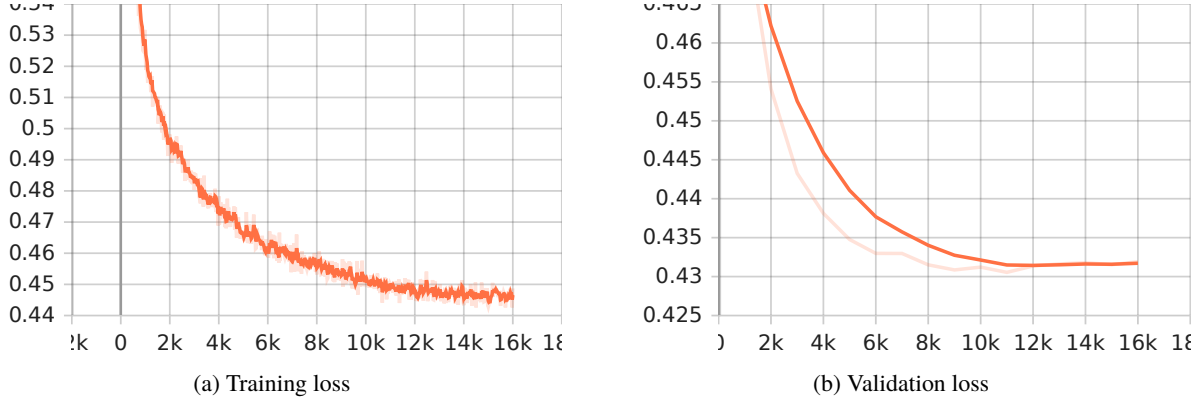


Figure 2: Training and validation loss over 16,000 steps (Weights & Biases). *Left*: Training loss decreases from 0.5244 (step 1k) to 0.4487 (step 16k). *Right*: Validation loss drops from 0.4757 (step 1k) to a minimum of **0.4305** at step 11,000, then stabilises at 0.4319 at the final checkpoint.

Table 4: Fine-tuning hyperparameters and compute details.

Hyperparameter	Value
Base model	microsoft/speecht5_tts
Vocoder	HiFi-GAN
Learning rate	1e-5
Train / Effective batch	16 / 128
Gradient accum. steps	8
Optimizer	Adam ($\beta_1=0.9$, $\beta_2=0.999$)
LR scheduler	Linear + 500 warmup steps
Total steps / Epochs	16,000 / ≈ 262
Sampling rate	16,000 Hz
Seed	42
Hardware	1 $\times$ V100-SXM2 (32 GB)
Training runtime	≈ 30.5 hrs
Framework	Transformers 4.38.1 / PyTorch 2.1.0
Best eval loss	0.4305 (step 11,000)
Final eval loss	0.4319 (step 16,000)

4.3 Training Dynamics

Figure 2 shows the loss curves over 16,000 steps. Validation loss drops from 0.4757 (step 1,000) to a minimum of **0.4305** at step 11,000, then stabilises at 0.4319. The plateau after step 9,000 is consistent with the limited data regime typical of low-resource fine-tuning. The slight uptick in training loss at step 16,000 (0.4487 vs. 0.4428 at step 15,000) with stable validation loss indicates no significant overfitting, attributable to the learning rate decaying to $\approx 1e-6$. The released model uses the **final checkpoint** (step 16,000, val. loss 0.4319), consistent with standard practice. This loss is within the 0.40–0.48 range reported for similar multilingual encoder-decoder fine-tuning on 20–40 hour African language datasets (Elmadany et al., 2025; Bakoubolo et al., 2026).

5 Evaluation and Results

5.1 Test Dataset

We constructed 35 test sentences in four groups (Appendix B): **G1** phonetic coverage ejectives, labialized consonants, 6th-order vowels (S1–S10); **G2** gemination and prosody minimal pairs (S11–S20); **G3** numbers, dates, and punctuation (S21–S25); **G4** conversational and domain-specific speech (S26–S35).

5.2 Results

Quality was assessed via ITU-T P.800 MOS (ITU-T, 1996) with forced transcription for intelligibility (Wester et al., 2015), collecting naturalness and intelligibility ratings (1–5) from **28 evaluators** (26 native, 2 fluent non-native Amharic speakers) and **30 community users**. As shown in Table 5, AmhTTS achieves naturalness MOS of **3.71** and intelligibility MOS of **4.02**, exceeding SpeechT5’s English baseline (Ao et al., 2022) (3.65 on 460 hrs) with only 25 hours of Amharic data. All 28 evaluators expressed willingness to use the system (“Yes”: 14; “Maybe”: 14).

Table 5: Overall MOS ($N=28$, 35 sentences).

Metric	Score	Ref. Data
Naturalness (per-sent.)	3.71	—
Intelligibility (per-sent.)	4.02	—
Naturalness. (self-reported)	3.61	—
Intelligibility. (self-reported)	3.93	—
SpeechT5 (Ao et al., 2022)	3.65	460 hrs
Tacotron (Wang et al., 2017)	3.82	24.6 hrs

6 Discussion

6.1 Comparison with Related Systems

As shown in Table 1, AmhTTS achieves naturalness MOS 3.71 and intelligibility MOS 4.02, exceeding the English SpeechT5 baseline (Ao et al., 2022) (3.65 on 460 hours) with only 25 hours of Amharic data. This demonstrates effective cross-lingual transfer of pre-trained encoder-decoder representations consistent with the broader transfer learning literature (Ren et al., 2021; Kim et al., 2021). AmhTTS is the first system to report native Amharic MOS and to operate directly on Ethiopic script without transliteration.

6.2 Results by Sentence Group

Table 6 reveals a clear hierarchy. G1 (phonetic coverage) achieved the highest scores (4.22/4.36), confirming solid ejective consonant handling. G4 (conversational) also scored strongly (3.73/4.18), reflecting good coverage of high-frequency vocabulary. G2 (gemination) shows a naturalness–intelligibility gap (3.58 vs. 3.96), suggesting evaluators recovered words even when gemination was imperfect, consistent with Sebsibe et al. (2004). G3 (numbers/dates) scored lowest (2.92/3.16), confirming numeral normalisation as the critical weakness.

Table 6: Mean MOS by sentence group.

Group	N	Nat.	Int.
G1: Phonetic Coverage	10	4.22	4.36
G2: Gemination/Prosody	10	3.58	3.96
G3: Numbers/Dates	5	2.92	3.16
G4: Conversational	10	3.73	4.18
Overall	35	3.71	4.02

6.3 Phonetic Error Analysis

Table 7 summarises phonetic error annotations across 980 observations. No errors were reported in 41.9% of cases. Gemination (17.9%) and ejective errors (16.3%) were the leading types, while number errors (4.9%) concentrated in G3. The lowest-scoring sentence, S21 (ዛሬ ሐምሌ ፩ ቀን ፳፻፲፯ ዓ.ም ነው።, Ge’ez numeral date), received naturalness 1.39 and intelligibility 1.36. Conversely, S5 (ኢትዮጵያ የብዙ ቋንቋዎች መገኛ ናት) achieved the highest scores (4.82/4.86), confirming strong synthesis on high-frequency vocabulary.

Table 7: Phonetic error distribution ($N=980$).

Error Type	Count	%
No errors detected	410	41.8
Other issues	175	17.9
Gemination unclear	175	17.9
Ejective sounds unclear	160	16.3
Number pronunciation unclear	60	6.1
Total	980	100.0

6.4 Failure Analysis

Gemination is not encoded at the character level in Ethiopic script; the model must infer consonant duration from context a task for which SpeechT5, pre-trained on English, has no prior. Non-autoregressive models with explicit duration predictors (Ren et al., 2021) offer a natural solution. **Numeral synthesis** failures stem from incomplete Ge’ez numeral handling in the normalisation pipeline, as evidenced by S21. Additional evaluator feedback identified: (1) male voice variants; (2) context-dependent homophone disambiguation; (3) improved prosody at punctuation boundaries; and (4) code-switched Amharic–Latin text support.

7 Conclusion

We presented **AmhTTS**, the first open, natively evaluated Amharic TTS system, fine-tuned from SpeechT5 (Ao et al., 2022) on a 25-hour single-speaker corpus using a three-stage Ethiopic normalisation pipeline. Human evaluation by 28 native speakers yielded a naturalness MOS of **3.71** and an intelligibility MOS of **4.02**, exceeding the English SpeechT5 baseline despite using a fraction of the training data. Current limitations include single-speaker coverage, suboptimal gemination (G2: 3.58/3.96), and weak numeral synthesis (G3: 2.92/3.16), attributable in part to the English-only SpeechT5 pre-training, which introduces systematic bias against ejective consonants and gemination features absent from its pre-training distribution. Future work will incorporate an explicit duration module, comprehensive numeral normalisation, and code-switched Amharic–Latin text support. The system has been publicly released and is actively used by visually impaired students, demonstrating its practical value for accessibility and digital inclusion in Amharic-speaking communities.

8 Limitations

Linguistic and Speaker Diversity. AmhTTS is trained on a single female speaker (Addis Ababa accent), limiting generalisation to regional varieties such as Gojjam and Wollo Amharic. Gemination handling remains suboptimal (G2: 3.58/3.96), and numeral synthesis requires a dedicated component (G3: 2.92/3.16). While our web application is currently active across multiple university disability centers in Ethiopia, the evaluator and initial user pool skew toward urban, educated academic environments, which may not fully capture rural user demographics or varying literacy levels upon wider nationwide deployment.

Transfer Learning and Infrastructure Constraints. SpeechT5 (Ao et al., 2022) is pre-trained exclusively on English, introducing a systematic bias against ejective consonants and gemination features absent from the pre-training distribution. This architectural foundation may structurally limit fine-tuning performance independently of data quantity. Furthermore, because our tool is currently deployed as a centralized web application, its utility is tightly coupled with internet stability. Optimization for lightweight, offline mobile-edge deployment remains a critical upcoming challenge to support users in low-connectivity areas.

9 Ethical Considerations

We acknowledge the dual-use risk of TTS technology, including the potential for audio deepfakes or malicious voice cloning. However, the primary motivation of this work is to improve accessibility for visually impaired Amharic speakers and promote digital inclusion in Ethiopian higher education. To address these risks transparently, the model weights, the AmhTTS corpus, and evaluation data are released under an open licence to advance low-resource, accountable research. No personally identifiable information was collected from users or evaluators; university center participants engaged with the application voluntarily with informed consent.

References

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, and 1 others. 2022. [SpeechT5: Unified-modal encoder-decoder pre-training for spoken language](#)

[processing](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5723–5738.

Rosana Ardila and 1 others. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of LREC*.

Justin E. Bakoubolo, Catherine Nana Nyaah Essuman, Messan Agbobli, Ahoefa Kansier, Eli Jean Doumeyan, Julie Pato, Notou Your Timibe, Emile KOGBEDJI Agossou, and Guedela Bakouya. 2026. [Eyaa-tom 26, yodi-mantissa and lom bench: A community benchmark for TTS in local languages](#). In *Proceedings of the 7th Workshop on African Natural Language Processing (AfricanNLP 2026)*, pages 264–270. Association for Computational Linguistics.

Edresson Casanova, Kelly Davis, Eren Gölge, Görkem Gökna, İulian Gulea, Logan Hart, Aya Aljafari, Joshua Meyer, Reuben Morais, Samuel Olayemi, and Julian Weber. 2024. [XTTS: a massively multilingual zero-shot text-to-speech model](#). *arXiv preprint arXiv:2406.04904*.

Edresson Casanova, Julian Weber, Christopher D Shulby, Arnaldo Candido Junior, Eren Gölge, and Moacir A Ponti. 2022. [YourTTS: Towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone](#). In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 2709–2720.

AbdelRahman Elmadany, Sang Yun Kwon, Hawau Olamide Toyin, Muhammad Abdul-Mageed, and 1 others. 2025. [Voice of a continent: Mapping Africa’s speech technology frontier](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Ethnologue. 2023. [Amharic](#).

ITU-T. 1996. Methods for subjective determination of transmission quality. Technical Report P.800, International Telecommunication Union.

Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. [Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech](#). In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 5530–5540.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Advances in Neural Information Processing Systems*, volume 33, pages 17022–17033.

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. 2023. [AfriSpeech-200: Pan-African accented speech dataset for clinical and general domain](#)

- [ASR](#). *Transactions of the Association for Computational Linguistics*, 11:1669–1685.
- Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Saranya Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2023. [Scaling speech technology to 1,000+ languages](#). In *arXiv preprint arXiv:2305.13516*.
- Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. [FastSpeech 2: Fast and high-quality end-to-end text to speech](#). In *International Conference on Learning Representations (ICLR)*.
- Yared Sebsibe, Solomon Kebede, and 1 others. 2004. Development of an Amharic text-to-speech system using cepstral method. In *Proceedings of the Workshop on Computational Approaches to Arabic Script-based Languages*, pages 83–89.
- Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, and 1 others. 2018. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783.
- Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, and 1 others. 2017. [Tacotron: A fully end-to-end text-to-speech synthesis model](#). *arXiv preprint arXiv:1703.10135*.
- Mirjam Wester, Cassia Valentini-Botinhao, and Gustavo Eje Henter. 2015. [Are we using enough listeners? No! — An empirically-supported critique of Interspeech 2014 TTS evaluations](#). In *Proceedings of Interspeech*, pages 3476–3480.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of EMNLP: System Demonstrations*, pages 38–45.

A Appendix

B Test Dataset: 35 Amharic Sentences with MOS Scores

Table 8: Full test dataset with per-sentence MOS scores. (Nat. = Naturalness; Int. = Intelligibility)

S	Sentence	Nat.	Int.
Group 1: Phonetic Coverage (Ejective & Labialized)			
1	ጳውሎስ በሊባኖስ ተራራ ላይ ቆመ። (ጳ, ቆ)	4.18	4.61
2	ጨረቃ በጨለማ ብርሃን ትሰጣለች። (ጨ, ቃ)	4.71	4.71
3	ጥንቃቄ ለጤና በጣም አስፈላጊ ነው። (ጥ, ጤ)	4.57	4.75
4	ፀሐይ ጠዋት በምስራቅ በኩል ትወጣለች። (ፀ, ሐ)	4.57	4.82
5	ኢትዮጵያ የብዙ ቋንቋዎች መገኛ ናት። (ጵ)	4.82	4.86
6	የጭቃ ሹም በፈረስ ላይ ተቀመጠ። (ጭ, ሹ)	4.21	3.96
7	ቅዳሜና እሁድ ገበያ በጣም ይሞላል። (ቅ, ሁ)	4.36	4.86
8	መጽሐፍ ማንበብ እውቀትን ይጨምራል። (ጽ, ሐ)	4.71	4.75
9	ውሃ ጠጪ ብሎ አዘዛት። (Labialization)	3.39	3.75
10	ፍየሉ ቁጥቋጦውን በላው። (ቁ, ቋ)	2.64	2.54
Group 2: Gemination & Prosody			
11	ገና አልመጣም። (ገና = “yet”)	2.79	2.86
12	ገና በዓልን በደስታ አሳለፍን። (ገና = “Christmas”)	3.25	3.32
13	አባባ ደብዳቤ ጻፈ። (Plosive ‘b’)	2.50	2.82
14	እሱ ትልቅ ሰው ነው። (‘S’ sounds)	4.54	4.71
15	መመላለስ ሰልችቶኛል። (Repeated)	3.43	3.86
16	እሷ በጣም ጎበዝ ተማሪ ናት። (Feminine pitch)	4.43	4.82
17	ወንድሜ ትናንት መጣ። (Gemination ‘n’)	3.68	4.29
18	ይህ ሻይ በጣም ይፋጃል። (Affricates)	3.11	3.86
19	አንበሳው በጫካው ውስጥ ይጮኻል። (ኸ)	3.82	4.39
20	ሰማዩ ዛሬ በጣም ሰማያዊ ነው። (Vowels)	4.25	4.64
Group 3: Numbers, Dates & Punctuation			
21	ዛሬ ሐምሌ ፭ ቀን ፳፻፲፮ ዓ.ም ነው። (Ge’ez numerals)	1.39	1.36
22	በክፍሉ ውስጥ 12 ተማሪዎች አሉ። (Arabic numerals)	4.61	4.82
23	ሰዓቱ አራት ከሠላሳ ሆኗል። (Time format)	3.00	3.14
24	ዋጋው 500 ብር ከ60 ሳንቲም ነው። (Currency)	2.96	3.25
25	ስልክ ቁጥሩ 0911-22-33-44 ነው። (Phone numbers)	2.64	3.21
Group 4: Conversational & Domain-Specific			
26	እንደምን አደሩ? ዛሬ እንዴት ነዎት? (Formal)	3.14	3.64
27	የአየር ሁኔታ ትንበያው ዝናብ እንደሚዘንብ ተናግሯል። (News)	4.00	4.11
28	ኢንተርኔት ዓለምን ወደ አንዲት መንደር ቀይሯታል። (Tech)	4.39	4.68
29	ጤና ይስጥልኝ! ስሜ ዋለልኝ ይባላል። (Intro)	2.86	3.57
30	ቡና መጠጣት ትወዳለህ ወይስ ሻይ? (Question)	2.93	3.68
31	የአፍሪካ አንድነት ድርጅት ዋና መሥሪያ ቤት አዲስ አበባ ነው። (Political)	4.21	4.57
32	ባቡሩ በአስር ደቂቃ ውስጥ ይደርሳል። (Transport)	3.89	4.50
33	ምሳ ለመብላት የት እንሂድ? (Casual)	3.18	3.75
34	ትምህርት ለትውልድ ዕድገት ወሳኝ ነው። (Educational)	4.29	4.61
35	ኳስ መጫወት የአካል ብቃት እንቅስቃሴ ነው። (Sports)	4.36	4.71

Marker Signal Void: Linguistic Deception Fingerprinting and Transformer Benchmarking on a Domain-Stratified Bangla Rumour Corpus

Farzana Zahid¹ Sadman Bin Rashid¹ Israt Jahan Anika¹ Md. Rakibul Islam¹

¹Ahsanullah University of Science and Technology

farzanazahid314@gmail.com, sadman.cse.20210204102@aust.edu

isratjahan7908@email.com, rakib.cse@aust.edu

Abstract

Rumour detection in Bangla remains severely under-resourced, with prior work conflating it with fake news classification and lacking domain-stratified analysis of the linguistic properties that distinguish misinformation from authentic content. We introduce MisInfo-11K, a domain-stratified corpus of 11,198 instances spanning seven topical domains, constructed from Bangla fact-checking portals and verified news outlets. Using 14 linguistically motivated markers, we show via chi-square and Kruskal-Wallis analysis that 13 of 14 exhibit significant cross-domain variation, revealing distinct deception strategies (e.g., authority mimicry in political rumours, fear amplification in health rumours). Building on this, we introduce the Marker Signal Void (MSV), a chi-square-weighted scoring framework identifying rumour instances with no net discriminative surface signal, establishing Macro-F1 = 0.9843 as the ceiling achievable by this 14-marker set. Benchmarking three pre-trained transformers (BanglaBERT, MuRIL, XLM-RoBERTa) against five traditional models, MuRIL achieves the strongest performance (Macro-F1 0.9800), while a marker-augmented Random Forest reaches 0.9750, surpassing XLM-RoBERTa. The gap between the MSV ceiling and transformer performance quantifies the contribution of contextual semantic representations beyond surface-level linguistic cues.

1 Introduction

The 2019 Padma Bridge lynchings in which crowds killed several people acting on a false Facebook rumour that children’s heads were needed as sacrificial offerings for the bridge’s construction illustrate that unverified claims in active circulation cause physical harm well before their veracity can be established (Shirina and Prodhan, 2020). Yet computational approaches to Bangla *rumour detection* remain absent from the literature: existing work addresses fake news detection binary

classification of fabricated articles which is categorically distinct from rumour detection, since rumours are unverified claims whose veracity is actively contested at propagation time (Zubiaga et al., 2018). Prior work has explored Bangla fake news detection using linguistic and psychological features (Mahmud et al., 2026), but this targets fake news rather than rumours, is not domain-stratified, and does not characterise cross-domain linguistic variation; no domain-stratified Bangla rumour corpus exists, and no theoretical framework has examined the limits of surface-level detection in this language.

We introduce **MisInfo-11K** through four contributions: (i) the first domain-stratified Bangla rumour corpus spanning seven topical domains, establishing the empirical foundation for cross-domain analysis; (ii) 14 validated linguistic markers producing domain-specific deception fingerprints that reveal what properties distinguish Bangla rumours from authentic content, via χ^2 and Kruskal-Wallis analysis; (iii) the **Marker Signal Void (MSV)**, a novel chi-square-weighted scoring framework quantifying, for our validated 14-marker set, the performance ceiling of surface-level detection and identifying which specific instances fall outside its reach; (iv) a benchmark of three pretrained transformers and five traditional models quantifying whether contextual representations contribute signal beyond surface markers — MuRIL achieves Macro-F1 0.9800 and a marker-augmented Random Forest reaches 0.9750, surpassing XLM-RoBERTa.

2 Related Work

Rumour Detection. PHEME (Zubiaga et al., 2016) and RumourEval (Gorrell et al., 2019) established rumour detection on Twitter, covering breaking-news events and formalising stance/veracity subtasks, while Bi-GCN (Bian

et al., 2020) showed propagation structure carries signal beyond text. A consistent finding is that linguistic features degrade across event domains (Zubiaga et al., 2018), directly motivating our domain-stratified analysis.

Bangla Fake News Detection. Hossain et al. (2020) introduced BanFakeNews ($\approx 50K$), the foundational Bangla benchmark, with follow-up work applying CNN-LSTM, BiGRU, and BERT-based classifiers; all prior work is confined to fake news classification on a domain-undifferentiated corpus, with none addressing rumour detection, domain stratification, or detection limits.

Linguistic Markers and Low-Resource Misinformation. Claim verbs, attribution patterns, and negation density discriminate misinformation across languages (Newman et al., 2003), but analyses remain single-domain, masking cross-domain variability; low-resource languages remain severely underserved (Chalehchaleh et al., 2024), with deception cues transferring unreliably across morphologically distinct languages, reinforcing the need for language-specific corpus construction.

Pretrained Language Models for Bangla. Bhattacharjee et al. (2022) introduced BanglaBERT (27.5GB Bangla web text, state-of-the-art on four Bangla NLU tasks); MuRIL (Khanuja et al., 2021) extends multilingual BERT with dedicated support for 17 South Asian languages including Bangla, while XLM-RoBERTa (Conneau et al., 2020) is the standard high-capacity multilingual baseline. We evaluate all three on the first domain-stratified Bangla rumour corpus for reproducible cross-family baselines.

3 Dataset: MisInfo-11K

3.1 Collection Methodology

Rumour instances were collected from three dedicated Bangla fact-checking portals — **Rumor Scanner**,¹ **Boom Bangladesh**,² and **Jachai**³ — via a multi-threaded, rate-limited scraper. URL-level deduplication within the training split prevented duplicate collection. Each fact-check headline is treated as a rumour instance (label 1).

¹<https://rumorsscanner.com/>

²<https://boombd.com>

³<https://jachai.org>

We use only the fact-check portal’s headline, not the article body, for two reasons. First, the headline reproduces the claim as it actually circulated on social media — this is the unit that misleads readers, and fact-check portals preserve it verbatim as the object of their verification. The article body, by contrast, is the fact-checker’s own debunking explanation and would trivially leak the rumour label if included. Second, using headlines on both sides keeps the comparison at a consistent structural level: non-rumour instances are likewise drawn from news headlines, since headline text drives engagement and exposure on social media, independent of the outlet’s editorial register. The task is therefore defined at the level readers actually encounter misinformation headline against headline rather than full-article content.

Non-rumour instances were collected from nine established Bangladeshi news outlets **Naya Diganta**, **NTV**, **MY TV**, **Manobzamin**, **RisingBD**, **BD Pratidin**, **Bangla Vision**, **Kaler Kantho**, and **Jugantor** — via RSS feeds and paginated HTML listings, filtered by a Unicode Bangla character check and a 15–300 character length constraint. Non-rumour labels were assigned programmatically and never sourced from fact-check platforms to prevent label leakage.

3.2 Domain Taxonomy

Seven domains were defined a priori based on the thematic coverage of Bangla fact-checking portals: political, health, religious, cultural, sports, celebrity, and international. Labels were assigned automatically via a Bangla keyword lexicon curated per category. Rumour instances may carry multiple domain labels reflecting co-occurring themes; non-rumour instances carry exactly one label derived from the source category. Instances with no keyword match default to political, reflecting political content’s empirical dominance in Bangla misinformation and its broad topical overlap with unlabelled instances. We acknowledge this fallback introduces a structural asymmetry: the political domain’s marker profile is a mixture of genuinely keyword-matched content and residual unclassified instances. However, we note that this pooling effect is directionally conservative rather than inflationary — merging heterogeneous, topically diffuse content into a single bucket tends to dilute rather than manufacture a distinctive marker signature, since unrelated instances regress the do-

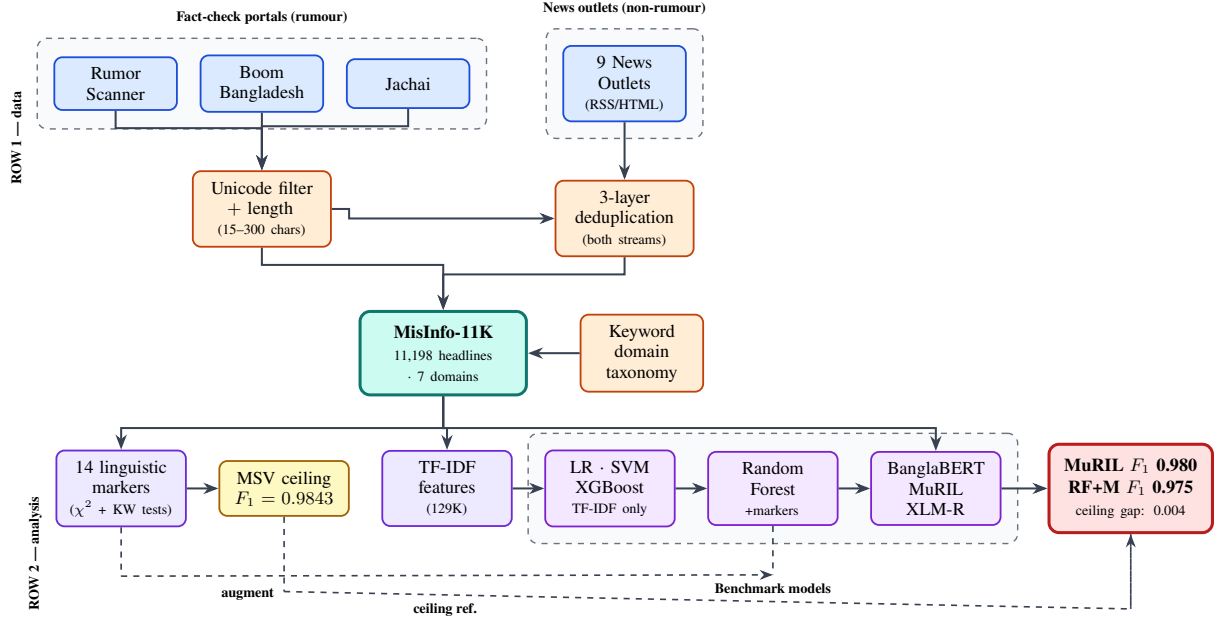


Figure 1: **MisInfo-11K pipeline.** **Row 1:** Rumour headlines from three fact-checking portals and non-rumour headlines from nine news outlets pass through quality-control filters and 3-layer deduplication (both streams) into the domain-stratified corpus. **Row 2:** Three parallel paths — (i) linguistic markers $\rightarrow$ MSV ceiling $F_1=0.9843$; (ii) TF-IDF $\rightarrow$ traditional classifiers with marker augmentation for Random Forest; (iii) transformer fine-tuning — converging on the benchmark results.

main’s mean activation rates toward the corpus-wide baseline. The Kruskal–Wallis result reported in Section 4.2 (13 of 14 markers significant across domains, $p < 0.05$) should therefore be read as, if anything, an underestimate of true cross-domain variation, since any homogenising effect of the fallback would work against detecting domain-specific signal rather than artificially creating it. We flag this as a taxonomy limitation in Section 7 (Limitations) and recommend that future work either introduce an explicit *Uncategorised* category or validate fallback assignments against a larger human-annotated sample.

3.3 Dataset Statistics

MisInfo-11K contains 11,198 instances across training (8,198), validation (1,400), and test (1,400) splits, each evaluation split balanced 50/50 between classes. Table 1 reports the full breakdown by split. Figure 2 shows domain distribution across both classes, revealing a notable asymmetry: political is the dominant rumour domain (2,407 instances) yet the most underrepresented in non-rumour (204), while sports and religious show the inverse pattern. This asymmetry motivates the domain-stratified analysis in Section 4.2.

3.4 Quality Control

Three layers of quality control were applied: exact-match deduplication on the text field to remove conflicting labels from headlines appearing across multiple sources; a URL and headline blacklist built from the training split to prevent any training instance from appearing in validation or test; and explicit cross-contamination checks confirming zero overlap across all three splits. Instances shorter than five characters, containing only ASCII, or matching known portal homepage titles were discarded.

Beyond deduplication, taxonomy reliability was validated through human annotation. Three annotators independently labelled a 200-instance stratified subset across all seven domains. Pairwise Cohen’s κ was computed for each domain, yielding a macro-averaged κ of 0.746 (*good* agreement; Landis and Koch 1977). Agreement was strongest for health ($\kappa = 0.861$) and sports ($\kappa = 0.847$), domains with unambiguous topical boundaries. Gold-standard labels for this subset were assigned by majority vote across the three annotators. To verify that rumour headlines reflect the circulating claim rather than the fact-checker’s own framing, we manually spot-checked a random sample of rumour instances for explicit meta-

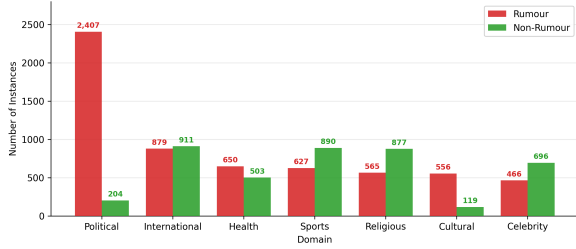


Figure 2: Domain distribution across rumour and non-rumour instances in the training set. Political is the dominant rumour domain (2,407 instances) yet the most underrepresented in non-rumour (204), revealing a structural asymmetry that motivates domain-stratified evaluation. Sports and religious show the inverse pattern.

linguistic markers (e.g., গুজব ‘rumour’, ভুয়া ‘fake’, সত্যতা যাচাই ‘fact-check verified’). We found such terms to be rare, supporting that the corpus predominantly captures the claim as circulated rather than the fact-checker’s verification framing.

Split	Total	Rumour	Non-R	Balanced
Train	8,198	3,998	4,200	✗
Validation	1,400	700	700	✓
Test	1,400	700	700	✓
Total	11,198	5,398	5,800	—

Table 1: MisInfo-11K dataset statistics by split. Non-R = Non-Rumour. The training split retains natural class imbalance (3,998 rumour vs. 4,200 non-rumour, 51.2% non-rumour); validation and test are balanced 50/50 to avoid confounding evaluation metrics with class prevalence. Training-set imbalance is addressed via class-weighted loss and a weighted random sampler (Section 5.2).

4 Linguistic Marker Framework

4.1 Marker Design and Validation

We design 14 linguistically motivated markers spanning four categories: *structural* (text length, word count, short title, long title, ends with question mark), *lexical* (claim verb, negation words, clickbait), *attribution* (has quote, unnamed source, named outlet), and *content* (has numbers, religious alarm, health claim). Markers were defined using Bangla-specific regular expressions validated against native speaker intuition and prior cross-lingual deception detection literature (Mahyoob et al., 2021; Zhou and Zafarani, 2020). Each marker was validated on the full combined dataset using chi-square tests for binary markers and Mann–Whitney U tests for continuous markers ($p < 0.0001$ for all 14). Table 2 reports discrimina-

tive weights $w_m = \chi_m^2 / \sum_{m'=1}^{14} \chi_{m'}^2$ and per-class activation rates. For the two continuous markers (word count and text length), the χ^2 column reports a variance-normalised effect size used solely for weight computation, while significance was measured using Mann–Whitney U tests.⁴

Figure 3 visualises binary marker activation rates and text length distribution by class.

Marker	χ^2	w_m	R	NR
Word count [†]	2881.07	0.311	9.83	7.30
Text length [†]	2494.79	0.269	65.80	49.55
Claim verb	1582.03	0.171	28.5%	1.6%
Long title	907.93	0.098	17.3%	0.9%
Has quote	287.90	0.031	5.4%	0.1%
Has numbers	254.17	0.027	10.2%	21.3%
Short title	237.01	0.026	1.2%	7.1%
Named outlet	151.66	0.016	2.8%	0.0%
Health claim	141.76	0.015	0.7%	4.2%
Negation words	87.59	0.009	16.0%	10.0%
Ends question	86.00	0.009	1.0%	3.8%
Unnamed source	64.43	0.007	1.8%	0.2%
Clickbait	50.87	0.006	1.4%	0.2%
Religious alarm	49.75	0.005	2.7%	0.9%

Table 2: Linguistic marker discriminative weights, ranked by χ^2 (chi-square statistic for binary markers; variance-normalised effect size for the two continuous markers[†] — see Section 4.1). $w_m = \chi_m^2 / \sum_{m'=1}^{14} \chi_{m'}^2$ is the normalised weight used in the WMS score (Eq. 1).

4.2 Domain Deception Fingerprints

We compute the mean activation rate of each marker separately for rumours in each of the seven domains, producing a 7×14 deception strategy profile matrix. Kruskal–Wallis tests confirm that 13 of 14 markers show statistically significant cross-domain variation ($p < 0.05$). Only negation_words fails to reach significance ($H = 10.34$, $p = 0.111$), rejecting the hypothesis that Bangla rumours employ a uniform linguistic strategy across topics.

Figure 4 shows three broad deception patterns by domain: *authority mimicry* (political/international, elevated claim_verb rates of 0.260/0.307) (Rashkin et al., 2017), *emotional provocation* (religious, highest religious_alarm rate at 0.196), and *fear amplification* (health, elevated health_claim/has_numbers co-occurrence) (Pérez-Rosas et al., 2018). Pairwise cosine similarities between domain profiles exceed 0.999 (range:

⁴Mann–Whitney U statistics: text length $U = 22,564,338$; word count $U = 23,117,884$; both $p < 0.0001$.

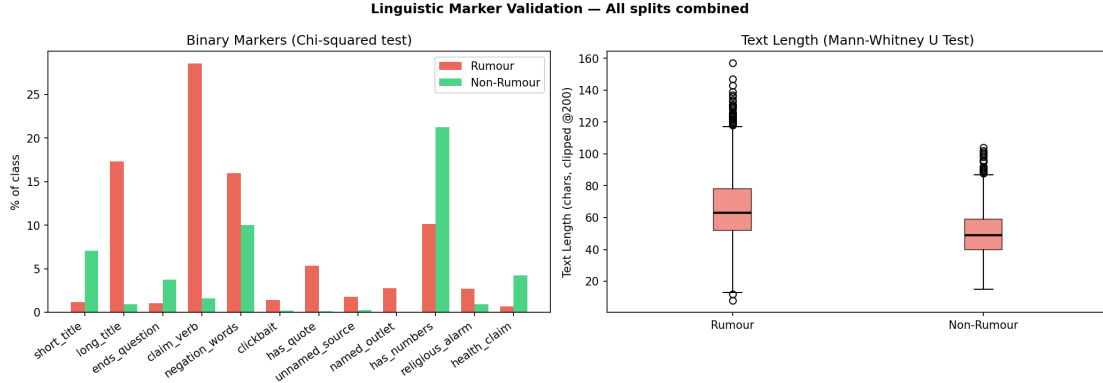


Figure 3: Left: binary marker activation rates by class (all χ^2 , $p < 0.0001$). Right: text length distribution by class (Mann–Whitney U, $p < 0.0001$). Rumours are systematically longer and more claim-verb dense, while non-rumours cite numbers and ask questions more frequently.

0.997–1.000),⁵ confirming that while domains differ in marker emphasis, they share a common underlying structure disrupted by domain-specific signals.

5 Experiments

5.1 Models

We benchmark eight models across two families. We restrict our benchmark to fine-tuned encoder models (BanglaBERT, MuRIL, XLM-RoBERTa) rather than large generative LLMs, since our focus is on establishing the surface-marker detection ceiling and quantifying the incremental value of contextual fine-tuning over hand-crafted features; zero-shot/few-shot LLM baselines are a natural extension we leave to future work.

Traditional Models. Five models combine character n-gram TF-IDF (2–5-grams) and word n-gram TF-IDF (1–3-grams), yielding 129,199 features:

1. Logistic Regression + TF-IDF
2. LinearSVC + TF-IDF
3. XGBoost + TF-IDF
4. Random Forest + TF-IDF + Markers
5. XGBoost + TF-IDF + Markers

Models 1–3 test surface pattern detection alone, while Models 4–5 test whether our 14 validated markers add discriminative signal beyond n-gram features.

Transformer Models. Three pretrained transformers are fine-tuned using a [CLS] $\rightarrow$

⁵Computed over the 72 = 21 pairwise domain profile vectors.

Dropout(0.3) $\rightarrow$ Linear(768 $\rightarrow$ 2) classification head:

- **BanglaBERT** (Bhattacharjee et al., 2022), pretrained on 27.5 GB of Bangla text.
- **MuRIL** (Khanuja et al., 2021), a multilingual model with dedicated South Asian language support pretrained on 17 Indian languages including Bangla.
- **XLM-RoBERTa** (Conneau et al., 2020), the standard 125M-parameter multilingual baseline.

5.2 Training Setup

All transformers were trained for 4 epochs using AdamW ($lr = 2 \times 10^{-5}$, weight decay = 0.01), linear warmup over 10% of steps, maximum sequence length of 128 tokens, and batch size 16.

Class-weighted cross-entropy loss and a weighted random sampler jointly address training-set imbalance (3,998 rumour vs. 4,200 non-rumour instances). Best checkpoint selection used validation Macro-F1, and all random seeds were fixed at 42.

Traditional models used validation-set grid search over $C \in \{0.01, 0.1, 1.0, 10.0\}$ for Logistic Regression and SVM. XGBoost and Random Forest used fixed robust defaults (200 estimators, maximum depth 6).

5.3 Results

Table 3 reports Macro-F1, ROC-AUC, and accuracy on the held-out test set for all eight models.

MuRIL achieves the highest test Macro-F1 of 0.9800 (validation: 0.9821), followed by BanglaBERT (0.9771; validation: 0.9807) and XLM-RoBERTa (0.9736; validation: 0.9750). All

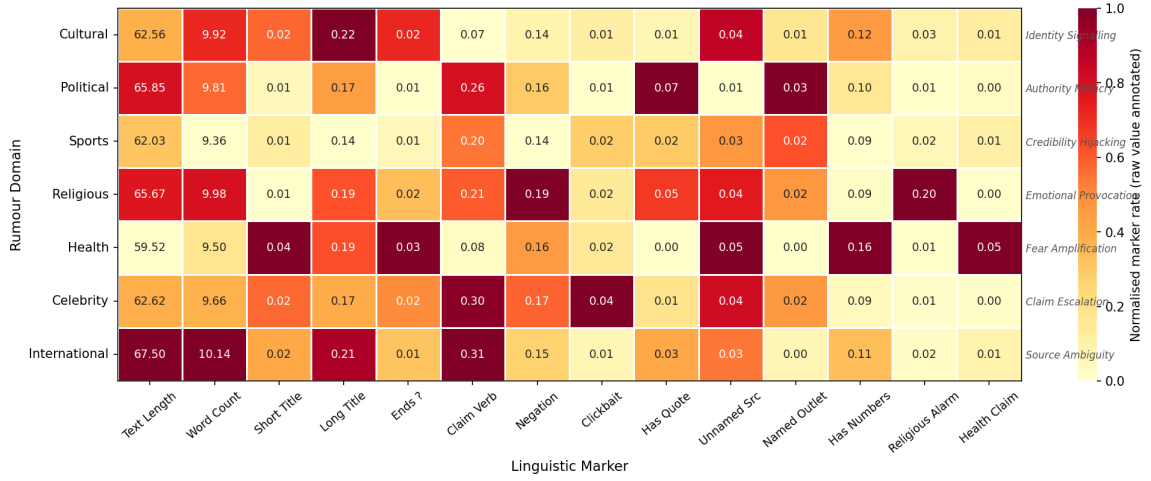


Figure 4: Domain deception strategy fingerprints. Colour intensity shows normalised marker activation rate, with raw values annotated. Strategy labels on the right axis reflect the dominant deception mechanism per domain.

validation-to-test Macro-F1 drops remain within 0.004 points (MuRIL: 0.0021; BanglaBERT: 0.0036; XLM-RoBERTa: 0.0014), indicating consistent generalisation.

Among traditional models, Random Forest augmented with our linguistic markers (0.9750) surpasses XLM-RoBERTa (0.9736), confirming hand-crafted markers carry genuine discriminative value beyond n-gram surface patterns. The +0.57 Macro-F1 gain of RF+Markers over the best TF-IDF-only model (LR and XGBoost, both 0.9693) quantifies the independent contribution of 14 validated markers.

Model	Macro-F1	ROC-AUC	Acc.
<i>Traditional Models</i>			
LR + TF-IDF	0.9693	0.9937	0.9693
SVM + TF-IDF	0.9643	0.9946	0.9643
XGBoost + TF-IDF	0.9693	0.9930	0.9693
RF + TF-IDF + M [†]	0.9750	0.9972	0.9750
XGB + TF-IDF + M [†]	0.9636	0.9934	0.9636
<i>Transformer Models</i>			
XLM-RoBERTa	0.9736	0.9937	0.9736
BanglaBERT	0.9771	0.9963	0.9771
MuRIL	0.9800	0.9971	0.9800

Table 3: Test-set results on the held-out test split ($n = 1,400$). Macro-F1 is macro-averaged F1 across the rumour and non-rumour classes; Acc. is overall accuracy. [†]M = augmented with our 14 linguistic markers. Best result per column in bold.

5.4 Cross-Dataset Generalisation

To assess transferability beyond Misinfo-11K, we evaluate our best-checkpoint BanglaBERT model

zero-shot on BanFakeNews (Hossain et al., 2020), a publicly available Bangla dataset comprising 7,000 authentic and 1,000 fake/satire news articles.

We sample 1,000 instances per class without any fine-tuning or domain adaptation, mapping BanFakeNews labels to our binary schema (authentic $\rightarrow$ non-rumour; fake/satire $\rightarrow$ rumour). We acknowledge this mapping is imperfect given our own argument that rumour and fake-news detection are categorically distinct tasks; we treat this transfer evaluation as a stress test of cross-task generalisation rather than a like-for-like benchmark. The model achieves Macro-F1 = 0.4728 and accuracy = 0.5325, with strong class asymmetry: recall on the fake class reaches 0.869, while authentic recall collapses to 0.196, producing 804 false positives against 131 false negatives.

This result is interpretable rather than anomalous, and arises from two compounding factors.

First, there is a severe input-length mismatch: MisInfo-11K consists of short claim headlines (mean 65.8 characters; truncated at 128 tokens), whereas BanFakeNews comprises full article bodies with mean length substantially exceeding this window. The model therefore operates on heavily truncated prefixes that fail to represent the article’s discriminative content.

Second, the two tasks differ categorically: rumour detection concerns unverified claims in active social-media circulation, while fake-news detection concerns fabricated article content with a different linguistic register, source structure, and propagation context (Zubiaga et al., 2018; Pérez-Rosas et al., 2018). These distinct feature distribu-

tions cannot be bridged through zero-shot transfer.

Rather than indicating a generalisation failure, this result provides direct empirical support for the core motivation of MisInfo-11K: Bangla rumour detection requires a dedicated headline-level corpus and task formulation that existing full-article fake-news resources cannot substitute.

6 Analysis

6.1 Marker Signal Void

We introduce the **Weighted Marker Score (WMS)** for an instance x :

$$\text{WMS}(x) = \sum_{m=1}^{14} w_m \cdot (f_m(x) - \mu_m^{\text{NR}}) \quad (1)$$

where $f_m(x)$ is the value of marker m , $w_m = \chi_m^2 / \sum_{m'=1}^{14} \chi_{m'}^2$ is its normalised discriminative weight (Table 2), and μ_m^{NR} is the mean of marker m over the non-rumour training split, computed without access to evaluation data to prevent leakage.

An instance lies in the **Marker Signal Void** when $\text{WMS}(x) \leq 0$: it carries no net positive marker signal relative to the non-rumour baseline, meaning surface-level linguistic cues are insufficient to distinguish it from authentic content.

While the ceiling formally bounds only the 14 markers validated here, these markers span all four linguistic categories established in cross-lingual deception-detection literature structural, lexical, attribution, and content-based cues (Mahyoob et al., 2021; Zhou and Zafarani, 2020) so the ceiling approximates, rather than exhaustively proves, the limit of surface-level detection more broadly. We do not claim exhaustiveness.

We find that 3.1% of test-set rumours (22 of 700) lie in the void. Figure 5 shows the WMS distribution by class: the two distributions overlap substantially near zero, confirming that a non-trivial subset of rumours is surface-linguistically indistinguishable from authentic content regardless of model capacity.

Examining the 22 void instances reveals three recurrent patterns:

1. *Meta-referential misinformation*: headlines from fact-checking organisations reporting their own annual detection statistics (e.g., “472 pieces of misinformation identified in March”), which carry the register of credible journalism.

2. *Government scheme impersonation*: rumours written in neutral administrative language with no claim verbs, vague attribution, or clickbait markers.
3. *Recurring death hoaxes*: bare declarative statements where the rumour keyword itself appears in both genuine fact-check headlines and fabricated claims, providing no discriminative signal.

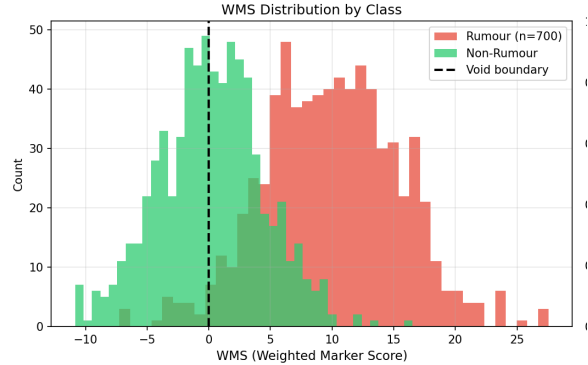


Figure 5: WMS distribution by class. The dashed line marks the void boundary ($\text{WMS} \leq 0$); 22 rumours (3.1%) fall below it. Overlap near zero confirms a subset of rumours is linguistically indistinguishable from authentic content at the surface level.

6.2 Detection Ceiling and Transformer Contribution

The 3.1% void rate establishes a hard theoretical ceiling on any system relying exclusively on the 14 surface markers. A perfect oracle achieves at most $\text{Macro-F1}_{\text{ceil}} = 0.9843$, corresponding to a theoretical recall ceiling of 0.9686 on the rumour class. All eight models operate below this ceiling. The ordering

TF-IDF only < TF-IDF + Markers < Transformer

holds consistently across the benchmark.

Explicit linguistic markers add +0.57 Macro-F1 points over distributional n-gram features (RF+Markers: 0.9750 vs. LR/XGB: 0.9693), while transformer pre-training adds a further +0.50 points over the best marker-augmented model (MuRIL: 0.9800 vs. RF+Markers: 0.9750).

Each component contributes independent, additive discriminative value.

MuRIL (FN = 10) recovers 12 of the 22 void rumours through contextual semantic representations; BanglaBERT (FN = 11) recovers 11; XLM-RoBERTa (FN = 14) recovers 8.

The 8–14 false negatives that persist across all three transformers constitute the irreducible difficulty of this dataset under text-only approaches (Zubiaga et al., 2018): rumours whose surface form is indistinguishable from authentic content even to contextual models.

6.3 Error Analysis

Table 4 summarises false positive and false negative counts for all three transformers on the test set.

MuRIL achieves the most favourable error profile (FN = 10, FP = 18), followed by BanglaBERT (FN = 11, FP = 21) and XLM-RoBERTa (FN = 14, FP = 23).

Model	Class	P	R	F1	FP	FN
MuRIL	Non-Rumour	0.9855	0.9743	0.9799	18	–
	Rumour	0.9746	0.9857	0.9801	–	10
BanglaBERT	Non-Rumour	0.9841	0.9700	0.9770	21	–
	Rumour	0.9704	0.9843	0.9773	–	11
XLM-RoBERTa	Non-Rumour	0.9797	0.9671	0.9734	23	–
	Rumour	0.9676	0.9800	0.9737	–	14

Table 4: Per-class results for transformer models ($n = 1,400$). P, R, and F1 denote precision, recall, and F1-score per class; FP and FN denote false positives and false negatives, reported once per model since each error is shared between the two classes (a false positive for Rumour is a false negative for Non-Rumour and vice versa).

False Negatives. Two categories dominate the errors shared by all three models: *meta-referential content* (Rumor Scanner/Jachai headlines reporting their own detection statistics, carrying the register of credible journalism and lying in the Marker Signal Void, $WMS \leq 0$), and *neutral-register impersonation* (government scheme rumours in bureaucratic language that activate no positive marker). Both expose a structural limitation: once a rumour adopts institutional or fact-checking register, neither marker-based nor distributional approaches can reliably flag it (Rashkin et al., 2017).

False Positives. The dominant pattern is *legitimate attributed speech* activating `claim_verb` ($w = 0.171$), where political/ international headlines quoting named parties are surface-identical to rumours quoting false claims; a second cluster is celebrity/ entertainment headlines whose dramatic vocabulary activates multiple lower-weight markers. This aligns with prior findings that epistemic markers misfire on legitimate attributed political

speech (Pérez-Rosas et al., 2018). Both error directions reveal consistent failure patterns across all three models.

7 Conclusion

We presented **MisInfo-11K**, the first domain-stratified Bangla rumour corpus (11,198 headlines, 7 domains), with an eight-model benchmark spanning traditional and transformer architectures. Our 14 validated linguistic markers (all $p < 0.0001$) reveal domain-specific deception fingerprints authority mimicry (political/international), emotional provocation (religious), and fear amplification (health) — with a +0.57 Macro-F1 gain from marker augmentation confirming their independent discriminative value. The Marker Signal Void analysis establishes Macro-F1 = 0.9843 as the ceiling for this marker set; MuRIL achieves 0.9800 (ROC-AUC = 0.9971), recovering 8–12 of the 22 void rumours through contextual representations. Zero-shot evaluation on BanFakeNews (Hossain et al., 2020) confirms rumour detection requires a dedicated headline-level resource. Future work includes propagation-graph integration (Bian et al., 2020), multimodal evidence fusion, and cross-lingual transfer to other South Asian languages.

Limitations

Domain and Temporal Scope. MisInfo-11K covers headlines from 2024–2026; models may not generalise to earlier periods or future data as misinformation strategies evolve.

Headline-Only Input. All models operate on headline text truncated to 128 tokens. The zero-shot collapse on BanFakeNews (Macro-F1 = 0.4728) confirms that article-level discriminative signals metadata, propagation context, and multimedia are absent from the current framework.

Formal Written Bangla. The corpus covers formal registers only. Informal social-media language dialectal variation, code-switching, WhatsApp forwards — is not represented, which may limit robustness on real-world misinformation content.

Taxonomy Reliability. Domain labels were assigned programmatically; only a 200-instance subset received human IAA validation ($\kappa = 0.746$). Domain labels are assigned via automatic keyword matching, with unmatched instances defaulting to political and rumour instances permitted multiple

domain labels while non-rumour instances receive exactly one; this asymmetry in label cardinality means the domain-level deception fingerprints in Section 4.2 should be read as indicative patterns rather than precise per-domain estimates.

Class Balance. We did not run a class-balanced downsampling ablation on the training split; given the modest imbalance (3,998 vs. 4,200) and our use of class-weighted loss / weighted sampling, we expect the effect to be small, but we flag this as an open empirical question.

Single-Language Scope. The marker framework is validated on Bangla only. Transfer to related South Asian languages (Hindi, Assamese, Sylheti) remains an open empirical question.

Data Availability. We will release MisInfo-11K including the train/validation/test splits, domain labels, and the 14 linguistic marker annotations — on GitHub upon publication to support reproducibility and further research on Bangla rumour detection.

References

- Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. [BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.
- Tian Bian, Xi Xiao, Tingyang Xu, Peilin Zhao, Wenbing Huang, Yu Rong, and Junzhou Huang. 2020. [Rumor detection on social media with bi-directional graph convolutional networks](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 549–556.
- Razieh Chalehchaleh, Reza Farahbakhsh, and Noel Crespi. 2024. [Multilingual fake news detection: A study on various models and training scenarios](#). In *Intelligent Systems Conference*, pages 73–89, Cham. Springer.
- Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. [Unsupervised cross-lingual representation learning at scale](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.
- Genevieve Gorrell, Elena Kochkina, Maria Liakata, Ahmet Aker, Arkaitz Zubiaga, Kalina Bontcheva, and Leon Derczynski. 2019. [SemEval-2019 task 7: RumourEval, determining rumour veracity and support for rumours](#). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 845–854, Minneapolis, Minnesota, USA. Association for Computational Linguistics.
- Md Zobaer Hossain, Md Ashraful Rahman, Md Saiful Islam, and Sudipta Kar. 2020. [BanFakeNews: A dataset for detecting fake news in Bangla](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2862–2871, Marseille, France. European Language Resources Association.
- Simran Khanuja, Diksha Kakwani, Satvik Golla, Gokul Bhatt, Amir Bhattacharyya, Bharat Paranjape, and Partha Talukdar. 2021. [MuRIL: Multilingual representations for Indian languages](#). *arXiv preprint arXiv:2103.10730*.
- J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.
- Tanjim Mahmud, Md Akash Rahman, Rimon Devnath, Mohammad Hossain, and Karl Andersson. 2026. [Detecting fake news in bangla using linguistic and psychological features](#). *ACM Transactions on Social Computing*, 9:1–46.
- Mohammad Mahyoob, Jeehaan Algaraady, and Musaad Alrahaili. 2021. [Linguistic-based detection of fake news in social media](#). *International Journal of English Linguistics*, 11(1):99.
- Matthew L. Newman, James W. Pennebaker, Diane S. Berry, and Jane M. Richards. 2003. [Lying words: Predicting deception from linguistic styles](#). *Personality and Social Psychology Bulletin*, 29(5):665–675.
- Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. [Automatic detection of fake news](#). In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3391–3401, Santa Fe, New Mexico, USA. Association for Computational Linguistics.
- Hannah Rashkin, Eunsol Choi, Jin Yea Jang, Svitlana Volkova, and Yejin Choi. 2017. [Truth of varying shades: Analyzing language in fake news and political fact-checking](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2931–2937, Copenhagen, Denmark. Association for Computational Linguistics.
- Sharifa Umma Shirina and Md Tabiur Rahman Prodhon. 2020. Spreading fake news in the virtual realm in Bangladesh: Assessment of impact. *Global Journal of HUMAN-SOCIAL SCIENCE: A Arts & Humanities-Psychology*, 20:17.
- Xinyi Zhou and Reza Zafarani. 2020. [A survey of fake news: Fundamental theories, detection methods, and opportunities](#). *ACM Computing Surveys*, 53(5):1–40.
- Arkaitz Zubiaga, Ahmet Aker, Kalina Bontcheva, Maria Liakata, and Rob Procter. 2018. [Detection and resolution of rumours in social media: A survey](#). *ACM Computing Surveys*, 51(2):32:1–32:36.

Arkaitz Zubiaga, Geraldine Wong Sak Hoi, Maria Liakata, and Rob Procter. 2016. PHEME dataset of rumours and non-rumours.

Does Live Browser Context Improve LLM-Assisted End-to-End Test Script Repair? An Empirical Study of Playwright MCP

Rifa Indra Setiawan¹, Divi Galih Prasetyo Putri¹,

¹Department of Electronic and Informatics Engineering, Universitas Gadjah Mada

Correspondence: divi.galih@ugm.ac.id

Abstract

Automated End-to-End test scripts are inherently fragile. Modifications to the User Interface (UI) cause automation test script failures, imposing significant maintenance overhead. Automated test script repair using Large Language Model (LLM) has shown promise, but existing approaches operate without real-time access to the application under test. To the best of our knowledge, no prior controlled study has compared LLM-based test-script repair with and without MCP-mediated live browser access while keeping the remaining repair settings consistent. Through a controlled experiment on a web application, we inject 30 failure scenarios across two categories of UI modifications (locator changes and DOM structural changes) and evaluate two LLM models (GPT 5.0 Mini and Claude Haiku 4.5) with two conditions (with and without MCP access). Configurations with MCP achieve a correct patch rate of 65%, compared with 25% without MCP. The plausible patch rate reached 98% (vs. 87%), the failure rate dropped from 13.3% to 1.7%, and MTTR decreased by 31 seconds. These results suggest that providing agents with MCP-mediated live browser access primarily enhances repair *correctness* rather than merely increasing re-execution success rates.

1 Introduction

Automated End-to-End (E2E) testing has become an important part of modern web development, particularly within agile and continuous integration/continuous delivery (CI/CD) workflows (Niu et al., 2025). By executing test cases automatically through frameworks such as Playwright and Selenium, teams can validate application behavior across full user flows with reduced manual effort (Melyawati et al., 2024). However, a persistent challenge undermines the long-term value of automation testing which is **test script fragility**. When front-end interfaces evolve, existing test

scripts break and require significant maintenance effort (Samad et al., 2021). Awad et al. (2024) report that testers can spend approximately 40% of their time updating and maintaining test suites as applications evolve. Prior empirical studies have consistently identified locator changes and DOM structural modifications as the most dominant causes of test script failure across multiple frameworks (Parry et al., 2022; Di Meglio et al., 2026). Even minor changes, such as renaming a CSS class or altering an element’s position in the DOM hierarchy, can cause complete script failure (Nass et al., 2023). These challenges are not transient. UI test automation will continue to face long-term difficulties in script maintenance, stability, and adaptation to interface changes (Nass et al., 2021).

To reduce this maintenance effort, automated test script repair mechanisms have been proposed to detect, diagnose, and repair broken test scripts with minimal manual intervention (Saarathy et al., 2024). Existing approaches rely on techniques such as heuristic-based locator matching, semantic information from previous test executions, and more recently, large language model (LLM)-based autonomous agents to recover failed test scripts (Grover and Sepuri, 2025; Qi et al., 2023; Bouzenia et al., 2025). Despite these advances, existing test repair approaches increasingly incorporate information from the current state of the application under test. Heuristic-based systems such as Healenium use historical locator information together with runtime information from the updated interface to identify replacement elements (Grover and Sepuri, 2025). Recent AI-assisted approaches may also use failure artifacts or information from the current user interface to guide repair decisions (Biswas, 2026). However, these approaches are typically evaluated as complete repair systems, making it difficult to determine the specific contribution of live and interactive browser context. Consequently,

it remains unclear whether giving an LLM-based repair agent direct access to the current browser state actually improves the semantic correctness of generated repairs, rather than merely increasing the likelihood that the modified test script passes execution. This distinction is important because a passing repair may weaken, remove, or circumvent the original verification logic, thereby masking genuine application regressions.

In this paper, we investigate whether MCP-mediated live browser access improves the quality of LLM-assisted test-script repair compared with artifact-based repair alone. MCP is a standardized protocol for exposing external tools and contextual information to AI applications (Sudakara, 2025). In this study, we use **Playwright** as the automation framework for executing and validating end-to-end web tests, together with the Playwright MCP Server. This implementation exposes Playwright’s browser interaction capabilities to the LLM-based repair agent, allowing it to navigate web pages, obtain structured accessibility snapshots, interact with interface elements, and observe the current browser state during the repair process, as illustrated in Figure 1 (Hou et al., 2026).

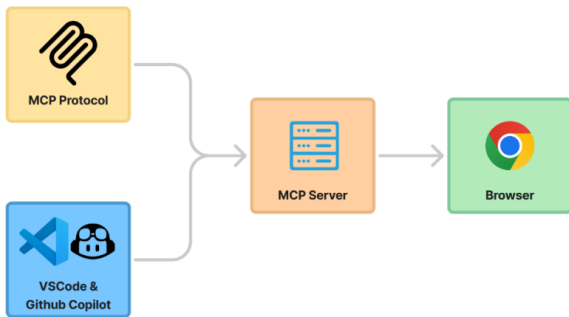


Figure 1: Implementation overview of Playwright MCP with an AI agent. The MCP server bridges the static LLM knowledge with browser context tools, allowing the agent to observe the live DOM state during self-healing.

We conduct a controlled experiment on a web application by injecting 30 failure scenarios across two categories, 15 scenarios related to locator changes and 15 scenarios related to DOM structural changes. Each scenario is tested under four configurations that combine two LLM models (GPT 5.0 Mini and Claude Haiku 4.5) with two conditions (with and without MCP access). Following the Automated Program Repair (APR) evaluation framework (Dikici and Bilgin, 2025), we measure three metrics: *plausible patch rate* (proportion of

repairs that restore test execution), *correct patch rate* (proportion of repairs that match manually constructed ground truth), and *Mean Time to Repair* (MTTR).

2 Related Work

Several approaches have been proposed to automatically repair broken test scripts, broadly categorizable into heuristic-based, semantic-based, and LLM-based methods. Grover and Sepuri (2025) compare traditional Selenium frameworks with the Healenium self-healing stack, which uses attribute-similarity heuristics to find replacement locators when the original locator fails. While Healenium significantly reduces failures caused by locator changes, it relies on syntactic matching without understanding the functional purpose of the target element. This creates a risk of *regression masking*, where a repaired script passes execution but the chosen element is functionally incorrect (Biswas, 2026). Similarly, Kim and Kouatly (2024) integrate Selenium with the ChatGPT API for self-healing, demonstrating improved maintenance efficiency, though the approach still depends on static context for repair decisions. Qi et al. (2023) propose Semter, which captures semantic information from test executions on an earlier application version and uses semantic similarity to locate target elements in updated versions. Semter achieves an average repair ratio of 84% across six web applications and can handle workflow changes caused by page additions or removals. However, its effectiveness depends on the quality of semantic information from initial executions and degrades when interface changes diverge substantially from the original version. Pham et al. (2022) review AI-augmented E2E testing tools and find that existing self-healing features remain limited to element-level locator matching, lacking the ability to handle complex functional or structural changes. More recently, LLM-based agents have shown promise for automated repair tasks. Bouzenia et al. (2025) introduce RepairAgent, an autonomous LLM-based agent for Automated Program Repair (APR) that successfully fixes 164 bugs on the Defects4J dataset through iterative reasoning. Although RepairAgent targets production code rather than test scripts, its APR framework provides a rigorous evaluation methodology applicable to code repair assessment. Most closely related to our work, Biswas (2026) implements an AI-assisted self-healing agent (Play-

wright Healer Agent) for E2E web testing, achieving a 73% reduction in false positives, 45% reduction in MTTR, and 91.3% test reliability. However, this approach performs inference solely from the LLM’s internal knowledge and static error logs, without the ability to access or verify the live page state, leaving it vulnerable to regression masking. A common limitation across all prior automated script repair approaches is the absence of real-time interaction with the application under test during the repair process. MCP addresses this gap by standardizing how AI systems integrate with external tools, data sources, and environments (Hou et al., 2026). MCP follows a client-server architecture where an MCP host (e.g., an IDE with an AI agent) communicates with MCP servers that expose domain-specific capabilities (Sudakara, 2025). LLM-based agents, which combine LLM reasoning with tool use, memory, and planning modules (Chowa et al., 2026), are particularly well-suited to leverage MCP’s capabilities. He et al. (2025) demonstrate that modern LLMs’ planning and reasoning abilities enable agents to handle complex tasks including code generation, testing, and script maintenance autonomously. However, LLMs are fundamentally reasoning engines rather than data repositories. LLM knowledge is static and they cannot directly observe live application state (Lee et al., 2024). MCP bridges this gap by providing grounded, real-time context that enables more informed repair decisions. Despite the conceptual promise of this integration, no prior work has empirically evaluated MCP’s contribution to automated script repair in automated testing. Our study addresses this gap through a controlled comparison of LLM-based repair with and without MCP-mediated live browser access. The experiment therefore evaluates the practical contribution of adding MCP-enabled live interaction to an otherwise comparable agent workflow.

3 Methodology

3.1 System Under Test

We conduct our experiments on a locally deployed web-based information system for managing teaching assistant recruitment. The application is developed in PHP using the Laravel framework. We focus on the student role, which involves a sequence of functions, including completing a profile, uploading academic transcripts, browsing available positions, registering for a vacancy, and managing

offers. Such interactions are representative of the user-oriented workflows typically evaluated in end-to-end web testing, where tests navigate through the application, perform user actions, and verify observable outcomes. Full access to the source code and local deployment environment enables us to introduce controlled interface changes for fault injection while preserving the application’s underlying business functionality.

3.2 Test Design

We design 15 End-to-End (E2E) test cases (TC01–TC15) using Playwright with TypeScript, organized across five functional areas: profile management (F1=4 TCs), transcript upload (F2=3 TCs), vacancy registration (F3=3 TCs), registration verification (F4=3 TCs), and offer management (F5=2 TCs). All test scripts follow the Page Object Model (POM) pattern, separating locator definitions, page interaction logic, and test specifications into distinct layers. Baseline scripts are validated through three consecutive executions with zero failures before experimentation.

3.3 Fault Injection

We inject 30 failure scenarios across two categories: Locator changes and DOM structural changes. The scenarios were designed to represent changes that commonly occur during web application evolution (Nass et al., 2023; Di Meglio et al., 2026). Examples of failure scenarios are shown in Table 1.

Locator changes (LC01–LC15). These scenarios modify element attributes that test scripts use for identification, such as changing data-testid formats (e.g., kebab-case to camelCase), renaming CSS classes, altering button text that affects accessible names, or modifying XPath-dependent text content. Each injection targets a different locator strategy to evaluate self-healing across varying fragility levels. The purpose was to evaluate whether the repair agent merely updates a broken locator value or identifies a more stable locator strategy when one is available.

DOM structural changes (DC01–DC15). These scenarios modify the HTML hierarchy or introduce new interface elements, such as wrapping elements in additional containers, adding confirmation steps before form submission, introducing input validation, or disabling buttons based on application state. DOM changes are generally more complex as they

ID	Type	Changes	Impact
LC03	LC	testId format: kebab-case → camel- Case	Selector miss
LC08	LC	CSS class renamed	Class mismatch
LC10	LC	Button text changed	Name mismatch
DC01	DC	Element wrapped in new <div>	Hierarchy break
DC10	DC	Confirmation step added	Missing step
DC13	DC	Button disabled on state	Logic change

Table 1: Representative examples from each category. LC represents locator changes and DC represents DOM changes. All injections are designed to reflect realistic front-end changes encountered during application maintenance, not artificial scenarios targeting specific repair strategies.

may require not only locator updates but also modifications to interaction logic.

3.4 Ground Truth Construction

For each of the 30 injection scenarios, we manually write the correct repair following Playwright’s locator stability hierarchy: testId (most stable) > role with accessible name > label > CSS/XPath (most fragile). Repairs follow two patterns: (1) *value update* which retains the baseline locator strategy and adjusting only the changed value when the strategy is already stable, or (2) *strategy upgrade* by replacing a fragile locator strategy with a more stable one. This produces 30 ground truth repairs, each documented as a unified diff file in a version-controlled repository.¹

3.5 Experimental Configurations

We evaluate four configurations that cross two independent variables:

MCP condition– With MCP: the AI agent has access to Playwright MCP Server, enabling real-time browser navigation, accessibility snapshots,

¹Repository: <https://github.com/self-healing-automation/self-healing-automation-MCP>

and live DOM interaction. **Without MCP:** the AI agent operates solely from test scripts, failure screenshots, error logs, and its internal knowledge. This represents the current state of LLM-based self-healing approaches without adding any external capabilities.

LLM model– GPT 5.0 Mini: a thinking-architecture model from the GPT-5 family with a 400K token context window, optimized for deep reasoning at reduced computational cost. **Claude Haiku 4.5:** a speed-optimized model with 4–5× lower latency than Claude Sonnet 4.5, designed for high throughput with competitive accuracy.

Config	MCP	Model
GPT + MCP	✓	GPT 5 Mini
GPT – MCP	×	GPT 5 Mini
Claude + MCP	✓	Haiku 4.5
Claude – MCP	×	Haiku 4.5

Table 2: Experimental configurations. Each configuration is tested on all 30 failure test scenarios (15 LC + 15 DC).

Both models are accessed through GitHub Copilot Chat in Visual Studio Code operating in *agent mode*, which enables autonomous file editing, command execution, and tool invocation. Table 2 summarizes the experimental design.

3.6 Prompting Strategy

To promote consistency across experimental sessions, we designed two standardized prompt variants corresponding to the MCP and non-MCP conditions. Both variants assigned the agent the same role as a QA automation engineer and specified an identical repair objective, technical context, Run–Analyze–Investigate–Fix–Verify workflow, and hard constraints. These constraints prohibited the agent from removing or weakening assertions, skipping test steps, introducing arbitrary waits, using forced interactions, or modifying authentication fixtures. No scenario-specific repair hints were included, and the same prompt variant was reused across all sessions within each condition.

Both conditions allowed the agent to execute and verify tests through Playwright and inspect the resulting error output, stack traces, and failure screenshots. They differed only in the investigation capabilities available to the agent. In the non-

MCP condition, browser information was mediated by the prescribed test execution and its generated artifacts. In the MCP condition, the agent additionally had direct access to the running browser through Playwright MCP, enabling agent-directed navigation, accessibility snapshots, and interface interaction during diagnosis.

Before each session, the test scripts were restored to their baseline state using version control, and a new agent session was initialized. The fault-injected test was first executed to confirm the expected failure. The agent then performed the repair workflow and was allowed up to three verification cycles. A session was classified as failed when the test remained unsuccessful after the third cycle. The complete prompt variants are provided in the appendix.

3.7 Evaluation Metrics

Following the APR evaluation framework (Dikici and Bilgin, 2025; Bouzenia et al., 2025), we evaluate repairs using three metrics:

Plausible patch rate. The proportion of scenarios where the repaired script executes successfully:

$$\text{Plausible Rate} = \frac{|\text{Passing Scenarios}|}{|\text{Total Scenarios}|} \times 100\% \quad (1)$$

Correct patch rate. The proportion of repairs that match the ground truth, either identically or through functional equivalence as determined by manual evaluation (Le et al., 2019):

$$\text{Correct Rate} = \frac{|\text{Ground Truth Matches}|}{|\text{Total Scenarios}|} \times 100\% \quad (2)$$

Mean Time to Repair (MTTR). The average time from the start of the agent’s analysis to successful verification:

$$\text{MTTR} = \frac{\sum \text{Repair Time}}{|\text{Repaired Scenarios}|} \quad (3)$$

Each repair is classified into one of four categories: **Correct (C)** indicates that the repaired test passes and matches ground truth; **Plausible Incorrect (PI)** indicates that the repaired test passed but differs from ground truth; **Regression Masking (RM)** indicates that the test passes because assertions or test logic have been removed, weakened, or bypassed; and **Fail (F)** indicates that the test does not pass after repair. Figure 2 illustrates these four

outcomes in the context of the sequential End-to-End execution flow. Each scenario is executed once per configuration due to token quota constraints on the AI agent platform, yielding 120 total sessions. We acknowledge this limitation and discuss its implications in Section 5.

4 Results and Discussion

Prior to evaluating repair performance, all 15 baseline test cases were executed in three consecutive validation runs and passed without any observed flaky behavior. Each of the 30 fault-injection scenarios subsequently triggered the expected test failure, confirming both the stability of the baseline test suite and the effectiveness of the injected changes.

Table 3 presents the aggregate results for each configuration across all 30 scenarios. Configurations with MCP consistently outperform their non-MCP counterparts on both quality metrics.

Config	Plaus.	Correct	Fail	MTTR
GPT – MCP	83%	30%	17%	3:50
Claude – MCP	90%	20%	10%	1:34
GPT + MCP	97%	70%	3%	2:39
Claude + MCP	100%	60%	0%	1:39

Table 3: Per-configuration results across 30 scenarios. Plaus. = plausible patch rate; Correct = correct patch rate; Fail = failure rate; MTTR in min:sec.

The largest observed difference between the condition "With MCP" and "Without-MCP" is in the correct patch rate. GPT improves the correct patch rate from 30% to 70% (+40pp) and Claude from 20% to 60% (+40pp). Plausible rate improvements are comparatively modest (GPT: +14pp; Claude: +10pp), indicating that the largest observed benefit of the MCP-enabled condition lies in repair correctness rather than merely in making scripts pass.

Claude + MCP achieves the highest plausible rate (100%) but a lower correct rate (60%) than GPT + MCP (70%). In the observed repair traces, Claude Haiku 4.5 more frequently applied an initially viable fix, whereas GPT-5 mini more frequently inspected alternative locator strategies before applying a patch. These observations may help explain the measured speed–correctness trade-off, although the limited number of runs prevents generalizing these behavioral differences to the models as a whole. Figure 3 illustrates this trade-off between repair speed and correctness across all configurations.

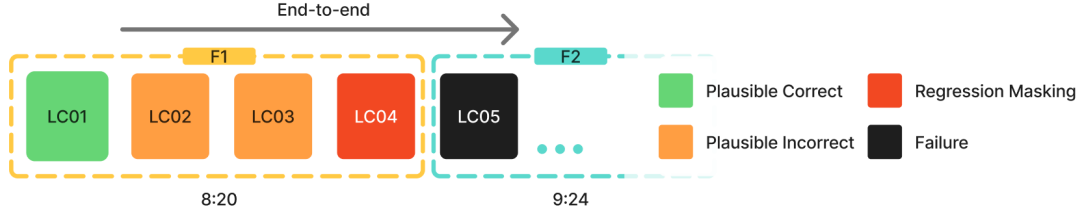


Figure 2: Visualization of the End-to-End self-healing execution flow.

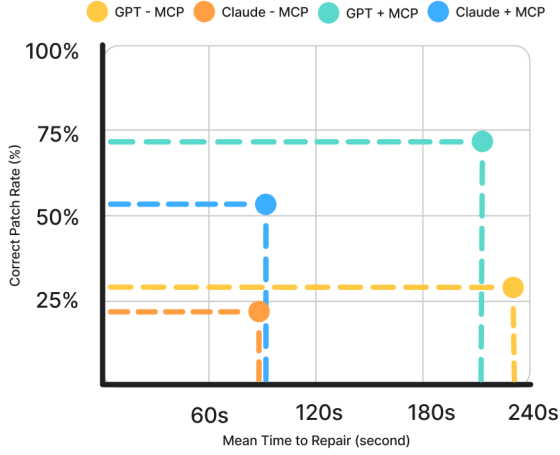


Figure 3: Efficiency diagram plotting MTTR against Correct Patch Rate.

Table 4 presents MTTR disaggregated by injection type and MCP condition.

Condition	Type	Total Time	MTTR
Without MCP	LC	61:51	2:03
With MCP	LC	54:25	1:48
Without MCP	DC	100:49	3:21
With MCP	DC	75:10	2:30

Table 4: MTTR by injection type and MCP condition (min:sec). LC = locator change; DC = DOM change.

DOM changes require substantially longer repair times than locator changes across both conditions, reflecting their greater complexity. Total repair time was 7 minutes 26 seconds lower for locator changes and 25 minutes 39 seconds lower for DOM changes in the MCP-enabled condition. Notably, Claude without MCP records the lowest individual MTTR (1:34 overall), as it skips browser interaction and selects fixes more aggressively. However, this speed comes at the cost of the lowest correct rate (20%) and highest regression masking rate, illustrating that repair efficiency must be evaluated jointly with repair quality.

With MCP. Agents navigate the SUT autonomously, take accessibility snapshots of relevant pages, and use live DOM data to inform locator selection. When navigating to an incorrect page, agents self-correct by recognizing the mismatch between expected and observed page content. For example, in scenario LC08, GPT + MCP first identifies that the CSS class has changed, then discovers a more stable data-testid attribute on the same element, and selects the testId strategy—matching the ground truth. This two-stage verification process (identify change → evaluate alternatives) is characteristic of the thinking-architecture model as we can see from Figure 4.

Without MCP. Agents rely entirely on static error logs, failure screenshots, and their internal knowledge. Without the ability to verify assumptions against the live DOM, agents tend to make more aggressive modifications, sometimes altering valid test logic or removing assertions to make scripts pass. In scenario LC03, Claude without MCP removes the file name verification step entirely after failing to locate the element—producing regression masking despite explicit prompt instructions prohibiting assertion removal. This pattern explains the higher regression masking rate in non-MCP configurations (7 instances vs. 1 with MCP).

Model differences. The two models exhibited different observed trade-offs between repair time and correctness. Claude Haiku 4.5 generally completed repairs faster, whereas GPT-5 mini achieved a higher correct patch rate in both conditions. In the inspected repair traces, GPT-5 mini more frequently examined alternative locator strategies before applying a patch, while Claude Haiku 4.5 more often applied an initial viable repair directly. Both models showed the same 40-percentage-point increase in correct patch rate when MCP was added, reaching 70% for GPT-5 mini and 60% for Claude Haiku 4.5.

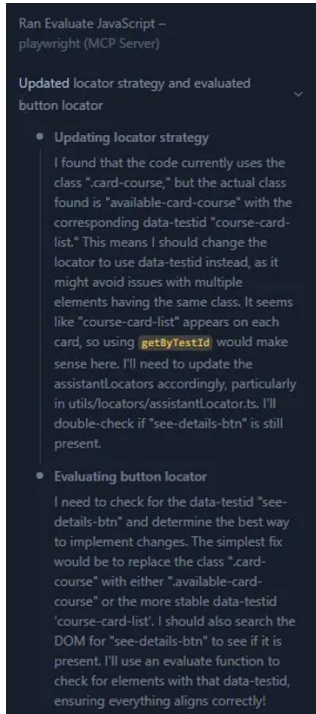


Figure 4: Qualitative example of GPT + MCP on LC08. The agent uses live DOM context to identify a stable data-testid alternative instead of only updating the broken CSS class.

5 Conclusion

We present an empirical evaluation of MCP-mediated live browser contribution to AI-agent-based self-healing in automated End-to-End testing. Through a controlled experiment with 120 sessions across 30 fault injection scenarios and four configurations, we demonstrate that providing AI agents with real-time browser context through MCP substantially contributes to improving repair quality.

Our key findings are twofold. First, the largest difference between conditions occurs in repair correctness: the correct patch rate increases from 25% to 65%, whereas the plausible patch rate increases by 11 percentage points. Second, the MCP-enabled condition also shows a lower failure rate (2% vs. 13%) and an average MTTR that is 31 seconds shorter, with the largest time savings on complex DOM structural changes (25 minutes 39 seconds total).

These results suggest that the integration of structured environmental context through protocols like MCP represents a promising direction for improving the reliability of AI-driven software maintenance tools. Future work should extend this evaluation to multiple applications with diverse architectures, incorporate additional failure types (e.g.,

business logic changes, flaky failures), add experimental repetitions with statistical significance testing, and explore the interaction between MCP and higher-capability LLM models.

Limitations

Our study has several limitations that should be considered when interpreting the results. The observed advantage of MCP is limited to a single locally deployed web application and two interface-related failure types: locator and DOM-structure changes. Its effect may differ for business-logic changes, API modifications, or non-deterministic failures. Moreover, each scenario was executed once per configuration due to GitHub Copilot quota and session limits; therefore, the results reflect single-run outcomes and do not capture run-to-run LLM variability or support statistical significance testing. An important limitation concerns the interpretation of the MCP comparison. The MCP-enabled condition provides the agent with direct access to live application information through browser navigation, accessibility snapshots, and interface interaction, whereas the non-MCP condition relies on information available through test execution artifacts. Therefore, the observed differences cannot be attributed specifically to the structure of the MCP protocol itself. An additional control condition providing equivalent browser information without MCP would be required to disentangle these factors.

Our work is conducted using Playwright MCP Server (v0.0.7) and GitHub Copilot Chat (v0.37.9) that are actively developed tools. The behavior may change in future versions. Our results should be interpreted in the context of the specific versions used during experimentation.

References

- Abdallah Awad, Mahmoud H. Qutqut, Ali Ahmed, Fatima Al-Haj, and Fadi Almasalha. 2024. [Artificial Intelligence Role in Software Automation Testing](#). In *2024 International Conference on Decision Aid Sciences and Applications (DASA)*, pages 1–6, Manama, Bahrain. IEEE.
- Saumen Biswas. 2026. Enhancing End-to-End Test Stability Through AI-Assisted Self-Healing: A Case Study of Playwright Healer Agent Implementation. *International Journal of Engineering & Scientific Research*, 14(01):1–12.
- Islem Bouzenia, Premkumar Devanbu, and Michael Pradel. 2025. [RepairAgent: An Autonomous,](#)

- LLM-Based Agent for Program Repair. In *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*, pages 2188–2200, Ottawa, ON, Canada. IEEE.
- Sadia Sultana Chowdhury, Riasad Alvi, Subhey Sadi Rahman, Md Abdur Rahman, Mohaimenul Azam Khan Raiaan, Md Rafiqul Islam, Mukhtar Hussain, and Sami Azam. 2026. *From language to action: A review of large language models as autonomous agents and tool users*. *Artificial Intelligence Review*, 59(2):71.
- Sergio Di Meglio, Luigi Libero Lucio Starace, Valeria Pontillo, Ruben Opdebeeck, Coen De Roover, and Sergio Di Martino. 2026. *Investigating the adoption and maintenance of web GUI testing: Insights from GitHub repositories*. *Information and Software Technology*, 189:107928.
- Sena Dikici and Turgay Tugay Bilgin. 2025. *Advancements in automated program repair: A comprehensive review*. *Knowledge and Information Systems*, 67(6):4737–4783.
- Savi Grover and Shruthi Sepuri. 2025. SELF-HEALING WEB AUTOMATION: AN EMPIRICAL COMPARISON OF TRADITIONAL SELENIUM FRAMEWORKS AND HEALENIUM STACK. *International Journal of Applied Mathematics*, 38(12).
- Junda He, Christoph Treude, and David Lo. 2025. *LLM-Based Multi-Agent Systems for Software Engineering: Literature Review, Vision, and the Road Ahead*. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–30.
- Xinyi Hou, Yanjie Zhao, Shenao Wang, and Haoyu Wang. 2026. *Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions*. *ACM Transactions on Software Engineering and Methodology*, page 3796519.
- Yunjin Kim and Rand Kouatly. 2024. *Self-healing automation testing with Selenium and ChatGPT API integration to improve the efficiency of the maintenance*. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, pages 339–344, Dubai, United Arab Emirates. IEEE.
- Xuan-Bach D. Le, Lingfeng Bao, David Lo, Xin Xia, Shanping Li, and Corina Pasareanu. 2019. *On reliability of patch correctness assessment*. In *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, pages 524–535.
- Hyunji Lee, Doyoung Kim, Kyoung-Woon On, and Minjoon Seo. 2024. *How Well Do Large Language Models Truly Ground?* *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1:2437–2465.
- Ni Luh Putu Melyawati, I Made Dwi Putra Asana, Ni Nyoman Suardiati Putri, Ketut Jaya Atmaja, and I Gede Iwan Sudipa. 2024. *Comparison of Automation Testing On Card Printer Project Using Playwright And Selenium Tools*. *Journal of Computer Networks, Architecture and High Performance Computing*, 6(3):1309–1320.
- Michel Nass, Emil Alégroth, and Robert Feldt. 2021. *Why many challenges with GUI test automation (will) remain*. *Information and Software Technology*, 138:106625.
- Michel Nass, Emil Alégroth, Robert Feldt, Maurizio Leotta, and Filippo Ricca. 2023. *Similarity-based Web Element Localization for Robust Test Automation*. *ACM Trans. Softw. Eng. Methodol.*, 32(3):75:1–75:30.
- Yingbei Niu, Soo See Chai, Kok Luong Goh, Kim On Chin, and Emily Sing Kiang Siew. 2025. *Continuous Integration and Continuous Delivery Based on ARM Cortex-M4 Embedded Software Automation Testing*. *Journal of Advanced Research in Computing and Applications*, 39(1):146–163.
- Owain Parry, Gregory M. Kapfhammer, Michael Hilton, and Phil McMinn. 2022. *A Survey of Flaky Tests*. *ACM Transactions on Software Engineering and Methodology*, 31(1):1–74.
- Phuoc Pham, Vu Nguyen, and Tien Nguyen. 2022. *A Review of AI-augmented End-to-End Test Automation Tools*. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–4, Rochester MI USA. ACM.
- Xiaofang Qi, Xiang Qian, and Yanhui Li. 2023. *Semantic Test Repair for Web Applications*. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1190–1202, San Francisco CA USA. ACM.
- Sutharsan Saarathy, Suresh Bathrachalam, and Bharath Rajendran. 2024. *Self-Healing Test Automation Framework using AI and ML*. *International Journal of Strategic Management*, 3(3):45–77.
- Abdus Samad, Tabrez Nafis, Shibli Rahmani, and Shahab Saquib Sohail. 2021. *A Cognitive Approach in Software Automation Testing*. *SSRN Electronic Journal*.
- Baradwaj Bandi Sudakara. 2025. *Leveraging MCP servers for context-aware playwright automation in cloud environments*. *Journal of Emerging Engineering Technologies*, 1(1):13–18.

A Appendix

```

1 ---
2 tools: ['playwright']
3 mode: 'agent'
4 ---
5
6 # Role
7 You are a QA automation engineer fixing one failing Playwright test.
8
9 # Context
10 Playwright Test (TypeScript), Page Object Model:
11 - `tests/` - spec files
12 - `pages/` - page classes
13 - `utils/` - locators
14 - `fixtures/` - setup (do not modify)
15
16 SUT: AssistLab at http://localhost:3000, pre-authenticated via fixtures.
17
18 # Investigation Approach
19 You have access to Playwright MCP browser tools. Use them to inspect
20 the running application directly – navigate to the relevant page,
21 capture accessibility snapshots, and read the actual DOM state before
22 proposing any fix. Runtime observation of the live application is
23 the primary source of evidence for this diagnosis. Also run the test
24 from the terminal as a QA engineer would. If a tool call fails with
25 a snapshot staleness error, capture a new snapshot and retry the
26 interaction.
27
28 # Task
29 Diagnose and fix one failing test in `{SPEC_PATH}`. Work in phases,
30 in order. Maximum 3 fix-verify cycles.
31
32 1. Run – execute `npx playwright test {SPEC_PATH}` and record the
33 exact error and failing step.
34 2. Analyze – read the spec and its dependencies; form an initial
35 hypothesis from the error.
36 3. Investigate – navigate the browser to the relevant page and
37 capture a snapshot. Compare the actual DOM state (elements,
38 attributes, data-testid) against what the locators in the code
39 expect. Report what you observed.
40 4. Fix – apply the smallest change that restores correctness,
41 grounded in the snapshot evidence.
42 5. Verify – re-run the failing test. If it still fails, return to
43 Investigate with new information. If the same approach fails
44 twice in a row, abandon that hypothesis and explore a different
45 angle – re-read the spec, check adjacent locators in the same
46 file, or inspect unrelated page sections that might have changed.
47
48 # Hard Rules
49 Violating any of these is unacceptable even if the test passes:
50 - No weakening or removing assertions
51 - No waitForTimeout or arbitrary sleeps
52 - No { force: true }
53 - No .skip, .only, or commenting out steps
54 - No changes to fixtures/ or auth setup

```

Figure 5: Prompt for test script repair using LLM and MCP

```

1 ---
2 mode: 'agent'
3 ---
4
5 # Role
6 You are a QA automation engineer fixing one failing Playwright test.
7
8 # Context
9 Playwright Test (TypeScript), Page Object Model:
10 - `tests/` - spec files
11 - `pages/` - page classes
12 - `utils/` - locators
13 - `fixtures/` - setup (do not modify)
14
15 SUT: AssistLab at http://localhost:3000, pre-authenticated via fixtures.
16
17 # Investigation Approach
18 You do not have browser automation tools in this session. Diagnose
19 the failure through static code analysis – trace through the spec,
20 page classes, and locator definitions, and reason carefully from
21 the error output. Read the files thoroughly: the specific locator
22 referenced in the error, the page class method that uses it, the
23 spec step that calls the method. Your diagnosis must be grounded
24 in concrete evidence from the code and terminal output. Also run
25 the test from the terminal as a QA engineer would, and read the
26 stack trace in full.
27
28 # Task
29 Diagnose and fix one failing test in `{SPEC_PATH}`. Work in phases,
30 in order. Maximum 3 fix-verify cycles.
31
32 1. Run – execute `npx playwright test {SPEC_PATH}` and record the
33 exact error and failing step.
34 2. Analyze – read the spec and its dependencies; form an initial
35 hypothesis from the error.
36 3. Investigate – navigate the browser to the relevant page and
37 capture a snapshot. Compare the actual DOM state (elements,
38 attributes, data-testid) against what the locators in the code
39 expect. Report what you observed.
40 4. Fix – apply the smallest change that restores correctness,
41 grounded in the snapshot evidence.
42 5. Verify – re-run the failing test. If it still fails, return to
43 Investigate with new information. If the same approach fails
44 twice in a row, abandon that hypothesis and explore a different
45 angle – re-read the spec, check adjacent locators in the same
46 file, or inspect unrelated page sections that might have changed.
47
48 # Hard Rules
49 Violating any of these is unacceptable even if the test passes:
50 - No weakening or removing assertions
51 - No waitForTimeout or arbitrary sleeps
52 - No { force: true }
53 - No .skip, .only, or commenting out steps
54 - No changes to fixtures/ or auth setup

```

Figure 6: Prompt for test script repair using LLM only (without MCP)

Assessing Large Language Models as Automated Writing Evaluation and Feedback Tools: A Human–AI Agreement Study on L2 English Writing

Kowshik Saha¹, Moyen Uddin², Md Tanvirul Islam Turad³, and Mizanur Rahman⁴

^{1,2,3}Department of Computer Science and Engineering

⁴Department of Humanities

Bangladesh University of Engineering and Technology

¹kowshiksahakabya@gmail.com, ²2005075@cse.buet.ac.bd, ³2005011@cse.buet.ac.bd,

⁴mizanurrahman@hum.buet.ac.bd

Abstract

Large language models (LLMs) are often suggested as tools for grading student writing and giving feedback, but it is unclear how far we can trust them on the separate skills that make up good writing. We compare two LLM graders (Gemini and Llama) with eight professional English teachers on 20 paragraphs written in second-language English by Bengali-speaking undergraduates in Bangladesh, each graded by two teachers, scoring content, organization, vocabulary, and grammar separately as well as the total. We then run a second stage in which teachers read each model’s written explanation, judge whether they agree with it, and rate the feedback. The LLMs match the teachers on the total score, content, and organization, but are unreliable on vocabulary and grammar, where the two models do not even agree with each other. We also show that one low agreement score is misleading: it comes from graders giving very similar marks (score clustering), not from real disagreement. In the second stage, teachers found the AI feedback acceptable and useful most of the time ($\geq 77.5\%$), with very few made-up errors (2.5%), and preferred Gemini’s stricter grammar score to their own 20% of the time. LLMs can help grade writing, but a teacher should audit their output with extra care on vocabulary and grammar.

1 Introduction

Teachers spend much of their time grading student writing and writing feedback on it. Large language models (LLMs) can now do both, which makes them attractive as low-cost tools (Mizumoto and Eguchi, 2023; Stahl et al., 2024). But a single overall score hides a question: a model may judge overall quality well and still be unreliable on the specific skills, such as grammar or word choice, that useful feedback has to address.

We therefore ask three questions. **(RQ1)** On which writing skills do LLM graders agree with

expert teachers, and on which do they disagree? **(RQ2)** When teachers see a model’s score together with its written explanation, are the disagreements acceptable, or are they real mistakes? **(RQ3)** Is the feedback the models produce useful, appropriately worded, and free of invented errors?

Each paragraph was graded independently by two teachers, which lets us measure how much teachers naturally differ. We test Gemini 3 Flash Preview (Google DeepMind, 2025) and Llama 3.3 70B Instruct (Meta AI, 2024), the latter through Groq (Groq, 2024). Alongside the gold-standard Quadratic Weighted Kappa (Cohen, 1968) we use Gwet’s AC2, which stays reliable under score clustering (Gwet, 2008). Our contributions are: (i) a skill-by-skill picture of when LLM grading matches teachers (total score, content, organization) and when it does not (vocabulary, grammar); (ii) evidence that a chance-corrected measure is needed, so that low agreement caused only by score clustering is not mistaken for real disagreement; and (iii) an audit showing that most score gaps are acceptable and that the feedback is useful, appropriately worded, and free of invented errors.

2 Related Work

Early automated essay scoring (AES) systems predicted scores from hand-picked text features (Attali and Burstein, 2006); later ones used neural networks (Uto, 2021); see Ke and Ng (2019) and Ramesh and Sanampudi (2022) for surveys. Recent studies ask an LLM to grade essays with no training examples or only a few (Mizumoto and Eguchi, 2023; Mansour et al., 2024), to judge how well a text holds together (Naismith et al., 2023), to place writers on a proficiency scale (Yancey et al., 2023), and to score and give feedback at once (Stahl et al., 2024), following the wider idea of an LLM as a judge (Zheng et al., 2023). Most report one overall agreement number. We instead report agreement

for each writing skill, pair QWK with Gwet’s AC2, which is less sensitive to score clustering (Gwet, 2008; Feinstein and Cicchetti, 1990), and ask teachers to check the models’ explanations and feedback, since feedback is what helps students improve (Hattie and Timperley, 2007).

3 Methodology

3.1 Corpus and Rubric

We use 20 student paragraphs written on the prompt “Fake News on Social Media.” The writers are first-year undergraduates at a public university in Bangladesh. Their first language is Bengali, so this is second-language English writing. We have no independent measure of their English proficiency. Each paragraph is graded on a rubric with four separate skills: Content & Relevance (max 3.0), Organization & Cohesion (max 2.0), Lexis & Academic Register (max 2.5; *vocabulary* for short), and Grammar & Mechanics (max 2.5), adding up to a total of 10.0 in steps of 0.25 (Appendix B). Both LLM graders get the same prompt (Appendix A) and no example gradings: Gemini 3 Flash Preview (gemini-3-flash-preview) through the Gemini API (Google DeepMind, 2025), and Llama 3.3 70B Instruct (Meta AI, 2024) through Groq’s llama-3.3-70b-versatile endpoint at temperature 0.3 (Groq, 2024). We also fix the order of the answer (Appendix C): a score and a written reason for each skill, then the total, then the feedback. This makes the model look at each skill carefully before it commits to a single overall score.

3.2 How Paragraphs Were Assigned to Teachers

We split the 20 paragraphs among 8 professional English teachers at two Bangladeshi public universities (4 men, 4 women) using a distributed cyclic incomplete block design, so that no teacher’s workload is heavy or uneven. One of the eight is an author of this paper and wrote the rubric; all eight were equally informed about the study, graded five paragraphs, and were blind to the AI’s output in Phase 1. The LLM graders ran first, so we used Gemini’s total to sort the corpus into low (below 5.0), average (5.0 to 6.0), and high (above 6.0) quality, giving 4, 12, and 4 paragraphs. This binning only balances workload across blocks and plays no part in the agreement analysis. The bins went into eight overlapping blocks, so that every paragraph is graded by *exactly two* teachers and

every teacher grades one low, three average, and one high quality paragraph. Phase 1 therefore gives 40 human ratings, two per paragraph: two lists of 20 for the human–human comparison, and all 40 paired against each model’s score, repeated for both raters. We keep both teacher scores rather than averaging them, so the spread between teachers is not lost.

3.3 Two-Phase Protocol

In **Phase 1 (blind scoring)**, teachers grade their paragraphs. A teacher has to finish Phase 1 before Phase 2 opens, so nothing the AI produced is visible to them while they are scoring. In **Phase 2 (audit)**, each teacher is shown the AI’s score together with its written explanation. Where the two scores differ on a skill, the teacher picks one of three answers: (1) the AI’s score is better, and its explanation pointed out something they had missed (we call this being *persuaded*); (2) their own score is better, and the AI’s explanation is unconvincing; or (3) the scores differ but both are reasonable. Teachers also rate on a 1 to 5 scale how easy the feedback is to act on and how appropriate its tone is, and report any error the AI flagged that is not actually there.

3.4 How We Measure Agreement

Quadratic Weighted Kappa (QWK; Cohen, 1968) is the standard measure in essay scoring and counts large score gaps as worse than small ones. Gwet’s AC2 with quadratic weights (Gwet, 2008) measures the same thing but is robust to score clustering. Pearson r and Spearman ρ show whether two graders move up and down together, even if one is consistently stricter. Adjacent Agreement (AA) is the share of paragraphs scored within 0.5 points for a single skill and within 1.0 point for the total, and Mean Absolute Error (MAE) is the average size of the gap. Because 20 paragraphs is a small sample, we also report 95% confidence intervals for QWK and AC2 from 1,000 bootstrap resamples. We resample paragraphs rather than rows, since the 40 rows are 20 paragraphs rated twice and the two rows for one paragraph are not independent.

4 Results and Analysis

4.1 Phase 1: The LLMs Stay Near the Middle

Figure 1 shows both LLMs keeping their scores close to the middle of the scale while teachers use the full range. On the total score, human marks are

widely spread (mean 5.975, SD 1.713), but Gemini (SD 0.939) and Llama (SD 0.743) stay in a narrow band. This is the worst on vocabulary and grammar.

4.2 Phase 1: Where the LLMs Do Well and Where They Fail

Table 1 gives every agreement measure, with confidence intervals, for each skill and pair of graders. On the *total* score, *content*, and *organization*, a human and an LLM agree about as much as two humans do. On the total score, human–AI QWK is 0.462 for Gemini and 0.422 for Llama against 0.490 human–human, and the human–AI intervals $[0.18, 0.62]$ and $[0.19, 0.57]$ overlap substantially with the human–human interval $[-0.09, 0.85]$, so these data cannot separate them. Human–AI AC2 is higher (0.825 and 0.821 against 0.738), and content behaves the same way. The scores then drop on *vocabulary* (0.225 and 0.212 against 0.433) and *grammar* (0.194 and 0.007 against 0.324).

4.3 Why One Measure Is Not Enough

No single measure tells the whole story, and three cases show why.

The *kappa paradox* (Feinstein and Cicchetti, 1990) is clearest on organization: the teacher pair has almost no QWK (0.052) but a reasonable AC2 (0.574). The teachers do agree closely on 80% of the paragraphs, and QWK is low only because of score clustering, not because they really disagree. The confidence intervals point the same way. For the two teachers, the QWK interval dips below zero on all five skills, so on the QWK scale we cannot rule out that they are only agreeing by chance. Their AC2 interval stays above zero on all five, so on the AC2 scale we can. One interval on 20 paragraphs would prove little, but the same split appearing on every skill is a sign that QWK is the shakier of the two measures at this sample size.

A model can also exhibit score clustering, as Figure 1 shows for vocabulary and grammar. Gemini’s vocabulary score is almost flat, and because it sits near the mark teachers usually give, it does well on the measures that only look at distance: against Llama, 82.5% versus 77.5% adjacent agreement and MAE 0.344 versus 0.556. But its Pearson r of 0.318 and Spearman ρ of 0.327 fall below the 0.435 and 0.375 between teachers, so it is barely telling the paragraphs apart.

Grammar then splits the two models in opposite ways. Gemini’s QWK (0.194) and AC2 (0.607) trail the teacher values (0.324 and 0.756), but its

r of 0.294 and ρ of 0.311 are close to their 0.338 and 0.285, so it sorts the paragraphs much as they do and simply marks harder. Llama is the reverse: an AC2 of 0.812 looks healthy, while a QWK of 0.007, an r of 0.010, and a ρ of 0.022 indicate strong score clustering around the typical teacher-assigned score.

4.4 Is a Teacher Still Needed?

The Gemini vs. Llama row of each block in Table 1 shows how much the two LLMs agree with each other. On the total score, content, and organization they agree *more* with each other than any pair involving a human does (total QWK 0.826, AC2 0.894). Agreement between models from different developers is a good sign, though it does not guarantee either is right. On vocabulary and grammar even this support is gone. The two models’ vocabulary QWK is 0.000, with an interval of $[-0.05, 0.03]$, the tightest in the study and centred on no agreement at all, and their grammar AC2 interval $[-0.16, 0.65]$ contains zero. On these two skills we cannot show that the models agree with each other better than chance.

4.5 Phase 2: What the Teachers Said

The audit tells us which Phase 1 gaps are real mistakes and which are only numbers behaving oddly (Tables 2 and 3). The acceptability rate, the share of judgments where the AI score matched the teacher, was accepted as better, or was called equally reasonable, is high everywhere (77.5% to 92.5%), so most score gaps are ones a teacher can live with rather than ones that would mislead a student. The most interesting result is on grammar: teachers said Gemini’s stricter score was better than their own 20% of the time, so its poor Phase 1 numbers partly reflect sound strictness rather than noise. The written feedback also scored well: mean actionability 3.90 for Gemini and 3.80 for Llama, mean tone 3.83 and 3.80, and an invented error in only 2.5% of cases for both models.

5 Conclusion

Given only the rubric and no worked examples, an LLM agrees with a teacher about as closely as two teachers agree with each other on *total quality*, *content*, and *organization*. It is not reliable on *vocabulary* and *grammar*, where the two models do not even agree with each other. A chance-corrected measure is needed, or a skill where everyone gives similar marks looks like a failure when it is not.

Score Distributions by Rater and Rubric Factor

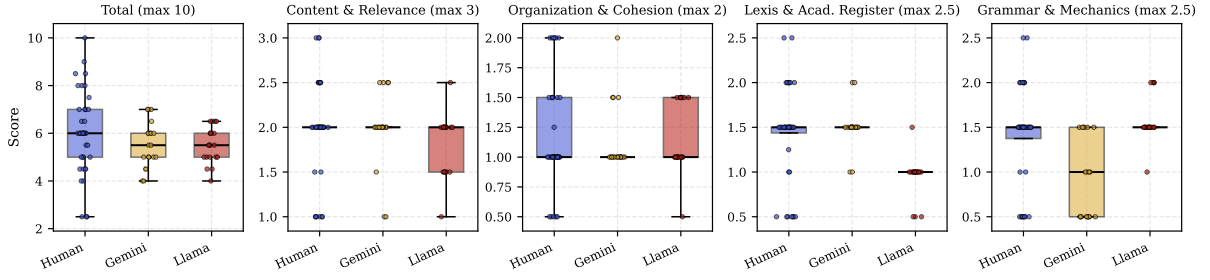


Figure 1: Score spread by grader and rubric skill (box plots, individual scores as dots). Both LLMs stay near the middle compared to the teachers, most strongly on Lexis and Grammar.

Skill	Grader Pair	QWK [95% CI]	AC2 [95% CI]	r	ρ	AA (%)	MAE
Total	Human vs. Human	0.490 [−0.09, 0.85]	0.738 [0.34, 0.92]	0.502	0.426	70.0	1.150
	Human vs. Gemini	0.462 [0.18, 0.62]	0.825 [0.69, 0.87]	0.582	0.544	57.5	1.200
	Human vs. Llama	0.422 [0.19, 0.57]	0.821 [0.68, 0.88]	0.622	0.587	62.5	1.175
	Gemini vs. Llama	0.826 [0.70, 0.91]	0.894 [0.83, 0.95]	0.849	0.863	100.0	0.375
Content & Rel.	Human vs. Human	0.420 [−0.06, 0.78]	0.808 [0.50, 0.94]	0.444	0.408	80.0	0.350
	Human vs. Gemini	0.484 [0.11, 0.73]	0.878 [0.75, 0.94]	0.501	0.521	85.0	0.300
	Human vs. Llama	0.412 [0.11, 0.64]	0.864 [0.75, 0.92]	0.485	0.478	87.5	0.350
	Gemini vs. Llama	0.657 [0.30, 0.85]	0.904 [0.78, 0.96]	0.727	0.674	100.0	0.200
Org. & Coh.	Human vs. Human	0.052 [−0.34, 0.47]	0.574 [0.12, 0.84]	0.052	0.093	80.0	0.438
	Human vs. Gemini	0.312 [−0.02, 0.58]	0.840 [0.73, 0.92]	0.353	0.343	90.0	0.269
	Human vs. Llama	0.368 [0.04, 0.63]	0.821 [0.68, 0.92]	0.407	0.380	92.5	0.269
	Gemini vs. Llama	0.583 [0.20, 0.82]	0.951 [0.62, 0.98]	0.586	0.651	100.0	0.125
Lexis & Reg.	Human vs. Human	0.433 [−0.15, 0.84]	0.792 [0.47, 0.96]	0.435	0.375	90.0	0.338
	Human vs. Gemini	0.225 [−0.02, 0.48]	0.871 [0.72, 0.94]	0.318	0.327	82.5	0.344
	Human vs. Llama	0.212 [0.03, 0.41]	0.674 [0.48, 0.78]	0.514	0.506	77.5	0.556
	Gemini vs. Llama	0.000 [−0.05, 0.03]	0.570 [0.00, 0.72]	0.000	0.000	75.0	0.550
Grammar & Mech.	Human vs. Human	0.324 [−0.18, 0.75]	0.756 [0.41, 0.92]	0.338	0.285	80.0	0.400
	Human vs. Gemini	0.194 [−0.02, 0.40]	0.607 [0.29, 0.76]	0.294	0.311	55.0	0.575
	Human vs. Llama	0.007 [−0.26, 0.29]	0.812 [0.53, 0.92]	0.010	0.022	77.5	0.425
	Gemini vs. Llama	0.093 [−0.04, 0.24]	0.303 [−0.16, 0.65]	0.301	0.297	55.0	0.650

Table 1: Phase 1 agreement measures for every skill and pair of graders. Confidence intervals come from 1,000 bootstrap resamples drawn at the paragraph level. AA = Adjacent Agreement, MAE = Mean Absolute Error.

Model	Skill	Persu. (%)	Accept. (%)
Gemini	Content & Rel.	5.0	87.5
Llama	Content & Rel.	2.5	87.5
Gemini	Org. & Coh.	2.5	85.0
Llama	Org. & Coh.	7.5	92.5
Gemini	Lexis & Reg.	5.0	90.0
Llama	Lexis & Reg.	10.0	82.5
Gemini	Grammar & Mech.	20.0	85.0
Llama	Grammar & Mech.	10.0	77.5

Table 2: Phase 2 persuasion and acceptability rates for each skill.

The audit shows that many apparent failures are differences a teacher accepts or cases where the AI was right, with few invented errors and feedback students can act on. We therefore suggest a teacher in the loop: let the model draft scores and feedback

Model	Action.	Tone	Halluc. (%)
Gemini	3.90	3.83	2.5
Llama	3.80	3.80	2.5

Table 3: Phase 2 mean rating for how easy the feedback is to act on (1–5), for its tone (1–5), and the share of judgements where the model reported an error that was not there (hallucination).

on all four skills, but keep a teacher auditing everything before the student sees it. Vocabulary and grammar need the most care, and our results indicate what to look for: score clustering may signal that the model has failed to distinguish a paragraph from the rest, so the teacher should check whether its score is appropriately differentiated rather than merely plausible on its own.

Limitations

Our study is small: 20 paragraphs, one writing prompt, two LLMs, and eight teachers. This is a resource constraint rather than a design choice. API costs add up quickly, and the human side is heavier still: a teacher scores every paragraph on four skills, then reads and audits the AI’s score, reasoning, and feedback for each. A bigger study would mean either more paragraphs per teacher, which brings back the fatigue the block design avoids, or more teachers, which we could not afford. What we offer instead is the design, which carries over unchanged to a larger corpus and panel. The results are preliminary, so the agreement numbers are a rough guide rather than a settled answer.

All the paragraphs argue a point on one social science topic and are second-language English written by Bengali speakers, so our findings may not hold for other kinds of writing, other first languages, or first-language English writers. This matters most for grammar, where a writer’s first language shapes which errors appear. We also have no independent measure of the writers’ proficiency, so we cannot separate the effect of ability from the effect of first language. Few-shot examples or fine-tuning could change the vocabulary and grammar results.

The audit is done by the same teachers who gave the Phase 1 scores, so they may be attached to their earlier marks. One of the eight teachers is an author of this paper and wrote the rubric. That teacher did not take part in the study’s design or in the analysis, did not know which agreement measures would be computed, and was informed about the study on the same terms as the other seven. The teacher appears in 5 of the 20 human–human pairs and contributes 5 of the 40 audit judgments per skill. Authorship is still an interest the other seven do not have, and it matters most in Phase 2, where the three response options differ visibly in how favourable they are even without knowledge of the metrics. At 5 of 40 judgments, however, no acceptability rate could shift by more than 12.5 points, and the observed rates range from 77.5% to 92.5%. We did not exclude these ratings, since dropping a quarter of the human–human pairs would leave the benchmark estimated on 15 paragraphs. A rubric’s author may also apply it more consistently than other raters, which could raise the human–human agreement we compare the models against, though the same rater also contributes to the human–AI comparison.

With only two ratings per paragraph, the agree-

ment estimates are uncertain: the intervals in [Table 1](#) are wide, and most differences between grader pairs are not resolved at this sample size. Finally, we test one version of each model, and results may change as models are updated.

Ethics Statement

The 20 paragraphs are pre-existing coursework from past assessments, not written for this study. A public university in Bangladesh provided them and authorised their research use, on the condition that it is not named. Individual student consent was not obtained. Before any analysis, names, student identifiers, and institutional details were removed. No paragraph is reproduced; all reported results are aggregate. No student’s grade or academic standing was affected. The AI-generated scores and feedback were never shown to students or used in any assessment decision.

Eight professional English teachers took part, four men and four women. They were informed of the study’s purpose beforehand, took part voluntarily, and could withdraw at any time. They received no financial compensation. They are not individually identifiable from the results. Formal ethics review was not sought. We relied on the safeguards described above: de-identification, non-reproduction, no effect on grades, and voluntary participation.

Only de-identified paragraph text was sent to the model APIs. Llama 3.3 70B was accessed through Groq, which does not use API inputs and outputs for model training. Gemini was accessed through the free tier of the Google Gemini API. Google’s terms permit free-tier inputs and outputs to be used to improve Google’s products, including human review; paid tiers exclude this. This is a shortcoming: the paid tier is the appropriate choice for student-authored text, and we recommend it for future work. No personally identifying information was transmitted, and the topic is general public policy; individual risk is minimal.

The training data of both commercial models is not publicly documented. Any biases they carry regarding writer background, dialect, or first language may reach their scores. A single-prompt, single-language corpus cannot detect such effects. This is a further reason for the teacher-in-the-loop use we recommend: a human auditor is the only safeguard against biases we cannot measure.

References

- Yigal Attali and Jill Burstein. 2006. [Automated essay scoring with e-rater® v.2](#). *Journal of Technology, Learning, and Assessment*, 4(3).
- Jacob Cohen. 1968. [Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit](#). *Psychological Bulletin*, 70(4):213–220.
- Alvan R. Feinstein and Domenic V. Cicchetti. 1990. [High agreement but low kappa: I. the problems of two paradoxes](#). *Journal of Clinical Epidemiology*, 43(6):543–549.
- Google DeepMind. 2025. [Gemini 3 flash preview](#). Gemini API documentation, Google AI for Developers. Model code: gemini-3-flash-preview. Accessed 2026.
- Groq. 2024. [Llama-3.3-70b-versatile model documentation](#). GroqCloud documentation. Endpoint llama-3.3-70b-versatile, served with TruePoint Numerics quantization. Accessed 2026.
- Kilem Li Gwet. 2008. [Computing inter-rater reliability and its variance in the presence of high agreement](#). *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.
- John Hattie and Helen Timperley. 2007. [The power of feedback](#). *Review of Educational Research*, 77(1):81–112.
- Zixuan Ke and Vincent Ng. 2019. [Automated essay scoring: A survey of the state of the art](#). In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 6300–6308.
- Watheq Ahmad Mansour, Salam Albatarni, Sohaila Eltanbouly, and Tamer Elsayed. 2024. [Can large language models automatically score proficiency of written essays?](#) In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 2777–2786, Torino, Italia. ELRA and ICCL.
- Meta AI. 2024. [Llama 3.3 70b instruct model card](#). Meta Llama Models repository. Accessed 2026.
- Atsushi Mizumoto and Masaki Eguchi. 2023. [Exploring the potential of using an ai language model for automated essay scoring](#). *Research Methods in Applied Linguistics*, 2(2):100050.
- Ben Naismith, Phoebe Mulcaire, and Jill Burstein. 2023. [Automated evaluation of written discourse coherence using GPT-4](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 394–403.
- Dadi Ramesh and Suresh Kumar Sanampudi. 2022. [An automated essay scoring systems: a systematic literature review](#). *Artificial Intelligence Review*, 55(3):2495–2527.
- Maja Stahl, Leon Biermann, Andreas Nehring, and Henning Wachsmuth. 2024. [Exploring LLM prompting strategies for joint essay scoring and feedback generation](#). In *Proceedings of the 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024)*, pages 283–298, Mexico City, Mexico. Association for Computational Linguistics.
- Masaki Uto. 2021. [A review of deep-neural automated essay scoring models](#). *Behaviormetrika*, 48(2):459–484.
- Kevin P. Yancey, Geoffrey Laflair, Anthony Verardi, and Jill Burstein. 2023. [Rating short L2 essays on the CEFR scale with GPT-4](#). In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA)*, pages 576–584.
- 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](#). In *Advances in Neural Information Processing Systems 36: Datasets and Benchmarks Track*, volume 36.

A Grading Prompt

Both LLM graders were given the same prompt, shown in [Figure 2](#), with no example gradings. The placeholder {{Grading Standard}} was replaced by the full rubric in [Appendix B](#), {{Output Format}} by the template in [Appendix C](#), and {{Student Submission Text}} by the paragraph being graded.

B Grading Rubric

The rubric was written by one of the eight teachers, who is an author of this paper. [Table 4](#) gives it in full: the four skills, the highest score each one can earn, what the grader should look at, and what a paragraph has to do to earn each score. The total score is the sum of the four skills (max 10.0) and is given in steps of 0.25. This whole rubric was inserted into the prompt in place of {{Grading Standard}}, and the teachers graded from the same rubric.

C Output Format

[Figure 3](#) gives the template that replaced {{Output Format}} in the prompt. It fixes the order of the answer: a score and a reason for each skill, then the total, then the two kinds of feedback, so that the model looks at each skill carefully before it commits to a single overall score.

```

You are an expert English writing evaluator. Grade the student submission
strictly based on the rubric provided.

## Target Audience & Tone:
The student is a first-year undergraduate student (a fresher). Keep your
feedback encouraging, direct, and highly logical. When explaining structural
issues, frame them pragmatically: focus on how clear paragraph organization
builds a cohesive, easy-to-follow argument. Avoid overly complex literary
jargon; instead, emphasize how grammatical errors impede clear communication
and how precise vocabulary strengthens the overall clarity and academic impact
of their writing.

Your goal is to provide a clear score breakdown, followed by specific
Corrective Feedback and Improvement Feedback.

## Feedback Definitions:
* Corrective Feedback: Identify specific errors in grammar, mechanics, and
  blatant structural failures. Quote the exact error from the text, explain
  logically why it is incorrect, and provide the corrected version.
* Improvement Feedback: Identify areas where the writing is acceptable but
  could be elevated to hit the next tier of the rubric. Focus on vocabulary
  enhancement, deeper analysis, and better transitions. Suggest specific ways
  to rewrite or expand these sections.

## Assignment Context
Title: Paragraph on Fake News on Social Media
Topic: Social Science

## Grading Guidelines
{{Grading Standard}}

## Maximum Score : 10.0

## Output Format
{{Output Format}}

## Student Submission
---
{{Student Submission Text}}
---
```

Figure 2: The zero-shot grading prompt given to both LLM graders.

Skill (Max)	Score	What earns this score
Content & Relevance (3.0)	<i>Focus</i>	Relevance, depth of argument, and specific evidence.
	3.0	Exceptional. Accurately defines fake news within the social media context (algorithms, virality) and provides a sophisticated impact analysis with a real-world example.
	2.0	Competent. Addresses the prompt well but may be slightly generic or miss the “social media” specific mechanisms (e.g. treats it like general rumors).
	1.0	Emerging. Vague or surface-level; fails to provide an example or meaningful impact analysis.
Organization & Cohesion (2.0)	<i>Focus</i>	Paragraph unity, topic sentences, and transitions.
	2.0	Strong “Point-Evidence-Explanation” (PEE) structure. Clear topic sentence and seamless transitions between the definition and the consequences.
	1.0	Functional. Has a beginning, middle, and end, but the logical flow is “clunky” or the topic sentence is weak.
	0.5	Disorganized. Ideas are scattered; lacks a cohesive argument or logical progression.
Lexis & Academic Register (2.5)	<i>Focus</i>	Word choice, technical terminology, and formal tone.
	2.5	Precise. Correctly uses technical terms (disinformation, algorithmic bias, echo chambers). Maintains a formal, objective stance (avoids “I think” or “it’s scary”).
	1.5	Appropriate. Standard vocabulary. Understandable but relies on common words rather than academic or field-specific terminology.
	0.5	Informal. Uses conversational language, slang, or overly emotional/subjective phrasing.
Grammar & Mechanics (2.5)	<i>Focus</i>	Syntax, punctuation, spelling, and capitalization.
	2.5	Highly Accurate. Little to no errors in tense, subject-verb agreement, or sentence structure. Proper use of complex sentences.
	1.5	Minor Issues. 1–3 slips that do not impede meaning (e.g. a missing comma or a singular/plural slip).
	0.5	Pervasive Errors. Significant issues with syntax or “Mechanics” (spelling/punctuation) that make the paragraph difficult to read.

Table 4: The full grading rubric. The listed values are the anchor points; graders could give any score in between in steps of 0.25.

Use proper spacing and indentation. Format your response exactly as follows:

Score Breakdown:

1. Content & Relevance: [Score] / [Max Score]
<<indentation>>Reason: [Reason for the score]
2. Organization & Cohesion: [Score] / [Max Score]
<<indentation>>Reason: [Reason for the score]
3. Lexis & Academic Register: [Score] / [Max Score]
<<indentation>>Reason: [Reason for the score]
4. Grammar & Mechanics: [Score] / [Max Score]
<<indentation>>Reason: [Reason for the score]

Overall Score: [Total Score] / 10.0

Corrective Feedback:

- <<Numbered list in the following format>>
- <<Number>> [Quote the specific error from the text]
- <<indentation>> Issue: [Briefly explain the grammatical or structural error]
- <<indentation>> Correction: [Provide the fixed version]

Improvement Feedback:

- <<Numbered list in the following format>>
- <<Number>> [Quote the specific sentence/phrase that could be improved]
- <<indentation>> Target Area: [Name the rubric category being addressed, e.g., Lexis & Academic Register]
- <<indentation>> Suggestion: [Explain how to elevate this specific text to reach a higher rubric tier, offering a concrete example. Suggest a better alternative for the sentence.]

Figure 3: The output format given to both LLM graders. Per-skill scores and reasons come before the total score, which comes before the feedback.

Improved Personalized Typing Predictions via Long Left Context

Soufia Bahmani and Keith Vertanen

Department of Computer Science
Michigan Technological University, Houghton, MI 49931, USA
{sbahmani, vertanen}@mtu.edu

Abstract

We study how to improve predictive typing via per-user personalization of a pretrained large language model (LLM). We evaluate personalization in a time-ordered setting, using each user’s earlier messages for adaptation as we try and predict their next message. We compare: (1) personalization versus a non-personalized base model, (2) differing amounts of left context, (3) one-time adaptation versus making small periodic updates, and (4) full fine-tuning versus parameter-efficient fine-tuning. We find both static and personalized LLMs significantly outperform an n-gram language model. This includes when the n-gram was mixed with a user model learned via prediction by partial matching (PPM). Using long recent user-specific context provided the largest personalization gains: conditioning on up to 1024 previous characters reduced the base model’s perplexity by 18%. We found fine-tuning model weights provided only a small additional gain.

1 Introduction

Typing can be slow and effortful for some people, including those with motor impairments (Cai et al., 2024), those facing situational constraints (e.g., typing on a mobile device while in motion), or those writing in a language they are learning (Sarsenbayeva et al., 2017; Haake et al., 2024). Within this space, Augmentative and Alternative Communication (AAC) provides a high-impact motivating case. AAC users often write letter-by-letter using alternative input methods such as eye tracking or single-switch (Beukelman and Light, 2020). Typical AAC text entry rates are 5–10 words per minute, an order of magnitude slower than spoken speech (Trnka and McCoy, 2008).

Text entry interfaces often try to reduce this burden by using a language model to autocorrect noisy input and by offering text predictions. However, a language model is usually trained once on large

generic corpora and then deployed unchanged for every user. This mismatch can be problematic for text entry as users may want to write specific text (e.g., names, contacts, or habitual phrases). Further, the user’s language usage can drift over time.

Previous text entry research has shown the value of personalization and using large language models (LLMs) (Adhikary and Vertanen, 2023; Gaines and Vertanen, 2025). We build on this work by studying progressive per-user LLM personalization under a chronological predict-then-update protocol.

We study progressive personalization of LLMs on real email data from the Enron corpus (Klimt and Yang, 2004). We ask how much text prediction improves when we adapt an LLM to an employee’s past messages. To measure prediction accuracy and typing effort, we use per-character perplexity and keystroke savings. Keystroke savings measures the percentage reduction in key presses compared to letter-by-letter typing in a text entry interface that suggests the next word (Trnka and McCoy, 2008).

We consider two personalization settings. In *one-time personalization*, we first adapt the model on the first half of the sent emails of a user (an Enron employee). We leave the model frozen while evaluating on the user’s second half of sent emails. In *progressive personalization*, we periodically update the model as we pass through the second half of the emails. Progressive personalization aims to adapt to ongoing changes in a user’s writing.

We make the following contributions:

1. We show an LLM without personalization outperforms a strong statistical baseline model adapted to a user’s writing.
2. We show that using large amounts of left context provides effective personalization without the need for updating the weights of an LLM.
3. We find parameter-efficient fine-tuning is faster and matches the prediction performance of full fine-tuning.

2 Related work

Language model personalization has long been recognized as a way to improve user-specific predictions. Early adaptive language modeling approaches included cache models (Kuhn and De Mori, 1990; Jelinek et al., 1991) and Prediction by Partial Matching (PPM) (Cleary and Witten, 1984). Later neural approaches used recurrent neural network language models (RNNLMs) that updated model weights as test text was processed (Mikolov et al., 2010).

Mikolov and Zweig (2012) conditioned an RNNLM on topic information extracted from preceding text. Krause et al. (2018) adapted RNNLMs based on long short-term memory (LSTM) networks during evaluation. The model first scores a text segment, then updates its parameters using gradient descent on that segment before predicting the next segment. These studies examined adaptation to recent text rather than individual-user predictive typing.

For individual-user personalization, Wen et al. (2013) fine-tuned RNNLMs using social-network text from users and their friends, and investigated user-oriented features for speech recognition. King and Cook (2020) compared continued training, interpolation with user-specific models, and recurrent-state priming based on users’ blog posts.

Fowler et al. (2015) studied personalization by simulating touchscreen typing of time-ordered Enron email. They interpolated a background bigram word language model with a user-specific unigram cache. The unigram cache was initialized from earlier emails and updated as typing progressed. Personalization reduced word error rate for noisy touch input compared to the background model alone. Uniform and exponentially decaying caches produced similar results. These caches captured which words a user frequently used, but did not model the contexts in which those words occurred.

Adhikary and Vertanen (2023) extended the study of Fowler et al. (2015) by using character n-gram background models and comparing unigram caching with PPM and a character RNNLM for rescore candidate word predictions. They evaluated typing of Enron email from 44 users, with and without spatial touch noise. They used fixed amounts of text per user for priming and evaluation, comparing primed models with flat-start models that had not observed the seed text. Increasing the background model’s n-gram order im-

proved keystroke savings in both noise conditions, while PPM-based adaptation provided further gains. Combining PPM with the RNNLM achieved the highest keystroke savings in both conditions.

Their RNNLM incorporated user history through hidden-state persistence rather than user-specific parameter updates. Adhikary and Vertanen (2023) observed little benefit from priming the RNNLM and suggested periodically updating neural model parameters on recent writing. We investigate these directions using one-time full-parameter fine-tuning and one-time or progressive LoRA. We also compare progressive conditions with and without seed-set adaptation.

Recent AAC research has explored LLMs for assistance beyond word prediction. KWickChat let users provide a small set of keywords and used an LLM to generate utterances based on those keywords, user profile information, and the conversational partner’s previous turns (Shen et al., 2022). Other work has studied AI-generated phrase suggestions and abbreviation expansion (Valencia et al., 2023; Cai et al., 2024).

For character-level AAC prediction, Gaines and Vertanen (2025) developed a search algorithm for obtaining next-character probabilities from sub-word LLMs. With a domain-adapted OPT-350M model (Zhang et al., 2022), their approach achieved lower average per-character perplexity than the evaluated n-gram, byte-level, and classifier-based alternatives, although inference was slower. Compared with the unadapted model, direct fine-tuning reduced average per-character perplexity and increased average simulated keystroke savings across their test sets. However, their adaptation targeted AAC-like language rather than individual users, and training and evaluation used isolated sentences without context from previous messages.

We extend past work by studying whether a user’s previous text can improve an LLM’s predictions. Specifically, we explore: 1) using previous text in the LLM’s left context, 2) performing one-time parameter adaptation, and 3) performing progressive parameter adaptation.

3 Dataset, procedures, and models

We used the Enron Personalization Validation Set.¹ This dataset contains the sent emails of Enron employees split into 45 development and 44 test users.

¹<https://github.com/google-research-datasets/EnronPersonalizationValidation>

It was assembled by Fowler et al. (2015) to evaluate language model personalization for touchscreen typing. Adhikary and Vertanen (2023) further normalized this dataset to evaluate PPM and recurrent neural network language model (RNNLM) personalization.² We use this latter version.

We split each user’s time-ordered sent emails into two equal-sized sets denoted *seed* (the first 50%) and *evaluation* (the second 50%). The seed set was the only user-specific text available before evaluation. All results were on the evaluation set. This approach follows Adhikary and Vertanen (2023) who evaluated with and without adapting on earlier messages (“primed” versus “flat start”). In the progressive setting, we processed the evaluation set in time-ordered chunks, scoring a chunk before using the chunk to adapt the model.

We used 10 of the 45 development users (denoted DEV) for choosing LLM size, left context length, and fine-tuning hyperparameters. We selected 10 users representative of the distribution over the per-user amount of email over the entire dataset. After fixing all choices, we used 44 test users (denoted TEST) for our final evaluation.

We lowercased all text, excluded phrases containing digits, and retained only the letters a–z, space, and apostrophe. We computed per-character perplexity over this 28-character vocabulary using the algorithm for subword-tokenized LLMs described by Gaines and Vertanen (2025).

We computed keystroke savings as the percentage of keystrokes that a user making no typing mistakes would save by making optimal use of the offered word predictions. We simulated a keyboard with three prediction slots. One slot was reserved for the literal characters typed. Thus, our simulated user could always obtain their desired text. The other two slots extended the currently typed text, searching for the most likely next words under the language model. Our simulator had no explicit word list, rather, we conducted a beam search under the language model looking for sequences ending in a space, comma, exclamation mark, question mark, or period. We assume one keystroke was required to select a prediction and produce any following space. Higher keystroke savings is better.

We tested LLMs from the OPT family (Zhang et al., 2022) as Gaines and Vertanen (2025) found it performed well compared to other model families on conversational style text. As a non-LLM

Language model	Perplexity ↓	Seconds per char
OPT-125M	2.73	0.068
OPT-350M	2.66	0.083
OPT-1.3B	2.57	0.082
OPT-2.7B	2.54	0.092
OPT-6.7B	2.51	0.096
OPT-13B	2.50	0.115

Table 1: Per-character perplexity and inference time on DEV for OPT language models of different sizes.

baseline, we used a 12-gram model with 408 M parameters trained on 21 B characters.³

For approaches that fine-tuned an LLM, we tuned hyperparameters on DEV. For all approaches, we tuned the dropout rate, learning rate, weight decay, and epochs. For LoRA (Hu et al., 2022) approaches, we also tuned adapter rank and alpha. For progressive personalization, we also tuned how many phrases we processed between each model adaptation. See Appendix C for details. The LoRA adapter reduced the average fine-tuning time from 66.9 to 30.8 minutes per user using four NVIDIA A100 GPUs.

4 Experiments

4.1 LLM model size

We first tested OPT LLMs of increasing size. We computed the perplexity of the evaluation split set for each user in the DEV set. The perplexity of each phrase was computed in isolation (i.e. we did not condition on previous phrases in a user’s data). As expected, LLMs with more parameters provided better predictions, though gains started to plateau for larger models (Table 1). Based on this, we decided to use the largest 13-billion-parameter model for our remaining experiments.

4.2 Amount of left context

Next, we explored the impact of how much context to use from a user’s previous emails. We retained recent text from previous phrases and trimmed it from the left to satisfy a specified *left context character limit*, normally removing whole words. The retained context could therefore begin partway through a sentence. As shown in Table 2, increasing the limit from 0 to 512 characters reduced perplexity from 2.50 to 2.05, a relative reduction of approximately 18%. Increasing the limit to 1024 characters provided only a further 1.5% reduction,

²<https://osf.io/45p3j/>

³We used the Dec 2019 huge model from <https://osf.io/ajm7t/>

Left context char	Perplexity ↓	Seconds per char
0	2.50	0.115
64	2.33	0.119
128	2.20	0.125
256	2.11	0.144
512	2.05	0.185
1024	2.02	0.275

Table 2: Per-character perplexity and inference time on DEV with increasing left context.

while inference time increased from 0.185 to 0.275 seconds per character. We therefore chose a left context limit of 512 for the main experiments. Section 4.8 compares context limits of 128, 256, 512, and 1024 characters on TEST for the baseline OPT-13B model without parameter adaptation (denoted Base), one-time LoRA, and seed-initialized progressive LoRA.

4.3 Static models

Our main evaluation results on TEST are shown in Table 3. We tested a static n-gram language model and a pretrained LLM (OPT-13B). As expected, the LLM performed substantially better, reducing perplexity from 3.00 to 2.05 and increasing keystroke savings from 54.8% to 67.5%. The LLM was able to condition on up to 512 characters from previous emails versus only 11 characters for the n-gram model. For comparison, the perplexity on TEST using the LLM using no left context was 2.46. This shows the LLM could leverage large amounts of left context to provide improved user-specific predictions even without user-specific adaptation of model parameters.

4.4 Mixture of n-gram + adaptive PPM

We compared LLM personalization with a statistical baseline model. Our baseline combined a fixed n-gram language model with an adaptive PPM model, similar to the mixture used by [Adhikary and Vertanen \(2023\)](#). PPM is an adaptive variable-order language model originally developed for text compression ([Cleary and Witten, 1984](#)). Similar to an n-gram model, PPM predicts the next symbol from recent context and uses lower-order contexts to smooth its probability estimates. A PPM model of order n conditions on the previous n symbols (in contrast, an n -gram model conditions on the previous $n - 1$ symbols). PPM maintains explicit context-dependent symbol counts that can be updated efficiently during use, making it straightfor-

ward to personalize and update continuously as new text is observed.

We used the character 12-gram model to generate an n-best list of 100 candidate word completions based on a user’s current text context. We then scored the n-best candidates using a user’s PPM model. We used a PPM model of order 11, which could condition on up to the previous 11 characters. For each model, we normalized the candidate probabilities to sum to one over the same n-best list. We then linearly interpolated these probabilities, assigning a weight of 0.75 to the n-gram model and the remaining 0.25 to PPM. We used the highest-ranked candidates under this mixture in our keystroke savings simulation.

For each user, we initialized a new PPM model and primed it on a user’s seed set. During evaluation, n-best scoring did not modify the PPM model. After each target word was committed, we updated PPM once with the committed word and its following space. Thus, the n-gram component of the mixture model remained static, whereas the PPM component used seed-set personalization followed by progressive adaptation on the evaluation set.

As shown in Table 3, compared to the static n-gram model, the mixture model reduced perplexity from 3.00 to 2.88. Keystroke savings also improved from 54.8% to 56.7%. However, despite adapting to a user’s text, the mixture model still performed substantially worse than OPT-13B without parameter adaptation. When conditioned on a user’s previous sent emails, OPT-13B achieved a perplexity of 2.05 and keystroke savings of 67.5%.

4.5 One-time personalization

We first tried full parameter fine-tuning of the LLM on each user’s seed set. While this only reduced the perplexity slightly, from 2.05 to 2.03, it did increase the keystroke savings by 0.5%. We found a LoRA adapter performed similar to full fine-tuning with the same perplexity of 2.03 with a slight decrease in keystroke savings from 68.0% to 67.9%.

4.6 Progressive personalization

We attempted to further improve the LoRA adapter learned from the seed data by progressively fine-tuning the adapter every 128 phrases during evaluation. While this slightly reduced perplexity from 2.03 to 2.02, it provided only a small improvement in keystroke savings.

To separate the contribution of progressive updates from seed-set adaptation, we also evaluated

Language model	Adaptation		Perplexity ↓	Keystroke savings ↑
	method	on seed? on eval?		
12-gram	none	no no	3.00 ± 0.047	$54.8 \pm 0.5\%$
12-gram + PPM-11	PPM	yes yes	2.88 ± 0.050	$56.7 \pm 0.5\%$
OPT-13B	none	no no	2.05 ± 0.038	$67.5 \pm 0.6\%$
OPT-13B	one-time full fine-tuning	yes no	2.03 ± 0.039	$68.0 \pm 0.6\%$
OPT-13B	one-time LoRA	yes no	2.03 ± 0.038	$67.9 \pm 0.6\%$
OPT-13B	one-time LoRA + progressive LoRA	yes yes	2.02 ± 0.039	$67.9 \pm 0.6\%$
OPT-13B	progressive LoRA(base-initialized)	no yes	2.03 ± 0.039	$67.8 \pm 0.6\%$
OPT-13B	cumulative-reset LoRA (base-initialized)	no yes	2.03 ± 0.040	$67.8 \pm 0.6\%$

Table 3: Per-character perplexity and keystroke savings of different language models and adaptation approaches. Results are means over the 44 TEST users. $\pm$ denotes 95% nonparametric user-level bootstrap confidence intervals and is the larger distance from the mean to either bootstrap endpoint. OPT-13B results conditioned on up to 512 characters of previous context. In the final row, each non-final block of 128 evaluation phrases was scored before being used for further LoRA fine-tuning.

base-initialized progressive personalization. In this approach, we started from the non-personalized OPT-13B checkpoint and did not adapt on the seed set. As with the normal progressive method, we updated the LoRA adapter after every 128 evaluation phrases. This did only slightly worse with a perplexity of 2.03 and a keystroke savings of 67.8%. This shows it is possible to learn a useful LoRA adapter even without a large amount of initial text.

4.7 Cumulative progressive personalization

We also evaluated a variant of the progressive method in which we periodically relearned the LoRA adapter from the cumulative set of all evaluation text seen thus far. The hope was that this would allow more robust updates to the adapter as the amount of text increased.

As in the base-initialized progressive condition described in the previous subsection, we did not adapt on the seed set, and the evaluation set was processed chronologically in 128-phrase chunks. The cumulative variant differed in how the adapter was updated. After scoring each non-final chunk, we reloaded the original non-personalized OPT-13B checkpoint, initialized a new LoRA adapter, and fine-tuned it on all evaluation phrases observed up to that point. In contrast, the base-initialized progressive method continued fine-tuning the existing adapter on only the newly scored chunk. Thus, the cumulative adaptation sets contained 128 phrases, 256 phrases, 384 phrases, and so on.

The cumulative approach obtained a mean per-character perplexity of 2.03 and mean keystroke savings of 67.84% on TEST. These mean results were similar to those of base-initialized sequential progressive LoRA. Compared with seed-initialized

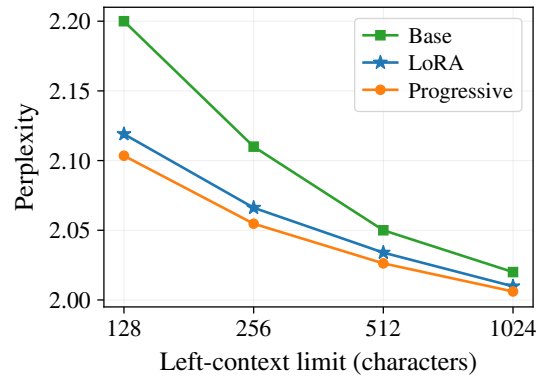


Figure 1: Perplexity at left-context limits of 128, 256, 512 and 1024 characters. Results on TEST for OPT-13B without fine-tuning (Base), personalized using a LoRA adapter on the seed set (LoRA), and personalized using a LoRA adapter on the seed set and updated during evaluation (Progressive).

progressive LoRA, the cumulative approach had higher mean perplexity and lower mean keystroke savings. The cumulative approach was also slower. Base-initialized progressive LoRA required an average of 38 seconds per fine-tuning update, while the cumulative approach needed 233 seconds.

4.8 Left context influence

Despite trying a variety of parameter fine-tuning approaches, we saw little gain beyond simply using the pretrained OPT-13B model conditioned on 512 characters. To investigate whether long left context was really limiting the upside to model parameter fine-tuning, we compared per-character perplexity for Base, one-time LoRA, and one-time LoRA + progressive LoRA at left context limits of 128, 256, 512, and 1024 characters. Base used pretrained OPT-13B without parameter adaptation at

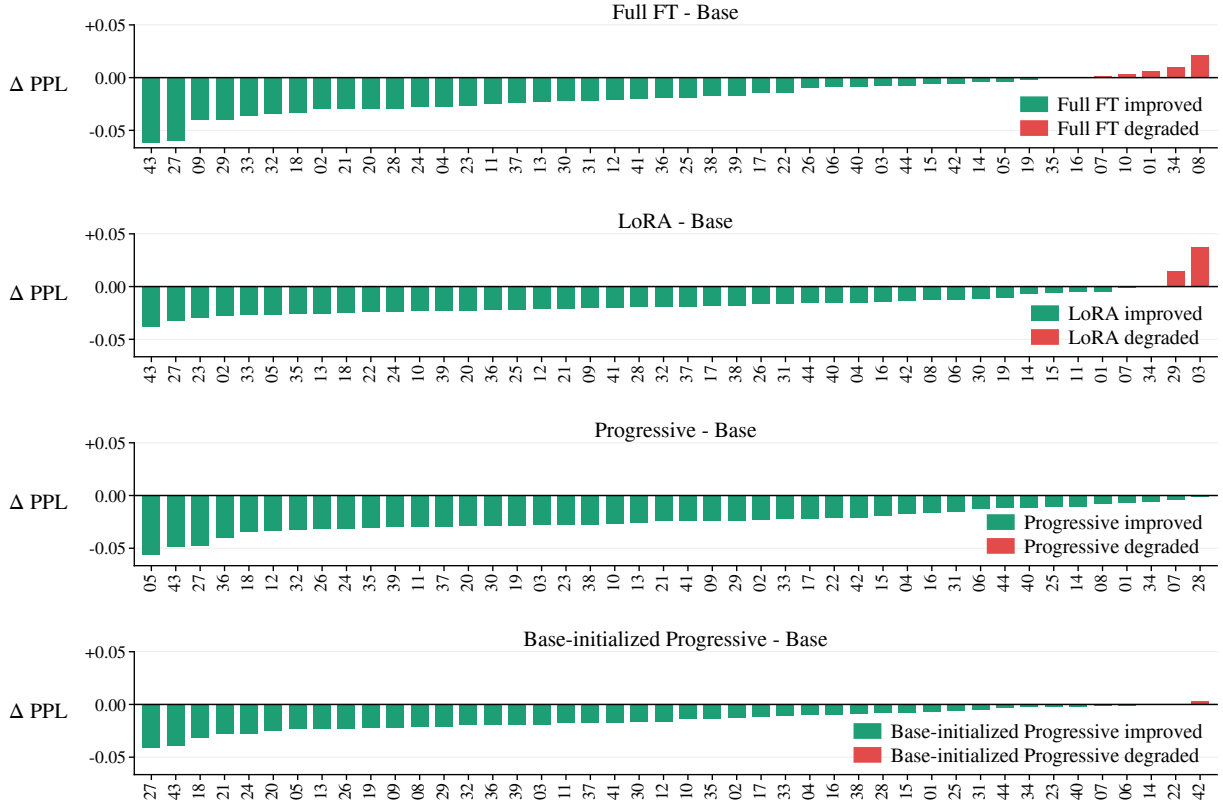


Figure 2: Per-user changes in per-character perplexity (PPL) relative to the non-personalized OPT-13B baseline (Base) for the 44 TEST users. From top to bottom, the panels show full-parameter fine-tuning, one-time LoRA, seed-initialized progressive LoRA, and base-initialized progressive LoRA. ΔPPL denotes the method’s perplexity minus that of Base; negative values indicate lower perplexity. Users are sorted independently from the largest to the smallest improvement in each panel.

each context limit. The left-context limit controlled how much text from previous phrases was supplied during prediction, not the length of the training examples. For each user, one-time LoRA used the same seed-trained adapter at all four context limits, without further fine-tuning.

For progressive LoRA, we ran the chronological predict-then-update procedure separately at 256 and 512 characters, using the same fine-tuning settings and 128-phrase update interval. At 128 and 1024 characters, we re-evaluated the saved adapter sequence from the 512-character run without repeating the training. The first chunk used the seed-trained adapter, and each later chunk used an adapter updated only on earlier chunks. Thus, no chunk was scored using an adapter trained on that chunk. As shown in Figure 1, perplexity decreased as left context increased. Both personalized models had lower mean perplexity than the non-personalized model at every tested left-context limit, and progressive LoRA had the lowest perplexity at all four context limits. At 1024 characters, mean perplexity was approximately 2.02

for Base, 2.01 for one-time LoRA, and 2.00 for progressive LoRA. Notably, with less left context, the perplexity advantage of the approaches that adapted model parameters increased. This suggests that parameter-based personalization may still be useful when shorter left context is desired to reduce inference costs or avoid retaining a user’s past writing between text entry sessions.

4.9 Per-user variability

We also inspected per-user changes in perplexity relative to the non-personalized OPT-13B baseline. Full fine-tuning and LoRA reduced perplexity for 38 and 41 of the 44 users, respectively. Progressive LoRA reduced perplexity for all users, while base-initialized progressive LoRA reduced perplexity for 43 of 44 users, with only a small increase for one user (Figure 2).

Thus, the improvements we saw were broadly distributed across users rather than being driven by a small number of users with unusually large gains. The base-initialized result also shows that chronological updates can provide broadly consistent per-

sonalization gains even without adaptation on the seed half, although progressive LoRA produced the most consistent per-user improvements. Figure 4 in Appendix B shows per-user keystroke savings for Base and the three seed-adapted conditions. Keystroke savings varied across users, whereas differences among these methods for a given user were generally small.

4.10 Paired comparisons between methods

We compared Base, one-time LoRA, one-time LoRA + progressive LoRA, and progressive LoRA (base-initialized). The one-time LoRA + progressive LoRA condition started from a seed-trained adapter, whereas progressive LoRA (base-initialized) started from Base without seed-set adaptation. These comparisons focused on the contributions of seed-set adaptation and progressive updates within LoRA, using Base as the reference. We evaluated all six method pairs for perplexity and keystroke savings, using a left context character limit of 512. Comparisons with full fine-tuning, the statistical baselines, and cumulative-reset personalization were outside the scope of this analysis.

For each method pair and metric, we calculated the difference between method A and method B for each of the TEST users, subtracting B from A. We report the mean of these 44 differences. To estimate a 95% percentile bootstrap confidence interval, we generated 100,000 bootstrap samples. Each sample contained 44 differences drawn with replacement from the original 44 user differences. We calculated the mean of each sample and used the 2.5th and 97.5th percentiles of these bootstrap means as the interval endpoints (Efron and Tibshirani, 1986).

The mean differences and bootstrap confidence intervals describe the size of the differences between methods and the uncertainty in these estimates. We used two-sided Wilcoxon signed-rank tests to assess the statistical significance of differences between methods (Wilcoxon, 1945). We calculated exact conditional p -values, excluding zero differences and assigning average ranks to tied absolute differences. We report W as the smaller of the rank sums for positive and negative differences, and n as the number of nonzero paired differences.

We applied Holm’s step-down Bonferroni correction across all 12 signed-rank tests (six method pairs for each of the two metrics) to control the family-wise error rate at 0.05 (Holm, 1979). Statistical significance was determined using the Holm-adjusted p -values, not the bootstrap intervals. The

bootstrap intervals are unadjusted and should be interpreted individually, not as providing simultaneous 95% coverage across all comparisons.

The signed-rank test assumes independent user differences that are symmetric about zero under the null hypothesis. It is not specifically a test of the mean difference.

Table 4 reports the paired mean differences, confidence intervals, comparison results, and per-user improvement counts. 11 of the 12 signed-rank comparisons were significant after Holm correction. The unadjusted bootstrap intervals showed a consistent pattern: they excluded zero for the 11 statistically significant comparisons and included zero for the remaining non-significant comparison.

All three adaptation conditions had lower mean perplexity and higher mean keystroke savings than Base. One-time LoRA + progressive LoRA achieved the lowest mean perplexity and highest mean keystroke savings among the four conditions. Relative to one-time LoRA, it reduced mean perplexity by 0.00772 and increased mean keystroke savings by 0.103%, improving perplexity for 34 users and keystroke savings for 36. Both comparisons were statistically significant, but the additional typing-effort benefit was small.

Progressive LoRA (base-initialized) also improved over Base without seed-set adaptation. Among the 12 comparisons at the 512-character context setting, the only nonsignificant result was its perplexity difference from one-time LoRA (exact conditional two-sided Wilcoxon signed-rank test: $W = 349$, $n = 44$, Holm-adjusted $p = 0.0896$). Its mean keystroke savings was 0.067% lower than one-time LoRA ($W = 285$, $n = 43$, Holm-adjusted $p = 0.0450$). All ten other comparisons had Holm-adjusted $p < 0.001$. Overall, the paired analysis supports small additional benefits from parameter adaptation beyond recent user-specific context, rather than large improvements in typing effort. All statistics were computed from unrounded per-user scores.

5 Discussion

Our results separate the benefit of using a stronger language model from the additional benefit of per-user adaptation. The mixture of an n -gram model with adaptive PPM improved both perplexity and keystroke savings relative to the static 12-gram model. This shows that a lightweight count-based model can benefit from adapting to a user’s text.

Metric	Method A	Method B	Mean Δ	95% CI	Result (Holm)	Users: A vs. B		
						Better	Worse	Tied
PPL↓	LoRA	Base	-0.016	[-0.020, -0.012]	A better	41	3	0
PPL↓	Progressive	Base	-0.024	[-0.028, -0.021]	A better	44	0	0
PPL↓	Progressive (base)	Base	-0.015	[-0.018, -0.012]	A better	43	1	0
PPL↓	Progressive	LoRA	-0.008	[-0.012, -0.004]	A better	34	10	0
PPL↓	Progressive (base)	LoRA	+0.002	[-0.003, +0.005]	Not sig.	19	25	0
PPL↓	Progressive	Progressive (base)	-0.009	[-0.012, -0.007]	A better	41	3	0
KS↑	LoRA	Base	+0.313	[+0.269, +0.357]	A better	43	1	0
KS↑	Progressive	Base	+0.416	[+0.363, +0.468]	A better	42	1	1
KS↑	Progressive (base)	Base	+0.246	[+0.197, +0.297]	A better	42	1	1
KS↑	Progressive	LoRA	+0.103	[+0.072, +0.135]	A better	36	8	0
KS↑	Progressive (base)	LoRA	-0.067	[-0.115, -0.021]	B better	18	25	1
KS↑	Progressive	Progressive (base)	+0.170	[+0.134, +0.209]	A better	42	0	2

Table 4: Paired comparisons of per-character perplexity (PPL) and keystroke savings (KS) over TEST users with a 512-character left-context. LoRA denotes one-time adaptation; Progressive and Progressive (base) denote seed- and base-initialized progressive LoRA. Differences are A minus B: negative PPL and positive KS differences favor A. KS differences are percentage points. CIs are unadjusted 95% paired-bootstrap intervals. The result identifies the method with the better observed mean for significant comparisons (two-sided Wilcoxon signed-rank test, Holm-adjusted $p < 0.05$ across 12 tests). “Not sig.” denotes a nonsignificant comparison.

However, the adaptive mixture was still substantially worse than the non-personalized OPT-13B baseline. Thus, in our setting, a pretrained LLM using long recent context provided a larger improvement than adapting the statistical language model. Since OPT-13B also used a much longer left context than the 12-gram model, this comparison reflects both the stronger pretrained model and its ability to condition on more of a user’s recent text.

The context-length comparison in Figure 1 shows how left context and model parameter adaptation interact. Perplexity decreased as the left-context limit increased for the baseline model, the LoRA-adapted model, and the progressively adapted LoRA model. The progressive LoRA method had the lowest mean perplexity at 128, 256, 512, and 1024 characters of left context, but the gap between the Base model and both adapted models narrowed as more context became available. The advantage of the progressive LoRA method over one-time LoRA adaptation also became smaller at the longer context limits, with the additional perplexity reduction halving between 512 and 1024 characters. This suggests that left context and parameter adaptation provide partly overlapping user-specific information. Adaptation was more useful when less recent text was available to the model, whereas long left context allowed the non-personalized model to recover more of this information without changing its parameters.

LoRA achieved predictive performance comparable to full-parameter fine-tuning while requiring less fine-tuning time. The progressive LoRA method, which updated the adapter every 128 phrases, further reduced mean perplexity and improved all 44 TEST users. However, the additional improvement in mean keystroke savings over one-time LoRA adaptation was small. Perplexity measures the probability assigned across all characters, whereas keystroke savings depend on whether the desired word appears in one of the available prediction slots. Small changes in the character probability distribution may therefore reduce perplexity without changing the word completions offered.

The base-initialized progressive model also improved over the non-personalized OPT-13B baseline and reduced perplexity for 43 of 44 users, despite receiving no adaptation on the seed set. This shows that chronological updates can provide useful personalization when earlier user data are not available. In contrast, cumulative progressive personalization did not improve over the simpler progressive procedure. Reinitializing the adapter and relearning it from all evaluation text observed thus far provided no additional benefit. Although the mean differences among the personalized OPT-13B models were small, seed-initialized progressive LoRA reduced perplexity for 34 of the 44 users compared with one-time LoRA. Compared with Base, it reduced perplexity for all 44 users.

6 Conclusion

We evaluated non-personalized, one-time personalized, and progressively personalized language models for predictive typing under a chronological predict-then-update protocol. A non-personalized OPT-13B model using long left context already outperformed both the static 12-gram model and the adaptive 12-gram + PPM mixture. Parameter adaptation provided smaller additional gains: LoRA closely matched full-parameter fine-tuning, while progressive LoRA obtained the lowest mean perplexity and the most consistent per-user improvements. Its relative advantage was larger when the available left context was shorter.

We also found that periodically updating the pretrained model using a user’s new messages improved predictions, even without first fine-tuning it on a collection of that user’s earlier messages. The limited change in keystroke savings among the adapted OPT-13B models showed that lower perplexity does not necessarily produce a comparable reduction in simulated typing effort. This is likely because small changes in a model’s predictive ability (as measured by perplexity), may not materially impact a keyboard’s top word predictions. Taken together, these results identify long recent user-specific context as the primary source of improvement, with efficient progressive adaptation providing a smaller, complementary benefit.

Limitations

Our work has a number of limitations. We used one global set of hyperparameters selected on development users and applied it to all TEST users, even though users had very different amounts of available text. Users with short, medium, and long histories may benefit from different fine-tuning hyperparameters, such as learning rate, number of epochs, and dropout.

We evaluated on a dataset containing sent emails of Enron employees. This data is now quite dated and may not reflect modern language usage. The genre of business email may also not be representative of other predictive typing settings such as informal person-to-person conversations that might occur on mobile devices, social media, or be needed in the everyday lives of AAC users. However, to our knowledge, there are no similar datasets containing the communications of a large number of people over a long period of time.

Our progressive adaptation method used an up-

date frequency of 128 phrases based on experiments we conducted on development data. But we did not study the trade-off between update frequency and computational cost in a deployed system. It is possible a different chunk size might offer a more acceptable end-user experience.

The LLM we tested had 13 B parameters and required a powerful GPU for adaptation and inference. The strong ability of left context to provide user personalization likely benefited from the “emergent” in-context learning abilities of the large model we used. Deploying such a large model on a user’s device, as might be necessitated to maintain user privacy, may require advances in mobile device hardware (e.g. GPU resources) and the software infrastructure to efficiently deploy powerful models on edge devices (e.g. model quantization). Alternatively, it would be interesting to investigate combining a large cloud-based model with an on-device personalized model.

We used an OPT LLM released in 2022. While past work showed the OPT family performed well compared to a variety of other model families (Gaines and Vertanen, 2025), there may be more recent LLMs that perform even better. While OPT performed well in English for our experiments, it may not be the best choice for users who need to write in other languages.

Our findings were based on simulating the keystroke savings of a hypothetical perfect user. This user made no typing mistakes and made optimal use of offered word predictions. Actual real-world user performance may differ and needs validation by observing users writing a substantial amount of text over a long period of time.

We did not systematically repeat the fine-tuning experiments with different random seeds. Our paired-bootstrap confidence intervals reflect uncertainty from sampling users, not variability across training runs. Repeating fine-tuning with different seeds would help assess the stability of the adaptation gains we saw in our experiments.

Acknowledgments

This material is based upon work supported by the NSF under Grant No. IIS-2402876.

References

Jiban Adhikary and Keith Vertanen. 2023. [Language model personalization for improved touchscreen typing](#). In *Interspeech 2023*, pages 1344–1348.

- David R. Beukelman and Janice C. Light. 2020. *Augmentative and Alternative Communication: Supporting Children and Adults with Complex Communication Needs*, 5th edition. Paul H. Brookes Publishing.
- Shanqing Cai, Subhashini Venugopalan, Katie Seaver, Xiang Xiao, Katrin Tomanek, Sri Jalsutram, Meredith Ringel Morris, Shaun K. Kane, Ajit Narayanan, Robert L. MacDonald, Emily Kornman, Daniel Vance, Blair Casey, Steve M. Gleason, Philip Q. Nelson, and Michael P. Brenner. 2024. [Using large language models to accelerate communication for eye gaze typing users with ALS](#). *Nature Communications*, 15(1):9449.
- John G. Cleary and Ian H. Witten. 1984. [Data compression using adaptive coding and partial string matching](#). *IEEE Transactions on Communications*, 32(4):396–402.
- Bradley Efron and Robert Tibshirani. 1986. [Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy](#). *Statistical Science*, 1(1):54–75.
- Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. [Effects of language modeling and its personalization on touchscreen typing performance](#). In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI ’15, pages 649–658. ACM.
- Dylan Gaines and Keith Vertanen. 2025. [Adapting large language models for character-based augmentative and alternative communication](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 15273–15291, Suzhou, China. Association for Computational Linguistics.
- Lisa Haake, Sebastian Wallot, Monika Tschense, and Joachim Grabowski. 2024. [Global temporal typing patterns in foreign language writing: Exploring language proficiency through recurrence quantification analysis \(RQA\)](#). *Reading and Writing*, 37(2):385–417.
- Sture Holm. 1979. [A simple sequentially rejective multiple test procedure](#). *Scandinavian Journal of Statistics*, 6(2):65–70.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- F. Jelinek, B. Meriello, S. Roukos, and M. Strauss. 1991. [A dynamic language model for speech recognition](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Pacific Grove, California, February 19–22, 1991*, pages 293–295.
- Milton King and Paul Cook. 2020. [Evaluating approaches to personalizing language models](#). In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2461–2469, Marseille, France. European Language Resources Association.
- Bryan Klimt and Yiming Yang. 2004. [The Enron corpus: A new dataset for email classification research](#). In *Machine Learning: ECML 2004*, volume 3201 of *Lecture Notes in Computer Science*, pages 217–226, Berlin, Heidelberg. Springer.
- Ben Krause, Emmanuel Kahembwe, Iain Murray, and Steve Renals. 2018. [Dynamic evaluation of neural sequence models](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2766–2775. PMLR.
- Roland Kuhn and Renato De Mori. 1990. [A cache-based natural language model for speech recognition](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6):570–583.
- Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. 2010. [Recurrent neural network based language model](#). In *Interspeech 2010*, pages 1045–1048.
- Tomáš Mikolov and Geoffrey Zweig. 2012. [Context dependent recurrent neural network language model](#). In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 234–239. IEEE.
- Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. [ZeRO: Memory optimizations toward training trillion parameter models](#). In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.
- Zhanna Sarsenbayeva, Niels van Berkel, Chu Luo, Vasilis Kostakos, and Jorge Goncalves. 2017. [Challenges of situational impairments during interaction with mobile devices](#). In *Proceedings of the 29th Australian Conference on Computer-Human Interaction*, OzCHI ’17, pages 477–481. Association for Computing Machinery.
- Junxiao Shen, Boyin Yang, John J. Dudley, and Per Ola Kristensson. 2022. [KWickChat: A multi-turn dialogue system for AAC using context-aware sentence generation by bag-of-keywords](#). In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI ’22, pages 853–867. Association for Computing Machinery.
- Keith Trnka and Kathleen McCoy. 2008. [Evaluating word prediction: Framing keystroke savings](#). In *Proceedings of ACL-08: HLT, Short Papers*, pages 261–264, Columbus, Ohio. Association for Computational Linguistics.
- Stephanie Valencia, Richard Cave, Krystal Kallarackal, Katie Seaver, Michael Terry, and Shaun K. Kane. 2023. [“The Less I Type, the Better”: How AI Language Models Can Enhance or Impede Communication for AAC Users](#). In *Proceedings of the 2023 CHI Conference on Human Factors in Computing*

Systems, CHI '23, pages 1–14. Association for Computing Machinery.

Tsung-Hsien Wen, Aaron Heidel, Hung-yi Lee, Yu Tsao, and Lin-shan Lee. 2013. [Recurrent neural network based language model personalization by social network crowdsourcing](#). In *Interspeech 2013*, pages 2703–2707.

Frank Wilcoxon. 1945. [Individual comparisons by ranking methods](#). *Biometrics Bulletin*, 1(6):80–83.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [OPT: Open pre-trained transformer language models](#). *arXiv preprint arXiv:2205.01068*.

A Enron dataset processing

We started with the version of the original Enron Personalization Validation Set provided by [Adhikary and Vertanen \(2023\)](#). For each Enron employee, the dataset has a time-ordered list of phrases from the employee’s sent emails. We kept only phrases that contained no digits. Each retained phrase was then converted to lowercase and filtered so that only letters, spaces, and apostrophes remained. Table 5 summarizes the total amount of data in the full DEV and TEST splits after our preprocessing. As shown in Table 5, the amount of text ranged widely between users. We split each user’s chronological stream in half, using the first half as seed data and the second half for evaluation.

Figure 3 shows the number of phrases remaining after preprocessing for each user in the full development and test partitions. To reduce the time needed for hyperparameter search, we selected DEV users from our full development set: dev03, dev06, dev09, dev14, dev21, dev24, dev27, dev34, dev36, and dev37. We used this subset for model selection and hyperparameter tuning. Our TEST set included all 44 users in the test partition.

Partition	Total			Per-user words		
	Users	Phrases	Words	Median	Min.	Max.
Development	45	109,696	1,069,532	13,585	2,581	177,216
Test	44	92,784	968,531	11,651	2,196	107,709

Table 5: Dataset statistics after preprocessing.

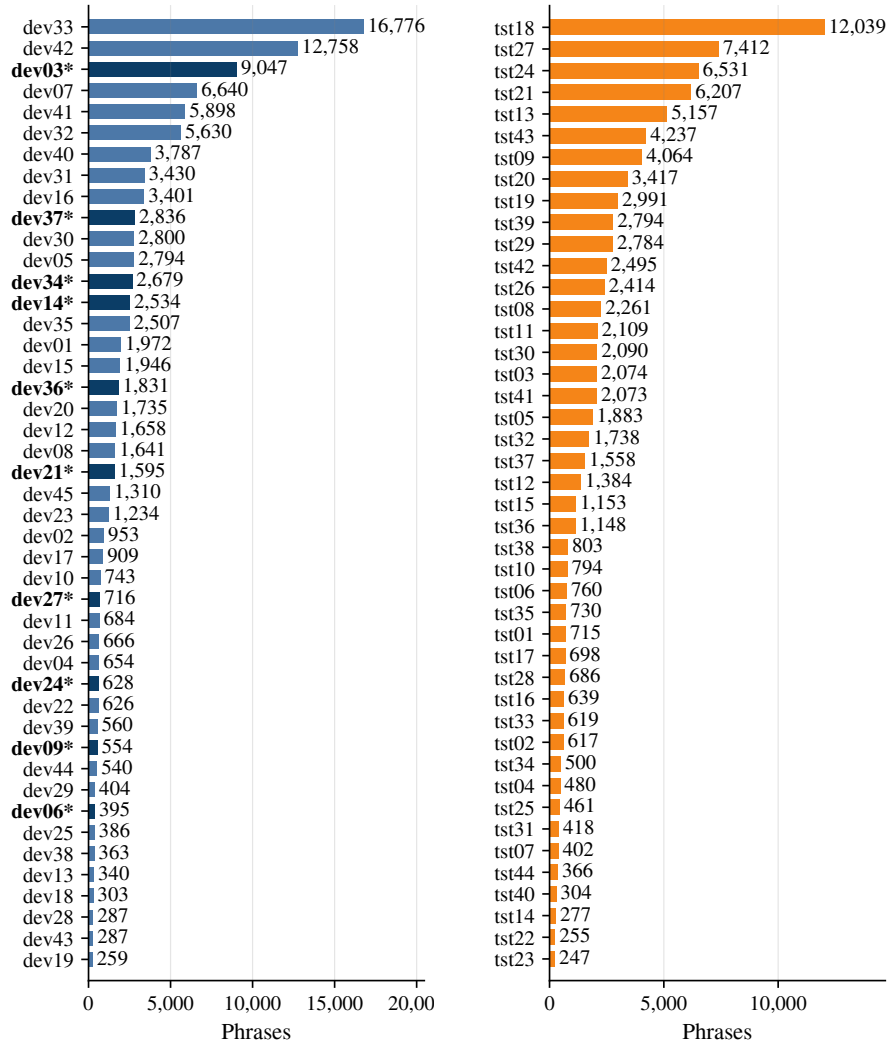


Figure 3: Number of phrases available after preprocessing for each user in the full development partition (left) and the TEST set (right). Users are ordered from most to fewest phrases. Asterisks and darker labels in the left panel mark the 10 users selected for the DEV set used for model selection and hyperparameter tuning.

B Per-user keystroke savings

Figure 4 shows per-user keystroke savings for the non-personalized OPT-13B baseline and the three seed-adapted conditions. Keystroke savings varied substantially across users, whereas differences among the methods for a given user were generally small.

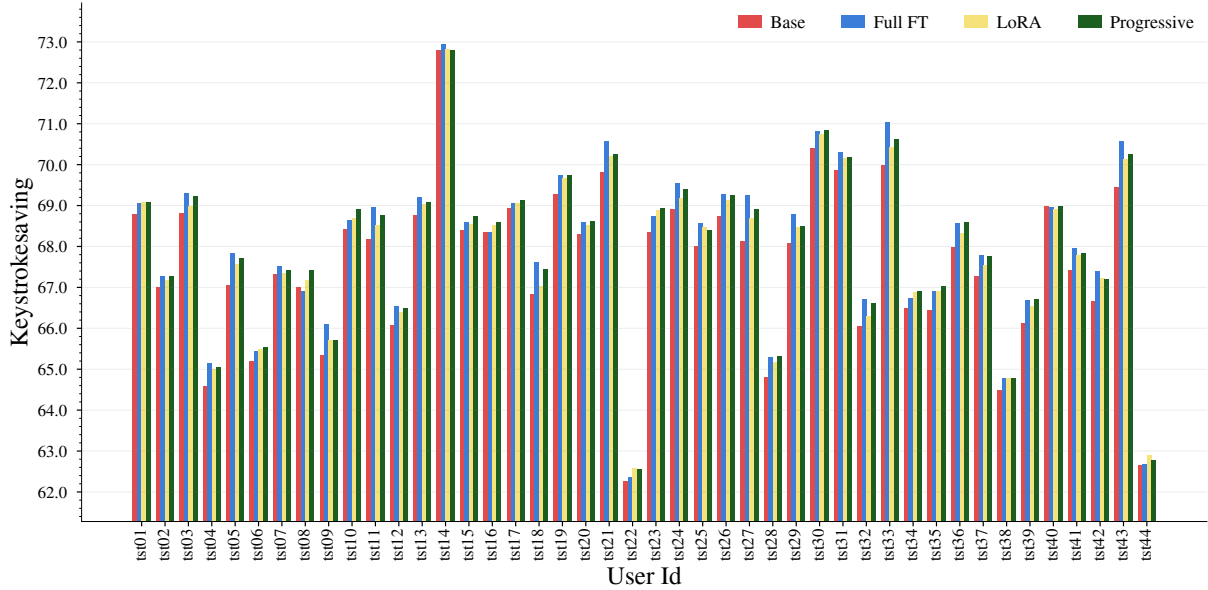


Figure 4: Per-user keystroke savings on TEST for the Base, full-parameter fine-tuning, LoRA fine-tuning, and progressive LoRA (Progressive). Higher values indicate a greater percentage reduction in keystrokes relative to letter-by-letter typing. Users are shown in test-user ID order.

C Hyperparameter tuning

We tuned hyperparameters for full fine-tuning, one-time LoRA, and seed-initialized progressive LoRA. For each method, we selected the settings that gave the lowest mean per-character perplexity across our DEV set, drawn from the full partition of 45 development users. Each tuning trial ran on four NVIDIA A100 GPUs with 40 GB of memory each. We used DeepSpeed ZeRO stage 3 (Rajbhandari et al., 2020) to reduce memory requirements on each GPU. All trials used OPT-13B with up to 512 characters of left context. Hyperparameter tuning used approximately two GPU-days per method. We measured LLM inference latency using FP16 on a single NVIDIA A100 GPU per evaluation process.

We varied the dropout rate, learning rate, weight decay, and the number of training epochs. We selected the learning rate from the candidate set [5e-6, 7e-6, 8e-6, 1e-5, 1.3e-5, 1e-4, 3e-4, 3e-3]. The learning-rate schedule had two additional tunable parameters: the number of warmup steps, selected from 0 to 10, and the stable fraction, selected from 0.00 to 0.80. The stable fraction controlled the fraction of training steps for which the learning rate was held fixed before decay.

The progressive LoRA method had a parameter *chunk size* controlling how often we re-fine-tuned the adapter. We started each of DEV users with a

model fine-tuned on their seed set using the best hyperparameters found for the non-progressive LoRA method. We then selected the chunk size from the candidate set [64, 128, 256, 512, and 1024]. Table 6 shows the values of our tuned hyperparameters for each of our methods.

Adaptation	Chunk size	LoRA rank	LoRA α	Dropout	LR schedule	Warmup steps	Stable fraction	Learning rate	Weight decay	Epochs
Full fine-tuning	–	–	–	0.20	WSD	2	0.15	1×10^{-5}	0.05	2
One-time LoRA	–	8	16	0.10	WSD	2	0.15	3×10^{-4}	0.00	2
Progressive LoRA (seed-initialized)	128	8	16	0.10	WSD	2	0.15	3×10^{-4}	0.00	1
Progressive LoRA (base-initialized)	128	8	16	0.10/0.05 ^a	Linear	2	–	3×10^{-4}	0.00	1
Cumulative progressive LoRA	128	8	16	0.10	Linear	2	–	3×10^{-4}	0.00	1

^aAdapter dropout was 0.10 for the first update and 0.05 for subsequent updates.

Table 6: Hyperparameters used for the five OPT-13B personalization methods. Chunk size is measured in phrases. For progressive methods, entries describe evaluation-stage updates and epochs are per update. LR denotes learning rate; WSD denotes warmup–stable–decay, and Linear denotes linear decay after warmup. Dropout applies to the model for full fine-tuning and to the adapter otherwise. Dashes indicate inapplicable settings.

Do Multilingual Encoders Produce Language-Consistent Semantic IDs?

Abhinav Bohra*

Amazon

bohraabhinav8@gmail.com

Anuj Bohra*

Rutgers University

anujbohra232003@gmail.com

Abstract

Semantic IDs (SIDs) compress item embeddings into discrete code sequences used in generative retrieval. We ask whether a multilingual encoder is sufficient for different-language renderings of the same product to receive language-consistent SIDs. Using Amazon ESCI listings rendered in English, Spanish, and Japanese, we test whether translations remain close to their English source, whether residual quantization is unusually sensitive to translation-induced movement, and whether multilingual or language-balanced quantizer fitting improves SID agreement. Multilingual E5 places translations measurably apart: under an English-heavy fit, a Japanese translation preserves the first SID code of its English counterpart in only 7.7% of cases, compared with 89.0% for an English rewording. Distance-matched product-directed controls produce nearly the same full-SID mismatch as translation, providing no evidence that the quantizer selectively amplifies language directions. Balancing the fitting mixture makes codebook use more uniform but further reduces cross-lingual prefix agreement: Spanish first-code consistency falls from 28.3% to 6.6%, while an English-only fit preserves it for 67.6% of Spanish translations. These results show that multilingual exposure and balanced codebook use alone do not guarantee language-consistent SIDs.

1 Introduction

Multilingual item catalogs are not naturally aligned parallel datasets. The same product may be represented differently across languages because descriptions are written, translated, or localized through independent processes. A Semantic ID (SID) computed from listing text is therefore an identifier for a particular textual representation, not automatically for the underlying product. This

distinction matters for multilingual retrieval and recommendation systems, where equivalent items must remain reachable across languages (Bonab et al., 2021).

SIDs are often treated as language agnostic when they come from a shared multilingual encoder. This assumption depends on two conditions: translations of the same product must remain close in encoder space, and whatever displacement remains must not push the product into a different discrete code prefix. The distinction matters because similarity in a continuous embedding space decreases gradually, whereas exact SID lookup does not: one changed code can send the item to a different branch of the tree.

We test three questions: whether translations remain close to the English source relative to same-language rewordings, whether the quantizer is unusually sensitive to translation-induced movement, and whether multilingual or balanced quantizer fitting improves cross-lingual SID agreement. We use product listings from the Amazon Shopping Queries Dataset (ESCI) (Reddy et al., 2022). Each listing is translated into Spanish and Japanese with NLLB (NLLB Team et al., 2022), while English paraphrases provide a same-language reference. We compare translation-induced displacement with distance-matched isotropic and product-directed controls, and repeat the English–Spanish analysis with Qwen2.5-3B-Instruct to reduce dependence on a single translation model.

Multilingual E5 already separates equivalent products by language. Residual quantization (Lee et al., 2022; Rajput et al., 2023; Vats et al., 2025a,b) turns this displacement into SID-prefix changes, but the product-directed control provides no evidence that the quantizer is especially sensitive to language directions. Under an English-heavy fit, E5 also assigns substantially less first-level capacity to Spanish and Japanese. A balanced mixture reduces this capacity gap but produces even lower

*Equal contribution.

prefix agreement across translations. LaBSE does not show the same capacity imbalance, indicating that this effect depends on the encoder–quantizer combination.

Together, these results show that encoder displacement, quantizer behavior, and balanced codebook use are distinct issues. Making codebook use more uniform across languages does not guarantee consistent SID prefixes, and multilingual exposure alone is insufficient for cross-lingual SID alignment.

2 Related Work

Residual quantization maps a continuous representation to a sequence of discrete codebook entries (Lee et al., 2022). In recommendation, TIGER (Rajput et al., 2023) used this idea to create semantic IDs for items and trained a sequence model to generate the ID of the next item.

Several recent papers approach multilingual alignment by changing how the identifier is built. MGR-CSC (Huang et al., 2025), for example, represents documents with multilingual keyword atoms and allows equivalent keywords to share atoms across languages. Parallel tokenizers (Al Kautsar and Koto, 2025) work at the subword level instead, aligning independently trained vocabularies so that translated words can receive the same token indices. Both approaches build alignment into the identifier itself. We study the unmodified pipeline, where the SID is whatever the quantizer produces from a multilingual embedding, because that is the configuration practitioners deploy when they assume the encoder handles languages for free.

The closest work on quantizer stability is DRQ (Wang et al., 2026), which studies how semantic IDs change when item embeddings are moved and uses code overlap and codebook capacity to evaluate the quantizer. We follow a similar idea, but we find that the direction of the movement matters. A random change can move an embedding by the same distance as a translation while still taking it through a region that is not typical of real product embeddings. We therefore add a product-directed control, where the movement follows a direction defined by another observed product embedding. This is useful because language representations are often concentrated in a narrow part of the embedding space (Ethayarajh, 2019; Gao et al., 2019). We also compare multilingual E5 with LaBSE (Feng et al., 2022), which is trained to place translations

close together in a shared embedding space.

3 Experimental Design

3.1 Dataset

We use ESCI because it provides a large real-world product-search catalog and includes US, Spanish, and Japanese locales, which motivate our three-language setting. However, ESCI does not provide a parallel catalog with a reliable cross-locale mapping for the same product. Comparing native locale listings would mix language with product identity, catalog composition, writing style, and field completeness. We therefore sample English listings and generate parallel Spanish and Japanese versions of the same product text.

We filter the US catalog by removing exact normalized title duplicates and obtain about 1M candidates. From these, we sample 20K products using seed 42. Each input contains the title, brand, color, and the first 120 words of the bullet points. The mean input length is 607 characters.

3.2 Translations and paraphrases

The main multilingual dataset uses NLLB-200-distilled-1.3B with greedy decoding for EN→ES and EN→JA. NLLB was run with a 128-token source and output cap. To measure the effect of source truncation, we reconstruct the exact truncated English input seen by NLLB and embed it as EN-TRUNC. To reduce dependence on a single translation model, we also translate the full 20K corpus with Qwen2.5-3B-Instruct; this replication focuses on English–Spanish.

English paraphrases provide a same-language reference. The light paraphrase applies sentence reordering and a small set of synonym substitutions. Another structural transformation was also tested but produces a smaller embedding displacement and is included only in the appendix. These rule-based paraphrases are controls, not substitutes for naturally written paraphrases.

3.3 Distance-matched controls

We use distance-matched controls to test whether translation-induced movement is unusually disruptive to SID assignment. If translation is uniquely amplified by the quantizer, it should produce more SID changes than equally distant realistic perturbations.

All source embeddings are L_2 -normalized. For source $\mathbf{x}$ and target cosine distance d , let $\theta =$

$\arccos(1 - d)$. The isotropic control samples a unit vector $\mathbf{u}$ orthogonal to $\mathbf{x}$:

$$\mathbf{x}_{\text{iso}} = \cos \theta \mathbf{x} + \sin \theta \mathbf{u}.$$

This control matches the translation distance but moves in an arbitrary direction in the embedding space.

For the product-directed control, we sample another product embedding $\mathbf{z}$ and project its direction onto the tangent space at $\mathbf{x}$:

$$\mathbf{v} = \frac{\mathbf{z} - (\mathbf{x}^\top \mathbf{z})\mathbf{x}}{\|\mathbf{z} - (\mathbf{x}^\top \mathbf{z})\mathbf{x}\|_2}, \quad \mathbf{x}_{\text{prod}} = \cos \theta \mathbf{x} + \sin \theta \mathbf{v}.$$

This construction does not guarantee that every intermediate point lies on the true data manifold, but its direction is defined by an observed product embedding and is therefore more realistic than an arbitrary isotropic direction.

3.4 Encoders and quantizer

The main encoder is multilingual-E5-large (Wang et al., 2024), a multilingual retrieval encoder, run in fp16 with 1024-dimensional outputs and the required passage: prefix. We use LaBSE (Feng et al., 2022), which is explicitly trained for cross-lingual alignment, as an independent encoder check. Embeddings are L_2 -normalized. The two embedding spaces differ sharply: E5 has a shared-offset proxy $\|\bar{\mathbf{x}}\|_2^2$ of 75.4% and an average random-pair cosine similarity of 0.754 ± 0.022 , compared with 32.6% and 0.321 for LaBSE.

The RQ-VAE uses three levels with 256 codes per level and a 256-dimensional latent space, Euclidean codebook assignment, and a cosine reconstruction loss. Inputs are centered before quantization and renormalized. Without centering and frequent dead-code recycling, the first level collapses to a single code (Appendix A). Remaining architecture and optimization settings are given in Appendix E.

3.5 Fitting mixtures

We use three fitting mixtures to test whether multilingual exposure and language balance improve SID consistency. Each quantizer is fitted on 16K products and evaluated on a 4K-product held-out set. Condition A is an English-only baseline. Condition B uses 80% English, 10% Spanish, and 10% Japanese, representing an English-heavy multilingual fit. Condition C uses equal proportions of all three languages to test whether balancing the fitting data improves cross-lingual agreement.

Product identities and counts remain fixed; only the rendering assigned during fitting changes. We train three seeds, 42, 123, and 789. The Qwen robustness arm uses separate English–Spanish mixtures of 100/0, 90/10, and 50/50; these are kept separate because they are not numerically identical to the three-language NLLB conditions.

3.6 Evaluation Metrics

We report paired cosine distance, five-fold cross-validated language-probe accuracy, Jaccard@10, full SID mismatch, and effective capacity:

$$K_{\text{eff}}(L) = 2^{H_L},$$

$$H_L = - \sum_c p_L(c) \log_2 p_L(c).$$

where $p_L(c)$ is the fraction of items assigned to code c at level L . Prefix consistency at depth d is the fraction of alternate renderings whose first d codes match those assigned to the corresponding English listing. This metric measures reachability under exact SID-prefix lookup; it is not an end-to-end generative-retrieval score.

4 Results

4.1 Encoder separation by language

	E5-large	LaBSE
Shared-offset proxy	75.4%	32.6%
Random-pair cosine	0.754 ± 0.022	0.321
$d(\text{EN}, \text{ES})$	0.081	0.145
$d(\text{EN}, \text{JA})$	0.113	0.165
ES / paraphrase	$7.4\times$	$13.7\times$
JA / paraphrase	$10.3\times$	$15.6\times$
Language probe	$99.9 \pm 0.1\%$	84.6%
Jaccard@10 (EN, ES)	0.41 ± 0.21	0.43 ± 0.21
Jaccard@10 (EN, JA)	0.33 ± 0.17	0.39 ± 0.19

Table 1: Encoder diagnostics on 20,000 parallel products. The table uses the original full-English source for comparability across encoders. Probe chance is 33.3%.

For E5, the Spanish and Japanese translations are 0.081 and 0.113 cosine distance from the English source, compared with 0.011 for the light English paraphrase. A linear classifier trained on the frozen embeddings reaches $99.9 \pm 0.1\%$ accuracy when predicting the language. The nearest-neighbor sets also change: Jaccard@10 is 0.41 for Spanish and 0.33 for Japanese. The translations therefore do not remain in the same local neighborhood as the English listing.

Language classification accuracy is lower for LaBSE, at 84.6%, but its translated pairs are not closer in cosine distance. The two measures capture different properties. Classification accuracy indicates whether language can be recovered from the embedding population, whereas paired distance measures whether two translations of the same product remain close.

4.2 Encoder displacement as prefix failure

Mix	Query translation	$d=1$	$d=2$	$d=3$
A	EN rewording	.901	.619	.396
A	ES	.676	.239	.081
A	JA	.643	.171	.037
B	EN rewording	.890	.605	.379
B	ES	.283	.055	.016
B	JA	.077	.002	.000
C	ES	.066	.017	.003
C	JA	.048	.009	.001

Table 2: E5 SID-prefix consistency on 4k held-out products, averaged over three seeds. A query succeeds at depth d when its first d codes match the SID assigned to the English listing.

Table 2 measures whether translations preserve the SID prefix assigned to the corresponding English listing. Under the 80/10/10 mixture, first code consistency is 89.0% for an English rewording, 28.3% for Spanish, and 7.7% for Japanese. Full SID agreement for Japanese is effectively zero.

This is where the continuous encoder shift becomes a discrete prefix failure. A translated embedding may remain close to the English embedding in cosine distance while crossing the boundary of the first code. An exact prefix lookup then follows a different branch, and the English-indexed product is no longer contained in that subtree.

4.3 Product-directed control

The isotropic control suggests stronger disruption from translation: full-SID mismatch is 0.981 versus 0.756 for Spanish and 0.995 versus 0.887 for Japanese. The product-directed control removes this gap, with 0.981 versus 0.976 for Spanish and 0.995 versus 0.998 for Japanese. We therefore find no evidence that the quantizer selectively amplifies language directions. Instead, product-supported movements cross code boundaries more often than isotropic ones. Because mismatch is already near saturation, small residual differences may remain hidden, but the stronger control does not support

Target distance	Translation	Isotropic	Product-directed
Spanish	.981	.756	.976
Japanese	.995	.887	.998

Table 3: Full SID mismatch under the E5 80/10/10 quantizer, using EN-TRUNC as the source. Both controls match each translation’s cosine distance. The product-directed control uses a direction defined by another observed product embedding.

the large effect suggested by the isotropic comparison.

4.4 Balanced fitting hurts prefix agreement

Encoder	Mix	$K_{\text{eff}}(\text{L1})$			Prefix@1	
		EN	ES	JA	ES	JA
E5	B	76	30	16	.283	.077
E5	C	49	42	42	.066	.048
LaBSE	B	101	91	90	–	–

Table 4: Effective first-level capacity on equal-size language pools and E5 first-code consistency with the English-indexed SID. Mixture B is 80/10/10; mixture C uses equal thirds.

Under the E5 80/10/10 mixture, effective first-level capacity is 76/30/16 for English, Spanish, and Japanese; equal-thirds fitting narrows this to 49/42/42. Prefix agreement, however, falls from 28.3% to 6.6% for Spanish and from 7.7% to 4.8% for Japanese. The English-only quantizer preserves the first code for 67.6% of Spanish and 64.3% of Japanese translations. Thus, balanced code use does not imply cross-lingual code agreement. Qwen shows the same trade-off: English–Spanish first-code agreement drops from 0.39 to 0.24 to 0.04 as the Spanish share of the fitting set increases (Appendix C).

5 Conclusion

Cross-lingual SID inconsistency begins in the encoder space, but exact-prefix lookup makes the effect much more severe. A modest change in cosine distance can alter the first code, after which the indexed product is no longer reachable through the same subtree. The product-directed control gives no evidence that the quantizer is unusually sensitive to language directions. However, the quantizer still reflects the geometry of its inputs. Under an English-heavy fitting mixture, E5 assigns substantially less effective first-level capacity to Spanish

and Japanese. Balancing the fitting mixture reduces this capacity gap, but it also lowers cross-lingual prefix agreement. By contrast, the English-only quantizer, despite seeing no non-English examples during fitting, preserves the first code for roughly two-thirds of the translations. These results suggest that multilingual exposure and balanced codebook use are not sufficient for SID alignment. Multilingual SID systems will likely need an objective that directly encourages parallel renderings to share assignments, rather than relying on a multilingual encoder or a balanced fitting corpus to produce that agreement indirectly.

Limitations

The Spanish and Japanese texts are machine translations rather than naturally written listings because ESCI does not provide reliable cross-locale matches for the same products. Consequently, the measured displacement may reflect both language differences and artifacts introduced by machine translation. Replication with Qwen reduces dependence on a single translation model but does not eliminate this confound. The study also covers only Spanish and Japanese as non-English languages, limiting broader generalization. Finally, prefix consistency measures exact SID reachability, not end-to-end retrieval quality; the detailed Qwen analysis is limited to English–Spanish.

Ethics Statement

The experiments use a public product-search dataset and contain no personal information. We measure differences between textual translations, not attributes of people. A failure to retrieve non-English listings could affect users in those locales, but this paper does not measure individual or population-level harm.

References

- Muhammad Dehan Al Kautsar and Fajri Koto. 2025. Parallel tokenizers: Rethinking vocabulary design for cross-lingual transfer. *arXiv preprint arXiv:2510.06128*.
- Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-market product recommendation. In *Proceedings of CIKM*.
- Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2. In *Proceedings of EMNLP-IJCNLP*.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of ACL*.
- Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.
- Yuxin Huang, Simeng Wu, Ran Song, Yan Xiang, Yantuan Xian, Shengxiang Gao, and Zhengtao Yu. 2025. Multilingual generative retrieval via cross-lingual semantic compression. In *Findings of EMNLP*.
- Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. 2022. Autoregressive image generation using residual quantization. In *Proceedings of CVPR*.
- NLLB Team et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Maheswaran Sathiamoorthy. 2023. Recommender systems with generative retrieval. In *Advances in Neural Information Processing Systems*.
- Chandan K. Reddy, Lluís Màrquez, Fran Valero, Nikhil Rao, Hugo Zaragoza, Sambaran Bandyopadhyay, Arnab Biswas, Anlu Xing, and Karthik Subbian. 2022. Shopping Queries Dataset: A large-scale ESCI benchmark for improving product search. *arXiv preprint arXiv:2206.06588*.
- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual E5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*.
- Xuesi Wang, Junjie Wang, Ziliang Wang, Weijie Bian, and Guanxing Zhang. 2026. Decoupled residual quantization for robust semantic IDs in recommendation. *arXiv preprint arXiv:2606.01844*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.
- Phil Wang. 2023. *vector-quantize-pytorch*. Software repository.
- Arpita Vats, Rahul Raja, Ashish Kattamuri, and Abhinav Bohra. 2025a. Quantum natural language processing: A comprehensive survey of models, architectures, and evaluation methods. *TechRxiv preprint*. doi:10.36227/techrxiv.176238247.72060075/v1.

Arpita Vats, Rahul Raja, Ashish Kattamuri, and Abhinav Bohra. 2025b. A systematic survey of quantum natural language processing: Models, encoding paradigms, and evaluation methods. In *Proceedings of the QuantumNLP: Integrating Quantum Computing with Natural Language Processing*, pages 53–64. Association for Computational Linguistics. doi:10.18653/v1/2025.quantumnlp-1.8.

A Level-1 Collapse and Mitigation

Our initial RQ-VAE runs assigned every item to a single first-level code. Later levels used roughly 45 codes each, reducing the observed ID space from 256^3 to approximately $45^2 \approx 2,000$ tuples. The training loss did not clearly reveal this failure, but the resulting first-level drift and utilization measurements were not meaningful.

Variant	Change	L1 alive	K_{eff}	Collision
V0	baseline	1	1	21%
V1	add centering	22	20	1.9%
V2	RVQ library, cosine	256	217	50%
V5	final configuration	256	102	4.3%
V7	reset every 2 epochs	8	6	99%

Table 5: Representative collapse ablations on 20,000 items. V5 is used in the main experiments.

The dominant shared direction in the E5 embeddings places the raw vectors in a narrow cone. Centering spreads the inputs presented to the quantizer, while per-step dead-code recycling prevents one early code from dominating the first level. Using cosine distance for codebook assignment kept more codes active, but it also produced highly correlated levels and a large full SID collision rate at this scale. The final configuration therefore uses Euclidean assignment with a cosine reconstruction loss. These choices address collapse in the present setting; they are not intended as evidence that the same configuration is optimal for larger catalogs.

B Low Displacement Surface Perturbations

The rule based Spanish and Japanese perturbations move the E5 embeddings by 0.0064 and 0.0239, compared with 0.081 and 0.113 for the NLLB translations. They produce little or no capacity asymmetry and substantially understate the cross-lingual separation observed with NLLB. Their full SID mismatch remains higher than that of the isotropic control, although neither condition is close to saturation.

Mix	Rendering	Spanish			Japanese		
		Trans.	Iso.	Δ	Trans.	Iso.	Δ
B	surface	42.8	24.7	+18.1	77.7	47.2	+30.5
C	surface	46.9	24.7	+22.2	78.6	47.6	+31.0
B	NLLB	99.0	82.0	+17.0	99.8	91.4	+8.4
C	NLLB	99.5	86.1	+13.4	99.9	94.0	+5.9

Table 6: full SID mismatch against the original isotropic control. These numbers are retained as measurements, not as evidence of language-selective quantizer amplification.

These results show why the realism of a displacement control must be considered before interpreting its margin. The isotropic comparison alone cannot support a language specific causal claim, because the product-directed control in Table 3 removes nearly all of the observed difference.

C Qwen Full-Corpus Replication

Qwen2.5-3B-Instruct translated the same 20,000-product corpus used in the main study. The reported downstream comparison uses English and Spanish with separate two-language mixtures: A_Q is 100/0, B_Q is 90/10, and C_Q is 50/50. The NLLB values in Table 7 are recomputed under the same binary mixtures and should not be compared directly with the three-language NLLB conditions in Table 2.

Metric	Qwen	NLLB
EN–ES distance	.0807	.0760
Distance / paraphrase	9.9×	9.3×
Jaccard@10	.453	.487
$K_{\text{eff}}(\text{L1}), B_Q: \text{EN/ES}$	50/11	49/17
Prefix@1 ES, $A_Q/B_Q/C_Q$	.39/.24/.04	.34/.13/.003
Prefix@1 EN rewording	.78–.83	.76–.83

Table 7: Full-corpus English–Spanish translator replication. The two-language mixtures are defined above and are separate from the main NLLB experiment.

Both translators exceed their isotropic control. Against the product-directed control, the remaining margins are only 0.001–0.006 at a mismatch rate near 0.99. We do not treat those differences as evidence of selective amplification.

D Direction and Category Checks

Back-translation produces high mismatch in both directions. In the reverse setting, however, the Spanish and Japanese source texts are themselves machine translations, and the audit uses only the isotropic control. We therefore use this experiment

only to rule out a simple one-way penalty against non-English inputs.

Direction	Translation	Isotropic	Margin
EN→ES	.994	.794	+.200
ES→EN	.989	.717	+.272
EN→JA	.996	.895	+.100
JA→EN	.993	.902	+.090

Table 8: Direction reversal under the E5 80/10/10 quantizer. The product-directed interpretation from the main paper also applies here.

We also divide the held-out English embeddings into 20 k -means clusters. Under mixture B, Spanish full SID mismatch ranges from 0.968 to 0.997 across clusters, with a standard deviation of 0.008. Japanese mismatch ranges from 0.998 to 1.000. The effect is therefore not limited to a single coarse product group in the embedding space.

E Compute and Reproducibility

The RQ-VAE encoder is `Linear(1024, 1024)`, `GELU`, `LayerNorm`, `Linear(1024, 256)`, and the decoder mirrors it. The objective adds a commitment term weighted by $\beta = 0.25$ to the cosine reconstruction loss. Each model trains for 20 epochs with batch size 1024 using Adam at a learning rate of 10^{-3} .

All experiments ran on a single NVIDIA L4 GPU with 24 GB of memory. NLLB translation of 20,000 products into Spanish and Japanese took approximately 20 minutes and was cached. With translations and embeddings cached, the main quantizer-training and analysis suite completed in approximately 3.3 minutes. The implementation uses PyTorch, NumPy, sentence-transformers, vector-quantize-pytorch, CTranslate2, Transformers, scikit-learn, and pandas.

When a Name Is Not a Name: A Benchmark Dataset and Distilled Reasoning for Culturally Entangled Bangla Homographs in Low-Resource LLMs

Md. Asaduzzaman Shuvo, Md. Adnanul Islam

United International University, Bangladesh

Emails: (ashuvo221104, mislam221096)@bscse.uui.ac.bd

Abstract

Many Bangla words are at once personal names and culturally loaded common nouns: মায়ী (*Maya*) is both a girl’s name and a word for affectionate compassion. Choosing the right reading demands cultural knowledge that is scarce in the pretraining data of modern language models. We introduce **Bangla Culturally Entangled Homograph (BanglaCEH)** disambiguation and build a Bangla benchmark of 1,516 expert-verified sentences and 3,032 labelled occurrences in which one word appears twice with two distinct readings, each labelled with a culturally grounded category and an explanation of the reasoning behind it. Across open- and closed-source models, we find a systematic dominant-meaning bias, models default to the common-noun sense and overlook the name. A Bangla-specific model fails under every prompting regime we test, showing that language-specific pretraining alone does not confer cultural grounding. We further show that contrastive chain-of-thought prompting can sharply reduce this bias without training, and that distilling cultural explanations teaches small (1–3B) models to reason toward the correct reading rather than memorise labels, cutting dominant-meaning bias from as high as 100% to under 5% and turning the failed Bangla-specific model into our strongest system.

1 Introduction

Word-sense ambiguity is a long-standing challenge in NLP, but a particularly difficult and understudied variant arises when a single word functions simultaneously as a personal name and as a culturally loaded common noun. In Bangla this is pervasive: parents routinely name children after words denoting prized emotions, virtues, or spiritual states, so one surface form carries both an individuating (name) reading and an abstract (concept) reading. The word মায়ী (*Maya*) is at once

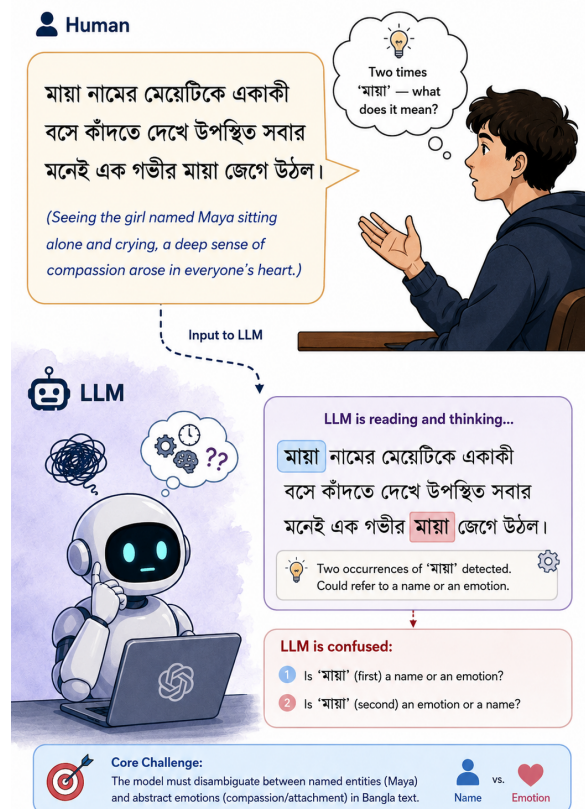


Figure 1: The Bangla Culturally Entangled Homograph (BanglaCEH) task. মায়ী (*Maya*) appears twice in one Bangla sentence first as a name, then as the emotion it is named after and the model must disambiguate the two using cultural knowledge alone.

a common girl’s name and a word for deep affectionate compassion; আরিফ (*Arif*) is both a boy’s name and, in Sufi theology, one who possesses intuitive knowledge of the divine. Resolving the intended reading demands cultural and pragmatic knowledge beyond surface lexical statistics.

We term this phenomenon the Bangla Culturally Entangled Homograph (BanglaCEH) and argue it is a sharp diagnostic of genuine cultural grounding. Unlike conventional word-sense disambiguation, a BanglaCEH cannot be resolved by fre-

quency or distributional cues: both readings coexist in the same cultural space, and disambiguation hinges on recognising when a form names a person versus invokes the concept it drawn from making BanglaCEH especially demanding for models trained on high-resource, Western-centric corpora.

LLMs are known to encode a Western-dominance bias and to struggle with low-resource cultures (Belay et al., 2025; Yu et al., 2025); language-specific competence does not entail cultural competence, and in-language prompting alone does not supply cultural context (Belay et al., 2025). For Bangla, recent work has produced capable models and benchmarks (Nahin et al., 2025; Raihan and Zampieri, 2025; Joy and Shatabda, 2026), but evaluation has centred on knowledge, reasoning, and classification, leaving cultural grounding in everyday lexical usage largely unexamined. Related studies show that fluent Bangla output can mask systematic register failures (Shuvo et al., 2026) and persistent hallucination (Adib et al., 2026) symptoms of shallow rather than grounded understanding.

We introduce the BanglaCEH task and build a benchmark of Bangla sentences in which one word appears twice with two distinct readings, each annotated with one of six culturally grounded categories and an explanation of the underlying cultural reasoning. Evaluating open and closed-source models, we uncover a systematic dominant-meaning bias: models default to the common-noun reading and overlook the personal-name reading (Figure 1). A Bangla-specific model fails under every regime we test, confirming that language-specific pretraining alone does not confer cultural grounding. We further show that contrastive chain-of-thought prompting can sharply reduce this bias without training, though its effect is strongly model-dependent, and that distilling cultural reasoning into small models teaches them to reason toward the correct reading rather than memorise labels.

Contributions:¹ (i) We formalise BanglaCEH disambiguation, a task isolating cultural grounding from surface lexical competence in a low-resource language. (ii) We release a BanglaCEH benchmark with token-level cultural categories and cultural-reasoning explanations. (iii) We evaluate open- and closed-source models under zero-

shot, few-shot, contrastive chain-of-thought, and knowledge-distillation regimes, revealing a consistent dominant-meaning bias and the failure of a Bangla-specific model across all settings. (iv) We show contrastive reasoning and distilled supervision mitigate this bias, and analyse when and why they succeed or fail.

2 Related Work

Word-sense disambiguation (WSD) is a longstanding NLP problem (Navigli and Ponzetto, 2010; Raganato et al., 2017), and remains especially hard for morphologically rich, low-resource languages where annotated corpora are scarce (Habtamu and Gizachew, 2024; Masethe et al., 2024); recent work has extended WSD evaluation to under-resourced settings through hybrid sense annotation (Goworek et al., 2025) and dedicated Arabic and Urdu resources (Khalilia et al., 2024; Saeed et al., 2019), while studies probing whether large language models truly grasp word senses report that surface fluency often masks shallow lexical understanding (Meconi et al., 2025; Ortega-Martín et al., 2023). Our BanglaCEH task differs from classical WSD in that the competing readings a personal name versus the culturally loaded concept it derives from are not lexically distinct senses but culturally entangled ones, requiring pragmatic and cultural knowledge rather than dictionary sense inventories. This connects our work to a rapidly growing literature on the cultural competence of LLMs, which finds that models encode a Western-dominance bias and underperform on the values and everyday knowledge of low-resource cultures (Belay et al., 2025; Yu et al., 2025), and that language-specific ability does not guarantee cultural grounding (Belay et al., 2025). For Bangla specifically, recent efforts have produced dedicated language models and knowledge benchmarks (Nahin et al., 2025; Raihan and Zampieri, 2025; Joy and Shatabda, 2026), yet cultural grounding in everyday lexical usage remains underexplored, and fluent Bangla generation can still hide systematic register and factuality failures (Shuvo et al., 2026; Adib et al., 2026). Finally, our mitigation strategy builds on chain-of-thought prompting (Wei et al., 2022) and on knowledge distillation of reasoning into smaller models (Magister et al., 2023; Hsieh et al., 2023), where a teacher model’s rationales are distilled into a compact student; unlike prior distillation work that tar-

¹Dataset and code are available at <https://github.com/ashuvo25/BanglaCEH>

gets mathematical or commonsense reasoning, we distil cultural reasoning, teaching small models to justify why a given occurrence is a name or a concept rather than merely to reproduce the label.

3 Dataset Construction

Task formulation: Each instance in our benchmark is a single Bangla sentence in which one word form appears twice with two distinct readings. Given the sentence and the target word, a model must assign each of the two occurrences one of six culturally grounded categories: Name, Concept, Emotion, Emotional State, Collective State, and Spiritual, and justify each assignment. Crucially, the two readings are not arbitrary lexical senses but culturally entangled ones: the same form functions once as a personal name and once as the emotion, concept, or spiritual state from which that name is conventionally drawn.

Word selection: We do not rely on a fixed name registry or lexical database. Instead, a native Bangla speaker manually curated words such as ময়া (Maya), আশা (Asha), and গগন (Gagan) that are both common Bangla given names and culturally loaded common nouns. This expert-driven selection deliberately targets forms with a genuine dual reading, which a generic name list would not isolate, at the cost of coverage reflecting a single annotator’s naming knowledge Section (7).

Multi-model generation: For each selected word, we generated sentences and accompanying cultural explanations using three state-of-the-art models as complementary generators: Claude Table 5, Kimi K3, and Gemini 3.5 Flash, producing 690, 480, and 346 instances respectively. Multiple generators reduce the stylistic and distributional bias of a single teacher and yield greater lexical and structural diversity.

Cross-model and human verification: We applied a two-stage verification protocol. First, outputs were cross-checked round-robin by a different model (Claude verified by Kimi, Kimi by Gemini, Gemini by Claude), flagging inconsistent labels or explanations without any model auditing its own output. Second, every instance was manually reviewed by two native Bangla-speakers (profile in appendix A), who corrected labels, explanations, and culturally inaccurate reasoning. Human verification is the final authority; the cross-model stage only surfaces candidate errors for the reviewer.

Annotations: Each verified instance contains

Word	গগন (<i>Gagan</i>)
Category	Name ↔ Concept
Sentence	সমুদ্রসৈকতে সূর্যাস্ত দেখতে দেখতে গগন এর মনে হলো, প্রাচীন শ্লোকে 'গগন' দিয়ে ঠিক এই আকাশ; নভোমণ্ডল-কেই বোঝানো হতো।
Gloss	<i>Watching the sunset, it struck Gagan that ancient verse used 'Gagan' for exactly this the celestial firmament.</i>
Occ. 1	Name a person, the grammatical subject
Occ. 2	Concept the celestial firmament
Cultural note	Bengali naming turns words for nature, deities, and virtues into personal names; গগন is thus once a living identity and once a cultural concept.

Table 1: An example BanglaCEH instance. The same form গগন (*Gagan*) appears twice with a name reading and a concept reading. The full bilingual schema is in Appendix C.

Category	Count	Label	Count
Name ↔ Concept	795	Name	1,152
Concept ↔ Emotion	193	Concept	988
Name ↔ Spiritual	179	E+C State	171
Name ↔ Emotion	178	Emotion	542
Emotion ↔ State	171	Spiritual	179
Instances	1,516	Labels	3,032

Table 2: BanglaCEH statistics entanglement categories (left) and token-level label count (right). E = Emotional, C = Collective

token-level labels for both occurrences, a natural-language explanation of why each occurrence carries its label, and a concise cultural-entanglement note describing the naming convention that links the two readings. These explanations form the supervision signal for our knowledge-distillation experiments Section (4). An example instance is shown in Table 1; the complete field schema and a raw JSON record are provided in Appendix C.

Statistics: The final benchmark contains 1,516 verified instances, each contributing two labelled occurrences. Table 2 reports the distribution over entanglement categories and over the six labels. Name and Concept are the most frequent labels, reflecting the prominence of name–concept entanglement in Bangla naming practice, while Emotional state and Spiritual form a rarer long tail that we find models handle least reliably in Section (5).

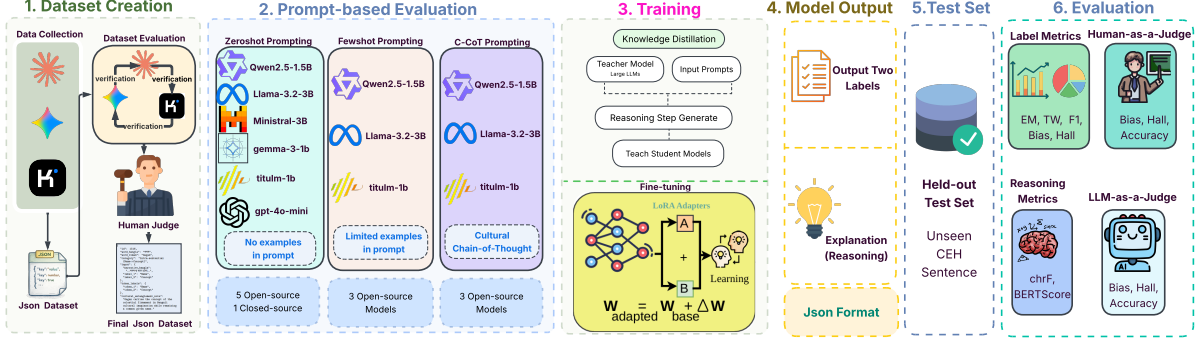


Figure 2: Overview of our methodology, spanning dataset creation, prompt-based evaluation (zero-shot, few-shot, C-CoT), knowledge-distillation fine-tuning with LoRA, and evaluation on held-out BanglaCEH sentences using automatic, human, and LLM-based judges.

4 Methodology

We evaluate the BanglaCEH task under four regimes of increasing supervision: zero-shot prompting, few-shot prompting, Cultural Chain-of-Thought (C-CoT) prompting, and knowledge-distillation (KD) fine-tuning. The first two establish how much cultural grounding models possess without task-specific adaptation; the latter two are our proposed interventions for mitigating the dominant-meaning bias. All regimes are evaluated on the same held-out test set to enable direct comparison. We split the benchmark into 78% training, 10% validation, and 12% test partitions with a fixed seed. Target words are disjoint across partitions. Figure 2 provides an overview of the full pipeline.

4.1 Prompting Baselines

Zero-shot: The model receives the sentence and an instruction to assign each of the two occurrences of the target word a label from the six categories, with no examples. This measures a model’s out-of-the-box cultural grounding. **Few-shot:** We prepend a small number of labelled in-context examples before the query, testing whether demonstrations alone without weight updates help the model recognise the entangled readings.

4.2 Cultural Chain-of-Thought

Standard chain-of-thought prompting elicits intermediate reasoning before a final answer (Wei et al., 2022), but it does not counteract a model’s prior tendency to collapse both occurrences onto the dominant sense. We introduce **Cultural Chain-of-Thought (C-CoT)** (Appendix D), a contrastive variant that explicitly forces the model to argue

both candidate readings for each occurrence before committing. For every occurrence, the model must (i) state why the form could be a Name, (ii) state why it could be the competing cultural category (Concept, Emotion, Spiritual, or State), and (iii) decide which reading the sentential and cultural context supports. By requiring the suppressed reading to be articulated before a decision is made, C-CoT directly targets the dominant-meaning bias rather than merely eliciting free-form rationale. C-CoT requires no training and is applied at inference time.

4.3 KD Fine-Tuning

Our final regime tests whether the cultural reasoning behind each label can be taught to a small model rather than merely prompted. We fine-tune compact open-source models on the human-verified cultural explanations in our benchmark, following a knowledge-distillation formulation in which a student model learns to reproduce the reasoning traces of stronger teacher models (Hsieh et al., 2023; Magister et al., 2023). Crucially, the training target is the full cultural explanation followed by the label, not the label alone: the model first generates the reasoning that justifies each occurrence’s reading and only then emits the final labels. This encourages the model to internalise the pattern of cultural reasoning rather than memorise a surface mapping from word to label. We use parameter-efficient fine-tuning via Low-Rank Adaptation (Hu et al., 2021) in its quantized form, QLoRA (Dettmers et al., 2023), which back-propagates gradients through a frozen 4-bit base model into low-rank adapters, enabling training of billion-parameter models on a single consumer

GPU. Loss is computed only over the target reasoning and label tokens, with the prompt masked, so that supervision is concentrated on the cultural reasoning the model must learn to produce. We select the checkpoint with the lowest validation loss and apply early stopping to prevent overfitting.

4.4 Models

We evaluate a mix of open- and closed-source models spanning a range of parameter scales. The open-source set comprises Qwen2.5-1.5B, Llama-3.2-3B, Gemma-3-1B, Ministral-3B, and the Bangla-specific TituLLM; the last is included specifically to test whether language-targeted pretraining confers cultural grounding. We include GPT-4o-mini as a strong closed-source reference. The prompting baselines are evaluated across all models, while C-CoT and knowledge-distillation (KD) fine-tuning are applied to a representative subset spanning small general-purpose and Bangla-specific models.

4.5 Evaluation Metrics

We evaluate along two tracks. The label track measures disambiguation accuracy through Exact Match (both occurrences correct), Average Score (Average Score is the percentage of total points earned over the maximum possible points, see Appendix(F.1)), together with Macro- and Micro-averaged F1 over the six labels. We additionally report a Hallucination rate the fraction of outputs producing a label outside the valid set and, as our central diagnostic, a Dominant-Bias rate measuring how often a model misses a Name label and defaults to the entangled cultural reading. The reasoning track evaluates the quality of generated cultural explanations against the gold explanations using chrF (Popović, 2015) and BERTScore (Zhang et al., 2019). In addition, we assess outputs through a Human-as-a-Judge and LLM-as-a-Judge evaluation.

5 Result Analysis

Table 3 reports performance across all four regimes. We organise our analysis around four findings: the severity of the dominant-meaning bias under prompting, the model-dependent behaviour of C-CoT, the decisive effect of KD fine-tuning, and the striking reversal of the Bangla-specific model.

5.1 Zero-Shot Prompting

Under zero-shot prompting, every open-source model shows a pronounced dominant-meaning bias, defaulting to the common-noun reading and misclassifying the personal-name occurrence. Bias ranges from 62.5% (Ministral) to 100% (Gemma, TituLLM), with exact match below 11% for all open models. The failure modes differ: Ministral and Llama produce valid but wrong labels, while others collapse entirely TituLLM hallucinates on 100% of inputs and Gemma shows complete (100%) bias, indicating the name reading is not even represented as a candidate. The closed-source GPT-4o-mini is markedly stronger 43.87% exact match, 0.65% hallucination yet retains a non-trivial 16.35% bias. This open–closed gap holds across every metric, confirming BanglaCEH disambiguation is genuinely hard: even a capable proprietary model without task-specific adaptation misreads the name as the entangled concept in roughly one in six cases. Uniformly low Macro-F1 (0.005–0.467) further shows zero-shot failure is disproportionate on rarer cultural labels, not merely the frequent name/concept pairing.

5.2 Few-Shot Prompting

Adding a few in-context examples improves most models over zero-shot, but the gains are inconsistent and leave the core bias largely unresolved. Gemma benefits most dramatically, rising from 0.91% to 31.29% exact match as its dominant-meaning bias falls from 100% to 39.5% evidence that demonstrations can teach a model to represent the name reading it previously ignored. Qwen and Llama show more modest gains (exact match 0.65%→8.39% and 10.97%→22.58%), with bias dropping into the low-20% range for both. Two models resist this trend: Ministral degrades under few-shot prompting exact match falls from 9.0% to 5.8% with bias barely moving (62.5%→60.6%), the demonstrations disrupting rather than guiding its predictions while TituLLM remains effectively non-functional, hallucinating on 100% of inputs in both regimes despite a nominal rise to 0.31% exact match, confirming its failure is one of task representation rather than a shortage of examples. GPT-4o-mini again leads, reaching 49.68% exact match with bias reduced to 2.88%, its strongest prompting result. Overall, few-shot prompting narrows but does not close the gap: even the best open-source configuration (Gemma, 31.29%) trails GPT-

		EM↑	Avg↑	Ma-F1↑	Mi-F1↑	chrF↑	BERT↑	Hall.↓	Bias↓
<i>Open-source</i>									
Qwen2.5-1.5B	Zero-shot	0.65	22.90	0.115	0.229	1.96	0.291	52.90	63.46
	Few-shot	8.39	36.45	0.227	0.365	17.70	0.301	54.84	23.08
	C-CoT	30.97	52.26	0.300	0.523	24.30	0.178	0.00	2.88
	QLoRA-KD	85.16	92.58	0.737	0.926	79.35	0.784	1.65	0.00
Llama-3.2-3B	Zero-shot	10.97	20.97	0.144	0.210	1.71	0.295	40.00	65.38
	Few-shot	22.58	40.00	0.239	0.400	16.20	0.289	34.19	22.12
	C-CoT	15.16	15.81	0.153	0.158	27.80	0.144	43.23	75.00
	QLoRA-KD	82.42	90.38	0.722	0.904	77.04	0.760	1.10	2.80
Ministral-3B	Zero-shot	9.00	22.60	0.158	0.226	2.19	0.251	41.30	62.50
	Few-shot	5.80	20.30	0.126	0.203	10.92	0.270	30.30	60.60
gemma-3-1b-it	Zero-shot	0.91	13.23	0.079	0.132	10.01	0.072	16.13	100.0
	Few-shot	31.29	54.19	0.339	0.542	17.59	0.285	7.74	39.50
titulum-1b	Zero-shot	0.11	2.33	0.005	0.006	2.33	0.032	100.0	100.0
	Few-shot	0.31	33.55	0.115	0.335	5.77	0.037	100.0	40.22
	C-CoT	1.14	2.77	0.012	0.014	13.23	0.048	89.12	92.13
	QLoRA-KD	87.91	92.58	0.758	0.926	79.50	0.790	1.65	4.20
<i>Closed-source</i>									
GPT-4o-mini	Zero-shot	43.87	64.19	0.467	0.642	19.10	0.299	0.65	16.35
	Few-shot	49.68	72.26	0.623	0.723	32.60	0.296	0.00	2.88

Table 3: BanglaCEH results across four regimes. ↑ higher is better; ↓ lower is better. EM = Exact Match, Avg = Average Score, Ma-F1 / Mi-F1 = Macro / Micro F1, BERT = BERTScore-F1, Hall. = Hallucination, Bias = Dominant-Bias (all %). C-CoT and QLoRA-KD were run on closed-source prompting reference. For each model, the best value per metric is in **bold**.

4o-mini by a wide margin, and dominant-meaning bias remains substantial for every open model motivating the reasoning- and training-based interventions we examine next.

5.3 Cultural Chain-of-Thought

C-CoT produces the largest prompting-time improvement we observe, but only for models able to follow its contrastive structure. On Qwen2.5-1.5B, it drives dominant-meaning bias from 63.46% (zero-shot) to 2.88%, eliminates hallucination entirely (0%), and raises exact match to 30.97% a level no other prompting configuration reaches for this model. Forcing the model to articulate the suppressed name-reading before committing counteracts the bias without parameter updates, supporting our hypothesis that the dominant-meaning failure is one of reasoning order rather than missing knowledge. The benefit does not transfer uniformly: on Llama-3.2-3B, C-CoT is counterproductive dominant-meaning bias rises to 75% and hallucination climbs to 43.23%, both worse than the zero- and few-shot baselines. The degradation coincides with a sharp rise in unparseable outputs: Llama’s extended contrastive reasoning often fails to terminate in the required label format, so otherwise valid predictions are discarded. C-

CoT’s effectiveness is thus contingent on sustaining format adherence over long reasoning chains an important limitation for smaller and lower-resource models. TituLLM remains largely non-functional under C-CoT (89.12% hallucination, 1.14% exact match); contrastive reasoning cannot compensate for a model that does not reliably represent the task. Together, these results show C-CoT can nearly eliminate dominant-meaning bias when a model can execute it, but not a universal remedy motivating the training-based approach we examine next.

5.4 QLoRA-KD Fine-Tuning

The QLoRA-KD regime yields the strongest results by a wide margin. All three fine-tuned models exceed 82% exact match and reduce dominant-meaning bias to at most 4.2% an order-of-magnitude improvement over their best prompting configuration. Qwen2.5-1.5B reaches 85.16% exact match with 0% dominant-meaning bias and 1.65% wrong predictions, while Llama-3.2-3B, which C-CoT had degraded, recovers to 82.42% exact match and 2.8% bias. This confirms that the dominant-meaning failure is not an inherent limitation of these models but a gap that targeted supervision can close. Crucially, the improve-

ment extends to reasoning quality, not merely label accuracy. For Qwen, BERTScore rises from 0.178 under C-CoT to 0.784 under QLoRA-KD and chrF from 24.3 to 79.35, indicating that the fine-tuned model generates cultural explanations closely aligned with the gold reasoning rather than producing correct labels for spurious reasons. Because the distillation target pairs each label with its cultural justification, the model learns the pattern of cultural reasoning that licenses a reading rather than a surface word-to-label mapping precisely the capability that prompting failed to elicit.

These gains are obtained with parameter-efficient adaptation on models of only 1–3B parameters. Notably, Qwen2.5 fine-tuned model surpasses the strongest closed-source prompting baseline (GPT-4o-mini few-shot, 49.68% exact match, 2.88% bias) by a large margin. We stress that this is not a like-for-like comparison GPT-4o-mini is evaluated only under prompting, not fine-tuned but it establishes that a small, openly available model, once taught cultural reasoning, can substantially outperform a much larger proprietary model prompted on the same task. The residual gap between Micro- and Macro-F1 (0.926 vs. 0.737 for Qwen) indicates that errors now concentrate almost entirely in the rarer cultural labels, which we identify as the primary remaining challenge (full per-label results and confusion matrices in Appendix H). An ablation study is provided in Appendix B.1, confirming the importance of reasoning supervision.

5.5 Human & LLM Judge Evaluation

Model	Human		GPT-5.4		$\Delta_{\max}$
	H.↓	B.↓	H.↓	B.↓	
Qwen2.5-1.5B	1.7	0.0	1.6	0.0	0.1
Llama-3.2-3B	1.1	2.5	1.1	2.8	0.3
titilm-1b	1.7	4.0	1.7	4.2	0.2

Table 4: Human and LLM (GPT-5.4) judge evaluation of the fine-tuned (QLoRA-KD) models. H. = hallucination rate (%), B. = dominant-meaning bias (%); ↓ = lower is better; $\Delta_{\max}$ denotes the maximum absolute disagreement between judges.

To verify that the fine-tuned models reach correct labels through culturally sound reasoning, we evaluate their outputs with two independent judging stages: two human judges (native Bangla speakers) and an LLM judge, GPT-5.4 (the prompt template is provided in Appendix E). Each

judge scores every output for hallucination and dominant-meaning bias; results are reported in Table 4. The two judging stages agree closely, differing by at most 0.3 percentage points on any metric, which both corroborates our automatic evaluation and supports the use of an LLM judge as a scalable proxy for human assessment. Consistent with the automatic results, all three fine-tuned models exhibit near-zero hallucination (1.1%–1.7%) and minimal dominant-meaning bias (0.0%–4.2%) under both judges, confirming that their post-distillation predictions are not only accurate but also judged culturally faithful by human experts.

6 Discussion

Our results separate two mechanisms for resolving culturally entangled homographs: surfacing knowledge already latent in a model’s parameters, and acquiring knowledge it lacks. Prompting, including contrastive C-CoT, engages only the former it helps only insofar as cultural grounding is already present and can be reorganised through reasoning order, which is why C-CoT nearly eliminates the dominant-meaning bias on Qwen yet leaves TituLLM, which lacks such grounding, effectively unchanged. Knowledge-distillation fine-tuning engages the latter: supervising on cultural explanations rather than labels alone induces the reasoning that connects a name to its source concept, so predictions follow from reasoning rather than memorised surface mappings. Under this regime accuracy and reasoning quality improve together rather than in isolation the models do not merely predict the right label but justify it with culturally faithful explanations, a coupling our human and LLM judges independently corroborate. These gains proved sensitive to how the adaptation was configured. Because the benchmark is small relative to adapter capacity, an initial higher-capacity setting overfit within a few epochs; reducing the adapter rank and learning rate and strengthening regularisation delayed overfitting and lowered the validation minimum, and combined with early stopping this configuration transferred unchanged across all three models, indicating that the recipe is not narrowly tuned to a single architecture (Appendix G).

The most consequential result is the reversal of the Bangla-specific model: non-functional under every prompting regime, yet best-in-class after distillation. This suggests that monolingual

pretraining supplies a useful substrate more faithful tokenisation of Bangla script and denser exposure to the relevant names and morphology but not task competence itself; the substrate instead allows limited reasoning supervision can be absorbed far more efficiently than by a general model with a weaker lexical prior. For low-resource, culturally rich languages that lack large instruction-tuned systems, this points to a reproducible recipe: pair a compact language-specific model with a small set of cultural reasoning traces and adapt it with parameter-efficient fine-tuning. Although our experiments focus on Bangla, the proposed framework is language-agnostic and could be evaluated on other languages with similar name–concept entanglement.

Two limitations qualify these conclusions. First, the benefit of C-CoT is model-dependent: it helps only when a model can sustain the required output format across long reasoning chains, and degrades performance where it cannot, as observed for Llama. Second, the persistent gap between Micro- and Macro-averaged F1 after fine-tuning shows that the rarer cultural labels remain the hardest, locating the current bottleneck in long-tail data scarcity rather than in the method, and marking targeted expansion of these categories as the clearest direction for future work.

7 Conclusion

We introduced Bangla Culturally Entangled Homograph (BanglaCEH) disambiguation, a task in which the same Bangla word must be read once as a personal name and once as the cultural concept it is drawn from, and released a benchmark of annotated instances paired with cultural-reasoning explanations. Across open- and closed-source models we found a systematic dominant-meaning bias models default to the common-noun sense and overlook the name and showed that language-specific pretraining alone does not fix it: the Bangla-specific model failed under every prompting regime. Yet the same model became the strongest system once fine-tuned on distilled cultural reasoning, reaching near-perfect disambiguation with the bias eliminated. Our results suggest that cultural grounding is not conferred by monolingual pretraining alone, but can be improved through supervision with reasoning-annotated examples. This provides a practical approach for improving cultural disambiguation in low-resource

languages. Evaluating its applicability to other languages with similar name–concept entanglement remains an important direction for future work.

Limitations

Our benchmark has several limitations. First, word selection was performed manually by a single native Bangla speaker rather than drawn from a name registry or lexical database; while this deliberately targets forms with a genuine dual reading, it means coverage reflects one annotator’s naming knowledge and may under-represent regional or less common names. Second, the sentences and cultural explanations were initially produced by large language models and subsequently verified by two native Bangla-speaking experts. Although every instance was manually reviewed, model-generated text may still exhibit subtle stylistic regularities that a fully human-authored corpus would not. Third, although every instance was independently reviewed by two native Bangla-speaking annotators and the resulting inter-annotator agreement was high (Cohen’s $\kappa = 0.95$), the annotation process still relied on a relatively small annotator pool. While disagreements were resolved through discussion using a shared annotation guideline, involving more annotators from diverse regional and linguistic backgrounds would further improve the robustness and generalizability of the benchmark. Finally, our fine-tuning experiments use parameter-efficient adaptation on small models under limited compute, so the reported gains may not transfer directly to larger models or full-parameter fine-tuning.

References

- Shefayat E Shams Adib, Ahmed Alfey Sani, Ekramul Alam Esham, Ajwad Abrar, Ishmam Tashdeed, and Md Taukir Azam Chowdhury. 2026. Benhal-lueval: A multi-task hallucination evaluation framework for large language models on bengali. *arXiv preprint arXiv:2605.31483*.
- Tadesse Destaw Belay, Ahmed Haj Ahmed, Alvin C Grissom II, Iqra Ameer, Grigori Sidorov, Olga Kolesnikova, and Seid Muhie Yimam. 2025. Culemo: Cultural lenses on emotion-benchmarking llms for cross-cultural emotion understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18894–18909.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: Efficient finetuning](#)

- of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.
- Roksana Goworek, Harpal Singh Karlcut, Hamza Shezad, Nijaguna Darshana, Abhishek Mane, Syam Bondada, Raghav Sikka, Ulvi Mammadov, Rauf Allahverdiyev, Sriram Satkirti Purighella, Paridhi Gupta, Muhinyia Ndegwa, Bao Khanh Tran, and Haim Dubossarsky. 2025. [SenWiCh: Sense-annotation of low-resource languages for WiC using hybrid methods](#). In *Proceedings of the 7th Workshop on Research in Computational Linguistic Typology and Multilingual NLP*, pages 61–74, Vienna, Austria. Association for Computational Linguistics.
- Robbel Habtamu and Beakal Gizachew. 2024. State-of-the-art approaches to word sense disambiguation: A multilingual investigation. In *Pan-African Conference on Artificial Intelligence*, pages 176–202, Cham. Springer Nature Switzerland.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. [Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Saman Sarker Joy and Swakkhar Shatabda. 2026. [Bn-MMLU: Measuring massive multitask language understanding in Bengali](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 12211–12230, San Diego, California, United States. Association for Computational Linguistics.
- Mohammed Khalilia, Sanad Malaysha, Reem Suwaileh, Mustafa Jarrar, Alaa Aljabari, Tamer Elsayed, and Imed Zitouni. 2024. [ArabicNLU 2024: The first Arabic natural language understanding shared task](#). In *Proceedings of the Second Arabic Natural Language Processing Conference*, pages 361–371, Bangkok, Thailand. Association for Computational Linguistics.
- Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. [Teaching small language models to reason](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada. Association for Computational Linguistics.
- Hlaudi Daniel Masethe, Mosima Anna Masethe, Sunday Olusegun Ojo, Fausto Giunchiglia, and Pius Adewale Owolawi. 2024. [Word sense disambiguation for morphologically rich low-resourced languages: A systematic literature review and meta-analysis](#). *Information*, 15(9).
- Domenico Meconi, Simone Stirpe, Federico Martelli, Leonardo Lavallo, and Roberto Navigli. 2025. [Do large language models understand word senses?](#) In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33897–33916, Suzhou, China. Association for Computational Linguistics.
- Shahriar Kabir Nahin, Rabindra Nath Nandi, Sagor Sarker, Quazi Sarwar Muhtaseem, Md Kowsher, Apu Chandraw Shill, Md Ibrahim, Mehadi Hasan Menon, Tareq Al Muntasir, and Firoj Alam. 2025. [TituLLMs: A family of Bangla LLMs with comprehensive benchmarking](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 24922–24940, Vienna, Austria. Association for Computational Linguistics.
- Roberto Navigli and Simone Paolo Ponzetto. 2010. Babelnet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225.
- Miguel Ortega-Martín, Óscar García-Sierra, Alfonso Ardoiz, Jorge Álvarez, Juan Carlos Armenteros, and Adrián Alonso. 2023. Linguistic ambiguity analysis in chatgpt. *arXiv preprint arXiv:2302.06426*.
- Maja Popović. 2015. chrF: character n-gram f-score for automatic mt evaluation. In *Proceedings of the tenth workshop on statistical machine translation*, pages 392–395.
- Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. [Word sense disambiguation: A unified evaluation framework and empirical comparison](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.
- Nishat Raihan and Marcos Zampieri. 2025. [TigerLLM - a family of Bangla large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 887–896, Vienna, Austria. Association for Computational Linguistics.
- Ali Saeed, Rao Muhammad Adeel Nawab, Mark Stevenson, and Paul Rayson. 2019. [A sense annotated corpus for all-words urdu word sense disambiguation](#). *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(4).
- Md. Asaduzzaman Shuvo, Mahedi Hasan, Md. Tashin Parvez, Azizul Haque Noman, and Md. Shafayet Hossain Ovi. 2026. [Polite on the surface, broken in practice: A curated dataset for fixing generation and register failures in low-resource bangla text generation](#). *Preprint*, arXiv:2605.22487.

Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. [Judging the judges: Evaluating alignment and vulnerabilities in LLMs-as-judges](#). In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pages 404–430, Vienna, Austria and virtual meeting. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Haeun Yu, Seogyong Jeong, Siddhesh Pawar, Jisu Shin, Jiho Jin, Junho Myung, Alice Oh, and Isabelle Augenstein. 2025. Entangled in representations: Mechanistic investigation of cultural biases in large language models. *arXiv preprint arXiv:2508.08879*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. volume 36, pages 46595–46623.

A Annotator Profile

The dataset was verified by two native Bangla-speaking annotators with graduate-level backgrounds in Computer Science and prior experience in Bangla NLP research. Both annotators independently reviewed the labels and cultural reasoning associated with each instance. Disagreements were resolved through discussion using a shared annotation guideline. The resulting inter-annotator agreement, measured using Cohen’s κ , was 0.95, indicating almost perfect agreement.

B Result Analysis Extend

B.1 Ablation Study

Table 5 examines the contribution of reasoning-aware knowledge distillation by comparing zero-shot inference, label-only distillation, and the full QLoRA-KD framework. Zero-shot prompting performs poorly on the BanglaCEH task, achieving only 0.65% Exact Match with high dominant-meaning bias (63.46%) and hallucination (52.9%), indicating that the pretrained model lacks sufficient cultural grounding. Introducing label-only

knowledge distillation substantially improves Exact Match to 41.33% and Macro-F1 to 48.12%, demonstrating that supervision from teacher predictions is beneficial. However, the model still exhibits considerable dominant-meaning bias (34.56%) and a high hallucination rate (36.11%), suggesting that labels alone are insufficient for learning the underlying cultural distinctions. In contrast, the full QLoRA-KD framework, which distills both cultural reasoning and labels, achieves 85.16% Exact Match and 73.7% Macro-F1 while reducing dominant-meaning bias to 0% and hallucination rate 1.65%. These results show that the largest gains are obtained by supervising the reasoning process rather than the prediction labels alone.

Method	EM $\uparrow$ %	Bias $\downarrow$ %	Hall. $\downarrow$ %	Macro-F1 $\uparrow$ %
Zero-shot	0.65	63.46	52.9	11.50
KD (Label-only)	41.33	34.56	36.11	48.12
QLoRA-KD	85.16	0	1.65	73.7

Table 5: Ablation study for Qwen2.5-1.5B. $\uparrow$ indicates higher is better and $\downarrow$ indicates lower is better.

C Dataset Schema and Example Record

Each instance in the BanglaCEH benchmark is stored as a JSON record with the fields listed below. The body of the paper (Table 1) shows a condensed view; here we give the complete bilingual schema and a full raw record.

```
{
  "id": 1516,
  "word_bangla": "গগন",
  "word_roman": "Gagan",
  "category": "Intra-sentential
    (Name↔Concept)",
  "input": {
    "narrative_bangla":
      "...গগন এর মনে হলো...",
    "label_1": "Name",
    "label_2": "Concept"
  },
  "token_labels": {
    "token_1": "Name",
    "token_2": "Concept"
  },
  "cultural_entanglement_note":
    "Gagan carries the concept of the
    celestial firmament in Bengali
    cultural imagination while remaining
    a common given name."
}
```

Figure 3: A raw BanglaCEH benchmark record (condensed). Bangla fields render in native script; the full schema additionally includes bilingual justifications, cultural-entanglement paragraphs, and full explanations omitted here for space.

D Cultural Chain-of-Thought Prompt

Figure 4 shows the full Cultural Chain-of-Thought (C-CoT) prompt template. Unlike standard chain-of-thought, it forces the model to argue both the name and the competing cultural reading for each occurrence before committing to a label, directly targeting the dominant-meaning bias.

E LLM-as-a-Judge Prompt

Figure 5 shows the prompt given to the LLM judge (GPT-5.4) for evaluating the fine-tuned models. To mitigate known judge biases (Zheng et al., 2023; Thakur et al., 2025), the judge model is distinct from all models under evaluation, and the human annotators were given an identical rubric to ensure the two judges measure the same quantities.

F Formula

F.1 Average Score

Each BanglaCEH instance contains two target occurrences. A prediction receives one point for each correctly classified occurrence (maximum two points per instance). The Average Score is computed as

$$\text{Average Score} = \frac{\text{Total points earned}}{\text{Number of instances} \times 2} \times 100.$$

```
[System]
You are an expert in Bangla cultural linguistics. Some Bangla words are simultaneously a person's NAME and a culturally meaningful concept, emotion, or spiritual state. Explain WHY each occurrence carries its meaning, then give the final labels.

[User]
The Bangla word 'গগন' (Gagan) appears TWICE in this sentence with two different meanings.
Sentence: {narrative}
Reason contrastively for EACH occurrence:
For Occurrence 1:
- Argument A: why it could be a Name.
- Argument B: why it could be an Emotion / Concept / Spiritual / State.
- Decision: which reading the context supports.
For Occurrence 2:
- Argument A: why it could be a Name.
- Argument B: why it could be an Emotion / Concept / Spiritual / State.
- Decision: which reading the context supports.
After reasoning, end with EXACTLY this block:
FINAL:
Occurrence 1: <label>
Occurrence 2: <label>

Valid labels: Name, Emotion, Concept, Spiritual, Emotional State, Collective State
```

Figure 4: The Cultural Chain-of-Thought (C-CoT) prompt template. The model must articulate a Name argument and a competing cultural-category argument for each occurrence before producing the final labels, counteracting the dominant-meaning bias.

Thus, Average Score reflects occurrence-level accuracy, whereas Exact Match requires both occurrences in an instance to be classified correctly.

G Hyperparameter Settings

Table 6 lists the final QLoRA fine-tuning configuration, applied without modification across all fine-tuned models. We arrived at these values by monitoring validation loss: an initial higher-capacity configuration (rank 32, $\alpha = 64$, learning rate 2×10^{-4} , dropout 0.05, weight decay 0.01) overfit within three epochs, so we reduced adapter capacity and learning rate and increased regularisation until the validation curve stabilised.

H Per-Label Analysis

To locate where residual errors concentrate after fine-tuning, we report per-label precision, recall, and F1 for the three QLoRA-KD models in Ta-

You are an expert evaluator of Bangla cultural linguistics, judging a model's performance on the Bangla Culturally Entangled Homograph (BanglaCEH) task, where one Bangla word appears twice in a sentence with two different meanings (e.g., once as a NAME and once as a CONCEPT, EMOTION, SPIRITUAL, or STATE reading). You are given a batch of items, each with the sentence, gold labels, gold reasoning, and the model output.

Valid labels: Name, Emotion, Concept, Spiritual, Emotional State, Collective State.

Per-item judgement:

- match_1: does the model's Occurrence-1 label equal the gold label?
- match_2: does the model's Occurrence-2 label equal the gold label?
- bias: did the model label a gold NAME occurrence as a non-name (cultural) reading?
- hallucinated: did the model output a label outside the valid set, or fail to produce a parseable label?

Aggregation (over N items):

$$\text{Exact\%} = 100 \times (\text{match_1 AND match_2}) / N$$

$$\text{Partial\%} = 100 \times (\text{exactly one match}) / N$$

$$\text{Bias\%} = 100 \times (\text{bias}) / (\text{items with a gold Name})$$

$$\text{Hall.\%} = 100 \times (\text{hallucinated}) / N$$

Output: return ONLY a JSON object, rounded to one decimal

```
{ "model", "hallucination_pct", "bias_pct", "exact_pct", "partial_pct" }
```

Figure 5: The LLM-as-a-judge prompt (GPT-5.4). The judge scores each output for label match, dominant-meaning bias, and hallucination, then aggregates into the percentage metrics reported in Table 4.

ble 7. Two patterns are consistent across all models. First, Name is recovered almost perfectly ($F1 \geq 0.98$ everywhere), confirming at the label level that the dominant-meaning bias central to our study is eliminated by distillation models no longer default away from the name reading. Second, the errors that remain fall almost entirely on Emotion and Spiritual ($F1$ as low as 0.59 and 0.62), while Name and Concept are handled reliably. Notably, Emotion is not among the rarest labels, suggesting its difficulty stems from semantic proximity to the State readings rather than from data scarcity alone. This concentration of error in the long tail accounts for the gap between Micro- and Macro-averaged $F1$, and indicates that the remaining challenge is one of data scarcity in culturally rarer readings rather than a limitation of the method itself.

Hyperparameter	Value
<i>Quantisation</i>	
Precision	4-bit NF4
Double quantisation	Yes
Compute dtype	fp16
<i>LoRA</i>	
Rank (r)	16
α	32
Dropout	0.1
Target modules	q, k, v, o, gate, up, down
<i>Optimisation</i>	
Learning rate	1×10^{-4}
LR schedule	cosine
Warm-up ratio	0.1
Weight decay	0.05
Optimiser	paged AdamW (8-bit)
Max gradient norm	0.3
<i>Training</i>	
Effective batch size	16 (grad. accum. 8)
Max epochs	8
Early stopping	val. loss, patience 3
Checkpoint	best val. loss
Max sequence length	768
<i>Hardware</i>	
GPU	1 $\times$ NVIDIA T4 (16 GB)
Decoding (inference)	greedy

Table 6: Final QLoRA-KD fine-tuning hyperparameters, applied identically across Qwen2.5-1.5B, Llama-3.2-3B, and TituLLM-1B.

Label	Qwen2.5-1.5B			Llama-3.2-3B			TituLLM-1B		
	P	R	F1	P	R	F1	P	R	F1
Name	1.00	1.00	1.00	0.99	0.97	0.98	1.00	0.97	0.99
Concept	0.92	0.91	0.91	0.97	0.85	0.91	0.98	0.91	0.94
Emotional State	0.97	0.97	0.97	0.80	0.95	0.87	0.89	1.00	0.94
Collective State	0.96	1.00	0.98	0.76	0.96	0.85	0.92	1.00	0.96
Emotion	0.61	0.85	0.71	0.65	0.55	0.59	0.93	0.74	0.82
Spiritual	0.89	0.47	0.62	0.77	0.94	0.85	0.57	0.89	0.70

Table 7: Per-label precision (P), recall (R), and F1 for the three fine-tuned (QLoRA-KD) models. Name achieves near-perfect F1 across all models; the rarest labels (Emotion, Spiritual) remain hardest.

ASCRIBE: Atomic and Significance-Based Reasoning for Thai Clinical SOAP Note Generation

Tarm Kalavantavanich^{1,*}, Teerawut Ponarchar^{1,*}, Pattaramanee Arsomngern^{1,*},
Jenta Wonglertsakul^{2,†}, Watcharakorn Chuthong¹, Chiraphat Boonnag¹,
Knot Pipatsrisawat¹, Titipat Achakulvisut^{1,3}

¹Looloo Health, Thailand, ²Kasetsart University, Thailand, ³Mahidol University, Thailand

*Equal contribution. †Work conducted during an internship at Looloo Health.

{tarm.kalavan, teerawut.p, pattaramanee.a, c.boonnag, knot}@loolootech.com

jenta.w@live.ku.th watcharakorn.c@looloohealth.com titipat.ach@mahidol.ac.th

Abstract

Automatic SOAP note generation can ease the documentation burden on physicians, but existing reasoning methods often omit clinically important information and generate unsupported content. Progress in Thai is further hindered by the lack of publicly available datasets. We propose ASCRIBE, a physician-inspired reasoning framework that ascribes a clinical-significance level to each extracted atomic fact in the conversation before summarization, making a general-purpose LLM a more reliable scribe. We also release ThaiClinicBench, the first de-identified Thai clinical summarization benchmark of real encounters, together with a synthetic training corpus derived from real clinical notes. As a prompt, ASCRIBE outperforms chain-of-thought prompting on GPT-5.4 and Gemini 3.1 Pro across the physician-aligned LLM-judge metrics and improves on standard prompting by up to 10.3 points on the completeness LLM-judge metric. As a GRPO reward, it enables a Gemma-4-E4B model trained solely on synthetic data to match Gemini 3.1 Pro in factual precision and surpass it in completeness. Code and data can be found at <https://github.com/loolootech/ascribe>.

1 Introduction

Comprehensive clinical documentation is critical for patient safety yet highly time-consuming, a serious burden in regions with low doctor-to-patient ratios such as Southeast Asia (Kanchanachitra et al., 2011). AI medical scribes commonly address this through a transcribe-summarize pipeline (Finley et al., 2018): an automatic speech recognition (ASR) system transcribes the encounter, and a summarization system converts it into a structured note such as the SOAP (Subjective, Objective, Assessment, Plan) format (Weed, 1968). However, transcripts are error-prone due to the spontaneous,

disfluent nature of clinical conversation. This problem is compounded in low-resource languages like Thai, whose conversational and dialectal nuances challenge even frontier large language models (LLMs) (Limkonchotiawat et al., 2025).

Prior work on clinical summarization primarily fine-tunes models on paired dialogue transcripts and summaries from English datasets (Yim et al., 2023; Abacha et al., 2023b). When such data are limited, recent efforts synthesize transcripts from patient profiles via agent simulation (Ge et al., 2025). For prompting techniques, while chain-of-thought (CoT) has been used to encourage clinical reasoning across medical tasks (Nachane et al., 2024; Wu et al., 2024), it often accumulates errors, resulting in poorly grounded outputs (Wu et al., 2025). Crucially, these efforts remain unvalidated for Thai: to our knowledge, no public Thai dataset supporting SOAP note generation exists, leaving both data and reasoning gaps unresolved.

To close the data gap, we adapt the transcript synthesis pipeline of Ge et al. (2025). By leveraging a real internal corpus of paired transcripts and SOAP notes, we synthesize a new Thai training corpus, S-ICOPD, which we release along with ThaiClinicBench, a de-identified benchmark of real encounters and physician-authored SOAP notes collected from a Thai clinic.

To close the reasoning gap, we introduce ASCRIBE (Atomic, Significance-Based Clinical Reasoning Framework for Medical Scribe), a clinically grounded reasoning framework that improves note quality both as a prompt template and as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) rewards. Inspired by physician history-taking, ASCRIBE decomposes a dialogue into atomic facts (Min et al., 2023; Chen et al., 2024b), ranks them by clinical significance, and generates the SOAP note from this reasoning trace (Figure 1). We evaluate notes

on Thai encounters with two physician-aligned LLM-as-a-judge metrics, ClaimEntailment (completeness) and SOAPPrecision (factual grounding), that also serve as verifiable RL rewards. As a prompt, ASCRIBE improves frontier LLMs by up to 10.3 points over standard prompting. As GRPO rewards, ASCRIBE lets a fine-tuned Gemma-4-E4B (Team et al., 2026) surpass Gemini 3.1 Pro on completeness by over 10 points, even when trained only on synthetic data.

In summary, our contributions are:

- ASCRIBE, a physician-inspired framework that ranks atomic clinical facts by significance and improves SOAP notes both as a prompt and as GRPO rewards.
- A Thai transcript-synthesis pipeline, with real-data-seeded synthetic transcripts matching real-transcript SFT performance and surpassing it on completeness under GRPO.
- Two released datasets: S-ICOPD, the synthetic training corpus, and ThaiClinicBench, the first medical-scribe benchmark from a Thai clinic.

2 Related Work

Clinical Note Generation. Early work developed section-aware SOAP note generation pipelines (Krishna et al., 2021), with subsequent studies releasing English datasets and benchmarks (Abacha et al., 2023b,a; Yim et al., 2023). Later work improved performance using in-domain pre-training (Grambow et al., 2022), extractive-abstractive summarization (Neupane et al., 2024), transcript synthesis (Ge et al., 2025), and prompting techniques (Kernberg et al., 2024). Yet LLM-generated notes still suffer from omission, hallucination, and poor out-of-domain generalization (Kernberg et al., 2024; Chen and Hirschberg, 2024).

Reasoning in Medical Tasks. LLMs show strong medical reasoning using chain-of-thought (CoT) prompting (Wei et al., 2022), with MedPaLM 2 reaching expert-level accuracy on medical licensing exams (Singhal et al., 2025). Recent work grounds this reasoning in clinical knowledge, such as anchoring it in knowledge seeds (Wu et al., 2024) or mirroring physicians’ diagnosis with few-shot examples (Nachane et al., 2024).

However, recent evidence shows that CoT often degrades performance on clinical text, with failures driven by hallucination and weak grounding rather than faithful reasoning (Wu et al., 2025); this extends to SOAP note generation from clinical dialogue (Faisal, 2026). Motivated by these findings, we instead enforce structured reasoning over atomic clinical facts, not free-form generation.

Evaluating Clinical Notes and RLVR. Conventional overlap metrics such as ROUGE correlate inconsistently with physician judgment and miss clinically critical omissions and hallucinations (Abacha et al., 2023c). Instead, prior work evaluates notes by detecting omitted atomic facts (Schumacher et al., 2023), grading errors by clinical impact (Asgari et al., 2025), or measuring claim-level precision and recall (Xie et al., 2024), the last of which was adapted into a GRPO reward for reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; Jhaveri et al., 2025). Unlike prior work that rewards only the final note, we also reward the reasoning trace by supervising clinical fact extraction and significance attribution.

3 Proposed Method

We formulate SOAP note generation as a sequence-to-sequence summarization task. Given a transcript of a physician–patient encounter, a language model produces a structured clinical note in SOAP format, comprising the Subjective findings reported by the patient, the Objective findings from physical exams and diagnostic results, the physician’s Assessment, and the management Plan. Concretely, we follow the SOAP-inspired four-division scheme of Yim et al. (2023) in our experiments, with each section emitted as a list of entries, which we refer to as *statements*.

We approach this task from two angles. First, we study LLM summarization. Since conventional prompting strategies such as CoT struggle to improve clinical note quality (Wu et al., 2025; Faisal, 2026), we propose a sequence of reasoning steps and a reasoning-driven fine-tuning method to address it. Second, we tackle clinical data scarcity in low-resource languages like Thai, by exploring synthetic transcript generation as a fine-tuning dataset and outlining how such encounter transcripts could be synthesized from a clinical note.

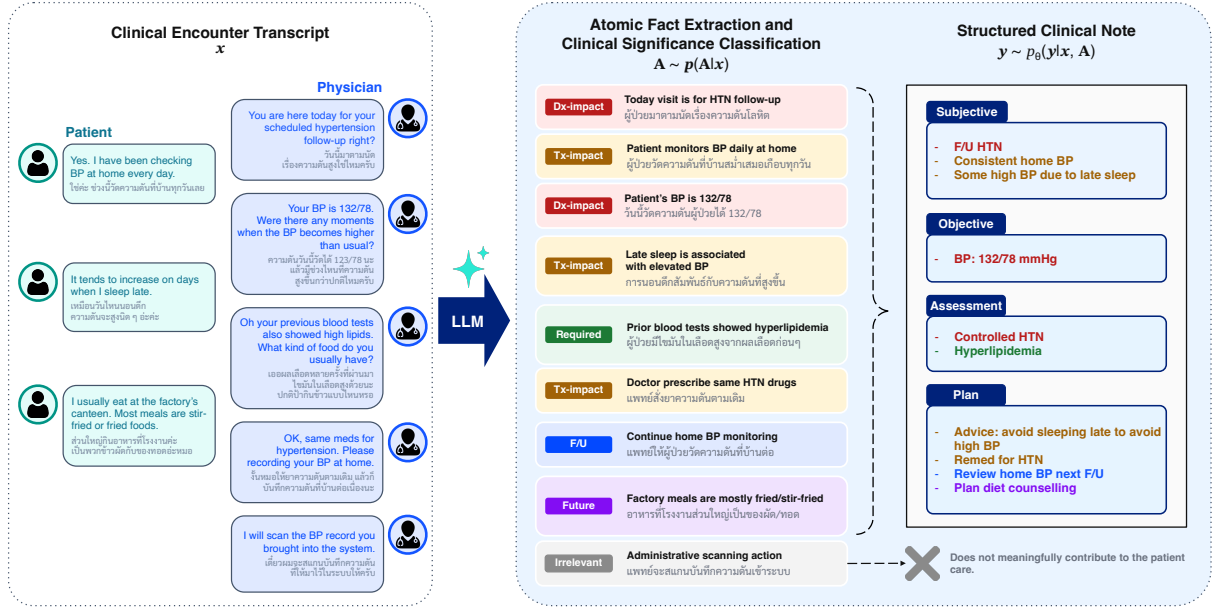


Figure 1: **ASCRIBE’s SOAP generation pipeline.** Atomic facts are extracted from the transcript x and classified by their clinical significance to form the reasoning trace A , which conditions generation of the final SOAP note y .

3.1 ASCRIBE: Reasoning over Atomic Facts and Clinical Significance

Physicians conducting history-taking listen for key facts as they emerge, directing their attention towards clinically impactful statements while filtering out irrelevant ones. Inspired by this process, we introduce ASCRIBE, a structured reasoning framework that decomposes clinical summarization into two steps: atomic fact extraction and clinical significance classification, as illustrated by Figure 1.

Atomic Fact Extraction. To ground the model’s reasoning in explicit, verifiable units of information, we first use an LLM to extract atomic facts from the transcript before summarization. Inspired by prior work on atomic facts and propositions (Min et al., 2023; Chen et al., 2024b), we define three conditions tailored to clinical text:

- *Standalone completeness:* Each fact must be a self-contained, natural sentence that is interpretable on its own, with all references resolved to explicit subject names.
- *One idea:* Each fact must contain a single idea, with compound statements split into separate facts unless a clause is not meaningful on its own (e.g., a causal or temporal clause), in which case it stays attached so the main fact remains complete.

- *Uniqueness:* A fact stated multiple times must appear only once, in its most complete and specific form.

In our experiments, we extract facts with an LLM whose instruction was tuned against LLM judges, referred to as the *Atomic Fact LLM Judges*, that validate generated atomic facts against a physician-curated gold set in a precision-recall fashion and achieve substantial physician agreement (Appendix B.1).

Clinical Significance Classification. We define six levels of clinical significance, ranked by how much a fact could impact patient safety and care as detailed in Table 1.

We co-designed these levels with certified physicians, drawing on causality found in clinical settings and the Problem-Oriented Medical Record (Weed, 1968), which underlies SOAP and is mandated in Thailand’s national medical-record standards (National Health Security Office (Thailand), 2020). The first three levels mirror what a SOAP note must support, namely the assessment, the plan, and mandated documentation, while the remaining levels extend this logic to longitudinal care. An LLM classifier reproduces the physicians’ labels with substantial agreement (Appendix B.2).

Structured Reasoning-based Generation. We combine the fact extraction and classification into

Level	Description
Diagnosis-impacting	Essential information supporting the diagnosis or assessment of the current visit (e.g., chief complaint, symptoms, clinical findings, disease progression, physical examination, signs, laboratory or imaging findings).
Treatment-impacting	Details regarding or influencing the management of the current visit (e.g., medications, procedures, follow-up plans, treatment recommendations).
Required	Patient information required by clinical standards, regardless of the current complaint (e.g., comorbidities, drug allergies, smoking/alcohol history).
Follow-up-relevant	Details not essential to the current visit but useful for subsequent follow-ups (e.g., advice given, referrals, treatment goals).
Future-reference	Details useful for future reference but unlikely to affect the current episode of care (e.g., irrelevant family history, social history, unrelated medical history).
Irrelevant	Does not meaningfully contribute to current or future clinical decision-making (e.g., redundant statements, vague complaints, hyper-specific context, administrative details, physician explanations).

Table 1: Clinical significance levels used for categorizing atomic facts. Levels are listed from most to least significant.

a single reasoning trace. Given a transcript x , the model p_θ first generates the trace A as a set of atomic facts with their evidence and clinical significance,

$$A = \{(f_i, e_i, s_i)\}_{i=1}^N, \quad (1)$$

where f_i is an atomic fact, e_i is a short supporting span from the transcript that grounds it, s_i is its predicted clinical significance level, and N is the number of extracted atomic facts. The model then generates the SOAP note y in the same decoding pass, conditioned on both the transcript and the reasoning trace,

$$y \sim p_\theta(y \mid x, A). \quad (2)$$

Conditioning on A encourages the model to identify and prioritize clinically impactful facts before producing the SOAP note, rather than summarizing directly or relying on free-form reasoning.

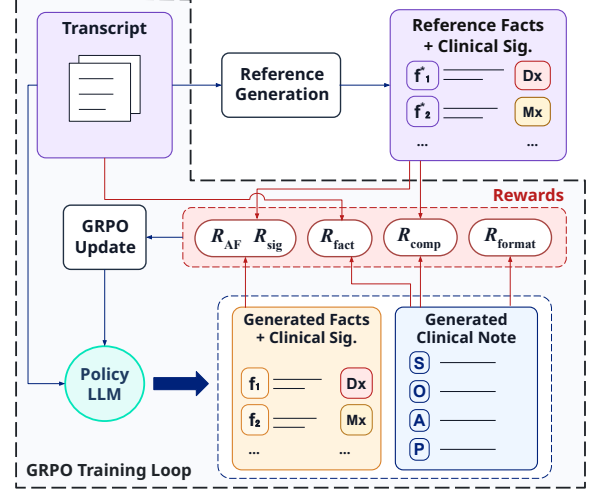


Figure 2: **GRPO training pipeline.** The generated ASCRIBE reasoning trace A and SOAP note from the policy LLM are scored by the five rewards of Section 3.2.2, which jointly drive the GRPO update. Certain rewards require reference facts whose generation details are in Section 4.4.

3.2 Fine-Tuning Small Language Models for SOAP Note Summarization

Due to their capability to be deployed in on-premise settings, Small Language Models (SLMs) are our candidate summarization models, for which we explore both supervised and reinforcement learning techniques.

3.2.1 Supervised Fine-Tuning (SFT)

We apply LoRA-based SFT (Hu et al., 2022) in two settings: standard prompting, which trains the model to generate only SOAP notes from clinical transcripts; and structured reasoning, where the target is prepended with a reference reasoning trace.

3.2.2 Reinforcement Learning with Verifiable Rewards (RLVR)

To align the model with ASCRIBE reasoning, we adopt GRPO (Shao et al., 2024) with LoRA as shown in Figure 2. We use our structured reasoning to define five reward functions that optimize both reasoning and final note quality, each scaled to the range $[0, 1]$, as follows:

Atomic Fact Reward. Let A and $A^* = \{(f_j^*, e_j^*, s_j^*)\}_{j=1}^M$ denote the sets of generated and reference atomic facts, respectively. We define two *Atomic Fact LLM Judges* to compare the two sets independently and identify semantically matching facts (Appendix B). We denote $c(i)$ as the index

of the reference fact $f_{c(i)}^*$ matched to the generated fact f_i assigned by the Atomic Fact LLM Judge, with $c(i) = \emptyset$ when no match exists. Atomic precision and recall are computed as:

$$\text{Prec}_{\text{atomic}} = \frac{\#\text{matched generated facts}}{N},$$

$$\text{Rec}_{\text{atomic}} = \frac{\#\text{matched reference facts}}{M}.$$

The final reward is their harmonic mean:

$$R_{\text{AF}} = \frac{2 \text{Prec}_{\text{atomic}} \text{Rec}_{\text{atomic}}}{\text{Prec}_{\text{atomic}} + \text{Rec}_{\text{atomic}}}. \quad (3)$$

Clinical Significance Reward. For each generated fact f_i and its matched reference fact $f_{c(i)}^*$, we compare s_i against the reference significance label $s_{c(i)}^*$. If no reference fact is matched, the prediction is considered incorrect. The reward is computed as:

$$R_{\text{sig}} = \frac{1}{N} \sum_{i=1}^N \mathbf{1}[s_i = s_{c(i)}^*]. \quad (4)$$

Format Reward. R_{format} checks that the generated note follows the required SOAP note structure; the reward is reduced if a generated note does not follow the format.

Completeness Reward. Following evaluation-driven GRPO for clinical summarization (Jhaveri et al., 2025), we use the ClaimEntailment metric described in Section 5 as the R_{comp} reward. This directly encourages generated SOAP notes to preserve clinically important facts from the encounter.

Factual Grounding Reward. Following Jhaveri et al. (2025), we use the SOAPPrecision metric outlined in Section 5 as the R_{fact} reward, which encourages clinical statements in the SOAP note y to remain factually supported by the encounter transcript.

3.3 Synthesizing Encounter Transcripts

As real clinical transcripts are scarce in low-resource languages like Thai, we explore synthetic transcript generation by adapting the pipeline of Ge et al. (2025) to our settings. Our pipeline takes a SOAP note as the seed patient profile, from which the physician–patient dialogue transcripts were synthesized in two stages. First, *intent planning* constructs a weighted directed intent graph that models probabilistic conversation flow

and selects an intent path grounded in the profile. Second, *transcript generation* synthesizes the dialogue based on that path. See Appendix C for full details.

Intent Planning. Following Ge et al. (2025), we construct an intent graph from a corpus of real physician–patient conversations. Each training transcript is first abstracted into an *intent skeleton*, a sequence of conversational intents discussed during the encounter. These intent skeletons are then aggregated into an intent graph $G = (V, E)$, where V represents intents and E represents transitions whose weights refer to the transition probabilities. For each seed patient profile, we gather candidate intent paths by performing depth-first search on G . To ensure the path is consistent with the profile, we limit the search space to intent nodes corresponding to the profile, then rank the resulting paths with an LLM judge, selecting the one with the highest score.

Transcript Generation. Given a patient profile and the chosen intent path, a doctor agent and a patient agent generate the encounter transcript, addressing one intent at a time. Generation for each intent continues until an LLM judge determines that the intent has been fulfilled or the per-intent turn limit has been reached. Each completed dialogue section is then validated against ten rules adapted from Ge et al. (2025) for Thai clinical conversations. If validation fails, the section is regenerated using the validator’s feedback. The pipeline proceeds to the next intent once the section passes validation or the retry budget is exhausted, retaining the final attempt.

4 Datasets

We use three datasets in this study: an internal outpatient corpus of Thai physician–patient encounters (ICOPD), a new real-world benchmark (ThaiClinicBench), and a synthetic training corpus (S-ICOPD) built from ICOPD. All real conversation transcripts are manually produced by human annotators. We describe each dataset and how the reference training targets are constructed.

4.1 Internal Clinical OPD Dataset (ICOPD)

The ICOPD dataset is a private corpus of 400 consented outpatient-department (OPD) encounters collected in Thailand. The conversations are in colloquial Thai, with dialectal medical terms common in Thai clinical settings. For each encounter,

we obtain the transcript, the physician-written note, and an LLM-constructed reference SOAP note for SFT (Section 4.4). The conversations average 328 words (1.34k characters). We split the corpus into training (300), validation (50), and test (50) sets.

4.2 ThaiClinicBench

We curate ThaiClinicBench, a benchmark of manually de-identified and consented OPD encounters collected from a medical clinic in Thailand. It contains 44 encounters (pairs of transcript and physician-written SOAP note). The conversations average 703 words (2.89k characters). All encounters serve as an out-of-distribution test set. To the best of our knowledge, this is the first Thai medical-scribe benchmark released for research.

4.3 Synthetic ICOPD Dataset (S-ICOPD)

Beyond the real encounters, we construct a synthetic ICOPD using the transcript synthesis pipeline (Section 3.3) as an additional training dataset. We seed it with the physician-written SOAP notes of ICOPD, each serving as the patient profile from which we synthesize an encounter transcript. Running it on the training and validation splits yields 295 and 50 synthetic transcripts, respectively, with 5 training cases excluded as the pipeline does not produce a usable transcript. Each synthetic transcript is then re-summarized with the pipeline of Section 4.4 to obtain its reference data, so that every text in S-ICOPD is model-generated.

4.4 Reference Training Target Generation

We generate a reference set of atomic facts and their significance labels for each encounter using Gemini 2.5 Pro with prompts in Appendix B. These serve as the reference facts behind R_{AF} and R_{sig} during RLVR, the input to evaluation metrics in Section 5, and the reasoning targets for SFT.

For SFT, we construct reference SOAP notes in two stages. We first generate a draft SOAP note from the transcript alone using Gemini 3.1 Pro conditioned on the proposed structured reasoning and then refine it with GPT-5.4 (Appendix A.4) to further improve overall quality.

5 Evaluation Metrics

Following Yim et al. (2023), we calculate ROUGE-L (Lin, 2004) and BLEURT (Sellam et al., 2020) to measure lexical and semantic similarity between generated notes and SOAP notes written by certified physicians. To better assess clinical qual-

ity, we adopt an LLM-judge framework inspired by DocLens (Xie et al., 2024), simplifying their Claim Precision pipeline by judging the generated note directly rather than extracting claims from it first. These adapted LLM judges compute two complementary metrics and are validated for alignment with physicians (details are provided in Appendix D).

ClaimEntailment (CE) measures completeness of the generated SOAP note by using reference atomic facts to evaluate whether each fact was entailed by the generated note (i.e., recall). We also introduce *CE-Impact*, which calculates CE only on the three highest clinical significance levels (*Diagnosis-impacting*, *Treatment-impacting*, and *Required*) to further measure the recall of clinically important facts.

SOAPPrecision (SP) measures factual grounding by verifying whether each statement in the generated SOAP note is explicitly supported by the encounter transcript (i.e., precision). It allows for paraphrasing and terminology normalization (e.g., from lay language to medical jargon) but penalizes unsupported, inferred, or hallucinated content.

6 Experimental Settings

Prompting Baselines. We use GPT-5.4 and Gemini 3.1 Pro as our baselines under three zero-shot prompting strategies: standard SOAP note generation (Yim et al., 2023), CoT, and our proposed structured reasoning (see Appendix A for prompt details).

Fine-tuning Baselines. We use Gemma-4-E4B-it (Team et al., 2026) as the base SLM for all fine-tuning experiments. We compare our solution against three GRPO baselines: **ROUGE-L** uses the ROUGE-L score between the generated and reference SOAP notes as the reward, representing a conventional lexical-overlap optimization baseline (Ou et al., 2026). **BGE-M3** uses BGE-M3 (Chen et al., 2024a) embedding similarity between the generated and reference SOAP notes as the reward, following prior work (Akarajadwong et al., 2025) that adopts it for fine-tuning Thai LLMs. **ClinicEval** uses Completeness and Factual Grounding rewards following the evaluation-integrated GRPO training for clinical summarization (Jhaveri et al., 2025). See Appendix E for training details.

7 Experimental Results

We evaluate the effectiveness of ASCRIBE for zero-shot prompting and SLM fine-tuning, compare SFT with GRPO, and analyze the contribution of the proposed reward. We further study the effectiveness of our synthetic training data, the relationship between conventional and clinical metrics, and robustness to ASR noise. Table 2 outlines our full experimental results.

Effectiveness of Structured Reasoning. The ASCRIBE prompt outperforms both standard and CoT prompting in the zero-shot setting. For Gemini 3.1 Pro, the ASCRIBE prompt improves SP/CE/CE-Impact over standard prompting by +3.4/+2.2/+3.7 on ICOPD and +2.6/+6.8/+9.1 on ThaiClinicBench. GPT-5.4 shows even larger gains of +4.7/+5.1/+7.2 and +5.8/+6.9/+10.3, respectively. The same trend largely holds for Gemma-4-E4B in the zero-shot setting. In contrast, CoT prompting yields smaller CE gains while reducing conventional metrics and, in some cases, SP. For SFT settings, structured reasoning generally improves CE and CE-Impact, particularly on ThaiClinicBench, but its effect on SP is mixed.

SFT vs. GRPO. SFT generally achieves higher ROUGE-L, which could be attributed to being supervised against curated reference SOAP notes, but the differences in BLEURT scores are less consistent. In contrast, training models with GRPO using clinically grounded rewards substantially improves SP, CE, and CE-Impact, reflecting that lexical similarity alone does not necessarily represent the clinical quality of the notes.

Effectiveness of RLVR. Among GRPO variants trained on real ICOPD data, the ASCRIBE reward achieves the best overall clinical performance, obtaining the highest SP, CE, and CE-Impact on ICOPD (87.1/62.1/71.2) and similarly strong results on ThaiClinicBench (89.4/64.0/72.1). Trained with the ASCRIBE reward, Gemma-4-E4B also outperforms proprietary Gemini 3.1 Pro across all LLM-judge metrics and GPT-5.4 on CE, despite having only 4.5B effective parameters.

Notably, baseline performance varies substantially with the reward design. Optimizing for ROUGE-L maintains high SP but severely reduces CE, while optimizing for BGE-M3 improves CE but obtains the lowest SP of any GRPO baseline. Crucially, optimizing for ClinicEval substantially improves CE while maintaining comparable level

of SP, demonstrating the value of directly optimizing for clinical quality. However, by rewarding the intermediate reasoning trace as well, training GRPO with ASCRIBE rewards improves both SP and CE.

Ablation of Structured Reasoning Prompt and Rewards. Focusing on the results of models trained on real data with GRPO reward, we found that the ASCRIBE prompt primarily improves SP, whereas our reward provides more consistent gains across the clinical metrics. Substituting the standard prompt with the ASCRIBE prompt improves SP by +1.3 on ICOPD and +1.4 on ThaiClinicBench, while CE remains comparable. Under the ASCRIBE prompt, replacing the ClinicEval reward with our reward further improves SP/CE/CE-Impact by +2.3/+0.3/+0.2 on ICOPD and +1.1/+0.5/+0.3 on ThaiClinicBench.

Effectiveness of Synthetic Training Data. Under SFT, training on synthetic data alone performs comparably to training on real data; on ThaiClinicBench, synthetic-only training with the standard prompt achieves SP/CE/CE-Impact of 83.7/49.1/56.1 versus 83.9/50.6/56.5. However, combining real and synthetic data lowers CE in every SFT settings, e.g., from 53.8 to 51.8 on ThaiClinicBench with our prompt despite a higher SP. These results suggest that synthetic data could substitute for real data in SFT, but naively mixing the two datasets does not bring further gains.

Under GRPO, synthetic-only training outperforms real-only training in CE and CE-Impact, improving from 62.1/71.2 to 64.9/75.7 on ICOPD and from 64.0/72.1 to 69.7/77.9 on ThaiClinicBench, with comparable SP. Mixed training yields similar CE, including the highest CE of 70.0 on ThaiClinicBench, but lower SP. These gains suggest that synthetic transcripts are also effective for RLVR.

We hypothesize that the performance degradation found in both SFT and GRPO settings stems from distributional differences between the corpora which were naively mixed. Real conversations contain disfluencies and colloquial expressions that might only be partially captured by synthetic transcripts.

Clinical vs. Conventional Evaluation. Frontier LLMs achieve strong clinical performance while also attaining competitive ROUGE-L and BLEURT scores. However, the trends across baselines show that conventional metrics do not con-

Model / Training	GRPO Reward	Training Data	Prompt	ICOPD					ThaiClinicBench				
				SP	CE	CE-Impact	ROUGE-L	BLEURT	SP	CE	CE-Impact	ROUGE-L	BLEURT
Frontier LLM prompting													
Gemini 3.1 Pro	–	–	Standard	83.5	52.6	63.3	0.3467	0.3366	86.6	51.6	58.8	0.3390	0.2488
	–	–	CoT	81.0	53.2	64.3	0.3387	0.3177	84.7	52.0	59.8	0.3205	0.2430
	–	–	Ours	<u>86.9</u>	<u>54.8</u>	<u>67.0</u>	0.3406	0.3220	<u>89.2</u>	<u>58.4</u>	<u>67.9</u>	0.3170	<u>0.2525</u>
GPT-5.4	–	–	Standard	90.5	54.5	64.0	<u>0.3413</u>	<u>0.3299</u>	92.5	55.7	64.3	<u>0.3023</u>	0.2623
	–	–	CoT	90.9	58.6	68.2	0.3169	0.3194	93.3	58.6	68.8	0.2809	0.2545
	–	–	Ours	95.2	59.6	71.2	0.3032	0.3237	98.3	62.6	74.6	0.2781	0.2627
Gemma-4-E4B-it fine-tuning													
Zero-shot	–	–	Standard	84.7	38.4	46.6	0.3180	0.3183	<u>87.7</u>	42.5	48.0	<u>0.3003</u>	0.2707
	–	–	Ours	<u>87.0</u>	<u>44.2</u>	<u>53.4</u>	0.3026	<u>0.3207</u>	87.0	<u>46.6</u>	<u>52.9</u>	0.2915	0.2803
SFT	–	Real	Standard	75.1	<u>46.6</u>	<u>57.3</u>	0.3420	<u>0.3224</u>	83.9	50.6	56.5	<u>0.3083</u>	<u>0.2593</u>
	–	Real	Ours	79.3	46.3	57.0	0.3395	0.2946	85.3	<u>53.8</u>	<u>59.5</u>	0.2961	0.2545
	–	Synth	Standard	<u>80.9</u>	43.4	53.4	0.3272	0.3159	83.7	49.1	56.1	0.2928	0.2506
	–	Synth	Ours	77.2	45.5	55.7	0.3276	0.3060	82.9	50.8	57.0	0.2863	0.2520
	–	Real+Synth	Standard	78.5	41.9	51.1	0.3433	0.3112	83.0	48.8	56.7	0.3050	0.2550
	–	Real+Synth	Ours	77.5	45.5	55.7	0.3491	0.3164	<u>86.7</u>	51.8	<u>59.5</u>	0.2993	0.2483
GRPO	ROUGE-L	Real	Standard	86.8	33.9	40.2	<u>0.3489</u>	0.3034	87.9	35.7	40.3	0.3149	<u>0.2725</u>
	BGE-M3	Real	Standard	81.4	48.8	59.3	0.3120	0.3366	84.4	52.3	58.8	0.2946	0.2687
	ClinicEval	Real	Standard	83.5	61.4	70.3	0.2293	0.3143	86.9	63.7	72.3	0.2052	0.2396
	ClinicEval	Real	Ours	84.8	61.8	71.0	0.2551	0.2979	88.3	63.5	71.8	0.2323	0.2467
	Ours	Real	Ours	87.1	62.1	71.2	0.2773	0.2898	89.4	64.0	72.1	0.2487	0.2450
	Ours	Synth	Ours	86.0	64.9	75.7	0.2478	0.2786	89.6	69.7	77.9	0.2143	0.2280
	Ours	Real+Synth	Ours	83.2	64.4	75.1	0.2306	0.3203	85.2	70.0	77.0	0.2105	0.2509

Table 2: Comparison of ASCRIBE prompting and fine-tuning against baseline methods on the ICOPD test split and ThaiClinicBench. *Ours* denotes the corresponding ASCRIBE prompt or reward, *Real* denotes training on ICOPD, and *Synth* denotes training on S-ICOPD. Higher values indicate better performance for all metrics. Underline marks the best result within each model/training block, and **bold** marks the best result within each approach (frontier LLM prompting vs. Gemma-4-E4B-it fine-tuning). SP, CE, and CE-Impact are reported on a 0–100 scale.

sistently reflect clinical quality. For example, SFT and ROUGE-L-optimized GRPO can obtain high lexical-similarity scores without corresponding improvements in CE. Conversely, ClinicEval GRPO substantially improves CE and CE-Impact at the cost of ROUGE-L. These results suggest that conventional and clinical metrics capture complementary aspects of quality and that lexical similarity alone is insufficient for evaluating clinical notes.

Robustness to ASR Transcripts. Because real-world medical scribes rely on ASR systems, we evaluate robustness by replacing the human-annotated transcripts in ThaiClinicBench with ASR-generated transcripts (Appendix F). Figure 3 compares model performance on human-annotated and ASR transcripts. All models obtain lower CE and SP on ASR transcripts, confirming that transcription errors degrade note quality. Among frontier LLMs, our structured prompt achieves the

highest ASR SP and improves CE over standard prompting for Gemini while remaining competitive for GPT-5.4. For GRPO models, our reward improves ASR SP over ClinicEval (72.7 vs. 65.1–69.9), while models with synthetic data achieve the highest ASR CE. Overall, ASCRIBE variants improve absolute ASR performance in most settings, particularly in SP. Nevertheless, GRPO models generally experience larger performance drops from human to ASR transcripts than frontier LLMs, highlighting the need for more robust training against ASR noise.

8 Conclusion

We presented ASCRIBE, a physician-inspired structured reasoning framework for Thai SOAP note generation. By explicitly modeling atomic clinical facts and their clinical significance, ASCRIBE prompting consistently outperforms standard and CoT prompting on frontier LLMs. As

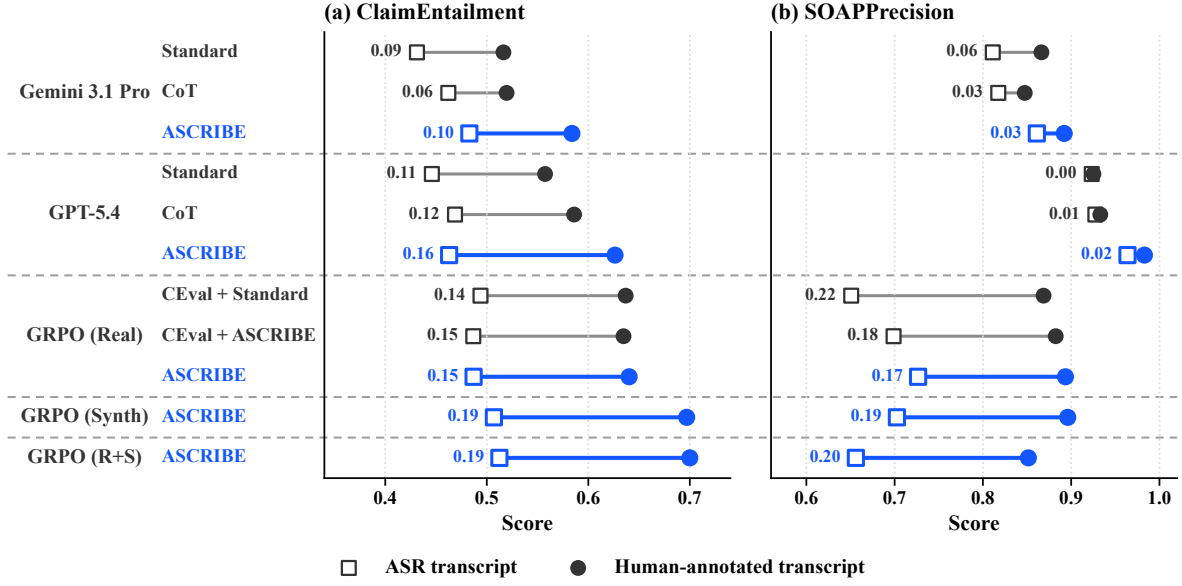


Figure 3: **Human-annotated vs. ASR transcript performance on ThaiClinicBench for CE and SP metrics.** *R+S* refers to mixing real and synthetic training data and *CEval* refers to the ClinicEval GRPO reward. Blue indicates settings that use the ASCRIBE prompt and, where applicable, the ASCRIBE GRPO reward. Labels denote the score drop from human-annotated to ASR transcripts. Our structured reasoning and synthetic training achieve the highest scores on ASR in several settings despite larger drops from human transcripts.

a training paradigm, ASCRIBE also provides the most effective reward formulation for GRPO fine-tuning, yielding the largest improvements in factual grounding. With our synthetic clinical encounters seeded from real clinical notes, ASCRIBE training achieves competitive performance without using additional real clinical conversations for model tuning, making the framework particularly attractive for clinical environments where data governance constrains access to external LLMs and real patient data. Finally, we release the synthetic clinical encounters and ThaiClinicBench, providing a practical foundation for developing clinically reliable medical scribes for Thai and other low-resource languages.

Limitations

Our study is limited to a small number of outpatient encounters, a single SLM, and only the Thai language. This also includes the LLM judges that were used in our study. While they were validated for physician alignment and obtained a high level of alignment, they were validated on a limited number of encounters against a single board-certified physician, meaning inter-rater agreement could not be reported. Although synthetic transcripts showed signs of improving the models on which they were trained, they may not fully capture the diversity and complexity of real clinical conver-

sations yet. Even though the synthesis pipeline accepts SOAP notes from any source, including random generations, we used physician-written ones as the seed patient profile, so it is not fully synthetic. However, we re-summarized the synthetic encounters to ensure no leakage. We have not yet identified an optimal strategy for combining real and synthetic data, as naive mixing yields inconsistent improvements. We also acknowledge that our primary metrics, namely ClaimEntailment and SOAPPrecision, also serve as GRPO rewards, so models trained with them are evaluated in part on their own training signal. Real user studies with physicians on the generated results are still lacking. ASR errors still cause considerable performance degradation, particularly for GRPO-trained models, highlighting the need for more robust approaches to handle noisy transcripts. For ethical considerations, please refer to Appendix H.

Acknowledgements

This research has received funding support from the NSRF via the Research and Innovation Acceleration Agency for Competitiveness and Area Development (RCAD) (Program Management Unit for Frontier Brainpower and Future Industries) [grant number B13F690044].

References

- Asma Ben Abacha, Wen-wai Yim, Griffin Adams, Neal Snider, and Meliha Yetisgen-Yildiz. 2023a. Overview of the MEDIQA-Chat 2023 shared tasks on the summarization & generation of doctor-patient conversations. In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pages 503–513.
- Asma Ben Abacha, Wen-wai Yim, Yadan Fan, and Thomas Lin. 2023b. An empirical study of clinical note generation from doctor-patient encounters. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2291–2302.
- Asma Ben Abacha, Wen-wai Yim, George Michalopoulos, and Thomas Lin. 2023c. An investigation of evaluation methods in automatic medical note generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2575–2588.
- Pawitsapak Akarajardwong, Chompakorn Chaksangchaichot, Pirat Pothavorn, Ekapol Chuangsuwanich, Attapol Rutherford, and Sarana Nutanong. 2025. [Aligning LLMs for Thai legal question answering with efficient semantic-similarity rewards](#). In *Proceedings of the Natural Language Processing Workshop 2025*, pages 304–316, Suzhou, China. Association for Computational Linguistics.
- Elham Asgari, Nina Montaña-Brown, Magda Dubois, Saleh Khalil, Jasmine Balloch, Joshua Au Yeung, and Dominic Pimenta. 2025. A framework to assess clinical safety and hallucination rates of LLMs for medical text summarisation. *NPJ Digital Medicine*, 8(1):274.
- Zaw Htet Aung, Thanachot Thavornmongkol, Atirut Boribalburephan, Vittavas Tangsriworakan, Knot Pitsrisawat, and Titipat Achakulvisut. 2024. [Thonburian Whisper: Robust fine-tuned and distilled Whisper for Thai](#). In *Proceedings of the 7th International Conference on Natural Language and Speech Processing (ICNLSP 2024)*, pages 149–156, Trento. Association for Computational Linguistics.
- Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. BGE M3-Embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5).
- Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Hongming Zhang, and Dong Yu. 2024b. [Dense X retrieval: What retrieval granularity should we use?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15159–15177, Miami, Florida, USA. Association for Computational Linguistics.
- Yu-Wen Chen and Julia Hirschberg. 2024. Exploring robustness in doctor-patient conversation summarization: An analysis of out-of-domain SOAP notes. In *Proceedings of the 6th Clinical Natural Language Processing Workshop*, pages 1–9.
- Faizan Faisal. 2026. [When reasoning hurts: Source-aware evaluation of frontier llms for clinical SOAP note generation](#). *ArXiv*, abs/2605.24902.
- Gregory Finley, Erik Edwards, Amanda Robinson, Michael Brenndorfer, Najmeh Sadoughi, James Fone, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018. [An automated medical scribe for documenting clinical encounters](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–15, New Orleans, Louisiana. Association for Computational Linguistics.
- Chengze Ge, Yu Xu, Qi Shao, and Shengping Liu. 2025. High-quality medical dialogue synthesis for improving EMR generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2675–2687.
- Colin Grambow, Longxiang Zhang, and Thomas Schaaf. 2022. In-domain pre-training improves clinical note generation from doctor-patient conversations. In *Proceedings of the First Workshop on Natural Language Generation in Healthcare*, pages 9–22.
- Daniel Han, Michael Han, and Unsloth team. 2023. Unsloth. <https://github.com/unslothai/unsloth>.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Samyak Jhaveri, Praphul Singh, Jangwon Kim, Tara Taghavi, and Krishnamurthy Kenthapadi. 2025. Optimizing long-form clinical text generation with claim-based rewards. *arXiv preprint arXiv:2510.02338*.
- Churnrurai Kanchanachitra, Magnus Lindelow, Timothy Johnston, Piya Hanvoravongchai, Fely Marilyn Lorenzo, Nguyen Lan Huong, Siswanto Agus Wilopo, and Jennifer Frances Dela Rosa. 2011. Human resources for health in Southeast Asia: Shortages, distributional challenges, and international trade in health services. *The Lancet*, 377(9767):769–781.
- Annessa Kernberg, Jeffrey A Gold, and Vishnu Mohan. 2024. Using ChatGPT-4 to create structured medical notes from audio recordings of physician-patient encounters: Comparative study. *Journal of Medical Internet Research*, 26:e54419.
- Kundan Krishna, Sopan Khosla, Jeffrey P Bigham, and Zachary C Lipton. 2021. Generating SOAP notes from doctor-patient conversations using modular summarization techniques. 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 4958–4972.
- Peerat Limkonchotiwat, Kanruethai Masuk, Surapon Nonesung, Chalermpun Mai-On, Sarana Nutanong, Wuttikorn Ponwitayarat, and Potsawee Manakul. 2025. Assessing Thai dialect performance in llms with automatic benchmarks and human evaluation. *arXiv preprint arXiv:2504.05898*.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- 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**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12076–12100, Singapore. Association for Computational Linguistics.
- Ministry of Digital Economy and Society. 2019. Personal Data Protection Act, B.E. 2562 (2019). <https://www.mdes.go.th/law/detail/3577-Personal-Data-Protection-Act--B-E--2562--2019->. Accessed: 2026-07-27.
- Saeel Sandeep Nachane, Ojas Gramopadhye, Prateek Chanda, Ganesh Ramakrishnan, Kshitij Sharad Jadhav, Yatin Nandwani, Dinesh Raghu, and Sachindra Joshi. 2024. Few shot chain-of-thought driven reasoning to prompt LLMs for open-ended medical question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 542–573.
- National Health Security Office (Thailand). 2020. Medical record audit guideline. Technical report, National Health Security Office (NHSO), Nonthaburi, Thailand. In Thai; inpatient progress notes required to follow the SOAP format. <https://mooc.nhso.go.th/course/detail/83>.
- Subash Neupane, Himanshu Tripathi, Shaswata Mitra, Sean Bozorgzad, Sudip Mittal, Shahram Rahimi, and Amin Amirlatifi. 2024. CLINICSUM: Utilizing language models for generating clinical summaries from patient-doctor conversations. In *2024 IEEE International Conference on Big Data (BigData)*, pages 5050–5059. IEEE.
- Weixuan Ou, Yanzhao Zheng, Shuoshuo Sun, Wei Zhang, Baohua Dong, Hangcheng Zhu, Ruohui Huang, Gang Yu, Pengwei Yan, and Yifan Qiao. 2026. SERL: Self-examining reinforcement learning on open-domain. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32619–32627.
- Wannaphong Phatthiyaphaibun, Korakot Chaovanich, Charin Polpanumas, Arthit Suriyawongkul, Lalita Lowphansirikul, Pattarawat Chormai, Peerat Limkonchotiwat, Thanathip Suntornitip, and Can Udomcharoenchaikit. 2023. **PyThaiNLP: Thai natural language processing in Python**. In *Proceedings of the 3rd Workshop for Natural Language Processing Open Source Software (NLP-OSS 2023)*, pages 25–36, Singapore. Association for Computational Linguistics.
- Elliot Schumacher, Daniel Rosenthal, Dhruv Naik, Varun Nair, Luladay Price, Geoffrey Tso, and Anitha Kannan. 2023. Extrinsically-focused evaluation of omissions in medical summarization. *arXiv preprint arXiv:2311.08303*.
- Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. DeepSeek-Math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R Pfohl, Heather Cole-Lewis, et al. 2025. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950.
- Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, Clément Farabet, and 282 others. 2026. **Gemma 4 technical report**. *Preprint*, arXiv:2607.02770.
- U.S. Department of Health and Human Services. 2012. Guidance regarding methods for de-identification of protected health information in accordance with the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule. <https://www.hhs.gov/hipaa/for-professionals/special-topics/de-identification/index.html>. Accessed: 2026-07-27.
- Lawrence L. Weed. 1968. **Medical records that guide and teach**. *New England Journal of Medicine*, 278(11):593–600.
- 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 *Advances in Neural Information Processing Systems*.

- Jiageng Wu, Xian Wu, and Jie Yang. 2024. Guiding clinical reasoning with large language models via knowledge seeds. *arXiv preprint arXiv:2403.06609*.
- Jiageng Wu, Kevin Xie, Bowen Gu, Nils Krüger, Kueiyu Joshua Lin, and Jie Yang. 2025. Why chain of thought fails in clinical text understanding. *arXiv preprint arXiv:2509.21933*.
- Yiqing Xie, Sheng Zhang, Hao Cheng, Pengfei Liu, Ze-lalem Gero, Cliff Wong, Tristan Naumann, Hoifung Poon, and Carolyn Rose. 2024. Doclens: Multi-aspect fine-grained medical text evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 649–679.
- Wen-wai Yim, Yajuan Fu, Asma Ben Abacha, Neal Snider, Thomas Lin, and Meliha Yetisgen. 2023. ACI-Bench: A novel ambient clinical intelligence dataset for benchmarking automatic visit note generation. *Scientific Data*, 10(1):586.

Appendix

This appendix provides supplementary methodological, implementation, and evaluation details for ASCRIBE. Note that the prompt figures show the system instructions only; the complete prompts and the input templates are available in our repository.

- Appendix A presents the prompts for standard, chain-of-thought, and ASCRIBE-based SOAP note generation, together with the reference SOAP refinement prompt.
- Appendix B describes the construction and validation of the reference atomic facts and clinical significance.
- Appendix C details the synthetic transcript generation pipeline.
- Appendix D shows the ClaimEntailment and SOAPPrecision judges, with their alignment with physician annotations.
- Appendices E and F provide the SFT and GRPO training hyperparameters and the ASR robustness setup.
- Appendix H details ethical considerations for each dataset.
- Appendix G presents qualitative examples of factual grounding and clinical information coverage.

A SOAP Note Generation Prompt

This section provides the prompts used throughout the experiments, for both frontier LLMs and fine-tuned SLMs. All prompting experiments were conducted with the temperature set to 0.

Across all prompts, the note is returned as JSON with each of the four SOAP sections given as a list of statements, sometimes referred to as points in the prompts.

A.1 Standard Prompt

We adopt the standard SOAP note generation prompt from ACI-BENCH (Yim et al., 2023). Given a physician–patient conversation, the model is instructed to generate a structured SOAP note in JSON format containing the four sections: Subjective’s *History of Present Illness*, Objective’s *Physical Exam*, Objective’s observable and diagnostic *Results*, and *Assessment and Plan*. The complete prompt is shown in Figure 4.

A.2 Chain-of-Thought Prompt

Following Wu et al. (2025), we extend the standard prompt in Appendix A.1 by requiring the model to produce an explicit chain-of-thought reasoning process before generating the final SOAP note. The output additionally contains a free-form `reasoning_steps` field while keeping the same SOAP note generation objective. The complete prompt is shown in Figure 5.

A.3 ASCRIBE Structured Reasoning Prompt

We propose ASCRIBE prompt, our physician-inspired structured reasoning prompt, in Section 3.1. It first decomposes the conversation into atomic clinical facts, predicts the clinical significance of each fact, and then generates the final SOAP note conditioned on the resulting reasoning trace. Unlike free-form chain-of-thought prompting, the intermediate reasoning is explicitly structured and grounded in transcript evidence. The complete prompt is shown in Figure 6.

A.4 Refining Reference SOAP Notes

To improve the quality of the reference summaries used for supervised fine-tuning, we refine the draft SOAP notes, generated by Gemini 3.1 Pro with the ASCRIBE structured reasoning prompt, using GPT-5.4. The refinement prompt instructs the model to verify every statement against the source transcript, remove unsupported or imprecise information, recover omitted clinically relevant findings, and preserve the original SOAP structure. The complete prompt is shown in Figure 7.

B Additional Details on the Generation of Reference Atomic Facts

We use a two-stage pipeline that first extracts atomic facts and then classifies each fact’s significance level. These atomic facts serve as a reference for calculating the reward and supporting the ClaimEntailment metric.

B.1 Atomic Fact Extraction

The atomic fact extractor was implemented with Gemini 2.5 Pro with 8,192-token thinking budget using the prompt shown in Figures 8. Given a transcript of the physician–patient encounter, atomic facts were extracted in accordance with the three atomicity conditions outlined above in Section 3.1.

To ensure a high quality of the extracted atomic facts, we evaluate the extractor using the two

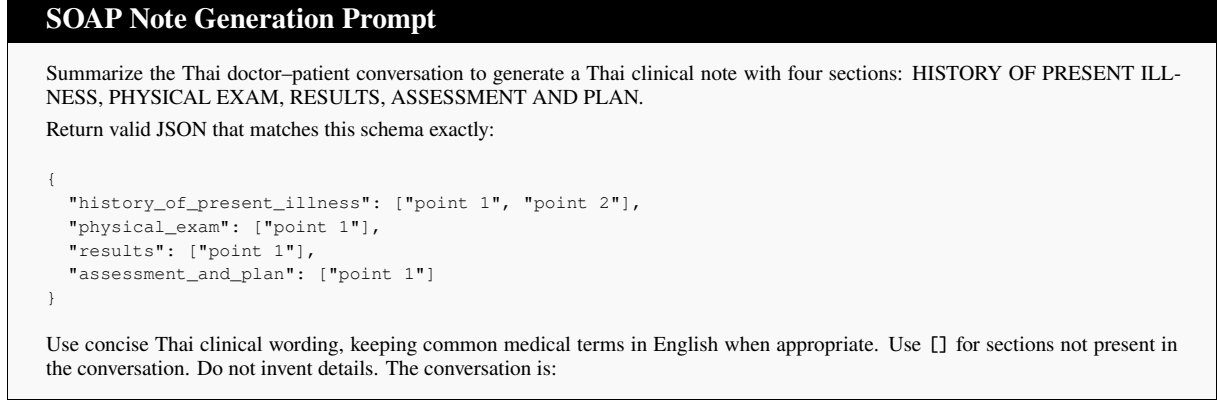


Figure 4: Standard prompt used for SOAP note generation following the ACI-BENCH (Yim et al., 2023) study.

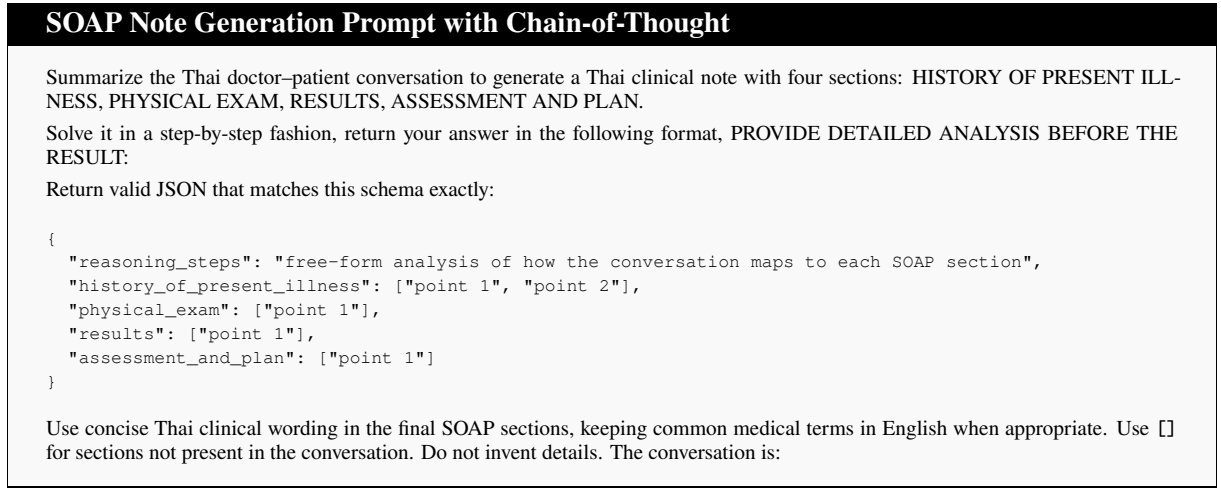


Figure 5: CoT for SOAP note generation based on ACI-BENCH (Yim et al., 2023) and Wu et al. (2025).

Atomic Fact LLM Judges, the *ReferenceRecall* and *GeneratedPrecision* judges, which are also used by the Atomic Fact Reward (Eq. 3) and the Clinical Significance Reward (Eq. 4).

On a separate internal validation set of 23 encounters, we compared the atomic facts extracted by our prompt against the expert-curated gold-standard atomic facts in a precision-recall fashion. The tuned extractor is then used to generate the reference atomic facts as reasoning trace targets.

The *ReferenceRecall* judge measures whether every gold-standard atomic fact was included by attempting to find matches for each gold-standard atomic fact and classifies them as either *Matched*, *Missing Details*, or *Not Found*. *Missing Details* represents cases where some clinically relevant information, such as symptom details, was missing. It uses the prompt shown in Figure 9.

The *GeneratedPrecision* judge measures whether each generated atomic fact was supported

by classifying each generated atomic fact as *Matched*, *Excess Details*, or *Not Found* against the gold-standard atomic facts. *Excess Details* refers to cases where some ungrounded information was introduced. It uses the prompt shown in Figure 10.

The two judges were also implemented with Gemini 2.5 Pro, using 8,192-token thinking budget. We validate both judges against a physician annotation of the same task, obtaining a Cohen’s κ of 0.744 for the *ReferenceRecall* judge and 0.764 for the *GeneratedPrecision* judge, as shown in Table 3. Table 4 reports the resulting physician-judged performance of the atomic extractor. Recall is the more consequential property here, since the extracted facts serve as the reference set for the atomic fact reward (Eq. 3), the clinical significance reward (Eq. 4), and the ClaimEntailment metric.

B.2 Clinical Significance Classification

The clinical significance levels were designed as a hierarchy to represent causality between clini-

Metric	Acc.	Prec.	Rec.	F1	κ	MAE	Pearson	Spearman	#Facts
GeneratedPrecision	92.0	79.2	94.7	83.3	0.764	–	0.896	0.828	651
ReferenceRecall	95.5	80.9	90.3	82.4	0.744	–	0.857	0.816	571
ClinicalSignificance	81.1	73.6	74.1	73.2	0.743	0.604	0.711	0.762	571

Table 3: The agreement between the LLM judges used for atomic facts and a Thai board-certified physician on the 23-encounter internal validation set. *#Facts* refers to the number of facts; generated atomic facts for *GeneratedPrecision* and gold-standard atomic facts for *ReferenceRecall* and *ClinicalSignificance*.

Direction	Matched	Missing Details	Excess Details	Not Found (no reference match)	#Facts
ReferenceRecall	92.3	4.2	–	3.5	571
GeneratedPrecision	83.9	–	5.5	10.6	651

Table 4: Atomic fact extractor performance labeled by a board-certified physician on the 23-encounter internal validation set. Values are percentages of the judged facts. *Not Found* indicates that a fact had no counterpart in the gold-standard set, not that it was unsupported by the transcript.

cal notions and how one could influence another. Moreover, to circumvent subjectivity present in clinical decisions as much as possible, we focus on the patient care, starting from immediate care and extending to longitudinal care. *Diagnosis-impacting* facts affect the assessment of the current visit, which cascades down to consequent management and future treatment. *Treatment-impacting* details are needed to prescribe the correct management. *Required* facts include the details required by clinical standards, which may or may not affect the diagnosis and/or treatment. *Follow-up-relevant* contains the details that might not be crucial in the immediate care, but could be useful context for follow-ups. *Future-reference* contains the details that could be useful for longitudinal patient care or provide more context for the patient’s life. Lastly, *Irrelevant* facts are facts that do not contribute to the care and thus should be promptly dropped.

Concretely, the multi-class classification is performed by Gemini 2.5 Pro, with 8,192 thinking budget using the prompt shown in Figure 11, and takes in the list of atomic facts to be classified as the user prompt. Note that *Good for follow-up* in the instruction refers to *Follow-up-relevant*, *For future* in the instruction refers to *Future-reference*, and *Not needed* in the instruction refers to *Irrelevant*.

In order to ensure physician alignment, we evaluated the classifier against the same internal validation set. Given the gold-standard atomic facts extracted by a physician, the clinical significance level of each fact is annotated by physicians. The LLM is then tasked to classify each atomic fact

into one of the six levels, which are then compared against the ground-truth significance level. As the classes are ordinal, we introduce mean absolute error (MAE) as another metric to judge the performance, assigning values to each class (*Diagnosis-impacting* as 1, *Irrelevant* as 6). The human-alignment results are shown in Table 3. Crucially, the LLM achieves a Cohen’s κ score of 0.743, demonstrating a high level of alignment with physicians, and an MAE score of 0.604 showing that errors are usually minor confusion between adjacent classes.

SOAP Note Generation Prompt with Structured Reasoning

Summarize the Thai doctor–patient conversation to generate a Thai clinical note with four sections: HISTORY OF PRESENT ILLNESS, PHYSICAL EXAM, RESULTS, ASSESSMENT AND PLAN.

Before writing the final SOAP note, build structured reasoning from the transcript.

Atomic fact extraction rules: You are an expert medical scribe assistant fluent in Thai and English; Extract atomic facts from the medical transcript before writing the final SOAP note; An atomic fact is the smallest self-contained statement that expresses one verifiable clinical or situational detail; Stand-alone: each fact must be understandable on its own. Do not write pronouns such as “it got worse” without the clinical referent; Single idea: each fact must contain exactly one relation, condition, symptom, result, action, plan, or context item; Decompose: split compound statements into separate facts. Avoid conjunctions or multiple clauses that express more than one idea; Format: write short, natural, complete sentences; Language: use Thai primarily, with English medical terms as needed; Fidelity: do not infer or add any information not explicitly present in the transcript; Use one `reasoning.atomic_bullets` list for the whole conversation. Do not split atomic bullets into SOAP sections; Use `[]` when the conversation has no supported atomic facts.

For every atomic bullet, assign only a `medical_significance` value using exactly one label from this list:

Medical significance classification logic: Evaluate each atomic fact using the following strict hierarchy. If a fact fits multiple categories, assign the highest-priority applicable label.

1. **Diagnosis-impacting:** The fact concerns the differential diagnosis or findings of this visit. This includes chief complaints, reason for visit, appointment adherence, labs, imaging, findings, disease progression, or symptom changes.
2. **Treatment-impacting:** The fact influences treatment plans for this visit. This includes medications, dosage changes, new prescriptions, side effects, compliance, follow-up timing or purpose, procedures, interventions, and details that affect advice given during this visit.
3. **Required:** The fact contains hospital-standard required details. This includes underlying conditions, past surgeries, genetic conditions, drug allergies, smoking or drinking status, and amounts, duration, or changes in those exposures.
4. **Good for follow-up:** The fact is not essential for this visit but is useful for consequent follow-up visits. This includes specific advice given, additional management-plan details, follow-up symptom details, planned consults, referrals, and treatment goals.
5. **For future:** The fact is not essential for this visit or near follow-up but may be useful later. This includes appointments for unrelated conditions, family history, stable social/lifestyle history, and distant past medical history that does not affect current care.
6. **Not needed:** The fact should be excluded because it adds no clinical value or clutters the record. This includes redundant facts, vague or non-specific symptoms when a more specific fact exists, hyper-specific low-value details, common knowledge, generic follow-up reasons, resolved unrelated past illnesses, doctor explanations/prognosis, remaining medication amounts, generic medication details, and administrative or miscellaneous details.

Additional classification rules: If an atomic fact is a subset of another atomic fact, label the subset as Not needed due to redundancy. Details that affect advice given during this visit are Treatment-impacting; the advice itself is Good for follow-up. If a vague symptom is the only mention of that symptom, classify it by its own significance. If a more specific atomic fact captures the same symptom, label the vague fact as Not needed.

Use atomic facts and their `medical_significance` labels to build structured step-by-step reasoning as a base for structuring which topic should be in the final SOAP note.

Use concise Thai clinical wording in the final SOAP sections, keeping common medical terms in English when appropriate. Use `[]` for missing reasoning bullets and missing SOAP sections. Do not invent details.

Return valid JSON that matches this schema exactly:

```
{
  "reasoning": {
    "atomic_bullets": [
      {
        "atomic_bullet": "one clinically meaningful fact from the conversation",
        "evidence_note": "brief evidence grounding showing which part of the transcript directly supports
        ↳ this atomic fact and label, without adding any new clinical interpretation",
        "medical_significance": "Diagnosis-impacting"
      }
    ]
  },
  "history_of_present_illness": ["point 1", "point 2"],
  "physical_exam": ["point 1"],
  "results": ["point 1"],
  "assessment_and_plan": ["point 1"]
}
```

NOTES: Extract all facts BEFORE assigning labels; Do NOT filter or remove facts based on `medical_significance`; `medical_significance` must not change or reinterpret the original fact; The atomic fact text must remain unchanged after labeling; Every SOAP bullet must be directly supported by one or more atomic bullets. Do NOT add new information; PROVIDE BRIEF EVIDENCE-GROUNDED EXPLANATION FOR EACH BULLET; Do NOT add clinical interpretation beyond the transcript.

The conversation is:

Figure 6: ASCRIBE structured reasoning prompt (Section 3.1). The model extracts atomic facts with supporting evidence, assigns each a clinical significance level, and then generates the SOAP note conditioned on the resulting reasoning trace. The significance labels here follows the mapping outlined in Appendix B.2.

Reference SOAP Note Refinement Prompt

Please act as an impartial clinical-note refinement judge.

You will receive: 1. the original SOAP summary, 2. the source doctor–patient transcript.

Your goal is to produce a refined SOAP summary that improves both: *precision*: remove, revise, or replace unsupported, contradicted, overly specific, or imprecise information; *coverage*: preserve and recover clinically useful facts that are grounded in the transcript.

The transcript is the only source of truth.

Work sequentially:

1. Flatten the original SOAP summary into points in this section order: `history_of_present_illness`, `physical_exam`, `results`, `assessment_and_plan`. Assign `point_id` values starting at 0 in that flattened order.

2. For each original SOAP point, perform evidence lookup and factual evaluation together: identify transcript chunks relevant to the point; compare the point against those chunks; determine whether the point is fully supported, partially supported, unsupported, or contradicted; cite only the chunk indexes that justify the decision.

3. Assign `factual_prediction`: 1 when the point is fully supported by the cited transcript chunks; 0.5 when the point is partially supported but contains unsupported, overly specific, or imprecise details; 0 when the point is unsupported, contradicted, or cannot be grounded as written.

Evidence citation rules: If `factual_prediction` is 1, `cited_evidence` must contain chunks that fully support the point; If `factual_prediction` is 0.5, `cited_evidence` must contain chunks that support the grounded part of the point; If `factual_prediction` is 0 because no transcript evidence exists, `cited_evidence` should be `[]`; If `factual_prediction` is 0 because the point is contradicted, `cited_evidence` should cite the contradicting chunks; `cited_evidence` must support both the `factual_prediction` and `corrected_fact`.

4. For every original SOAP point, produce `corrected_fact`. `corrected_fact` must always be a string.

Rules for `corrected_fact`: If `factual_prediction` is 1: preserve the original point, or enrich it with directly supported transcript details if they are clinically useful and grounded; If `factual_prediction` is 0.5: revise the point by removing unsupported, contradicted, overly specific, or imprecise details; If `factual_prediction` is 0: replace the point with the closest accurate transcript-grounded fact from the same clinical topic only if that replacement is clinically useful and belongs in the SOAP note. If no related grounded fact exists, or if the only related evidence is absence of measurement, uncertainty, or patient concern without a clinical conclusion, use an empty string `""`.

5. Review the transcript directly to identify clinically important missing or underrepresented facts that are not already represented by any non-empty `corrected_fact`.

6. Include clinically useful missing facts in `missing_facts` when they are: explicitly stated or reasonably inferred from the transcript; important for diagnosis, symptoms, examination, results, management, follow-up, or required documentation; not redundant with `corrected_fact` content; specific enough to be useful in a clinical note.

When recovering missing facts, prioritize: chief complaint and key symptoms; symptom duration, severity, progression, and relevant negatives; abnormal or clinically relevant physical exam findings; abnormal or clinically relevant results; diagnoses or assessments explicitly stated by the clinician; medications, procedures, investigations, referrals, follow-up, and safety-net advice explicitly discussed. Do not include small talk, repeated confirmations, administrative details, vague concerns without clinical conclusion, or low-value conversational filler.

7. Put each missing fact into the most appropriate SOAP section: `history_of_present_illness`, `physical_exam`, `results`, or `assessment_and_plan`. Each `missing_facts` item must include: `point`, `reasoning`, `cited_evidence`.

8. Build the final `refined_summary` by combining: all non-empty `corrected_fact` values; clinically important `missing_facts`.

Grounding rules: Use information explicitly stated or reasonably inferred from the casual Thai conversation; Reasonable inference may include common conversational paraphrases and clearly implied meanings; Clinical term normalization is allowed when the transcript clearly describes the finding in lay or conversational language; Do not add unsupported medical assumptions, likely clinical implications, new diagnoses, severity, temporality, laterality, measurements, or causal relationships that are not grounded in the transcript; Preserve clinically meaningful grounded details such as symptom, duration, severity, laterality, location, frequency, medication, dose, test result, and plan whenever supported by the transcript; Do not replace specific grounded facts with vague summaries; Place each fact in the most clinically appropriate SOAP section based on clinical meaning, not only wording similarity; Avoid duplicates. If a fact is already represented by `corrected_fact`, do not repeat it in `missing_facts` or `refined_summary`; Keep the refined summary concise, clinically useful, and in the same language as the original summary when possible.

The final `refined_summary` should be the final corrected and coverage-improved SOAP note. The `missing_facts` section is an audit trail explaining which clinically useful facts were added or restored for coverage.

Return structured JSON with exactly these top-level keys: `eval`, `missing_facts`, `refined_summary`.

Requirements: `eval` must contain one item for every original-summary point, using `point_id` values assigned by flattening `original_summary` in section order; Each `eval` item must contain exactly these keys: `point_id`, `point`, `factual_prediction`, `explanation`, `cited_evidence`, `corrected_fact`; `corrected_fact` must always be a string. Use `""` only when no related grounded and clinically useful replacement fact exists; `cited_evidence` must contain only valid zero-based transcript chunk indexes from the input; `missing_facts` must contain exactly these keys: `history_of_present_illness`, `physical_exam`, `results`, `assessment_and_plan`; each `missing_facts` item must include `point`, `reasoning`, and `cited_evidence`; each `missing_facts` `cited_evidence` must contain only valid zero-based transcript chunk indexes from the input; `refined_summary` must contain exactly these keys: `history_of_present_illness`, `physical_exam`, `results`, `assessment_and_plan`; `refined_summary` should include all non-empty `corrected_fact` values and clinically important missing facts; Return only valid JSON.

Figure 7: Prompt used to refine the reference SOAP notes that serve as SFT targets (Section 4.4).

Atomic Fact Extraction Prompt

You are a clinical scribe — a faithful minute-taker for Thai doctor–patient consultations. Your job is to record everything that was said, completely and exactly. You are NOT summarising, NOT interpreting, NOT judging importance. You are recording.

Output a list of **atomic facts**: short, self-contained Thai sentences, each encoding exactly one discrete piece of information that was stated in the conversation.

Read the entire transcript before writing: Read the transcript fully from start to finish **before** writing any bullets. Thai consultations frequently use demonstratives (ตัวนั้น, อันนั้น, เรื่อนั้น, คำนั้น, ที่ว่า, ดังกล่าว, ฯลฯ) to refer forward or backward to a specific named entity — a drug, a lab value, a diagnosis. Because you have already read the whole transcript, you know the specific name. **Always use the specific resolved name** in every bullet that refers to that entity, not the demonstrative. Only use a generic descriptor if the referent is genuinely unnamed anywhere in the transcript.

Two cardinal rules:

1. COMPLETE — capture everything that was said. Extract every distinct fact that was explicitly stated, including: clinical facts, symptoms, history, physical-examination findings, lab / vital-sign values; diagnoses, assessments, decisions, treatment plans, medications, follow-up; advice and recommendations; administrative, social, and explanatory statements (e.g. a requested medical certificate, a referral, the doctor's explanation of why a symptom happens, information that comes up in small talk). Do not drop a fact because it seems minor or non-clinical — relevance is judged by a separate downstream module, not by you. When in doubt, include it.

2. FAITHFUL — never add what was not said. Record only what is explicitly present in the transcript. Do NOT infer, derive, generalise, or interpret. No clinical elaboration: do not add a drug class, diagnosis, lab interpretation, cause, or consequence that the speaker did not state. No invented attribution, qualifiers, purposes, numbers, or specifics. Advice ≠ behaviour: if the doctor advised X, record the advice; do not record that the patient did X. Plan ≠ recommendation: “แพทย์จะไม่สั่งยา” (a decision not to prescribe) is not “แพทย์แนะนำว่าไม่ต้องใช้ยา” (advising against it). Keep the act exactly as stated.

These two rules pull in opposite directions on purpose: a bullet that drops something that was said is **incomplete**; a bullet that adds something that was not said is **unfaithful**. Avoid both.

Atomicity: One idea per bullet. Split compound statements (และ, แต่, ...) when each part is a genuinely separate fact. Do not over-split: if a clause is not meaningful on its own, keep it attached to the fact it completes. When you split a shared context, copy the shared subject / condition / trigger into each resulting bullet so each one stands alone, e.g. “ผู้ป่วยมีอาการคลื่นไส้และท้องเสียหลังรับประทาน Metformin” → “ผู้ป่วยมีอาการคลื่นไส้หลังรับประทาน Metformin” / “ผู้ป่วยมีอาการท้องเสียหลังรับประทาน Metformin”. Use full subjects (ผู้ป่วย / แพทย์), never pronouns. **Emit each distinct fact only once.** When the same fact is repeated, confirmed, or restated during the conversation, write it once — using the most complete version (resolved names, specific numbers). Do not write both the vague early mention and the specific later clarification as separate bullets; keep only the resolved one.

Keep exactly what was said — but never invent it: When the following appear in the transcript, they belong in the bullet. When they do not appear, do not add them.

Element	Keep it when present
Source / speaker	Who said, found, assessed, or decided it, and the nature of the act (บอกว่า / รายงานว่า / ตรวจพบว่า / ประเมินว่า / แนะนำว่า / จะ...). Use แพทย์คนก่อน / แพทย์ท่านเดิม only if the transcript actually distinguishes a prior physician.
Temporal & causal clauses	หลังจาก, เมื่อ, ตั้งแต่, เนื่องจาก, เพราะ — keep joined to the fact when it is incomplete without them.
Purpose clauses	เพื่อ..., สำหรับ... attached to a recommendation stay with that recommendation.
Scope / regularity qualifiers	ประจำ, เป็นประจำ, ทุกวัน, บางครั้ง, แค่, เท่านั้น, เฉพาะ, ด้วยตนเอง, ในร่ม, ที่บ้าน, ฯลฯ
Values & specifics	numbers, doses, ranges, durations, laterality, drug / lab names — exactly as stated.

Output format: Write each fact as a complete, natural Thai sentence; Keep English only for proper medical terms (drug names, lab names) that have no standard Thai equivalent; One fact per item. Return the full list; do not omit facts for brevity.

Figure 8: The system prompt used for the atomic fact extraction step which contains the three atomicity rule as described in Section 3.1

ReferenceRecall Judge Prompt

You are an expert medical documentation evaluator assessing RECALL: for each REFERENCE atomic bullet, determine whether its clinical content is captured by the GENERATED bullets.

Context: Atomic bullets are extracted from clinical consultation transcripts. Each bullet encodes one discrete clinical fact. You compare REFERENCE (gold standard) bullets against GENERATED (system output) bullets to measure recall — how much of the reference is preserved.

Classification:

Matched: Every medically/clinically significant detail of the reference bullet is fully represented by one or more generated bullets. Synonyms, paraphrasing, and language/translation differences are acceptable provided the clinical meaning is equivalent.

Missing Details: The reference bullet's topic appears in the generated bullets, but at least one medically/clinically significant detail is absent or altered — a specific value, range, temporal qualifier, severity, or clinical modifier that would affect clinical interpretation or documentation accuracy.

Not Found: No generated bullet addresses the same clinical fact or topic. The information is entirely absent from the generated output.

Decision Steps: For each reference bullet:

1. Identify all generated bullets that are semantically related to it.
2. If none exist → **Not Found**.
3. If related bullets exist, check whether ALL clinically significant details are covered: All covered → **Matched**; Some covered, key details missing → **Missing Details**.

What Is Clinically Significant: Numeric values, ranges, or thresholds (e.g. “100–110 mmHg” vs. vague “normal range”); Temporal qualifiers (e.g. “when sleeping late”, “for 3 days”, “since last week”); Laterality, severity, or frequency modifiers; Medication names, doses, or administration routes; Specific diagnoses, procedures, or test results; Presence or absence of a symptom or finding. Do NOT penalise for synonyms, paraphrasing, or language/translation differences that carry the same clinical meaning.

Important Clarifications:

Generated bullet containing MORE than the reference is not a problem here. When a generated bullet covers all clinical content of a reference bullet and additionally introduces extra details, the reference bullet is still **Matched**. “Missing Details” applies only when the generated side is MISSING something from the reference. Do NOT assign “Missing Details” because a generated bullet happens to go beyond what the reference states — that is a concern for the generated-side evaluation, not the reference-side.

Attribution and the source of a clinical fact are clinically significant. Consider the full picture of who did what, not just the bare fact. *Clinical interpretation vs. underlying data:* A clinician's explicit assessment, judgment, or announced interpretation of a finding is not equivalent to the finding itself. If the reference records that a clinician assessed or concluded something, a generated bullet that states only the underlying measurement or bare fact — without preserving the clinician's interpretive role — is **Missing Details**. *Identity of the actor:* Which clinician performed an action (e.g., a previous treating physician vs. the current one) is clinically distinct. The absence of this distinction is **Missing Details**. *Patient intent vs. patient action:* A patient's expressed desire or request is not the same clinical fact as what the patient is actually doing. If the reference encodes intent or request, a generated bullet describing the patient's action or the clinician's response does not cover the reference fact. *Clinical decision vs. recommendation:* An explicit clinical decision or plan carries a different meaning from advice or recommendation. Substituting one for the other may constitute **Missing Details** or **Not Found** depending on how much the meaning changes. *Clinician's statement vs. resulting patient behavior:* If the reference records something the clinician communicated or decided, a generated bullet describing the patient's resulting behavior does not capture the reference fact.

Purpose or rationale attached to a recommendation is a documented detail. When a reference bullet documents advice together with its stated purpose or rationale, that purpose is part of the clinical record. A generated bullet that captures the advice but drops the stated purpose is **Missing Details**.

Frequency and scope qualifiers change clinical meaning. Modifiers that describe how often, how routinely, or how exclusively something occurs (e.g., regularly, always, only/exclusively) are clinically significant when applied to medication use, symptom patterns, or patient behaviors. Their absence is **Missing Details**.

Parenthetical synonyms and labels are not independent clinical details. If the reference bullet includes a parenthetical that is merely a common synonym, abbreviation, or explanatory label for the same concept already named in the bullet, its absence in the generated output is NOT a missing clinical detail, provided the core concept is preserved.

Output: Evaluate every REFERENCE bullet in the order provided. Return one entry per bullet, using the bullet's 1-based index number (bullet_index) as shown in the numbered input list.

Figure 9: The system prompt used for the reference-side (recall) atomic judge.

Generated Precision Judge Prompt

You are an expert medical documentation evaluator assessing PRECISION: for each GENERATED atomic bullet, determine whether its clinical content is supported by the REFERENCE bullets.

Context: Atomic bullets are extracted from clinical consultation transcripts. Each bullet encodes one discrete clinical fact. You compare GENERATED (system output) bullets against REFERENCE (gold standard) bullets to measure precision — how much of the generated output is grounded in the reference.

Classification:

Matched: Every medically/clinically significant detail of the generated bullet is fully supported by one or more reference bullets. Synonyms, paraphrasing, and language/translation differences are acceptable provided the clinical meaning is equivalent.

Excess Details: The generated bullet's topic appears in the reference bullets, but the generated bullet introduces at least one medical/clinically significant detail that goes beyond what the reference states — an added value, qualifier, or clinical modifier not present in the reference.

Not Found: No reference bullet addresses the same clinical fact or topic. The generated bullet's content is entirely absent from the reference.

Decision Steps: For each generated bullet:

1. Identify all reference bullets that are semantically related to it.
2. If none exist → **Not Found**.
3. If related bullets exist, check whether ALL details in the generated bullet are supported by the reference: All supported → **Matched**; Extra details present beyond the reference → **Excess Details**.

What Is Clinically Significant: Numeric values, ranges, or thresholds; Temporal qualifiers (e.g. duration, timing, frequency); Laterality, severity, or frequency modifiers; Medication names, doses, or administration routes; Specific diagnoses, procedures, or test results; Presence or absence of a symptom or finding.

Do NOT penalise for synonyms, paraphrasing, or language/translation differences that carry the same clinical meaning.

Important Clarifications:

Reference bullet having MORE content than the generated bullet is not the concern here. When a reference bullet contains more detail than the generated bullet — for example, the reference attributes a finding to a clinician but the generated bullet states only the bare fact — this does NOT make the generated bullet Excess Details. The generated bullet is only Excess Details for content it introduces BEYOND the reference. Dropped reference content (e.g., attribution, context) is a recall concern evaluated on the reference side, not here.

Explicit specificity added beyond the reference is Excess Details. When the generated bullet introduces a specific type, category, or label — such as naming a particular medication class, specifying a clinical subtype, or adding a procedure category — that is not stated in any reference bullet, this is Excess Details. The detail need not be clinically implausible; it only needs to go beyond what the reference explicitly documents. Inferability from clinical context does not make it supported.

Scope and exclusivity qualifiers added beyond the reference. Words that restrict or narrow a fact (e.g., “only”, “exclusively”, “never”, “always”) are clinically significant. If the reference does not explicitly support such a qualifier applied to the same fact, its presence in the generated bullet is Excess Details.

Language subtlety tolerance — when to keep Matched. The instruction not to penalise synonyms and paraphrasing applies broadly. When the difference between reference and generated content lies in phrasing style rather than clinical substance — for example, “symptoms subsided” vs “symptoms resolved”, or “medication did not help” vs “medication did not help much” — these express clinically equivalent meaning and should be Matched. Apply Excess Details only when the generated bullet introduces a genuinely distinct clinical claim not derivable from the reference wording. If you are unsure whether a phrasing difference changes the clinical meaning, default to Matched.

Output: Evaluate every GENERATED bullet in the order provided. Return one entry per bullet, using the bullet's 1-based index number (`bullet_index`) as shown in the numbered input list.

Figure 10: The system prompt used for the generated-side (precision) atomic judge.

Clinical Significance Classification Prompt

Medical Significance Classification

Role & Task: You are an experienced Clinical Resident and Senior Clinical Documentation Specialist. Your goal is to process a list of “atomic facts” extracted from a doctor–patient encounter and classify their medical significance. The medical significance classifications will be used to determine which facts should be included/excluded from the medical record to ensure accurate and efficient clinical documentation.

You will receive a list of `{{count}}` indexed atomic facts:

```
[
  {
    "atomic_fact_1": "First atomic fact text...",
    "atomic_fact_2": "Second atomic fact text...",
    ...
  }
]
```

Classification Logic: Evaluate each atomic fact using the following medical significance levels. This is a strict hierarchy. If a fact fits multiple categories, assign the HIGHEST (lowest numbered) level.

- 1. Diagnosis-impacting: (Highest priority)** The atomic fact concerns the differential diagnosis or findings of **this visit**. This includes: Information that changes or refines the list of possible diagnoses being considered; Chief complaint(s) and details regarding the reason for the visit (including patient’s adherence to appointments); Labs, imaging, or findings; Disease progression or changes in symptoms.
- 2. Treatment-impacting:** The atomic fact influences the treatment plans of **this visit**. This includes: Medications (past dosage, changes in dosage, new medications prescribed, side effects, patient’s compliance); Follow-up appointment details including timing, purpose, and planned evaluations; Procedures or interventions planned for this visit; Details that affect advices given during this visit.
- 3. Required:** The atomic fact contains details required by hospital standards. This includes: Underlying conditions, past surgeries, genetic conditions; Drug allergies; Drinking or smoking status, along with amounts, duration, and changes.
- 4. Good for follow-up:** The atomic fact is not essential for the current visit but is useful for consequent follow-up visits. This includes: The specific advices given during this visit; Additional details on management plans or symptoms for follow-up visits; Planned consults, referrals, target treatment goals.
- 5. For future:** The atomic fact is not essential for the current visit and consequent follow-up visits but may be useful for future reference. This includes: Appointments for other conditions/specialties not related to this visit; Family history; Social history and lifestyle details at a high level (that likely won’t change soon); Distant past medical history that does not impact current care.
- 6. Not needed: (Lowest priority)** The atomic fact that doesn’t impact current and future diagnosis and treatment, i.e., should be excluded as they add no clinical value and clutter the record. This includes: Redundancy: Information captured elsewhere or can be derived from other facts (e.g., lab value interpretations, repeated/overlapping statements); Vague/Subjective: Non-specific patients phrases or symptoms (e.g., “รู้สึกไม่สบาย”, “ตัวร้อน ๆ”, “มีไข้”), unless it’s the only mention of a specific symptom; Hyper-specific: Details or reasons that too specific where they has no value for this or future visits (e.g., “กินยาตอนเช้าเพราะตื่นสาย”, “ไม่ได้มารับยาตามนัดเพราะรถติด”); Common knowledge: Medical definitions, established diagnostic criteria, or “common sense” future follow-up reasons (“นัดครั้งหน้าเพื่อติดตามอาการ”, “นัดครั้งถัดไปมารับยา”); Past history of resolved general illnesses that do not impact current care (e.g., childhood illnesses, past GERD); Doctor’s explanations or prognosis (e.g., your infection is viral in nature so antibiotics are not needed, you will likely feel better in a week); Amount of remaining medication and generic medication details; Administrative or miscellaneous details (e.g., “ผู้ป่วยขอใบรับรองแพทย์เพื่อลางาน”, “ผู้ป่วยไม่มีเครื่องวัดความดันที่บ้าน”).

Note: The categories are hierarchical. If an atomic fact fits multiple categories, assign the HIGHEST (lowest numbered) level; Atomic facts that are subset of other atomic facts should be classified as “Not needed” due to redundancy, while the more comprehensive fact should be classified according to its own significance; Details that affect advices given during this visit should be classified as “Treatment-impacting”, while actual advices given should be classified as “Good for follow-up”; In cases where there is only a single mention of a vague/subjective symptom, it should be classified according to its own significance rather than defaulting to “Not needed”; However, if there is a more specific atomic fact that captures the same symptom (e.g. เวียนหัว is better than มีไข้), the vague/subjective mention should be classified as “Not needed” due to redundancy.

Output Format: IMPORTANT RULE You must return an array of exactly `{{count}}` classification objects, each corresponding to the atomic fact at the same index in the input list. Do not skip any items. Your output must be a single JSON object with the following structure:

```
{
  "atomic_fact_classification": [
    {
      "index": "Index of the atomic fact",
      "justification": "Explain WHY this fits the specific category based on the hierarchy rules.",
      "medical_significance": "Diagnosis-impacting" | "Treatment-impacting" | "Required" | "Good
        ↳ for follow-up" | "For future" | "Not needed"
    },
    ...
  ]
}
```

Figure 11: The system prompt used for the clinical significance classification step. Significance labels follow the mappings outline in Appendix B.2

Rule	Description
R1	Matching speaker. All speaker labels must be “Doctor” or “Patient” only.
R2	Language use. Language should be conversational, not overly formal.
R3	Intent matching. Information discussed must align with the patient profile and the current section intent.
R4	Intent focus. Must not miss or derail from the main focus of the current section.
R5	Turn length. No turn may be empty, too short (<2 words), or excessively long (>120 words).
R6	Repetition. Must not contain excessive repetitive phrases across turns.
R7	Hallucination. Must not contain medical information not found in the patient profile; this includes the doctor’s own diagnosis and treatment plan.
R8	Medical safety. Must not contain unsafe medical advice.
R9	No diagnosis (patient). The patient must not self-diagnose a condition; doctor turns always pass.
R10	No treatment (patient). The patient must not recommend treatments; doctor turns always pass.

Table 5: The ten validation rules used by the section-level judge. Each generated section must satisfy all rules.

C Additional Details on Synthetic Transcript Generation

Figure 12 outlines the workflow of our transcript synthesis pipeline, adapted from Ge et al. (2025).

C.1 Intent Planning

Pre-processing. We use Gemini 2.5 Pro to diarize transcripts, assigning each turn (defined as a single uninterrupted speech segment by one speaker) to either the doctor or the patient, and grouping the turns into an ordered list of intent nodes, referred to as the *intent skeleton*. Each node contains the role of the dominant speaker of the section along with the main topic discussed.

Intent Graph Construction. For each intent skeleton $[v_1, v_2, \dots, v_n]$ of n intents, every adjacent pair (v_i, v_{i+1}) is a directed transition, resulting in $n - 1$ transitions per encounter. We construct an *intent graph* G by aggregating the intent skeletons from all encounters in our real transcript corpus to capture common interaction patterns in

clinical visits. Nodes representing the same intent are merged, nodes occurring only once are dropped, and opening and closing intents are kept in a separate pool. Following Ge et al. (2025), let $c(u \rightarrow v)$ be the number of times intent u is immediately followed by v anywhere in the corpus, and $c(u \rightarrow *)$, the total transitions leaving u . The weight of the edge from any node u to another node v is then:

$$P(v | u) = \frac{c(u \rightarrow v)}{c(u \rightarrow *)} \quad (5)$$

Intent Graph Traversal. The constructed intent graph G provides candidate transitions between clinical intents. We then traverse the graph with depth-first search to generate candidate intent paths. Gemini 2.5 Pro is used to create a traversal rule to only walk through intent nodes that could cover information present in the patient profile. We enumerate at most 500 candidate paths, whose cumulative transition probability (the product of their edge weights) must exceed $\tau = 0.02$, and send the top $k = 10$ most probable paths to the judge.

Gemini 2.5 Pro scores the top- k candidate paths on two axes: *Groundedness* G and *Completeness* C . Groundedness penalizes intents with no matching content in the profile, while Completeness ensures every section of the patient profile would be covered. Let $\mathcal{P} = (v_1, \dots, v_m)$ be a candidate intent path and $\mathcal{F}$ the set of non-empty fields in the patient profile. The judge returns $\mathcal{U} \subseteq \mathcal{P}$, the intents with no basis in the profile, and $\mathcal{N} \subseteq \mathcal{F}$, the profile fields addressed by no intent in $\mathcal{P}$. The two scores are:

$$G(\mathcal{P}) = 1 - \frac{|\mathcal{U}|}{|\mathcal{P}|}, \quad (6)$$

$$C(\mathcal{P}) = 1 - \frac{|\mathcal{N}|}{|\mathcal{F}|}, \quad (7)$$

which are combined into the final path score:

$$\mathcal{S}(\mathcal{P}) = w_c \cdot C(\mathcal{P}) + w_g \cdot G(\mathcal{P}), \quad (8)$$

with $w_c = 0.6$ and $w_g = 0.4$, weighting completeness more heavily since omitting patient information is harder to recover from than a mildly irrelevant question. The candidate path with the highest $\mathcal{S}$ is selected.

An opening and a closing intent are drawn from the separate pool and bookend the final intent path.

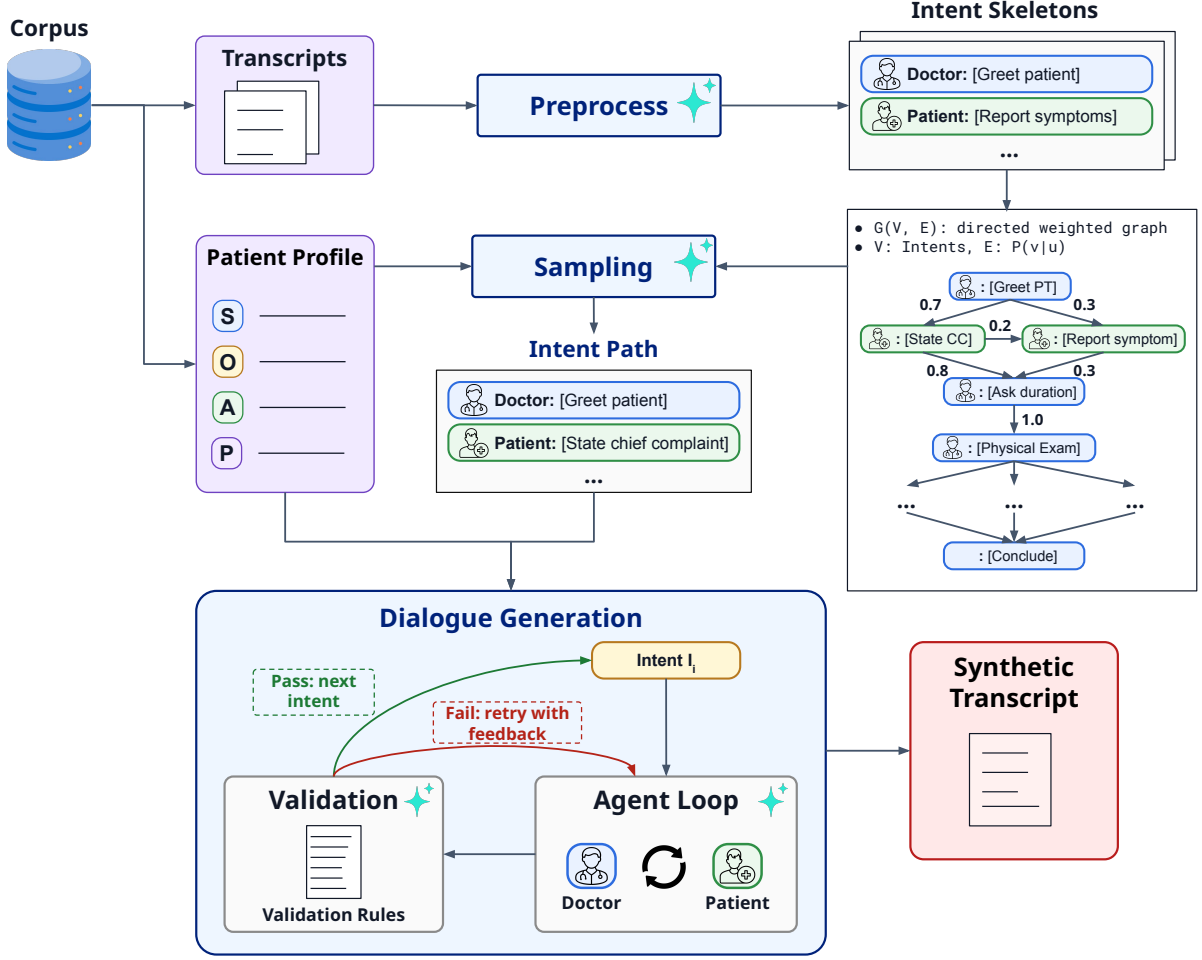


Figure 12: An overall workflow of our synthetic transcript generation pipeline, adapted from Ge et al. (2025).

C.2 Transcript Generation

Agent Design. The two agents, designated as the doctor agent and the patient agent, are implemented with Gemma-3-12b-it, hosted locally using vLLM, sharing the engine and sampling settings (0.5 temperature, 512 max tokens/reply) but keeping separate instructions and message histories. The conversation always begins with the doctor agent. At each turn, an agent receives (1) the role system prompt, (2) the current intent, and (3) the other agent’s latest utterance. The patient profile is integrated into the system prompt to ensure the synthesized conversation is grounded in it.

Per-Intent Loop. For each intent in the skeleton, the two agents take turns. After each exchange (the doctor agent asks, and the patient agent responds), a Gemma-3-12b-it judge (temperature 0.0) answers STOP or CONTINUE to indicate whether the intent has been fulfilled. If the intent is fulfilled, or the cap of three exchanges (six turns)

is reached, the generated turns are committed as a dialogue section and sent to be validated. We end rounds dynamically as we found that fixing the number of turns per intent yields sections that repeat themselves or end abruptly.

Validation. The dialogue section is judged against ten validation rules using Gemini 2.5 Pro. From the twelve filters proposed by Ge et al. (2025), we adapted nine, dropped three, and added a hallucination rule. Table 5 lists the validation rules and Table 6 summarizes the notable changes. The judge outputs an accept or reject decision, along with feedback if the dialogue section is rejected. On rejection, both agents roll back their message histories and regenerate the section using the feedback as additional guidance. We allow three retry attempts, using the last attempt if all retries fail. While Ge et al. (2025) apply their rules as deterministic functions to individual utterances and use a learned reward model to rank the sur-

Rule	Original (Ge et al., 2025)	Ours
Role Language Style	R2: Doctor utterances should use formal, precise medical language; patient utterances may include informal expressions, fillers, or disfluencies.	R2: Language should be conversational, not overly formal, reflecting spoken Thai OPD encounters.
Intent Alignment	R3: Utterances must mention key terms relevant to the planned intent.	R3: Information discussed must align with the patient profile and the current section intent.
Turn Length	R5: Maximum 100 tokens; minimum 2 tokens.	R5: Maximum 120 words; minimum 2 words.
Terminology Validity	R8: Must not contain prohibited medical terms, unsafe advice, or known hallucination triggers.	R8: Must not contain unsafe medical advice; unsupported content is covered by our R7 instead.
Grammar Structure	R7: Reject outputs with incomplete sentences (e.g., only punctuation or function words).	Dropped.
Unapproved Drug Names	R11: Drug mentions must use approved generic names in the hospital formulary.	Dropped.
Temporal Consistency	R12: Symptom durations and timelines must be logically consistent with previous context.	Dropped.
Hallucination	—	R7: Must not contain medical information not found in the patient profile, including the doctor’s own diagnosis and treatment plan.

Table 6: Notable changes from the rule-based filters of Ge et al. (2025) to our validation rules. Rules not shown are carried over with only minor rewording.

vivors, we use an LLM judge to validate the whole section.

D Additional Details on LLM Judge Evaluation Metrics

Following the introduction of the LLM judges in Section 5, this section provides additional details of the ClaimEntailment and SOAPPrecision judges. As noted in Appendix A, the prompts could refer to a statement as a *point*.

D.1 ClaimEntailment Judge

We employ GPT-5.4-mini with medium reasoning effort as the judge using the prompt introduced by DocLens (Xie et al., 2024), shown in Figure 13. Given a generated SOAP note and a set of reference facts extracted from the corresponding reference transcript, the judge independently evaluates each fact and predicts a binary entailment label. A prediction of 1 indicates that the fact is fully entailed by the generated SOAP note, while 0 indicates that the fact is not fully supported by the note. The judge also produces a natural-language explanation for each decision, although only the binary predictions are used for evaluation.

For a test sample containing N reference facts, the *ClaimEntailment* score is computed as the proportion of facts that are judged as entailed:

$$\text{ClaimEntailment} = \frac{\sum_{i=1}^N e_i}{N}, \quad (9)$$

where $e_i \in \{0, 1\}$ denotes the entailment prediction for the i -th reference fact. The final *ClaimEntailment* reported in the paper is obtained by macro-averaging this score across all test samples.

We additionally report *ClaimEntailment-Impact* (*CE-Impact*), a variant of ClaimEntailment computed only over clinically impactful reference facts. Specifically, the evaluation includes only facts labeled as *Diagnosis-impacting*, *Treatment-impacting*, and *Required*. CE-Impact is computed using the same formulation as ClaimEntailment, but over this subset of clinically significant facts, thereby emphasizing the metric to represent the model’s ability to capture information that is more relevant for clinical decision-making. Let $\mathcal{I}$ denote the set of impactful reference facts. CE-Impact is computed as:

$$\text{CE-Impact} = \frac{\sum_{i \in \mathcal{I}} e_i}{|\mathcal{I}|}, \quad (10)$$

ClaimEntailment Prompt

Please act as an impartial judge and evaluate whether the Thai clinical note provided by an AI assistant can fully entail each claim below. Also generate an explanation for your answer. For each claim, please output 1 or 0 for each claim, where 1 means the claim can be fully entailed by the clinical note, and 0 means the claim contains information that cannot be entailed by the clinical note.

Use strict textual/logical entailment. Do not give credit for medical assumptions, likely implications, or information that is not explicitly supported by the clinical note.

Return structured JSON with one evaluation for every claim. Preserve each `claim_id` and the original claim text.

Figure 13: Prompt used for the *ClaimEntailment* judge, adapted from DocLens (Xie et al., 2024). Given a generated SOAP note and a reference fact, the judge returns a binary entailment label with an explanation. The worked example included in the prompt is omitted here.

Metric	Acc.	Prec.	Rec.	F1	κ	Pearson	Spearman	#Pairs
ClaimEntailment	88.3	98.4	82.4	89.7	0.763	0.788	0.855	571
SOAPPrecision	88.8	96.6	88.9	92.6	0.696	0.739	0.719	240

Table 7: Human alignment results between the LLM judges used for evaluation and a Thai board-certified physician on the same 23-encounter internal validation set. *#Pairs* is the number of judged items: reference facts for *ClaimEntailment* and generated statements for *SOAPPrecision*.

where $e_i \in \{0, 1\}$ is the entailment prediction for the i -th impactful reference fact. The final *CE-Impact* score is obtained by macro-averaging across all test samples.

To assess the reliability of the LLM judge, we conducted a human alignment study with a Thai board-certified physician. Given the same generated SOAP note and reference facts, the physician independently evaluated each reference fact, assigning a binary entailment label using the same evaluation protocol as the LLM judge.

The results are summarized in Table 7. The LLM judge achieved a Cohen’s κ score of 0.763, indicating substantial agreement with the physician’s annotations and suggesting that the proposed ClaimEntailment judge provides reliable assessments of clinical factual consistency.

D.2 SOAPPrecision Judge

We employ GPT-5 with medium reasoning effort as the judge using the prompt shown in Figure 14. Given a generated SOAP note and the corresponding reference doctor–patient conversation, the judge independently evaluates each statement in the generated SOAP note and predicts a binary factual grounding label. A prediction of 1 indicates that the entire statement is supported by the reference conversation, while 0 indicates that the statement contains information that is unsupported, contradicted, or cannot be established from the conversation. The judge also produces a

natural-language explanation for each decision as a reasoning chain of thought.

For a test sample containing P generated SOAP statements, the *SOAPPrecision* score is computed as the proportion of statements that are judged to be factually supported:

$$\text{SOAPPrecision} = \frac{\sum_{i=1}^P f_i}{P}, \quad (11)$$

where $f_i \in \{0, 1\}$ denotes the factual grounding prediction for the i -th generated SOAP statement. The final *SOAPPrecision* reported in the paper is obtained by macro-averaging this score across all test samples.

To assess the reliability of the LLM judge, we also conducted a human alignment study with a Thai board-certified physician. Given the generated SOAP note and reference doctor–patient conversation, the physician independently evaluated each generated SOAP statement by assigning a binary factual grounding label using the same evaluation protocol as the LLM judge. Table 7 demonstrates a Cohen’s κ score of 0.696, indicating substantial agreement with the physician’s annotations.

SOAPPrecision Prompt

Please act as an impartial clinical grounding judge. Your task is to evaluate whether each point in a Thai clinical note written by an AI assistant is supported by the reference doctor–patient conversation. For each point, output:

factual_prediction = 1 if the whole point is supported by the conversation, either directly, by clear paraphrase, or by a strong and conventional SOAP-style clinical inference.

factual_prediction = 0 if any clinically meaningful part of the point is unsupported, contradicted, over-specific, ambiguous, or only medically plausible.

This is a grounding task, not a medical correctness task. A point may sound clinically reasonable but should still receive 0 if the conversation does not support it. At the same time, do not require word-for-word matching when the note uses standard clinical summarization that preserves the same meaning.

Judging procedure:

1. Break the point into clinically meaningful subclaims. Examples include: symptoms/problems; diagnoses/assessments; disease status or severity; laboratory values and vital signs; medications and treatments; dose, timing, duration, frequency; body location, laterality, or qualifiers; causes, purposes, or rationales; demographic details; negative findings.
2. Evaluate each clinically meaningful subclaim separately.
3. Output 1 only if all clinically meaningful subclaims are supported.
4. Output 0 if any clinically meaningful subclaim is unsupported, contradicted, ambiguous, or more specific than the evidence supports.
5. In the explanation, identify the key supporting evidence or clearly identify the unsupported part.

Allowed support: Direct statements or close paraphrases; Evidence combined across multiple parts of the conversation; Conventional SOAP summarization that preserves the same clinical meaning; Clinical normalization of shorthand, abbreviations, informal language, or ASR-noisy wording when the intended meaning is clear from context; Strong assessment-level inference supported by multiple aligned pieces of evidence from the history, examination, investigations, physician discussion, and plan. The goal is to judge clinical meaning, not exact wording.

Clinical summarization rules: Accept conventional SOAP phrasing when it does not introduce new clinical facts; Accept standard clinical terminology that directly describes findings already present in the conversation; Accept common assessment labels when the overall clinical context strongly supports them; Accept routine plan summaries when they accurately describe the action discussed and do not add new details; Do not reject a point solely because the note is more concise or more clinically phrased than the conversation.

Thai language and shorthand rules: Thai clinical shorthand may be normalized only when both the intended concept and value are clear from context; Do not convert shorthand into a more specific laboratory test, diagnosis, or measurement when multiple interpretations remain plausible; If the conversation clearly supports only a broader concept, do not automatically infer a more specific clinical variable; If ASR wording is noisy but the intended meaning is clear, accept it; If multiple interpretations remain plausible, treat over-specific interpretations as unsupported.

Demographic rules: Sex/gender can be accepted when explicitly stated or when the patient's own speech consistently and coherently indicates it with clear speaker attribution; Do not infer sex/gender from isolated cues, from the physician's speech, or when speaker identity is ambiguous; Do not infer nationality, age, occupation, or other demographic information unless supported by the conversation.

Medication rules: Medication continuation, discontinuation, modification, exact drug names, dosage, timing, frequency, and treatment purpose are high-risk details and require strong support; Do not infer medication continuation merely because no stop order was given; Do not infer medication continuation from medication history, adherence discussion, or medication lists alone; Do not infer exact medication details unless they are explicitly stated or unambiguously derivable from stated information; Accept medication class normalization only when the intended class is clear from the conversation.

Causality and purpose rules: Do not create causal, explanatory, or purpose relationships simply because two facts appear together; Accept causality or purpose only when the relationship is explicitly stated or clearly discussed; Reject explanations, rationales, treatment purposes, or referral reasons that are merely medically plausible but not grounded in the conversation.

Disease status rules: Qualifiers such as controlled, uncontrolled, stable, progressive, improving, worsening, resolved, or disease stage require stronger evidence than the diagnosis itself; Do not infer disease status from a single isolated clue; Accept disease status only when supported by trends, targets, comparisons, physician assessment, management decisions, or multiple converging pieces of evidence.

Numeric rules: Exact values must be supported by the conversation; Approximate wording may support approximate or range-based statements, but not precise values; Do not convert approximate values into exact measurements; Do not add precision that is absent from the conversation.

Negative finding rules: Do not infer absence of symptoms, absence of disease, absence of investigations, absence of risk factors, or absence of history from silence; Accept negative findings only when explicitly stated, directly answered, or clearly communicated; For bundled negative findings, each negative finding must be supported.

Bundled point rules: Judge the whole point; Every clinically meaningful component must be supported; However, do not treat routine restatements of clearly described clinical actions as new unsupported facts; New diagnoses, medications, tests, causes, purposes, severities, timelines, disease-status qualifiers, or other meaningful additions require support.

Follow-up rules: Follow-up intervals must be supported by the conversation; Follow-up purposes must be explicitly stated or strongly tied to the physician's discussion and plan; Do not add future investigations, monitoring goals, or follow-up reasons that were not discussed.

Decision calibration: Be strict for high-risk factual details such as medications, doses, exact values, causality, disease status, future plans, and treatment rationale; Be flexible for standard SOAP summarization, common clinical terminology, and clearly grounded shorthand; Under uncertainty about a clinically meaningful detail, output 0; If **factual_prediction** is 0, explicitly identify the unsupported, contradicted, ambiguous, or over-specific component; If the explanation concludes that the point is clearly supported, **factual_prediction** should normally be 1.

Return structured JSON with one evaluation for every point, preserving each original **point_id** and point text.

Figure 14: Prompt used for the *SOAPPrecision* judge. Given a generated SOAP note and the encounter transcript, the judge decides whether each clinical statement in the note is factually supported by the transcript. The worked example included in the prompt is omitted here.

E Additional Training Details

All experiments were conducted on one NVIDIA H100 NVL GPU. For ROUGE-L, we use PyThaiNLP’s newmm tokenizer (Phatthiyaphai-bun et al., 2023). For BLEURT, we use bleurt-oss-21/BLEURT-20.

Supervised Fine-tuning. We fine-tune Gemma-4-E4B-it (Team et al., 2026) using supervised fine-tuning (SFT) with LoRA (Hu et al., 2022) ($r = 32$, $\alpha = 64$) using the Unsloth framework (Han et al., 2023). Training uses a maximum sequence length of 8,192, a learning rate of 9×10^{-5} with a cosine decay scheduler, AdamW (Loshchilov and Hutter, 2019) optimizer, batch size of 16, and 6 epochs, and selects the checkpoint with the lowest validation loss.

GRPO. We train GRPO on Gemma-4-E4B-it (Team et al., 2026) with LoRA (Hu et al., 2022) ($r = 32$, $\alpha = 64$) on the Unsloth framework using a learning rate of 8×10^{-6} and the AdamW optimization with a cosine decay scheduler. Training uses a batch size of 16 with 8 sampled generations per prompt with 0.4 temperature, a maximum completion length of 4,096 tokens, and 300 optimization steps. All LLM judges used as reward functions in the GRPO training are evaluated by Grok-4.3 using the same instructions to lower training costs. For the weights of each reward presented in Section 3.2.2, we use $\lambda_{\text{format}} = 3$, $\lambda_{\text{AF}} = 2$, $\lambda_{\text{sig}} = 1$, $\lambda_{\text{comp}} = 4$, and $\lambda_{\text{fact}} = 3$ as the weights of R_{format} , R_{AF} , R_{sig} , R_{comp} , and R_{fact} , respectively. The final reward is computed as their weighted sum:

$$R_{\text{all}} = \lambda_{\text{format}}R_{\text{format}} + \lambda_{\text{AF}}R_{\text{AF}} + \lambda_{\text{sig}}R_{\text{sig}} + \lambda_{\text{comp}}R_{\text{comp}} + \lambda_{\text{fact}}R_{\text{fact}}. \quad (12)$$

These weights are used for all ASCRIBE-based GRPO models.

For R_{sig} , the Atomic Fact LLM Judge may match a generated fact to multiple reference facts. Let $C(i) = \{c(i)_1, \dots\}$ denote the set of reference fact indices matched to the generated fact f_i . The predicted significance label s_i is considered correct if it matches the significance label of any matched reference fact. Otherwise, the prediction is considered incorrect.

F Additional Details on ASR Robustness Experiments

In Figure 3, we compare LLM-judge metrics on manually transcribed and ASR transcripts to evaluate robustness to transcription noise. ASR transcripts are generated using a proprietary Whisper-based model (Aung et al., 2024) fine-tuned on an internal voicebank containing Thai clinical speech, which obtains a WER of 29.22 on the ThaiClinicBench dataset.

G Qualitative Results

We present two representative encounters to examine how structured reasoning affects the clinical information preserved in the generated SOAP notes. For each example, we report the generated notes and provided a fact-level analysis showing which facts are entailed. All English translations are attached for readability and are made by Claude Opus.

In the first example (Tables 8 and 9), all models capture the main symptoms, including dizziness, nausea, and headache. However, the structured prompt with Gemini 3.1 Pro preserves more contextual details, such as working with chemicals throughout the day and normal limb strength, achieving 8/11 entailed facts compared with 6/11 for free-form CoT. The GRPO model trained on real encounters also captures normal limb strength and the lack of nasal protection, but omits the two-day symptom duration and full-day chemical exposure, yielding 7/11 entailed facts.

The second example (Tables 10, 11 and 12) compares different GRPO reward and training-data settings. All three models recover the major symptoms, examination findings, viral assessment, and symptomatic treatment. Nevertheless, the model trained with our rewards on real encounters retains the most complete set of details, including pain during phonation and the instruction to monitor for tonsillar swelling or exudate after returning home, achieving 22/28 entailed facts. The synthetic-only model performs similarly on 21/28 facts but misses several contextual details, whereas the CLINICEVAL model omits more follow-up- and encounter-specific information, achieving 17/28 facts.

Source transcript

เออ มีอยู่โอเค เมื่อก่อนมันเป็น queue นัดหมาย มันไม่มีเสียงแบบว่า รอ มีนัดหรือครับ สวัสดิ์คะ สวัสดิ์ครับ เป็นอะไร มาหรือครับ อิมมันเวียนหัวแล้วก็คลื่นไส้ เพราะว่า ประมาณ เป็นน้ำแอลกอฮอล์หรืออะไรสักอย่าง อ่า แต่เราทำงานเราไม่ได้ปิดจมูก อ้อ ทำโรงงาน แบบ สารเคมีแล้วเวียนศีรษะ ไซ้ ๆ ๆ ที่ทำเนี่ยทั้งวันเลยวันแรกเค้าให้พี่มาช่วย อ้อ ปวดหัวไหม ปวดหัว แล้วก็คือ 2 วันแล้ว ข้างเดียวหรือ 2 ข้าง ข้างเดียว ข้างไหน ข้าง ข้าง ๆ นี้ ขวา ไซ้ อ้อ แขนขา มีแรงปกติดี มีแรงปกติ

Uh, yes, okay. It used to be an appointment queue. There was no sound, like, wait, do you have an appointment? Hello. Hello. What brings you in? Um, I feel dizzy and nauseous, because, roughly, it's alcohol or something, ah. But when we work, we don't cover our nose. Oh, you work in a factory, with chemicals, and then got dizzy. Yes, yes, yes, doing this all day; on the first day they had me come help. Oh, any headache? Headache, and it's been 2 days. One side or both sides? One side. Which side? This side, right here. Right. Yes. Oh, arms and legs have normal strength. Normal strength.

	Gemini 3.1 Pro CoT	Gemini 3.1 Pro Ours	GRPO Gemma-4-E4B Ours (Real)
CE	54.5 (6/11 claims)	72.7 (8/11 claims)	63.6 (7/11 claims)
HPI	ผู้ป่วยมีอาการเวียนศีรษะ คลื่นไส้ และปวดศีรษะข้างขวา เป็นมา 2 วัน; มีประวัติทำงานในโรงงานและสูดดมสารเคมี (คล้ายแอลกอฮอล์) ตลอดทั้งวันโดยไม่ได้ใส่หน้ากากป้องกัน; ปฏิเสธอาการแขนขาอ่อนแรง. The patient has dizziness, nausea, and a right-sided headache for 2 days; has a history of working in a factory and inhaling chemicals (alcohol-like) all day without wearing a protective mask; denies limb weakness.	ผู้ป่วยมีอาการเวียนศีรษะ คลื่นไส้ และปวดศีรษะข้างขวาข้างเดียว เป็นมา 2 วัน; ผู้ป่วยสงสัยว่าอาการเกิดจากการสูดดมสารเคมี (แอลกอฮอล์หรือสารอื่น) จากการทำงานในโรงงาน; ผู้ป่วยทำงานสัมผัสสารเคมีทั้งวันในวันแรกที่ไปช่วยงาน โดยไม่ได้ใส่หน้ากากป้องกัน; แขนขามีแรงปกติดี. The patient has dizziness, nausea, and a one-sided right headache for 2 days; the patient suspects the symptoms are caused by inhaling chemicals (alcohol or another substance) from working in a factory; the patient worked with chemical exposure all day on the first day of helping out, without wearing a protective mask; limb strength is normal.	มีอาการเวียนหัวและคลื่นไส้; อาการอาจเกี่ยวข้องกับน้ำแอลกอฮอล์หรือสารบางอย่าง; ผู้ป่วยทำงานในโรงงานที่เกี่ยวข้องกับสารเคมีและไม่ได้ปิดจมูกขณะทำงาน; มีอาการเวียนศีรษะขณะทำงานทั้งวัน; มีอาการปวดหัวเป็นมา 2 วันแล้ว; แขนขาไม่มีแรงปกติ. Has dizziness and nausea; the symptoms may be related to alcohol or some substance; the patient works in a factory involving chemicals and does not cover the nose while working; has dizziness while working all day; has had a headache for 2 days; limbs do not have normal strength.
PE	–	Motor power: normal.	แขนขามีแรงปกติ.
	–	Motor power: normal.	Limbs have normal strength.
Results	–	–	–
	–	–	–
A&P	–	Headache and dizziness, likely secondary to chemical exposure.	–
	–	Headache and dizziness, likely secondary to chemical exposure.	–

Table 8: Qualitative comparison of SOAP notes generated by different methods from the same source transcript. CE is the ClaimEntailment score, computed against the same set of 11 transcript-derived reference claims. To preserve space, statements are separated with semi-colons (;). Within each section, the top half is the Thai note as generated and the bottom half is a literal English translation.

#	Reference claim	Gemini CoT	Gemini Ours	Gemma Ours
1	ผู้ป่วยมีอาการเวียนหัว (The patient has dizziness)	✓	✓	✓
2	ผู้ป่วยมีอาการคลื่นไส้ (The patient has nausea)	✓	✓	✓
3	ผู้ป่วยระบุว่าอาการเกิดจากการทำงาน กับสารเคมีคล้ายน้ำแอลกอฮอล์ (The patient states the symptoms came from working with alcohol-like chemicals)	✗	✗	✗
4	ผู้ป่วยไม่ได้ปิดจมูกขณะทำงาน (The patient did not cover their nose while working)	✗	✗	✓
5	ผู้ป่วยทำงานในโรงงาน (The patient works in a factory)	✓	✓	✓
6	ผู้ป่วยทำงานกับสารเคมีทั้งวัน (The patient works with chemicals all day)	✗	✓	✗
7	วันที่เกิดอาการเป็นวันแรก ที่ผู้ป่วยถูกให้มาช่วยทำงาน (The day symptoms began was the patient's first day being brought in to help)	✗	✗	✗
8	ผู้ป่วยมีอาการปวดหัว (The patient has a headache)	✓	✓	✓
9	ผู้ป่วยมีอาการมา 2 วันแล้ว (The patient has had symptoms for 2 days)	✓	✓	✗
10	ผู้ป่วยปวดหัวข้างขวา (The patient has a right-sided headache)	✓	✓	✓
11	ผู้ป่วยมีแรงแขนขาปกติ (The patient has normal limb strength)	✗	✓	✓
Total entailed claims		6/11	8/11	7/11
ClaimEntailment score		54.5	72.7	63.6

Table 9: Claim-level entailment comparison of the generated SOAP notes in Table 8. ✓ indicates that the reference claim is entailed by the generated SOAP note under the ClaimEntailment metric, whereas ✗ indicates that it is not fully entailed. Scores are reported as percentages.

Source transcript
หวัดดีครับ หวัดดีค่ะ วันนี้เป็นไรมาเอ่ย เจ็บคอค่ะ ครับ ไปทำอะไรมาแล้วเจ็บคอ อืม แบบไปเที่ยวไปเจอคนอะไรมีไหม ไม่ ๆ ไม่มีค่ะ ไม่ ที่ทำงานมีใครไม่สบายไหม ไม่ มีค่ะ ไม่มีเลยเนาะ เป็นมากี่วันแล้วครับ เมื่อวาน เมื่อวานเนาะ กลืนน้ำลายได้ไหม กลืนน้ำลายแล้วเจ็บเลยไหม กลืน ๆ ได้แต่ว่าเวลากินน้ำก็จะรู้สึกเจ็บนิดนึง อืม มีไข้ไหมครับ ไม่มีค่ะ ไม่มีไข้เนาะ ขอดูคอนิดนึงนะ อ่า ทำ อา แป้นะ ทำเสียงอา อา เจ็บเนาะเหมือนแบบมันสะอึก อะเงยหน้าขึ้นนิดนึง โอเค โอเค คอแดงนิดนึงนะ แล้วก็ คอไม่มีหนองนะครับ มี ไอไหม มีนิดนึง มีปวดเมื่อยตามตัวไหมครับ ปวด ปวดเมื่อยตามตัวไข้ไหมครับ อ้อ อ้อ แต่ไอ ๆ ด้วยนิดนึงเนาะ มีค่ะ มีไข้ไหมที่บ้าน ดูตัวรุม ๆ มีไหม ไม่ ไม่มีเลยเนาะ โอเค เอ่อ โอเคเดี๋ยวผมให้ยาไปนะ นะครับ มีไอนี้มีเสมหะหรือไอแห้ง ๆ ไอแห้ง ๆ ไอแห้ง ๆ เนาะ แต่ไอเยอะจนรำคาญเลยไหม หรือว่าแบบไม่ไอเยอะ ไม่ค่อยมีนิดเดียว นิดเดียวเลยเนาะ นิดเดียว แต่ว่าที่ดูคอแล้ว คอไม่มี ไม่มีหนองในคอนะ แต่ว่าแนะนำว่าถ้ากลับบ้านไปสังเกตหน้าปากดูคอได้ว่า ตรง tonsil ข้าง ๆ ในคอมันมีบวมมี ๆ หนองบ้างไหม คือบางคนเนี่ยพอมันอาการเป็นวันแรกอะ มันจะแยกยากกว่าเป็น bacteria หรือเป็น virus นะครับ แต่จากอาการเนี่ยค่อนข้างเป็น virus โทษนะ ขอดูลำต่อน้ำเหลืองนิดนึงนะ โอเค คลำต่อน้ำเหลืองบริเวณคอนะครับ cervical lymph node เนี่ยไม่มีเนาะครับ ไม่มีต่อน้ำเหลืองที่คอโตนะ ซึ่งก็อาจจะมันแนวโน้มเป็นอาการติดเชื้อ virus ได้มากกว่า ซึ่งพอเป็นการติดเชื้อ virus เนี่ย เราจะแนะนำให้กินยารักษาตามอาการแล้วพักผ่อนเยอะ ๆ เอาใบลาไหม ลากันวันนี้นะพรุ่งนี้ วันเดียวพอ พรุ่งนี้ วันเดียวพรุ่งนี้ พรุ่งนี้ พรุ่งนี้โอเค ได้เดียวหมอให้เอายา ยาหยุคนี้ไปด้วยนะ หมายถึงว่าตามปกติมันแหละ นั่นแหละเนาะ นะครับ โอเค ได้ มีข้อสงสัยอยากสอบถามตอนไหนเพิ่มเติมไหมครับ ไม่ ไม่มีเลยเนาะ เดี๋ยวลองกินยานะ โอเคค่ะขอบคุณมากค่ะ ครับหวัดดีครับ Hello. Hello. What brings you in today? A sore throat. Okay. What were you doing before the sore throat? Um, like, did you travel or meet anyone? No, no, I didn't. No. Is anyone at work sick? No, there isn't. No one at all, right? How many days has it been? Since yesterday. Yesterday, right. Can you swallow saliva? Does it hurt when you swallow? I can swallow, but when I drink water it hurts a little. Um, any fever? No. No fever, right. Let me take a look at your throat. Ah, say ah. Hold on, say "ah". Ah. It hurts, right, like a hiccup. Tilt your head up a little. Okay, okay, your throat is a little red, and, but there's no pus in your throat. Any cough? A little. Any body aches? Yes. Body aches, right? Oh, oh, and a bit of a cough too, right? Yes. Any fever, feeling feverish at home? No. None at all, right. Okay, uh, okay, I'll give you some medicine. Is the cough with phlegm, or dry? Dry. Dry cough, right. But is the cough so bad that it bothers you, or not much? Not much, just a little. Just a little, right. Just a little. But from looking at your throat, there's no pus in your throat. But I recommend that when you get home, you look at your throat in front of the mirror to see whether the tonsils at the sides of your throat are swollen or have any pus. For some people, on the first day of symptoms, it's hard to tell whether it's bacteria or a virus. But from the symptoms, it's likely a virus. Excuse me, let me feel your lymph nodes. Okay, palpating the lymph nodes in the neck, no cervical lymph nodes. No enlarged lymph nodes in the neck, right. So it's more likely to be a viral infection. And with a viral infection, we recommend taking medicine for the symptoms and getting plenty of rest. Do you want a leave note? How many days, today or tomorrow? One day is enough, tomorrow. One day, tomorrow. Tomorrow. Tomorrow, okay. Sure, I'll have you take the medicine, and this "stop medicine" too. I mean, the usual. Right, okay. Okay, sure. Any other questions you'd like to ask? No, none at all, right. Try taking the medicine. Okay, thank you very much. Bye.

Table 10: Source transcript of the second qualitative example, used for the SOAP notes in Table 11. The top half is the Thai transcript and the bottom half is a literal English translation.

	ClinicEval	Ours (Real)	Ours (Synth)
CE	60.7 (17/28 claims)	78.6 (22/28 claims)	75.0 (21/28 claims)
HPI	ผู้ป่วยมาด้วยอาการเจ็บคอ (Sore throat); ไม่ทราบว่าไปเที่ยวหรือเจอคนที่มีอาการหรือไม่ (No recent travel or exposure to sick people); ไม่พบผู้ป่วยไม่สบายที่ทำงาน (No sick colleagues at work); อาการเป็นมาตั้งแต่เมื่อวาน (Symptoms started yesterday); กลืนน้ำลายได้ แต่เจ็บเล็กน้อยเวลากินน้ำ (Can swallow saliva, but feels slight pain when drinking water); ไม่มีไข้ (No fever); มีอาการไอเล็กน้อย (Mild cough); มีอาการปวดเมื่อยตามตัว (Body aches); ไม่มีไข้ที่บ้าน (No fever at home); ลักษณะไอเป็นไอแห้ง (Dry cough), ไอไม่เยอะมาก (Not much cough); จากการตรวจคอ ไม่พบหนองในคอ (No pus observed in the throat); ผู้ป่วยไม่แน่ใจว่าไอแห้งหรือมีเสมหะ แต่ระบุว่าไอแห้ง (Patient unsure if phlegm or dry cough, but stated dry cough). The patient presents with a sore throat; unknown whether they traveled or met anyone with symptoms; no sick people found at work; symptoms since yesterday; can swallow saliva, but slight pain when drinking water; no fever; mild cough; body aches; no fever at home; the cough is dry, not much coughing; on throat examination, no pus found in the throat; the patient is unsure whether the cough is dry or productive, but states it is dry.	ผู้ป่วยมาด้วยอาการเจ็บคอ; อาการเริ่มเป็นเมื่อวาน; ไม่เคยไปเที่ยวหรือเจอคนป่วย; ไม่มีคนในที่ทำงานป่วย; กลืนน้ำลายได้แต่เจ็บเล็กน้อยเวลากินน้ำ; ไม่มีไข้; ทำเสียงอาแล้วเจ็บเหมือนสะอึก; มีไอเล็กน้อย; มีอาการปวดเมื่อยตามตัว; ไม่มีไข้ที่บ้าน; ไอเล็กน้อย ไม่ค่อยมี; คอไม่มีหนอง. The patient presents with a sore throat; symptoms started yesterday; has not traveled or met sick people; no one at work is sick; can swallow saliva but has slight pain when drinking water; no fever; saying “ah” hurts, like a hiccup; mild cough; body aches; no fever at home; slight cough, not much; no pus in the throat.	ผู้ป่วยมาด้วยอาการเจ็บคอ; อาการเป็นมาตั้งแต่เมื่อวาน; ผู้ป่วยไม่ได้ไปเที่ยวหรือเจอคนอื่นที่ทำให้เจ็บคอ; ที่ทำงานไม่มีใครไม่สบาย; กลืนน้ำลายได้แต่เจ็บ; เวลากินน้ำรู้สึกเจ็บนิดนึง; ไม่มีไข้; มีไอเล็กน้อย; มีปวดเมื่อยตามตัว; ที่บ้านดูตัว รุม ๆ ไม่มี; ไอแห้ง ๆ; ไอไม่เยอะ นิดเดียว. The patient presents with a sore throat; symptoms since yesterday; the patient did not travel or meet others who caused the sore throat; no one at work is sick; can swallow saliva but it hurts; feels slight pain when drinking water; no fever; mild cough; body aches; no feeling feverish at home; dry cough; not much cough, just a little.
PE	ตรวจคอ: คอแดงเล็กน้อย (Throat: Mild redness); ตรวจคอ: ไม่พบหนองในคอ (Throat: No pus); คลำต่อมน้ำเหลืองบริเวณคอ: ไม่พบต่อมน้ำเหลืองโต (Cervical lymph node palpation: No enlarged cervical lymph nodes). Throat exam: mild redness; throat exam: no pus found in the throat; neck lymph node palpation: no enlarged lymph nodes.	ตรวจคอ: คอแดงเล็กน้อย; ตรวจคอ: ไม่มีหนองในคอ; คลำต่อมน้ำเหลืองบริเวณคอ (cervical lymph node): ไม่มีต่อมน้ำเหลืองโต. Throat exam: mildly red; throat exam: no pus in the throat; neck lymph node (cervical lymph node) palpation: no enlarged lymph nodes.	ตรวจคอพบคอแดงนิดนึง; คอไม่มีหนอง; คลำต่อมน้ำเหลืองบริเวณคอ cervical lymph node ไม่มีโต. Throat exam shows slight redness; no pus in the throat; palpation of the neck lymph nodes: cervical lymph nodes not enlarged.
Results	จากอาการและผลการตรวจ ค่อนข้างมีแนวโน้มเป็น Viral infection มากกว่า Bacterial infection; แนะนำให้สังเกตอาการที่หน้าปากคอ ว่า มี tonsil บวมหรือมีหนองหรือไม่. Based on the symptoms and examination, more likely a viral infection than a bacterial infection; advised to check in front of the mouth and throat whether the tonsils are swollen or have pus.	จากอาการและผลการตรวจ ค่อนข้างเป็น virus; แนวโน้มเป็นอาการติดเชื้อ virus มากกว่า. Based on the symptoms and examination, likely a virus; more likely a viral infection.	อาการค่อนข้างเป็น virus; มีแนวโน้มเป็นอาการติดเชื้อ virus มากกว่า. Symptoms likely viral; more likely a viral infection.
A&P	วินิจฉัยเบื้องต้น: น่าจะเป็น Viral infection; แนะนำให้รับประทานยาตามอาการ; พักผ่อนให้มาก; แนะนำให้ลาหยุด 1 วัน; ให้ยาตามปกติ; หากมีข้อสงสัยให้สอบถามเพิ่มเติม. Preliminary diagnosis: likely a viral infection; advised to take symptomatic medication; rest a lot; advised to take 1 day of leave; medication given as usual; ask if there are further questions.	วินิจฉัย: คาดว่าติดเชื้อไวรัส (Viral infection); แนะนำให้กินยารักษาตามอาการและพักผ่อนเยอะ ๆ; แนะนำให้สังเกตหน้าปากดูคอว่า มี tonsil บวมหรือมีหนองหรือไม่เมื่อกลับบ้าน; ให้ใบลา 1 วัน (พรุ่งนี้); ให้ยาตามปกติ. Diagnosis: suspected viral infection (Viral infection); advised to take symptomatic medication and rest a lot; advised to check the throat in front of the mouth for swollen tonsils or pus after returning home; 1-day leave note given (tomorrow); medication given as usual.	แนะนำให้สังเกตหน้าปากดูคอว่า tonsil ข้าง ๆ ในคอมันมีบวมมีหนองบ้างไหม; แนะนำให้กินยารักษาตามอาการและพักผ่อนเยอะ ๆ; ให้ใบลา 1 วัน (พรุ่งนี้); ให้ยาหยุดนี้ไปด้วย. Advised to check the throat in front of the mouth for whether the tonsils at the sides are swollen or have pus; advised to take symptomatic medication and rest a lot; 1-day leave note given (tomorrow); also given this “stop medicine”.

Table 11: Qualitative comparison of SOAP notes generated by three GRPO models from the source transcript in Table 10. CE is the ClaimEntailment score, computed against the same set of 28 transcript-derived reference claims. To preserve space, statements are separated with semi-colons (;). Within each section, the top half is the Thai note as generated (English in parentheses was generated by the model) and the bottom half is a literal English translation.

#	Reference claim	Clinic Eval	Ours Real	Ours Synth
1	ผู้ป่วยมีอาการเจ็บคอ (The patient has a sore throat)	✓	✓	✓
2	ผู้ป่วยบอกว่าไม่ได้ไปเที่ยว (The patient says they did not travel)	✗	✓	✓
3	ที่ทำงานของผู้ป่วยไม่มีคนไม่สบาย (No one at the patient's workplace is sick)	✓	✓	✓
4	ผู้ป่วยเริ่มมีอาการเจ็บคอเมื่อวาน (The patient's sore throat started yesterday)	✓	✓	✓
5	ผู้ป่วยกลืนน้ำลายได้ (The patient can swallow saliva)	✓	✓	✓
6	ผู้ป่วยรู้สึกเจ็บคอนิดนึงเวลากินน้ำ (The patient feels slight throat pain when drinking water)	✓	✓	✓
7	ผู้ป่วยไม่มีไข้ (The patient has no fever)	✓	✓	✓
8	ผู้ป่วยรู้สึกเจ็บคอเมื่อทำเสียงอา (The patient feels throat pain when saying "ah")	✗	✓	✗
9	ผู้ป่วยรู้สึกเหมือนสะอึกเมื่อทำเสียงอา (The patient feels like hiccuping when saying "ah")	✗	✓	✗
10	แพทย์ตรวจพบคอแดงนิดนึง (The physician finds slight throat redness)	✓	✓	✓
11	แพทย์ตรวจไม่พบคอมีหนอง (The physician finds no pus in the throat)	✓	✓	✓
12	ผู้ป่วยมีอาการไอนิดนึง (The patient has a slight cough)	✓	✓	✓
13	ผู้ป่วยมีอาการปวดเมื่อยตามตัว (The patient has body aches)	✓	✓	✓
14	ผู้ป่วยมีอาการไอแห้ง (The patient has a dry cough)	✓	✗	✓
15	ผู้ป่วยไอไม่เยอะ (The patient does not cough much)	✓	✓	✓
16	แพทย์จะให้ยาแก่ผู้ป่วย (The physician will give the patient medication)	✓	✓	✓
17	แพทย์แนะนำให้ผู้ป่วยสังเกตว่า ต่อมทอนซิลบวมหรือมีหนองหรือไม่เมื่อกลับบ้าน (The physician advises the patient to check whether the tonsils are swollen or have pus after returning home)	✗	✓	✗
18	แพทย์อธิบายว่าในวันแรกของอาการ จะแยกได้ยากกว่าเป็นการติดเชื้อแบคทีเรียหรือไวรัส (The physician explains that on the first day of symptoms, it is hard to tell a bacterial from a viral infection)	✗	✗	✗
19	แพทย์ประเมินจากอาการว่าค่อนข้างเป็นการติดเชื้อไวรัส (The physician assesses from the symptoms that it is likely a viral infection)	✓	✓	✓
20	แพทย์ตรวจคลำต่อมน้ำเหลืองบริเวณคอ แล้วไม่พบว่าต่อมน้ำเหลืองโต (The physician palpates the neck lymph nodes and finds no enlarged lymph nodes)	✓	✓	✓
21	แพทย์อธิบายว่าการที่ต่อมน้ำเหลืองไม่โต มีแนวโน้มว่าเป็นการติดเชื้อไวรัสได้มากกว่า (The physician explains that non-enlarged lymph nodes make a viral infection more likely)	✗	✗	✗
22	แพทย์แนะนำให้กินยารักษาตามอาการ สำหรับการติดเชื้อไวรัส (The physician advises symptomatic medication for the viral infection)	✓	✓	✓
23	แพทย์แนะนำให้พักผ่อนเยอะ ๆ สำหรับการติดเชื้อไวรัส (The physician advises plenty of rest for the viral infection)	✓	✓	✓
24	แพทย์เสนอใบลาให้ผู้ป่วย (The physician offers the patient a leave note)	✗	✗	✓
25	ผู้ป่วยต้องการใบลา 1 วันสำหรับวันพรุ่งนี้ (The patient wants a 1-day leave note for tomorrow)	✗	✗	✗
26	แพทย์จะออกใบลาให้ผู้ป่วย (The physician will issue a leave note for the patient)	✗	✓	✓
27	ผู้ป่วยไม่มีข้อสงสัยสอบถามเพิ่มเติม (The patient has no further questions)	✗	✗	✗
28	แพทย์แนะนำให้ผู้ป่วยลองกินยา (The physician advises the patient to try taking the medication)	✗	✓	✓
Total entailed claims		17/28	22/28	21/28
ClaimEntailment score		60.7	78.6	75.0

Table 12: Claim-level entailment comparison among ClinicEval, Ours trained on real encounters, and Ours trained on synthetic encounters. ✓ indicates that the reference claim is entailed by the generated SOAP note under the ClaimEntailment metric, whereas ✗ indicates that it is not fully entailed. Scores are reported as percentages.

H Ethics & Data Governance Statement

ICOPD consists of consented outpatient encounters collected exclusively for internal AI research. The dataset contains real physician–patient conversations and is used only for model development and evaluation within our institution. It is not publicly released or distributed to external researchers.

S-ICOPD is a fully synthetic dataset generated from safe, de-identified clinical information derived from the ICOPD training data. It does not contain original physician–patient conversations or audio recordings. We will release S-ICOPD under a controlled-access, research-only data license to support reproducible research while minimizing the disclosure of real patient interactions.

ThaiClinicBench was constructed from clinical encounters collected with signed informed consent from all participating patients and physicians. Participants were informed that their clinical dialogue data would be de-identified and used for AI research and dataset development. Participation was voluntary, and the collected data were used only for the purposes described in the consent materials.

To mitigate the associated privacy risks, all transcripts and reference notes underwent manual de-identification before model development or analysis. Direct identifiers and potentially identifying contextual information, based on the HIPAA Safe Harbor ([U.S. Department of Health and Human Services, 2012](#)), such as names, contact information, locations, dates, institutional identifiers, and other encounter-specific details, were removed or replaced with synthetic placeholders. Audio recordings are not included in the released dataset. The de-identification procedure was designed with reference to Thailand’s Personal Data Protection Act ([Ministry of Digital Economy and Society, 2019](#)) and the HIPAA Safe Harbor framework ([U.S. Department of Health and Human Services, 2012](#)).

ThaiClinicBench will be distributed under a controlled-access, research-only data license. Users must agree not to attempt to identify participants, link the data with external information for identification purposes, redistribute the original dataset, or use it for clinical decision-making. The dataset is intended solely for research on clinical language processing and medical-scribe systems. Requests concerning privacy or removal of released data may be directed to the ThaiClinicBench authors.

KuMMLU: A Multitask Language Understanding Benchmark for Central Kurdish

Akam Nawzad

Ovanya

Kurdistan Region - Iraq

akam.nawzad@ovanya.com

akam.nawzad@ukh.edu.krd

Abstract

Language models are now tested across many languages, but Kurdish is mostly missing from these efforts. We present KuMMLU, a multiple-choice benchmark for Central Kurdish (Sorani) built from the national examinations of the Kurdistan Region of Iraq. It contains 3,272 questions across 18 subjects: 2,256 native examination questions, with answers taken from the official keys published by the Ministry of Education, and 1,016 questions from six subjects translated from English MMLU. We evaluate 32 language models, including proprietary systems, open-weight models, and multilingual models built for wide language coverage. The best models reach about 88% accuracy, while the five multilingual baselines score between 15.8% and 37.2%, at most modestly above the 25% chance level and well below the strongest general-purpose systems; this includes the one model whose published language list contains Kurdish. Every model scores higher on the translated subset than on the native questions. We examine this gap in light of subject difficulty and possible overlap with training data, and release the benchmark to support Kurdish language technology.

1 Introduction

Kurdish is spoken by tens of millions of people, yet it remains under-served by language technology (Esmaili, 2012; Ahmadi et al., 2024). It has three main varieties, Northern Kurdish (Kurmanji), Central Kurdish (Sorani), and Southern Kurdish, which differ in vocabulary, morphology, and writing system (Haig and Öpengin, 2014). Sorani is written in a modified Perso-Arabic script and holds official status in Iraq (Ahmadi et al., 2024). Most tools and datasets for Kurdish target basic tasks such as tokenization, transliteration, part-of-speech tagging, or named entity recognition (Ahmadi, 2020; Morad et al., 2024; Abdullah et al., 2024), and there is no standard way to measure what a modern language

model knows in Kurdish.

Multiple-choice benchmarks based on MMLU (Hendrycks et al., 2021) have become a common tool for this kind of measurement. Versions now exist for Arabic (Koto et al., 2024), Turkish (Yüksel et al., 2024), Korean (Son et al., 2025), and many languages at once (Singh et al., 2025), but not for Kurdish. Translated benchmarks are also known to reward Western-centric knowledge and to carry translation artifacts (Singh et al., 2025), so native questions are preferable when they are available.

We address this gap with KuMMLU, a benchmark for Central Kurdish drawn from national examinations. Our contributions are:

- A benchmark of 3,272 Sorani multiple-choice questions across 18 subjects, most of them written natively in Kurdish and answered from official examination keys.
- An evaluation of 32 language models, covering proprietary, open-weight, and multilingual systems.
- An analysis of the gap between native and translated questions, and of how multilingual models built for wide language coverage perform on Kurdish.

2 Related Work

MMLU and its variants. MMLU (Hendrycks et al., 2021) measures knowledge and reasoning through multiple-choice questions in 57 subjects. Later work adapts the format to specific languages. Some versions translate the English questions, while others collect native questions from local exams. KMMLU (Son et al., 2025) and ArabicMMLU (Koto et al., 2024) both argue that native questions capture linguistic and cultural content that translation loses, and TurkishMMLU (Yüksel et al., 2024) builds its set from high-school curricula. Global-MMLU (Singh et al., 2025) studies cul-

tural bias across languages and improves the translated MMLU with human annotation. KuMMLU follows the native-collection approach and adds a translated subset for comparison.

Kurdish NLP. Work on Kurdish has focused on foundational resources and tools. Esmaili (2012) describes the main obstacles, including dialect diversity and script variation. The Kurdish Language Processing Toolkit (Ahmadi, 2020) provides pre-processing and transliteration, and Ahmadi et al. (2024) survey language and speech technology for Central Kurdish. Recent task-specific systems include part-of-speech tagging for Northern Kurdish (Morad et al., 2024) and named entity recognition for Sorani built by fine-tuning transformer encoders (Abdullah et al., 2024). To our knowledge, no prior benchmark measures the general knowledge of language models in Kurdish.

Multilingual models. Several models are trained for broad language coverage. mT0 and BLOOMZ (Muennighoff et al., 2023) extend multitask fine-tuning to many languages, and the Aya models (Üstün et al., 2024; Dang et al., 2024) target instruction following across dozens to over a hundred languages. We include these models in our evaluation because they are the systems most likely to have seen Kurdish during training.

3 The KuMMLU Benchmark

3.1 Sources

The native questions come from the national examinations of the Kurdistan Region of Iraq, at the Grade 9 and Grade 12 levels. These are the two grade levels at which the Kurdistan Region holds centralized national examinations; they are set by the KRG Ministry of Education and cover the standard school curriculum. We recorded the correct answers manually from the official answer keys published by the Ministry, so each answer reflects the position taken by the examining authority and was not decided by us. A second person then checked the recorded answers against the published keys to catch transcription errors.

To connect KuMMLU to the original MMLU, we added a translated subset of six subjects: high school mathematics, philosophy, college chemistry, college physics, college mathematics, and college medicine. We translated these from English MMLU (Hendrycks et al., 2021) with Gemini 2.5 Flash and kept the original MMLU answer labels.

Two native Sorani speakers reviewed the translated questions and corrected fluency and terminology errors. This subset lets us compare native and translated questions inside one benchmark, with the caveats discussed in Limitations.

3.2 Annotation and quality control

The exam material was collected as page images and passed through optical character recognition. Two native Sorani speakers then checked the questions against their source images and corrected recognition errors, spacing, and spelling. Each question was checked by one speaker, and a second speaker reviewed the flagged items along with a random sample of the rest. We wrote short guidelines that told the checkers to keep the original wording, to leave technical terms and formulas unchanged, and to flag any item whose text or answer could not be recovered from the source. Cases that were unclear were resolved by discussion between the two speakers.

We then removed duplicate questions that appeared more than once in the source material, and discarded items with fewer than two usable options. We also normalized Arabic-script code points that OCR had introduced in place of their Kurdish equivalents, while keeping genuine Arabic quotations such as Qur’anic verses unchanged. As a further check, we used Gemini 3.1 Pro to flag questions with likely spelling or OCR problems, and gave those flags to the native speakers, who reviewed each flagged item and decided whether to correct it.

3.3 Statistics

The final benchmark has 3,272 questions in 18 subjects: 2,256 questions (12 subjects) written natively in Kurdish, and 1,016 (6 subjects) translated from English MMLU. The per-subject counts are in Appendix A. Of the 3,272 questions, 3,264 have four options; seven Kurdish literature questions have three and one mathematics question has five, as in the source papers. After the option shuffling described in Section 4.1, the correct answer falls in each position for 24.7–25.6% of the four-option questions.

4 Experiments

4.1 Setup

We evaluate each model in a zero-shot setting, which is the common protocol for instruction-tuned

Model	T	All	Nat	Tr	Model	T	All	Nat	Tr
Gemini 3.1 Pro	P	88.1	85.5	93.6	GPT-4o	P	46.1	44.2	50.3
Gemini 3.6 Flash	P	87.8	84.6	95.1	Qwen3 235B [‡]	O	45.6	42.0	53.5
GPT-5.6 (sol)	P	82.9	78.0	93.8	Llama 4 Scout [‡]	O	44.4	42.9	47.8
Gemini 2.5 Pro	P	75.9	73.4	81.7	Llama 3.3 70B [‡]	O	41.6	39.1	47.3
Claude Opus 4.8	P	75.4	70.5	86.2	Mistral Large	O	38.4	36.1	43.7
Kimi K3 [‡]	O	72.6	66.7	85.6	Aya Expanse 32B	M	37.2	35.8	40.2
Qwen3.7 Max	P	69.5	65.8	77.9	Mistral Medium 3.1	P	36.2	34.2	40.6
Claude Sonnet 5	P	68.7	62.4	82.6	Qwen3 8B	O	33.9	32.6	36.8
GPT-5.6 (luna) [‡]	P	68.5	60.5	86.3	Mistral Small 24B	O	32.1	30.5	35.7
Gemini 2.5 Flash	P	68.1	67.3	69.8	Aya Expanse 8B	M	32.1	31.2	34.0
Claude Sonnet 4	P	64.9	58.8	78.4	Ministral 8B	O	31.4	30.2	34.3
Gemini 2.5 Flash-Lite	P	62.3	59.9	67.7	Ministral 14B	O	31.3	30.0	34.2
Gemma 4 31B	O	57.9	53.6	67.2	Aya-101	M	29.5	29.0	30.5
Llama 4 Maverick	O	57.5	54.8	63.6	mT0-XXL [‡]	M	29.4	28.7	30.8
DeepSeek V4 Flash	O	53.8	51.7	58.4	Ministral 3B	O	27.5	27.2	28.3
Command A	O	48.3	45.0	55.6	BLOOMZ-7B1 [†]	M	15.8	14.5	18.8

Table 1: Accuracy (%) on KuMMLU, ordered by overall accuracy. Type: P = proprietary, O = open-weight, M = multilingual baseline. **All**, **Nat**, and **Tr** are overall, native-subset, and translated-subset accuracy. The chance level is approximately 25%. GPT-5.6 (sol) and GPT-5.6 (luna) are two configurations of GPT-5.6 offered by the provider. [†]BLOOMZ-7B1 left 37% of items unparsed; see text. [‡]Response rate between 83% and 98%; see Appendix F.

models on MMLU-style benchmarks. Each question is presented with its options, and the model is asked to reply with the option number. The instruction is written in English and the question and its options are kept in Kurdish; the exact template is given in Appendix B. We read the chosen option from the model’s output. Because language models are known to favour particular option positions (Zheng et al., 2024), we shuffle the options for each question with a fixed seed before evaluation and remap the correct answer accordingly. Reasoning (“thinking”) is set to low effort for GPT-5.6 (sol and luna), Claude Opus 4.8, Claude Sonnet 5, Claude Sonnet 4, Kimi K3, and the Gemini 2.5 models, and left at the provider default for Gemini 3.1 Pro and Gemini 3.6 Flash; for all other models it is disabled. We do not tune inference settings per model. API requests use temperature 0.

Proprietary and open-weight chat models are queried through the OpenRouter API; the multilingual baselines (mT0-XXL, BLOOMZ-7B1, and the Aya models) are run locally in bf16 on a single A100 GPU with the same prompt and shuffling. Accuracy counts an unparsed answer as incorrect. Response rates were above 98% for 25 of the 32 models and above 90% for all but two, BLOOMZ-7B1 (63%) and Qwen3 235B (83%). Per-model response rates are given in Appendix F, which lets missing or unparsable responses be separated from wrong answers. For the API-served models, most unparsed responses were empty replies.

4.2 Models

We test 32 models in three groups. The first is proprietary chat models from several providers. The second is open-weight models across a range of sizes. The third is multilingual models built for wide language coverage: mT0-XXL and BLOOMZ-7B1 (Muennighoff et al., 2023), Aya-101 (Üstün et al., 2024), and Aya Expanse in the 8B and 32B sizes (Dang et al., 2024). Aya-101 is the only model in our set whose published language list contains Kurdish, though it lists the Latin-script (Kurmanji) form, not the Arabic-script Sorani used here.

4.3 Results

Table 1 reports overall accuracy together with accuracy on the native and translated subsets. The strongest models reach about 88%. Accuracy then falls off across the open-weight models, and the multilingual baselines fall in the lower part of the table, at most modestly above the chance level (approximately 25%, since most questions have four options).

Two patterns hold across the whole table. First, every model scores higher on the translated subset than on the native questions. The six translated subjects come from English MMLU and overlap with material these models have likely seen in English, so this gap may reflect knowledge acquired elsewhere rather than Kurdish knowledge itself. The size of the gap is not uniform. It is smallest for the weakest models, which perform near chance

on both subsets (1.1 points for Ministral 3B), and largest among models in the middle and upper-middle of the table (up to 25.8 points for GPT-5.6 (luna)). Among the strongest models the picture is mixed: Gemini 3.1 Pro shows a gap of 8.1 points, while GPT-5.6 (sol) and Claude Opus 4.8 show gaps of 15.8 and 15.7 points. The Gemini models, whose family produced the translated subset, show smaller gaps than non-Gemini models of similar overall accuracy, but this comparison cannot isolate a possible advantage from Gemini-generated translations.

Second, the multilingual models built for many languages do poorly. mT0-XXL and Aya-101 score 29.4% and 29.5%, close to chance, and Aya Expanse 32B reaches only 37.2% despite being a newer and larger model. Aya-101 lists Kurdish among its languages but does not outperform Aya Expanse, which does not. This indicates that a language appearing on a training list does not by itself produce usable ability in that language. BLOOMZ-7B1 scores 15.8%, below chance: it failed to produce a parsable answer on 37% of items, and on the items it did answer it scored 25%, at chance. Its low aggregate therefore reflects both frequent failures to follow the answer format on Kurdish-language inputs and chance-level accuracy on the answers it gave.

Within the open-weight groups, larger models in a family tend to score higher, but even the best open model trails the strongest proprietary systems by a wide margin. One open model, Qwen3 235B, returned a parsable answer on only 83% of items, mostly because its replies were empty, and its accuracy on those items (54.7%) is well above its 45.6% overall score. Accuracy also varies by subject. Averaged across all models, the three Kurdish-language subjects are the hardest: rhetoric (33.7%), grammar (36.1%), and literature (40.8%) are the lowest of the 18 subjects, while general-knowledge subjects such as religious studies and social sciences are the highest. Full per-subject figures are in Appendix C. Subject sizes vary (Table 2), so estimates for the smallest subjects are less precise: rhetoric has only 49 questions, and the two highest-scoring subjects, social sciences and religious studies, have 55 and 68. The low scores on the Kurdish-language subjects do not rest on rhetoric alone, however, since grammar (282 questions) and literature (289) are among the largest subjects in the benchmark. This pattern fits the earlier observation: the subjects that are most specific

to Kurdish, and least likely to have a counterpart in higher-resource languages, are where models struggle most. Appendix D shows example questions, and Appendix E examines the errors in more detail.

5 Conclusion

We introduced KuMMLU, a multiple-choice benchmark for Central Kurdish built from national examinations, and used it to evaluate 32 language models. The best proprietary models reach about 88%, while multilingual models built for wide coverage score at most modestly above chance, and every model does better on translated questions than on native ones. These results give a first clear picture of where language models stand on Kurdish and provide a reference point for future work. We plan to extend KuMMLU to Kurmanji, so that a second Kurdish variety is covered.

Limitations

The benchmark covers Central Kurdish in the Arabic script only; Kurmanji and the Latin script are left for future work. The questions come from national school examinations, so the benchmark reflects that curriculum and not the full range of Kurdish knowledge.

The comparison between the native and translated subsets should be read with care, for three reasons. First, the two subsets differ in difficulty and domain: the native questions are from Grade 9 and Grade 12 school exams, while the translated questions come from MMLU’s high-school and college-level subjects. Part of the native–translated gap may therefore reflect difficulty, not language. Second, the translated subjects are taken from English MMLU, which is widely available online and is likely present in the training data of many models (Oren et al., 2024), so high scores there may reflect memorized English questions rather than Kurdish ability. The native exam papers also circulate online and could appear in training data, though we are not aware of a released, machine-readable version of them. Third, the translated questions were produced with a Gemini model, and Gemini models are among those we evaluate. The per-model gaps in Table 1 do not suggest that this gives them an advantage on the translated subset, but they cannot rule it out. We therefore report the gap as an observation and do not claim a single cause.

On the evaluation itself: we shuffle options once with a fixed seed, which reduces but does not re-

move position effects; averaging over several permutations, or scoring by option likelihood, would be more thorough, and the latter would also measure format-sensitive models such as BLOOMZ more fairly. The open-weight models are served through an API aggregator and may be quantized, so their scores may not exactly match the full-precision weights. The proprietary models are queried at a fixed point in time and may change. We evaluated each model once, using the inference settings described in Section 4.1 without performance-based tuning, so results may differ under other settings such as higher reasoning effort or few-shot prompting. The instruction is written in English; a Kurdish instruction could change the results, particularly for weaker models that already struggle with the answer format. Finally, the translated subset was checked by native speakers but may still contain translation errors. We did not formally measure inter-annotator agreement during checking; unclear cases were resolved by discussion, as described in Section 3.2.

Ethics Statement

The native questions come from national examination papers and their official answer keys, which are public documents, and the translated questions come from the publicly released MMLU benchmark. It contains no personal or private information. Two native Sorani speakers took part in the checking process; they were university students who volunteered for the project and were informed of its purpose and intended use. We release the benchmark, the evaluation code, and the per-model results at <https://huggingface.co/datasets/akam-ot/KuMMLU> under the CC BY-NC 4.0 license for research use, with documentation of the sources, the annotation process, and the native/translated split. Because publicly released benchmarks can be absorbed into future training data, the release includes a canary string and asks that the data not be used for training, so that the benchmark stays useful for evaluation.

Acknowledgements

We thank Amin Khattab Ahmed and a second native Sorani speaker, who wished to remain anonymous, for volunteering to check the questions and translations.

References

- Abdulhady Abas Abdullah, Srwa Hasan Abdulla, Dalia Mohammad Toufiq, Halgurd S. Maghdid, Tarik A. Rashid, Pakshan F. Farho, Shadan Sh. Sabr, Akar H. Taher, Darya S. Hamad, Hadi Veisi, and Aras T. Asaad. 2024. NER-RoBERTa: Fine-tuning RoBERTa for named entity recognition (NER) within low-resource languages. *arXiv preprint arXiv:2412.15252*.
- Sina Ahmadi. 2020. [KLPT – Kurdish language processing toolkit](#). In *Proceedings of the Second Workshop for NLP Open Source Software (NLP-OSS)*, pages 72–84. Association for Computational Linguistics.
- Sina Ahmadi, Daban Jaff, Md Mahfuz Ibn Alam, and Antonios Anastasopoulos. 2024. [Language and speech technology for Central Kurdish varieties](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 10034–10045, Torino, Italia. ELRA and ICCL.
- John Dang, Shivalika Singh, Daniel D’souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, et al. 2024. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.
- Kyumars Sheykh Esmaili. 2012. Challenges in Kurdish text processing. *arXiv preprint arXiv:1212.0074*.
- Geoffrey Haig and Ergin Öpengin. 2014. Kurdish: A critical research overview. *Kurdish Studies*, 2(2):99–122.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*.
- Fajri Koto, Haonan Li, Sara Shatnawi, Jad Doughman, Abdelrahman Boda Sadallah, Aisha Alraeesi, Khalid Almubarak, Zaid Alyafeai, Neha Sengupta, Shady Shehata, Nizar Habash, Preslav Nakov, and Timothy Baldwin. 2024. [ArabicMMLU: Assessing massive multitask language understanding in Arabic](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5622–5640. Association for Computational Linguistics.
- Peshmerge Morad, Sina Ahmadi, and Lorenzo Gatti. 2024. [Part-of-speech tagging for Northern Kurdish](#). In *Proceedings of the Joint Workshop on Multiword Expressions and Universal Dependencies (MWE-UD) @ LREC-COLING 2024*, pages 70–80, Torino, Italia. ELRA and ICCL.
- Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, et al. 2023. [Crosslingual generalization through multitask finetuning](#). In *Proceedings of the*

61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15991–16111. Association for Computational Linguistics.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. 2024. Proving test set contamination in black-box language models. In *International Conference on Learning Representations (ICLR)*.

Shivalika Singh, Angelika Romanou, Cl  mentine Fourrier, David Ifeoluwa Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiawat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Sebastian Ruder, Wei-Yin Ko, Antoine Bosselut, Alice Oh, Andre Martins, Leshem Choshen, Daphne Ippolito, and 4 others. 2025. [Global MMLU: Understanding and addressing cultural and linguistic biases in multilingual evaluation](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18761–18799, Vienna, Austria. Association for Computational Linguistics.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. [KMMLU: Measuring massive multitask language understanding in Korean](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104, Albuquerque, New Mexico. Association for Computational Linguistics.

Ahmet   st  n, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D’souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, Freddie Vargus, Phil Blunsom, Shayne Longpre, Niklas Muennighoff, Marzieh Fadaee, Julia Kreutzer, and Sara Hooker. 2024. [Aya model: An instruction finetuned open-access multilingual language model](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15894–15939. Association for Computational Linguistics.

Arda Y  ksel, Abdullatif K  ksal, L  tfi Kerem Senel, Anna Korhonen, and Hinrich Sch  tze. 2024. [TurkishMMLU: Measuring massive multitask language understanding in Turkish](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7035–7055. Association for Computational Linguistics.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. In *International Conference on Learning Representations (ICLR)*.

A Subjects

Table 2 lists the subjects in KuMMLU and the number of questions in each.

Subject	#Q
<i>Native (2,256)</i>	
Chemistry	336
Kurdish literature	289
Kurdish grammar	282
Biology	259
History	256
Economics	178
Geography	166
Physics	161
Mathematics (Gr. 9 and 12)	157
Religious studies	68
Social sciences	55
Kurdish rhetoric	49
<i>Translated from MMLU (1,016)</i>	
Philosophy	311
High school mathematics	264
College medicine	164
College chemistry	99
College mathematics	99
College physics	79
Total	3,272

Table 2: Subjects in KuMMLU and their question counts.

B Prompt Template

Each question is presented with the template below, shown here for a four-option question. The instruction is in English and the question and options are kept in Kurdish. For the API-served models the first line is sent as the system message and the rest as the user message; for the locally run multilingual baselines the whole template is given as a single prompt. The number of options (N) is three, four, or five depending on the question, and the final line asks for a number from 0 to N−1.

You are taking a multiple choice exam in Kurdish. Respond with ONLY the number of the correct answer. No explanation.

Question: {question}

Choices:

0: {option 0}

1: {option 1}

2: {option 2}

3: {option 3}

Answer with a single number from 0 to {N−1}:

C Per-Subject Accuracy

Table 3 reports accuracy for each subject, averaged over the 32 models. The three Kurdish-language subjects are the lowest-scoring.

Subject	Mean acc. (%)
Kurdish rhetoric	33.7
Kurdish grammar	36.1
Kurdish literature	40.8
Geography	44.6
History	45.4
College chemistry	48.0
Economics	49.1
Biology	49.7
College mathematics	53.6
High school mathematics	56.7
Chemistry	57.1
Physics	58.4
College physics	59.7
Philosophy	59.9
Mathematics (Gr. 9 and 12)	61.6
College medicine	63.2
Social sciences	63.4
Religious studies	65.3

Table 3: Per-subject accuracy, averaged over all 32 models.

D Example Questions

Table 4 gives English glosses of four native questions, one from each of grammar, rhetoric, history, and chemistry. The correct option follows the official answer key. The last column gives how many of the 27 API-served models answered correctly.

E Error Analysis

This analysis uses the per-question outputs of the 27 models served through the API; the multilingual baselines are not included.

Where the Kurdish-language subjects lose points. The drop on the Kurdish-language subjects is not limited to weak models. The five strongest models lose 16 to 30 points on grammar, literature, and rhetoric compared with the other native subjects (Gemini 3.1 Pro: 74.2% vs. 89.9%; GPT-5.6 (sol): 56.5% vs. 86.2%). Of the 620 questions in these three subjects, 70 (11.3%) are answered correctly by at most three of the 27 models, compared with 4.0% of the other native questions and 2.7% of the translated ones.

Reading these hardest items shows three recurring types. Grammar questions often ask for the role of a Sorani attached pronoun (for example, whether *-m* is the agent or a prepositional complement) or for a rewrite that moves a clitic, which requires a correct analysis of the Sorani clitic system. Literature questions ask who wrote a quoted verse, or which statement matches the curriculum’s account of a literary movement or journal; here

models tend to choose a generally plausible statement or “all of the above” instead of the answer the curriculum teaches, knowledge that is likely to be scarce in higher-resource-language training data. Rhetoric questions ask models to identify or count the rhetorical devices in a verse.

Errors are shared. Wrong answers are concentrated. On questions that at least five of the ten strongest models answer incorrectly (counting parsable answers only), on average 73% of their wrong answers on the Kurdish-language subjects, and 79% on the other subjects, go to the same distractor. This is consistent with shared distractor preferences, although it does not by itself establish their cause.

Position bias. Option shuffling places the correct answer in each position for about a quarter of the questions (Section 4.1), and the strongest models’ predictions are also close to uniform: Gemini 3.1 Pro, GPT-5.6 (sol), and Kimi K3 choose each position between 23.7% and 26.8% of the time. Several open-weight models show strong preferences instead: Ministral 14B chooses the first option in only 4.0% of four-option questions and the third in 52.0%, and Llama 3.3 70B chooses the third option 46.8% of the time. Because the shuffle makes these preferences independent of the correct answer, they should not systematically inflate scores, although a single fixed ordering cannot rule out chance effects in either direction.

F Response Rates

Most models returned a parsable answer on nearly every question. Table 5 lists the models whose response rate fell below 98%; all other models were above 98%.

Model	Response rate (%)
BLOOMZ-7B1	62.9
Qwen3 235B	83.3
GPT-5.6 (luna)	91.7
Llama 4 Scout	93.8
Llama 3.3 70B	94.0
Kimi K3	95.9
mT0-XXL	96.2
<i>All other models</i>	>98

Table 5: Models with response rate below 98%. A low rate means the model often did not return a parsable option number.

Subject	Question and options (English gloss)	Models correct
Grammar	In the sentence <i>kam qutabî be yekem derçû?</i> (‘Which student came first?’), what is <i>kam</i> ? (a) interrogative pronoun; (b) interrogative adverb; (c) interrogative adjective ; (d) adverb of manner	13/27
Rhetoric	In the verse ‘ <i>Ey biraderan, ba biçîne yarî / Newcewanîman lê bida diyarî</i> ’, the imperfect paronomasia (<i>yarî/diyarî</i>) lies in: (a) the kind of letters; (b) the number of letters ; (c) the order of letters; (d) the kind and number of letters	10/27
History	The forces of Sheikh Ubeydullah of Nehri, for liberating the part of Kurdistan under Qajar rule, were divided into how many columns? (a) three ; (b) four; (c) five; (d) six	13/27
Chemistry	In which cell does an electric current drive a non-spontaneous redox reaction? (a) fuel cell; (b) voltaic cell; (c) electrolytic cell ; (d) dry cell	17/27

Table 4: Example native KuMMLU questions with English glosses. Bold marks the answer in the official key.

A Theoretical Model of Hallucinations in NLP Large Language Models

Vicente Ivan Sanchez Carmona, Yuming Zhang, Junyu Zhou, Shanshan Jiang, Bin Dong

Ricoh Software Research Center (Beijing) Co., Ltd

{Vicente.Carmona, Yuming.Zhang1, Junyu.Zhou}@cn.ricoh.com

{Shanshan.Jiang, Bin.Dong}@cn.ricoh.com

Abstract

NLP Large Language Models have had a deep impact not only on research but also on society. However, there is one problem underlying their own nature which forbids them to be fully trusted and accountable: Hallucinations – a phenomenon that calls for a scientific account in the form of a theoretical model that represents the underlying rationale of how it emerges. In this paper, we present such a model. In the same spirit as in other fields proposing models of natural phenomena – such as rainfall – or social phenomena – such as knowledge creation in organizations – we propose a theoretical model that serves to explain hallucinations by synthesizing in an abstract and structured framework how the interrelations and interactions of more than 50 factors give rise to this phenomenon helping us to better understand its complexity and idiosyncrasies.

1 Introduction

Large Language Models (LLMs) have become ubiquitous systems across several domains such as Law (Nguyen, 2023), Medicine (Sorin et al., 2023), or Finance (Wu et al., 2023). However, one major obstacle to becoming fully trusted by users is the generation of hallucinations – nonfactual or non-sense outputs (Kaddour et al., 2023; Huang et al., 2025). Hallucinations can have negative impacts on society by, for example, imputing false crimes to people (Lomas, 2025), inventing law cases (Taylor, 2024), or suggesting users to ingest potentially poisonous chemicals (Eichenberger et al., 2025).

Previous works have shed light on possible causes of hallucination such as noisy training data (Ji et al., 2023), stochastic decoding methods (Huang et al., 2025), or LLMs not faithfully learning the data distribution (Lin et al., 2022). In turn, mitigation methods have been proposed such as improved decoding (Shi et al., 2024) and improved training (Lightman et al., 2024) methods.

These efforts have made LLMs less brittle to hallucinations. However, there is a fundamental, open problem: The scarcity of theoretical accounts that substantiate this phenomenon and bring conceptual understanding. Hallucinations from LLMs are not just a simple byproduct of the LLMs’ training data nor do they arise from the effect of isolated components; hallucinations are a complex phenomenon emerging from an intricate, systematic pattern of dependencies and interactions among the LLMs’ underlying components – including the data.

Let us consider two phenomena as illustrative examples. The phenomenon of rain is not independently caused by single factors; it is a set of interrelated factors including evaporation, condensation, temperature, humidity, atmospheric pressure, etc. that make rain to emerge (Pinho and Andrade, 1998; Nugent et al., 2019). In the social sciences, the phenomenon of knowledge creation in organizations is dependent on a framework of interrelated factors across cognitive, relational, and structural dimensions (Nahapiet and Ghoshal, 1998). As Poincaré argued, science not only advances by accumulating facts derived from works studying single factors, but by structuring them into models that allow us to theorize relations and patterns (Poincaré, 1913) – models that can enrich the NLP field to continue its scientific development.

In this paper, we propose a theoretical model of hallucinations in NLP LLMs. Our model aims to conceptually explain the emergence of this phenomenon via an abstract, high-level representation of how factors from LLMs systematically interrelate to cause hallucinations. Besides explanation, our theoretical model can serve multiple roles for future work. First, as a structured framework of previous works providing a visual aid for better informed surveys. Second, as a roadmap for planning both future experiments based on theorized relations and more complex mitigation methods that take into account the dependencies among fac-

tors. Third, as an abstract space where researchers can advance the conceptual knowledge of this phenomenon by discerning and deliberating proposed relations and by further theorizing on new factors and relations, thereby refining our comprehension of this phenomenon.

2 Background and Previous Works

2.1 Hallucinations in LLMs

The phenomenon of hallucinations has been framed as a tripartite concept (Huang et al., 2025; Zhang et al., 2025a; Ji et al., 2023): 1) Nonfactual hallucinations: LLMs’ generations are factually wrong or contradict information in well-established KBs or encyclopedias; 2) unfaithful hallucinations: LLMs’ generations contradict the input prompt or the generated output; and 3) unverifiable hallucinations: LLMs’ generations that cannot be verified due to the lack of reliable sources. We focus on the first two types of hallucinations since we believe are the most relevant in NLP research and applications.

Previous works have studied how candidate factors have played a role in the generation of hallucinations in NLP LLMs, and deep learning models, such as instruction-tuning (Pal et al., 2023), using maximum-likelihood for training (Dziri et al., 2022), or LLMs learning spurious word statistics from pretraining data (Kang and Choi, 2023), among others. However, the literature is lacking a theoretical model connecting these factors, via causal and associative links, into a structured and abstract representation depicting how the phenomenon of hallucinations arises.

Additionally, other works have studied the problem of hallucinations in LLMs from a theoretical standpoint based on either formal analyses of LLMs training process (Kalai et al., 2025), formal linguistic formulations of the generations of LLMs (Bouyamourn, 2023), or statistical models of the risk of hallucinations in LLMs (Gumaan, 2025). Nonetheless, these accounts are different in nature from our theoretical model. They are better categorized as formal mathematical or linguistic models which lie in different layers of abstraction compared to our proposed model: Whereas our model uses natural language and a visual aid as the main representational devices for encompassing the structure and logic of the phenomenon of hallucinations, those previous works use theorems and proofs as their main devices to establish necessary conditions for modeling hallucinations.

2.2 What is a Theoretical Model?

A theoretical model is an abstract, high-level, and conceptual representation of a set of systematically interrelated factors or concepts via causal (and other types of) links, which may or may not be grounded on empirical evidence, that depicts how a phenomenon emerges (Saunders et al., 2024) (Saunders et al., 2015). This class of model qualifies as a type of scientific explanation due to being a conceptual and general explanation of a phenomenon (Bhattacharjee, 2019). Although a theoretical model due to its conceptual nature is inherently theoretical, the theorized relationships among factors offer opportunities for empirical validation in future works (Swanson and Chermack, 2013).

3 Modeling Basis and Method for Building Our Theoretical Model

3.1 Choice of Theoretical Model

We use a variant of the theoretical model proposed by (McCain, 2022; Swanson and Chermack, 2013) because 1) it depicts dependencies and interactions between factors; 2) natural language can be used as the representational device for describing the relations underlying the model, as is common in the literature (N. and Rajendran, 2015; Eisend and Kuss, 2019); and 3) it explains how factors can be a direct/indirect cause/influence of the dependent variable – the generation of a hallucination by an LLM; i.e. it shows how factors are responsible for LLMs hallucinations (Figure 1 and Section 4).

3.2 Our Adapted Method for Building our Theoretical Model

Building a theoretical model comprises a subset of the steps for building scientific theories (N. and Rajendran, 2015). Hence, we borrow established theory-building methods from the Social Sciences due to their mature stage after decades of research (Bhattacharjee, 2019; Bacharach, 1989; Eisend and Kuss, 2019). Specifically, we adapt the deductive methods of (Swanson and Chermack, 2013; Whetten, 1989; N. and Rajendran, 2015), as well as the inductive methods of (Meza et al., 2023; Eisenhardt and Graebner, 2007) since it is difficult that a single method fits all possible scenarios for building a model and so methodological tailoring is required (Eisenhardt, 1989; Swanson and Chermack, 2013).

More concretely, we first review the NLP literature of LLM hallucinations; then, we select factors which are plausible causes of hallucination while

theorizing or extracting dependencies among them based on both the empirical evidence shown in the respective works and our domain knowledge. This bottom-up strategy allows us not only to synthesize in a structured way previous works, but also to find generalizable patterns while reconciling any possible inconsistency by crossing works and validating the effect of each factor on the generation of hallucinations. Moreover, this strategy also facilitates the extraction or proposal of explanations of why each factor causes the target phenomenon (Eisenhardt, 1989). This results in an emergent model grounded on previous works revealing causal (and other) patterns. After that, we reflect on the rationale underlying the proposed conceptual structure to deduce what possible factors are missing in order to have a coherent and logical model that accounts for a viable explanation of how hallucinations are generated. We then proceed to search for these factors in the literature and add them to our model following the procedure described above. If there are factors for which we were not able to find any previous evidence of their role in causing hallucinations, then we theorize, factor by factor, how they are related to the existing model built so far; this theorizing comes from domain knowledge and from efforts to keep our model both congruent with the existing dependencies and parsimonious. We then revisit our full model to search for possible missing links and to double-check the validity of each connection, as advised by (Rivard, 2021). We thus obtain our model of hallucinations (Figure 1).

3.3 Assumptions and Boundaries

We provide boundaries and assumptions under which our model is aimed to work. Our model is aimed for decoder-only Transformer NLP LLMs pretrained in English language (or multilingual where English is one of the main languages). We only consider the standard training process of LLMs: Pretraining, SFT, human-alignment (further assumptions in Section A.1).

3.4 Types of Relationships

Relations between factors depend on the domain (Salmons, 2019). We identify 6 types of relations: 1) *causal*, which corresponds to relations where the antecedent causes the consequent; 2) *influence* or *associative*, where the antecedent has an influence on the consequent, i.e. the consequent’s state value depends on the antecedent, but the antecedent does not cause the consequent; 3) *embodiment*, which

means that the antecedent’s effect will be embodied by the consequent; 4) *instantiation*, where the consequent is a specific type of factor of the antecedent; 5) *mediation*, where a factor is an intermediary between its antecedent and its consequent (e.g. B in $A \rightarrow B \rightarrow C$) and the effect of its antecedent depends on the state of the current factor; and 6) *interaction*, where two factors mutually modify the strength of their effects making them weaker or stronger depending on their state values. These relationships are differentiated in Figure 1 by color.

4 Theoretical Model of Hallucinations: Explaining Hallucinations

The phenomenon of hallucinations in NLP LLMs can be explained from factors from pretraining, Supervised Fine-tuning (SFT), human alignment training, the architecture of the LLM, and those arising at inference time. Figure 1 depicts our model connecting all those factors showing that some can be direct or indirect causes. We elaborate on this model by accounting for the logic behind what we consider the main candidate causes. In Section A.2, Tables 3 to 15 show all relations in Figure 1, and Tables 1 and 2 show examples of hallucinations, or scenarios where hallucinations can arise, for selected factors.

4.1 Causes of Hallucination from the Pretraining Stage

4.1.1 The Influence of Pretraining Data on Hallucinations

According to the type of **domain of pretraining data**, the states of factors such as **data quality**, **subjective information**, and **amount of pretraining data** may induce hallucinations (Ji et al., 2023; Huang et al., 2025; Li et al., 2024b). For example, social media documents, compared to scientific documents, might be of questionable quality due to containing biases, inaccurate info, noise, low quality of writing (syntactically and semantically), but they are abundant on the web. Thus, the way how the choice of domain data may cause hallucinations is via these features affecting the **parameters** state of an LLM: When trained on social media data, the embedding representations of entities and concepts will not be accurate nor rich in concise information; parameters at higher layers of the LLM may not encode proper grammar rules nor meaningful context information; thus, when these representations are used at inference time with an input prompt contain-

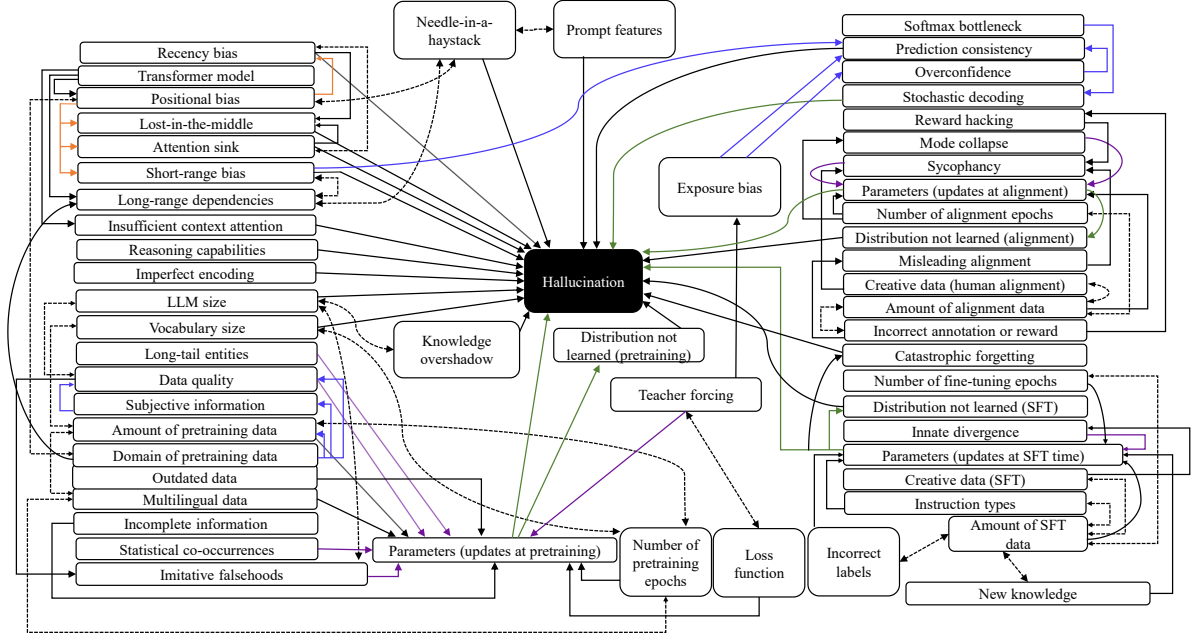


Figure 1: Our theoretical model of hallucinations in NLP LLMs. Single-head arrows indicate these relations: causal (black), influence (blue), embodiment (purple), instantiation (orange), or mediation (green). Double-head, dotted arrows indicate interaction. Description of all relations can be found in Tables 3 to 15 in Section A.2.

ing a novel context accompanying a target entity, the LLM will not be able to accurately extract information, or pay the correct attention weight to each token, leading to a nonfactual hallucination.

Hallucinations are also elicited by other data features. For instance, **outdated data** will allow outdated context information to be encoded in the entities embeddings, leading to a hallucination whenever an LLM provides this outdated information (Huang et al., 2025). Also, data containing **long-tail entities** will result in *poor* embeddings with insufficient context information, resulting in LLMs hallucinating due to the lack of knowledge (Xu et al., 2025). Moreover, **multilingual data** has been shown to impair high-resource language LLMs consequently leading to hallucinations: The variations of linguistic patterns of other languages may increasingly conflict with English patterns the higher the **amount of pretraining data** is used (Chang et al., 2024); i.e. the amount of pretraining data interacts with the multilinguality of the data (scenario in Table 1). Another possible cause of hallucination is the **incomplete information** of entities (Yang et al., 2023; Banerjee et al., 2024), i.e. when no document provides information about a specific aspect of an entity. Thus, when an LLM is queried about a missing aspect, it may respond

based on similar entities resulting in a nonfactual hallucination if, for example, the information requested is the birthday of a person. An additional cause is the frequent co-occurrence of two entities in the corpus – so-called **statistical co-occurrence** (Kang and Choi, 2023). The LLM may learn that regardless of their context, these entities should appear together; e.g., when queried with "What is the capital of Canada?" an LLM may incorrectly reply "Toronto" if Canada and Toronto appeared so frequently together overriding the correct association of capital_of(Canada, Ottawa) in the middle and late-layer self-attention heads (Yu et al., 2024).

4.1.2 The Influence of Pretraining on Hallucinations

Four main causes of hallucination arise as part of, or as a consequence of, pretraining: **Imitative falsehoods**, **teacher forcing**, **loss function**, and the **distribution of the data not being learned**. Inaccurate, noisy information and biases, such as social stereotypes, can be categorized as *falsehoods*, i.e. statements which truth value may not be bounded by reality; pretraining using data with falsehoods may lead to the phenomenon of **imitative falsehoods** (Lin et al., 2022) where an LLM not only memorizes falsehoods but it learns a latent model to mimic them, and by doing so it will halluci-

nate whenever given as input entities associated to such falsehoods. On the other hand, during pretraining, to predict the next token in a sequence, an LLM always receives the golden previous sequence; this **teacher-forced** expectation to receive the correct input sequence will forbid the LLM to learn to recover from wrong predictions; thus, at inference time, incorrect predictions may accumulate deriving into hallucinations (Dziri et al., 2022). Alongside this, the **loss function** (or pretraining objective) qualifies as a plausible factor inducing hallucinations (McKenna et al., 2023). The autoregressive nature of learning incentivizes the prediction of tokens based on statistical and linguistic patterns from the data; it does not enforce factuality (scenario in Table 1).

After pretraining, another cause may arise – indirectly caused by the **number of pretraining epochs** – namely, that the **pretraining distribution was not learned** or was not accurately encoded (Lin et al., 2022); this is a multivariate distribution encompassing variables such as statistics, attributes, and relations of entities and concepts, linguistic and semantic patterns, etc. Thus, not thoroughly capturing this distribution may translate into hallucinations due to insufficient context information or inaccurate patterns learned, such as an LLM not accurately learning the structure and semantics of the perfect present tense possibly due to insufficient exposure (example in Table 1).

4.2 Causes of Hallucination from the Supervised Fine-tuning Stage

4.2.1 The Role of SFT Data on Hallucinations

Similar to the pretraining scenario, some features from the SFT data can induce hallucinations in LLMs: Instances with **incorrect labels** can lead the training of an LLM to a **parameter** space where inputs are systematically mapped to wrong labels (Ji et al., 2023), leading the LLM to hallucinate by generating incorrect outputs as in the case of wrongly translated English-French sentences. Moreover, if the data contains **new knowledge** (Gekhman et al., 2024) unseen during pretraining, to accurately learn it, the training process may significantly alter the pretraining distribution encoded in the **parameters**, leading to **catastrophic forgetting** (Li et al., 2024a) which may cause hallucinations by altering or forgetting internalized knowledge (Zhai et al., 2023).

Moreover, another candidate cause of halluci-

nations is **creative data** (Dziri et al., 2022), such as open-ended dialogues, which enforce the phenomenon of **innate divergence** in LLMs by teaching them to be creative (we elaborate in Section 4.2.2). Finally, if the **amount of data** available for SFT is scarce it will not allow an LLM to accurately picking all the right statistical cues due to an insufficient backpropagation signal to fix errors and learn from them – **parameters** will not be accurately adjusted to learn the tasks correctly – specially for **instruction types** from complex tasks such as multi-hop QA or from different unrelated tasks (Li et al., 2024b); but even for simple tasks, insufficient data can lead to hallucinations (Kang et al., 2025) (example in Table 1).

4.2.2 The Role of SFT on Hallucinations

One of the contributing causes of hallucination from the SFT stage is the phenomenon of **innate divergence** (Ji et al., 2023), generated by dialogue data (and the like), which encapsulates the LLMs’ behavior of inherently diverging from the input context for the sake of creativity. This behavior may result in nonfactual hallucinations when dialogues are grounded in an ill-defined domain with low-quality data, as in the case of recreational data for chatbots where responses are expected to be not only creative but also *funny* and *entertaining* which may contain factually inaccurate statements. Another relevant cause of hallucination is the insufficient (or excessive) **number of fine-tuning epochs** which indirectly contributes to the **SFT distribution not being properly learned** (Pal et al., 2023) by leading to a **parameter** space where the LLM fails to pick necessary statistical, linguistic, and semantic patterns underfitting (or overfitting) the data which may lead to hallucinations (example in Table 2).

4.3 Causes of Hallucination from the Human Alignment Stage

4.3.1 The Impact of Alignment Data on Hallucinations

Features from the data for aligning LLMs to human preferences, similar to the SFT or pretraining stages, can also elicit hallucinations: **Incorrect annotations or rewards**, for DPO (Rafailov et al., 2023) or RLHF (Ouyang et al., 2022) methods, may lead the aligned LLM to prioritize certain aspects – such as fluency, avoiding biases, or being polite – over factuality (Chaduvula et al., 2026). The systematic application of these annotations (or

rewards) will lead to an underlying latent model of the data (or reward) known as **misleading alignment** (Zhang et al., 2025a) (when using DPO) or **reward hacking** (Wang et al., 2025) (when using RLHF), which in turn promotes the internalization of the phenomenon of **sycophancy** (explained in Section 4.3.2). Additionally, another reinforcing factor of sycophancy are **creative data** (Dziri et al., 2022) with an effect similar to that from the SFT stage, which we believe rather than be embodied as the phenomenon of innate divergence, it will be subsumed by the sycophancy phenomenon itself since the goal is to align to human criteria. Lastly, a similar effect from the **amount of data** available for alignment takes place as in previous stages.

4.3.2 The Impact of Alignment to Human Preferences on Hallucinations

Training an LLM to align to human preferences may allow certain factors to cause hallucinations. First, the **sycophancy** phenomenon (Wang et al., 2025) will form a dual internalization of a *servile* behavior possibly at the cost of factuality: Pleasing human values and generating creative outputs. This may give rise to a hallucination whenever an input prompt touches a contentious aspect such as: "Why men seem to hold more job positions in engineering than women?" where a possible response from an aligned LLM might be "that is an interesting but biased insight" when, in fact, the LLM could generate a neutral research-oriented and evidence-based factual response learned from reliable sources.

On the other hand, while an insufficient **number of alignment epochs** may contribute to **not learning the data distribution**, overtraining will lead the **parameters** to encode noise from the data, thus embodying the phenomenon of **mode collapse** (Hamilton, 2024), leading to hallucinations by not learning to generalize to different scenarios. For example, given prompts for alignment from educational scenarios such as "You are an empathetic LLM. Respond to the query while guiding through each step: how to prove the Pythagoras theorem?" the LLM may learn to associate being "empathetic" with educational contexts; thus, at inference time when the LLM receives an input with a medical scenario such as "You are an empathetic LLM. Respond to the query: I have these symptoms... How can I know if this is cancer?" the LLM may hallucinate procedural responses of medical with those of educational contexts and reply "Step 1, can you recognize the variables of the theorem?".

4.4 Causes of Hallucination from the LLM Architecture

Factors in the **Transformer model** such as the attention mask, context window size, the positional encoding, as well as issues such as attention dilution or vanishing gradients contribute to some types of **positional bias** (Wu et al., 2025b), such as **lost-in-the-middle**, **attention sink**, **short-range bias**, or **recency bias**, and to the issues of **long-range dependencies** and **insufficient context attention**; these factors contribute to the generation of hallucinations. **Lost-in-the-middle** (Liu et al., 2024) is the phenomenon where tokens placed in the middle of a long input context do not receive as big attention weights as tokens placed at the beginning or end of the input; this can lead to unfaithful hallucinations, for example, in lengthy input documents if the answer of a given query is placed in the middle. One possible reason for this phenomenon to arise is the presence of **recency bias** (Zhou et al., 2024) which occurs when an LLM puts more attention on recent tokens in the input sequence rather than on prior tokens; this bias may also lead to hallucinations if, for example, the LLM is tasked with summarizing a document and at some point the most recent tokens in the input sequence are those generated by the LLM in which case the LLM may put more focus on such generations. Another possible contributing factor to the lost-in-the-middle phenomenon is the phenomenon of **attention sink** (Gu et al., 2025; Xiao et al., 2024) which happens when some tokens at the beginning of an input prompt receive most of the attention as if they were *pulling* the attention towards them, possibly causing hallucinations when key information required to answer a query is placed towards the end of the input sequence. On the other hand, the **short-range bias** (Kazemnejad et al., 2023) happens due to the similar positional encoding vectors of proximal tokens, which affects their alignment and attention scores influencing them to be higher, leading to bigger attention weights for localized clusters of tokens and to similar hallucinations as the other positional biases. Additionally, an indirect cause of hallucinations is the factor of **long-range dependencies** (Assogba and Ren, 2025; Wu et al., 2025a) where two semantically related but distant tokens share a small, rather than a big, attention weight due to their distance. This may lead to hallucinations in long input sequences by making the LLM to focus on specific localized clusters of the sequence

whereas attention should be put across the whole sequence including both distant tokens which will result in the LLM not properly contextualizing the vectors.

Moreover, it has been shown that some LLMs pay more attention to their parametric memory than to the input context when answering a query or summarizing a document – a phenomenon known as **insufficient context attention** (Shi et al., 2024); this can effectively lead to a hallucination if the correct response to the query or the information required for the summary is contained only in the input context and any information retrieved from parametric memory may contradict it.

A further consideration is any limitation in the **reasoning capabilities** of LLMs which may also induce hallucinations (Zhang et al., 2025a). For example, failure to carry out a complex reasoning throughout input documents to answer complex historiographical queries can lead to hallucinations (example in Table 2). Another contributing factor to the generation of hallucinations is the **LLM size** (number of parameters) (Huang et al., 2025): Large LLMs may generate less hallucinations than small LLMs because the former may have a better capacity for faithfully learning the data distribution.

Additionally, we identify two factors that to our best knowledge have remained unexamined in the LLM literature. Similar to encoding issues in previous NLG models (Ji et al., 2023), the **imperfect encoding** of contextual information in the embeddings of entities and concepts may cause hallucinations; this information is not copy-pasted from the documents, it is constructed and embedded which can lead to loss or deformation of information at any step during the mapping process; this can lead to an incomplete internalization of an entity – not all its features and relations are included in its embedding. Thus, when an LLM is queried about such missing information of an entity it will hallucinate it by responding with information from a similar entity. The second factor we consider is the **vocabulary size** of an LLM. We contend that the finer-grained the vocabulary of an LLM,¹ the less hallucinations: In an ideal scenario each concept has its own embedding which accurately and solely captures its intended contextual meaning, provided that enough **amount of pretraining data** is available; in a less ideal scenario where concepts share

token-level embeddings, the overlap in contexts may lead to ambiguity and in turn to hallucination (example in Table 2).

4.5 Causes of Hallucination Due to Factors at Inference Time

Another set of critical factors contributing to the generation of hallucinations are those arising at inference time. We start our account with the phenomenon of **softmax bottleneck** (Chang and McCallum, 2022) where inside the set of k highest-probability candidate tokens for a prediction, some tokens will be semantically related among them but they may be unrelated to the input query. As demonstrated and analyzed in (Chang and McCallum, 2022), given an input query such as “When the king met the woman, protocol dictated that the [MASK] should be the one to greet first”, any of the two tokens ‘woman’ or ‘king’ would be a faithful choice; nevertheless, due to the softmax bottleneck, a plausible set of $k = 4$ candidate tokens would be ‘king’, ‘queen’, ‘woman’, ‘man’ which is generated due to the geometrical configuration $queen - king = woman - man$. Connected to this phenomenon, the effect of **stochastic decoding** can contribute to a faithfulness hallucination (Dziri et al., 2022) by choosing either the token of ‘queen’ or ‘man’ since methods such as top-k or top-p add randomness to avoid the *likelihood trap*² leading to a more human-like, creative output.

Another cause of hallucination is the phenomenon we term **prediction consistency**, also known as **hallucination snowballing** (Zhang et al., 2024), where LLMs incorrectly predict the first tokens in a sequence generation and rather than predicting subsequent tokens that correct the previous tokens, the LLMs seem to *prefer* to keep consistency in the generation continuing the hallucination (example in Table 2). Aligned to this, LLMs have been shown to exhibit some predictions with **overconfidence** (Zhang et al., 2025a), in the form of high likelihood or probability, despite being hallucinations. This factor may play a role in the phenomenon of **prediction consistency** by reinforcing the hallucinated sequence. Overconfidence when predicting a hallucination may be caused by a latent pattern learned at training time; for example, given the query “in the USA robots can already operate [MASK]”, LLMs may respond “patients” where the correct answer is “vehicles”;

¹The more the number of meaningful entries in the vocabulary added prior to pretraining.

²High-likelihood sequences are more prone to lead to lower-quality text.

this may occur not because of simple contextual features in pretraining data, but because at a higher semantic level the LLM built a representation of robots having near-human capabilities, as reported in some publications and news, leading to confidently predict "patients" to fit in the discourse level the narrative that robots have such advanced abilities. Equally significant is the phenomenon of **exposure bias** (Ji et al., 2023): Due to the mechanism of **teacher forcing** (Section 4.1.2), LLMs were trained by always receiving a gold input sequence for generating the next token; at inference time, the interruption of this scheme may lead to an unpredictable behavior that can influence the quality of the generations. We believe this breach between training and inference could influence the factor of **prediction consistency**: The LLM's divergence from predicting a factual sequence may be stimulated or exacerbated by the **exposure bias** effect by not being able to control or correct the generation due to not having been trained to do so.

In addition, **prompt features** also play a role on the hallucinations of LLMs. Specifically, the **prompt length** can increase the risk of generating hallucinations (Liu et al., 2024; Rawte et al., 2024): Prompts reaching the maximum input length permissible can lead to the problems of long-range dependencies or lost-in-the-middle where key contextual information may be neglected to some extent, forbidding a fully-grounded answer generation, as explained in Section 4.4. Furthermore, **query specificity** or **context precision** can lead to hallucinations as well (Janiak et al., 2025): Vague queries or incomplete contexts can lead to inaccurate answers; for example, the query "What is the best way to remove stains?" may lead to a hallucinated answer from a user's point of view if the LLM replies "you can remove stains from clothes with product xyz" whereas the expected answer was to do so from the fabric of a sofa. Additionally, we posit that an LLM receiving an **unseen instruction type** (an instruction not seen at training time) can also induce a hallucination simply by not knowing what action or task to carry out, as in the prompt: "let's play Werewolf" which is a game that is not frequently used for training LLMs.

A further cause of hallucinations is the phenomenon of **knowledge overshadow** (Zhang et al., 2025b) occurring when prompts contain more than one entity: The highest-frequency entity in pretraining data becomes the *dominant* entity, overshadowing the less frequent ones, which can result in a

hallucination if the main entity in the query is not the dominant one. As pinpointed in (Zhang et al., 2025b), given a query such as "Between the Arc de Triomphe and the Musée d'Orsay, which landmark is more popular in France?" if France overshadows the Arc de Triomphe and the Musée d'Orsay, then the LLM may interpret the query as requesting information about a popular landmark in France for which the LLM may output one of the most popular such as the Louvre or the Eiffel Tower. The last prompt feature we consider can actually have a positive impact on LLMs: Adding **in-context examples** to guide an LLM's generation by showing both the principles of the task and the expected response can reduce the chance of hallucination (Pal et al., 2023). For instance, if a prompt contains an unseen instruction such as "write a poem about the Earth" but the in-context examples demonstrate the expected output as in "write a poem about the Sun. The Sun is a beautiful star which warms us at day" then the LLM will use these examples to better contextualize the query by extracting the intent, inferring the input-output process, and mimicking the output.

Ultimately, the last salient cause of hallucinations from inference time we address is the problem of **needle-in-a-haystack**: Finding a piece of key information that can serve as an answer to a question – a *needle* – within a wide body of information in the context – a *haystack* (Banerjee et al., 2024). It has been shown that the accuracy of retrieving a needle declines a) with increased context length, b) when the needle is placed at the beginning or middle of the haystack, and c) with smaller needle lengths (Bianchi et al., 2025; Schuster et al., 2025). Thus, depending on these attributes is the difficulty of retrieving the target piece of information from a lengthy input context and failure to do so translates into a hallucination.

5 Conclusions

We presented a theoretical model of hallucinations in NLP LLMs: An abstract structure with its underlying rationale of how a set of interconnected factors gives rise to this phenomenon. A theoretical model, while well grounded on empirical findings, needs not be purely empirically driven – hence its name *theoretical* – as it will also abstract the domain knowledge of researchers whom, on intellectual grounds, theorize about relationships and factors from which new findings can be derived via

future experiments. We claim, the NLP field has studied phenomena mainly experimentally, leaving a plethora of scattered discoveries; synthesizing these works into conceptual representations, as done here, can provide us with a better conceptual understanding of how each phenomenon emerges, thus advancing our field.

Limitations

One natural limitation of this work is the possibility of not being exhaustively comprehensive and thus not including one or more causes of hallucination in NLP LLMs. As new research is published, new causes emerge. Nevertheless, the proposed theoretical model can be easily updated and modified by other researchers in the field satisfying any field-related idiosyncrasies from where the new research has been published. Thereof, we leave as an open, future work the continuous update of this model by all interested researchers.

The theoretical model may embody hypothetical relations or factors which may be proven inaccurate in future works. This is precisely one objective of this model. Any theorizing is not randomly proposed; it comes from our domain knowledge; it is to the best of our intellectual effort that we consider such hypotheses to hold true, aligned to the evidence from previous works, in the hope that future work will prove them correct or incorrect, as in the way any scientific field advances.

References

- Yannick Assogba and Donghao Ren. 2025. [Evaluating long range dependency handling in code generation llms](#). *Preprint*, arXiv:2407.21049.
- Samuel B. Bacharach. 1989. [Organizational theories: Some criteria for evaluation](#). *The Academy of Management Review*, 14(4):496–515.
- Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. 2024. [Llms will always hallucinate, and we need to live with this](#). *Preprint*, arXiv:2409.05746.
- Anol Bhattacharjee. 2019. *Social Science Research: Principles, Methods and Practices - (Revised edition)*. University of Southern Queensland.
- Owen Bianchi, Mathew J. Koretsky, Maya Willey, Chelsea X. Alvarado, Tanay Nayak, Adi Asija, Nicole Kuznetsov, Mike A. Nalls, Faraz Faghri, and Daniel Khashabi. 2025. [Hidden in the haystack: Smaller needles are more difficult for llms to find](#). *Preprint*, arXiv:2505.18148.
- Adam Bouyamourn. 2023. [Why LLMs hallucinate, and how to get \(evidential\) closure: Perceptual, intentional, and extensional learning for faithful natural language generation](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3181–3193, Singapore. Association for Computational Linguistics.
- Sindhuja Chaduvula, Ahmed Y. Radwan, Azib Farooq, Yani Ioannou, and Shaina Raza. 2026. [Reducing hallucinations in llms via factuality-aware preference learning](#). *Preprint*, arXiv:2601.03027.
- Haw-Shiuan Chang and Andrew McCallum. 2022. [Soft-max bottleneck makes language models unable to represent multi-mode word distributions](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8048–8073, Dublin, Ireland. Association for Computational Linguistics.
- Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Benjamin K. Bergen. 2024. [When is multilinguality a curse? language modeling for 250 high- and low-resource languages](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4074–4096, Miami, Florida, USA. Association for Computational Linguistics.
- Nouha Dziri, Sivan Milton, Mo Yu, Osmar Zaiane, and Siva Reddy. 2022. [On the origin of hallucinations in conversational models: Is it the datasets or the models?](#) In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5271–5285, Seattle, United States. Association for Computational Linguistics.
- Audrey Eichenberger, Stephen Thielke, and Adam Van Buskirk. 2025. A case of bromism influenced by use of artificial intelligence. *Annals of Internal Medicine: Clinical Cases*, 4(8).
- Martin Eisend and Alfred Kuss. 2019. *Research Methodology in Marketing: Theory Development, Empirical Approaches and Philosophy of Science Considerations*. Springer International Publishing, Cham.
- Kathleen Eisenhardt and Melissa Graebner. 2007. [Theory building from cases: Opportunities and challenges](#). *The Academy of Management Journal*, 50:25–32.
- Kathleen M. Eisenhardt. 1989. [Building theories from case study research](#). *The Academy of Management Review*, 14(4):532–550.
- Zorik Gekhman, Gal Yona, Roei Aharoni, Matan Eyal, Amir Feder, Roi Reichart, and Jonathan Herzig. 2024. [Does fine-tuning LLMs on new knowledge encourage hallucinations?](#) In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 7765–7784, Miami, Florida, USA. Association for Computational Linguistics.

- Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. [When attention sink emerges in language models: An empirical view](#). In *International Conference on Learning Representations*, volume 2025, pages 97114–97144.
- Esmail Gumaan. 2025. [Theoretical foundations and mitigation of hallucination in large language models](#). Preprint, arXiv:2507.22915.
- Sil Hamilton. 2024. [Detecting mode collapse in language models via narration](#). In *Proceedings of the First edition of the Workshop on the Scaling Behavior of Large Language Models (SCALE-LLM 2024)*, pages 65–72, St. Julian’s, Malta. Association for Computational Linguistics.
- Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025. [A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions](#). *ACM Trans. Inf. Syst.*, 43(2).
- Denis Janiak, Jakub Binkowski, Albert Sawczyn, Bogdan Gabrys, Ravid Shwartz-Ziv, and Tomasz Jan Kajdanowicz. 2025. [The illusion of progress: Re-evaluating hallucination detection in LLMs](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34728–34745, Suzhou, China. Association for Computational Linguistics.
- Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. [Survey of hallucination in natural language generation](#). *ACM Comput. Surv.*, 55(12).
- Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. [Challenges and applications of large language models](#). Preprint, arXiv:2307.10169.
- Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. 2025. [Why language models hallucinate](#). Preprint, arXiv:2509.04664.
- Cheongwoong Kang and Jaesik Choi. 2023. [Impact of co-occurrence on factual knowledge of large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7721–7735, Singapore. Association for Computational Linguistics.
- Katie Kang, Eric Wallace, Claire Tomlin, Aviral Kumar, and Sergey Levine. 2025. [Unfamiliar finetuning examples control how language models hallucinate](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3600–3612, Albuquerque, New Mexico. Association for Computational Linguistics.
- Amirhossein Kazemnejad, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Payel Das, and Siva Reddy. 2023. The impact of positional encoding on length generalization in transformers. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA. Curran Associates Inc.
- Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024a. [Revisiting catastrophic forgetting in large language model tuning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA. Association for Computational Linguistics.
- Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024b. [The dawn after the dark: An empirical study on factuality hallucination in large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.
- 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.
- 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.
- Natasha Lomas. 2025. [Chatgpt hit with privacy complaint over defamatory hallucinations](#). TechCrunch. Last accessed: 2026-05-24. <https://techcrunch.com/2025/03/19/chatgpt-hit-with-privacy-complaint-over-defamatory-hallucinations/>.
- Kevin McCain. 2022. *Understanding How Science Explains the World*. Understanding Life. Cambridge University Press.
- Nick McKenna, Tianyi Li, Liang Cheng, Mohammad Hosseini, Mark Johnson, and Mark Steedman. 2023. [Sources of hallucination by large language models on inference tasks](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2758–2774, Singapore. Association for Computational Linguistics.
- Rosemary D. Meza, James C. Moreland, Michael D. Pullmann, Predrag Klasnja, Cara C. Lewis, and Bryan J. Weiner. 2023. [Theorizing is for everybody](#):

- Advancing the process of theorizing in implementation science. *Frontiers in Health Services*, Volume 3 - 2023.
- Elangovan N. and Raju Rajendran. 2015. [Conceptual model: A framework for institutionalizing the vigor in business research](#). In *Third National Conference on Indian Business Management*.
- Janine Nahapiet and Sumantra Ghoshal. 1998. [Social capital, intellectual capital, and the organizational advantage](#). *The Academy of Management Review*, 23(2):242–266.
- Ha-Thanh Nguyen. 2023. [A brief report on lawgpt 1.0: A virtual legal assistant based on gpt-3](#). *Preprint*, arXiv:2302.05729.
- Alison Nugent, David DeCou, Shintaro Russell, and Christina Karamperidou. 2019. *Atmospheric Processes and Phenomena*. UH Pressbooks The University of Hawaii.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. 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 Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. [Med-HALT: Medical domain hallucination test for large language models](#). In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–334, Singapore. Association for Computational Linguistics.
- Suani T.R. Pinho and Roberto F.S. Andrade. 1998. [An abelian model for rainfall](#). *Physica A: Statistical Mechanics and its Applications*, 255(3):483–495.
- Henri Poincaré. 1913. *The Foundations of Science: Science and Hypothesis, The Value of Science, Science and Method*. The Science Press, New York and Garrison, N. Y. Translated by George Bruce Halsted. Project Gutenberg release date: May 16, 2012 [eBook #39713]. Press of the New Era Printing Company, Lancaster, PA.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Vipula Rawte, S. M Towhidul Islam Tonmoy, S M Mehedi Zaman, Prachi Priya, Aman Chadha, Amit P. Sheth, and Amitava Das. 2024. ["sorry, come again?" prompting – enhancing comprehension and diminishing hallucination with \[pause\]-injected optimal paraphrasing](#). *Preprint*, arXiv:2403.18976.
- Suzanne Rivard. 2021. Theory building is neither an art nor a science. it is a craft. *Journal of Information Technology*, 36(3):316–328.
- Janet E. Salmons. 2019. *Little Quick Fix: Find the Theory in Your Research*. SAGE Publications Ltd.
- Mark Saunders, David Gray, Paul Tosey, and Eugene Sadler-Smith. 2015. *Concepts and theory building*, pages 35–56. SAGE Publications.
- Mark N. K. Saunders, Philip Lewis, and Adrian Thornhill. 2024. *Research Methods for Business Students, 9th edition*. Pearson.
- Thomas Schuster, Marian Lambert, Nico Döring, and Julius Trögele. 2025. Needle-in-the-haystack testing llms with a complex reasoning task. In *Engineering Applications of Neural Networks*, pages 254–266, Cham. Springer Nature Switzerland.
- Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. [Trusting your evidence: Hallucinate less with context-aware decoding](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.
- Vera Sorin, Eyal Klang, Miri Sklair-Levy, Israel Cohen, Douglas B. Zippel, Nora Balint Lahat, Eli Konen, and Yiftach Barash. 2023. [Large language model \(chatgpt\) as a support tool for breast tumor board](#). *npj Breast Cancer*, 9.
- Richard Swanson and Thomas Chermack. 2013. *Theory Building in Applied Disciplines*. Berrett-Koehler Publishers, Inc.
- Josh Taylor. 2024. [Melbourne lawyer referred to complaints body after ai generated made-up case citations in family court](#). The Guardian. Last accessed: 2026-05-24.
- Chaoqi Wang, Zhuokai Zhao, Yibo Jiang, Zhaorun Chen, Chen Zhu, Yuxin Chen, Jiayi Liu, Lizhu Zhang, Xiangjun Fan, Hao Ma, and Sinong Wang. 2025. [Beyond reward hacking: Causal rewards for large language model alignment](#). *Preprint*, arXiv:2501.09620.
- David A. Whetten. 1989. [What constitutes a theoretical contribution?](#) *The Academy of Management Review*, 14(4):490–495.
- Longyun Wu, Dawei Zhu, Guangxiang Zhao, Zhuocheng Yu, Junfeng Ran, Xiangyu Wong, Lin Sun, and Sujian Li. 2025a. [LongAttn: Selecting long-context training data via token-level attention](#). In *Findings of the Association for Computational*

- Linguistics: ACL 2025*, pages 19367–19380, Vienna, Austria. Association for Computational Linguistics.
- Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhajan Kam-badur, David Rosenberg, and Gideon Mann. 2023. [Bloomberggpt: A large language model for finance](#). *Preprint*, arXiv:2303.17564.
- Xinyi Wu, Yifei Wang, Stefanie Jegelka, and Ali Jad-babaie. 2025b. [On the emergence of position bias in transformers](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 67756–67781. PMLR.
- Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024. [Efficient streaming language models with attention sinks](#). In *International Conference on Learning Representations*, volume 2024, pages 21875–21895.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2025. [Hallucination is inevitable: An innate limitation of large language models](#). *Preprint*, arXiv:2401.11817.
- Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023. [A new benchmark and reverse validation method for passage-level hallucination detection](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 3898–3908, Singapore. Association for Computational Linguistics.
- Lei Yu, Meng Cao, Jackie CK Cheung, and Yue Dong. 2024. [Mechanistic understanding and mitigation of language model non-factual hallucinations](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7943–7956, Miami, Florida, USA. Association for Computational Linguistics.
- Yuxiang Zhai, Shengbang Tong, Xiao Li, Mu Cai, Qing Qu, Yong Jae Lee, and Yi Ma. 2023. [Investigating the catastrophic forgetting in multimodal large language model fine-tuning](#). In *Conference on Parsimony and Learning (Proceedings Track)*.
- Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. 2024. How language model hallucinations can snowball. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Ahn Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025a. [Siren’s song in the ai ocean: A survey on hallucination in large language models](#). *Computational Linguistics*, pages 1–45.
- Yuji Zhang, Sha Li, Cheng Qian, Jiateng Liu, Pengfei Yu, Chi Han, Yi R. Fung, Kathleen McKeown, Chengxiang Zhai, Manling Li, and Heng Ji. 2025b. [The law of knowledge overshadowing: Towards understanding, predicting, and preventing llm hallucination](#). *Preprint*, arXiv:2502.16143.
- Hanzhang Zhou, Zijian Feng, Zixiao Zhu, Junlang Qian, and Kezhi Mao. 2024. Unibias: Unveiling and mitigating llm bias through internal attention and ffn manipulation. In *Advances in Neural Information Processing Systems*, volume 37, pages 102173–102196. Curran Associates, Inc.

A Appendix

A.1 Further Assumptions and Boundaries

We further delimit the scope under which our theoretical model is valid. We do not consider outer indirect factors such as the manufacturer of the LLM which may influence its programming quality. Furthermore, we view an LLM as a standalone system, and so we do not consider external sources or methods such as Retrieval Augmented Generation (RAG). Additionally, since our model only concerns with explanation of hallucinations, we do not consider any mitigation method. In addition, we define an input prompt as the composition: prompt = in-context learning examples + context + query, where 1) the operator ‘+’ means text concatenation, 2) a query can also be an instruction, and 3) the only element that must always appear in a prompt is the query (the other elements are optional). In this paper, if no specific element is specified in the example prompts provided, then it is assumed that the prompt only contains a query or an instruction.

A.2 Tables

Origin	Factor	Example or Scenario
Pretraining data	Multilingualism	As mentioned in Section 4.1.1, depending on the amount of multilingual tokens in the data is the magnitude of the effect of multilingualism on causing hallucinations such as input queries with complex English syntactic structures eliciting hallucinations due to the LLM’s inability of correctly parsing them caused by the impairment mentioned in Section 4.1.1.
Pretraining	Loss function	As mentioned in Section 4.1.2, the loss function does not incentivize factuality; thus, even when pretraining using only factual data, hallucinations can still occur when the LLM is queried with novel concepts or entities because this leads the LLM to infer new information from its parametric memory, and to do so, the LLM will simply repeat the learned generation process of predicting tokens based on patterns recovered from data.
Pretraining	Not learning the pretraining data distribution	As mentioned in Section 4.1.2, one possible consequence of not accurately learning the pretraining data distribution is the LLMs’ inability to faithfully capture grammar rules or tenses; thus, with a query in present perfect tense such as "Who has been the president of the USA for the last 3 years?" An LLM may interpret this as a past tense structure, and answer it with the name of the president 3 years ago which may have been another person if in that year there was a change of administration.
SFT data	Amount of SFT data	As mentioned in Section 4.2.1, even for simple tasks, insufficient data can lead to hallucinations; for example, given the prompt: "Translate the following question to French: what is the capital of France?" the LLM may misinterpret it as a question seeking for a capital city and respond "Paris" which is the correct answer to the question "what is the capital of France?" picked from pretraining, but it is an incorrect answer to the target instruction requesting a translation.

Table 1: Examples of hallucinations, or cases when hallucinations may arise, for selected factors.

Origin	Factor	Example or Scenario
SFT	Not learning the SFT data distribution	As mentioned in Section 4.2.2, not accurately picking the right statistical cues and patterns due to overfitting can lead to hallucinations; for example, given the instruction "Write a list of the top 40 most famous songs of 2026" an LLM may return a list of only 10 songs if it overfitted to the SFT examples requesting for lists of 10 items across diverse domains such as the 10 best movies or the 10 most popular cars.
LLM architecture	Reasoning capabilities	As mentioned in Section 4.4, failure in the capability to carry out a complex reasoning for complex queries, such as some historiographical queries, can lead to providing an incorrect answer and thus a hallucination; for example, for the query "What were the motivations of those whose actions contributed to the onset of the First World War?" an LLM needs to first figure out the actions that caused the first world war, then find whom were behind those actions, and finally discover their motivations.
LLM architecture	Vocabulary size	As mentioned in Section 4.4, a smaller and coarser vocabulary can lead to some scenarios of hallucination, as for example, with the query "Is magpie the favorite among people?" since 'magpie', the bird, may be tokenized as 'mag' (short for magazine) and 'pie' (the dessert) with no concise way to disambiguate due to the unclear context; but, this ambiguity may not happen if the vocabulary included 'magpie'.
Inference time	Prediction consistency	As pinpointed in (Zhang et al., 2024), given an input query such as "is 2^{10} greater than 10^3 ?" LLMs may incorrectly respond "no" and then give a reason such as "because the base of 10 grows faster than the base of 2", continuing the hallucination despite knowing the correct computations, which can be verified by querying the LLMs with "what is 2^{10} ?", "what is 10^3 ?", and then "is 1024 greater than 1000?" where LLMs respond "yes".

Table 2: Continuation of examples of hallucinations, or cases when hallucinations may arise, for selected factors.

Factors		Relation	Proposition
Antecedent	Consequent		
Data quality	Imitative falsehoods	causal	The presence of inaccurate facts, biases, noisy data in the pretraining corpus, i.e. <i>falsehoods</i> , will cause LLMs to incorporate the ability to imitate such falsehoods into their behavioral repertoire
Data quality	Parameters (updates)	embodiment	The quality of features and contexts info from entities and concepts, which will be learned and encoded in the parameters, depend on the data quality
Subjective information	Data quality	influence	For some domains, subjective beliefs from authors may take the form of biases or inaccurate facts which can in turn contribute to the pool of falsehoods
Amount of pretraining data	Parameters (updates)	causal	The number of pretraining documents will affect the quality of the knowledge in the parameters (e.g. the less the documents, the less the contextual information coverage)
Domain of pretraining data	Data quality	influence	The domain of the pretraining corpus may influence the likelihood of observing biases, inaccurate facts, certain writing style, vagueness, ambiguity, or noise in the corpus
Domain of pretraining data	Subjective information	influence	The domain of the pretraining corpus may influence the objectivity of the data (e.g. scientific domains may be more objective than social media domains)
Domain of pretraining data	Amount of pretraining data	influence	The domain of the pretraining corpus may be related to the amount of available data (e.g. social media documents are more abundant than scientific documents)
Long-tail entities	Parameters (updates)	embodiment	Highly infrequent entities will result in their embeddings having poor quality (not storing enough contextual information about such entities)
Outdated data	Parameters (updates)	causal	An outdated pretraining corpus will cause the embeddings of the entities and concepts to contain outdated information
Multilingual data	Parameters (updates)	causal	Variations in the linguistic patterns of other languages in the pretraining corpus may conflict with English patterns causing conflicts in the encoded information in the parameters

Table 3: Edges in our theoretical model of hallucinations (pretraining stage) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Incomplete information	Parameters (updates)	causal	Missing information of entities or concepts in the pretraining corpus will cause a poor or incomplete contextualization of their embeddings
Statistical co-occurrences	Parameters (updates)	embodiment	The more frequently two or more entities co-occur in the pretraining corpus, the higher the chance of capturing this statistical association in the parameters
Number of pretraining epochs	Parameters (updates)	causal	Under some conditions, a sufficient number of pretraining epochs will cause the parameters to optimally or accurately encode contextual information
Imitative falsehoods	Parameters (updates)	embodiment	The expression of the behavior of imitative falsehoods will be embodied by the LLM via the update of parameters after pretraining
Teacher forcing	Parameters (updates)	embodiment	The forced <i>expectation</i> of the LLM to receive always a golden input sequence to predict the next token will be embodied in the parameters
Loss function	Parameters (updates)	causal	The autoregressive loss function causes the prediction of tokens to be based on statistical/linguistic patterns which leads the parameters to a certain space to accurately learn those patterns
Parameters (updates)	Distribution not learned	mediation	Antecedent factors embodied in the parameters may cause or influence the LLM to not properly learn the distribution of the pretraining data, and is via the current configuration of the parameters that this takes effect
Parameters (updates)	Hallucination	mediation	The parameters are one way through which the previous antecedents can take effect. After pretraining, the parameters will contain the combined effects of all the antecedent causes of hallucination which will be triggered after the LLM receives an input prompt
Distribution not learned	Hallucination	causal	If the distribution of the pretraining data is not properly learned, it will lead the LLM to not properly recovering the linguistic or statistical patterns required to generate a factual or faithful prediction

Table 4: Continuation of edges in our theoretical model of hallucinations (pretraining stage) and their corresponding relation type and propositional statement.

Factors			Relation	Proposition
Antecedent	Consequent			
Incorrect labels	Parameters (updates)		causal	The pairing of inputs with incorrect labels, derived from a wrong annotation scheme or human error, will be learned and encoded in the parameters
Amount of SFT data	Parameters (updates)		causal	Insufficient instances to learn a specific domain or task may lead to a sub-optimal parameter configuration
New knowledge	Parameters (updates)		causal	New knowledge in the SFT data (not seen during pretraining) may either shift, modify, or significantly alter the pretraining distribution encoded in the parameters
Instruction types	Parameters (updates)		causal	If the SFT data contains instructions that are significantly complex, and/or the diversity of instruction types is high, then these instances may lead the training to a sub-optimal parameter space causing the tasks represented in these instances to not be properly learned
Creative data (SFT)	Innate diverge	di-	causal	Instances which show the LLM to behave in <i>creative</i> ways by purposefully diverging from accurate or factual information will lead to the phenomenon of innate divergence
Innate divergence	Parameters (updates)		embodiment	The repeated exposure to <i>creative</i> instances, such as open-ended dialogues, can lead to embodying the behavior of purposefully generating creative sequences, possibly diverging from factuality

Table 5: Edges in our theoretical model of hallucinations (SFT stage) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Number of fine-tuning epochs	Parameters (updates)	causal	Insufficient training epochs can lead to underfitting while the opposite can lead to overfitting; any scenario will shift the parameters to a specific parameter space reflecting characteristics of such a scenario such as learning noise from the data
Parameters (updates)	Distribution not learned	mediation	Similar to the pretraining scenario, particular states of antecedent factors of the parameters may lead to not properly or accurately recovering all the statistics and patterns from the SFT data, leading to an imprecise embodiment of the SFT data distribution
Parameters (updates)	Catastrophic forgetting	causal	Learning new knowledge at SFT time can lead to a parameter configuration which causes previously learned abilities or knowledge to be destroyed, overridden, or forgotten
Parameters (updates)	Hallucination	mediation	The encoded effects of the antecedents of the parameters will indirectly and effectively lead LLMs to hallucinate when an input prompt meets the conditions for triggering one or several of them
Catastrophic forgetting	Hallucination	causal	Significant alteration or forgetting of previous knowledge from pretraining can lead to hallucinations
Distribution not learned	Hallucination	causal	Incorrectly or poorly modeled statistical, linguistic, and semantic patterns from the data will lead to hallucinations when the concepts or tasks involved in such patterns are invoked via an input prompt

Table 6: Continuation of edges in our theoretical model of hallucinations (SFT stage) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Incorrect annotation or reward	Misleading alignment	causal	Systematically skewed annotations which favor a specific response type may lead to an underlying model of misleading alignment by showing the LLM to align to aspects such as fluency or politeness that may take over factuality
Incorrect annotation or reward	Reward hacking	causal	Systematically skewed rewards which favor a specific response type may lead to an underlying model of reward hacking by incentivizing the LLM to align to aspects such as fluency or politeness that may take over factuality
Amount of alignment data	Parameters (updates)	causal	The number of instances will be partly responsible for determining the optimality in the final configuration of the parameters in the parameter space
Creative data (human alignment)	Sycophancy	causal	Instances which favor some type of creative outputs, such as responses in the form of funny dialogues, may promote the sycophantic behavior of LLMs by rewarding the LLMs to generate outputs that may jeopardize factuality
Misleading alignment	Sycophancy	causal	The underlying misleading alignment in the data to favor human preferences or values over factuality can result in the phenomenon of sycophancy
Reward hacking	Sycophancy	causal	The underlying alignment in the skewed reward function to favor human preferences or values over factuality can result in the phenomenon of sycophancy

Table 7: Edges in our theoretical model of hallucinations (alignment to human preferences stage) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Sycophancy	Parameters (updates)	embodiment	The behavior of sycophancy will be embodied in the parameters which will be activated with any input prompt that asks to generate an output that follows the human preferences learned from alignment training
Number of alignment epochs	Parameters (updates)	causal	The number of epochs is a direct guide to the parameter configuration in the parameter landscape
Number of alignment epochs	Mode collapse	causal	The more the number of epochs for training, the more certain it is the occurrence of the mode collapse phenomenon due to overfitting
Mode collapse	Parameters (updates)	embodiment	The behavior associated with the phenomenon of mode collapse will be embodied in the parameters which will be invoked when an input prompt refers to or contains any element or pattern overfitted during training
Parameters (updates)	Distribution not learned	mediation	Similar to the pretraining and SFT scenarios, the antecedent factors of the parameters may lead to not properly or not accurately recovering all the statistical cues as well as the linguistic and semantic patterns from the alignment data; this in turn, will lead to not properly learning and embodying the data distribution
Parameters (updates)	Hallucination	mediation	The embodiment of all the antecedents of the parameters can indirectly and effectively lead LLMs to hallucinate when an input prompt meets the conditions for triggering one or several of them
Distribution not learned	Hallucination	causal	Incorrectly or poorly learned statistical, linguistic, and semantic patterns can lead to hallucinations when the concepts or tasks involved in such patterns are elicited via an input prompt

Table 8: Continuation of edges in our theoretical model of hallucinations (alignment to human preferences stage) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Transformer model	Positional bias	causal	Features from the Transformer model such as positional encoding, attention model, mask type, computation of K, V matrices, or use of softmax for normalization may impose a certain type of positional bias
Transformer model	Long-range dependencies	causal	Factors such as attention dilution, weakened gradient signals, softmax attention decay, context fragmentation, or positional encoding can contribute to the occurrence of long-range dependencies
Positional bias	Recency bias	instantiation	Recency bias is a type of positional bias where most recent tokens or tokens near the end of an input sequence receive more attention
Positional bias	Short-range bias	instantiation	Short-range is a type of positional bias where tokens in a localized cluster inside an input sequence receive bigger attention weights due to their proximal position
Positional bias	Attention sink	instantiation	Attention sink is a type of positional bias where tokens at the beginning of an input sequence receive bigger attention weights
Positional bias	Lost-in-the-middle	instantiation	Lost-in-the-middle is a type of positional bias where tokens in the middle of an input sequence receive a significantly smaller attention weight than those at the beginning or end
Recency bias	Lost-in-the-middle	causal	Recency bias is a plausible cause of the phenomenon of lost-in-the-middle by directing the attention towards tokens at the end of an input sequence
Attention sink	Lost-in-the-middle	causal	Attention sink is a plausible cause of the phenomenon of lost-in-the-middle by directing the attention towards tokens at the beginning of an input sequence
Recency bias	Hallucination	causal	Recency bias may cause hallucinations if the key information to be attended to in order to generate a factual or faithful answer is placed at the begging of an input sequence
Short-range bias	Hallucination	causal	Short-range bias may cause hallucinations if the pieces of information to be attended to, in order to generate a factual or faithful answer, are placed distantly enough to be in different localized clusters of tokens

Table 9: Edges in our theoretical model of hallucinations (due to the architecture of LLMs) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Domain of pretraining data	Long-range dependencies	causal	Documents from domains such as web content may lead to the long-range dependencies problem since they do not systematically contain semantically-valid long-range dependencies due to their simplified or inconsistent writing style or due to being spuriously concatenated to yield larger documents
Attention sink	Hallucination	causal	Attention sink may cause hallucinations if the key information to be attended to, in order to generate a factual or faithful answer, is placed towards the end of the input sequence
Lost-in-the-middle	Hallucination	causal	Lost-in-the-middle may cause hallucinations if the key information to be attended to in order to generate a factual or faithful answer is placed at the middle of the input sequence
Transformer model	Insufficient context attention	causal	Features from the Transformer model such as positional encoding, attention model, mask type, or computation of K, V matrices can impose a restriction to pay more attention to the parametric memory than to the input context
Insufficient context attention	Hallucination	causal	When the LLM pays more attention to its parametric knowledge than to the input context it may induce unfaithful hallucinations if this knowledge contradicts or deviates from the target information of the input context to be extracted as an answer
Reasoning capabilities	Hallucination	causal	Limitations in terms of reasoning ability directly impact on hallucinations as the complexity of the input query increases
Imperfect encoding	Hallucination	causal	Due to intrinsic limitations in the mechanism for building the embedding representations of entities and concepts, the encoding may not derive in a perfect <i>memory</i> of an entity or a concept which may cause hallucinations at inference time when the LLM is asked about an imperfectly encoded attribute
LLM size	Hallucination	causal	The number of learnable parameters in an LLM has an impact on the generation of hallucinations due to the amount of factual information that can be stored and the capacity for faithfully capturing the data distribution
Vocabulary size	Hallucination	causal	The level of granularity of the vocabulary may impact on the occurrence of hallucinations generated; the finer-grained the vocabulary the less hallucinations

Table 10: Continuation of edges in our theoretical model of hallucinations (due to the architecture of LLMs) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Softmax bottleneck	Stochastic decoding	influence	Some of the candidate tokens with the highest likelihood or probability to be chosen as a prediction may not be a faithful choice and depending on the decoding method it is the likelihood for generating a hallucination
Stochastic decoding	Hallucination	mediation	Any stochastic decoding method may induce hallucinations, where sampling-induced randomness acts as a mediator due to selecting tokens within the high-probability set constrained by the softmax bottleneck
Prediction consistency	Hallucination	causal	In a sequence generation, the factuality of the first tokens may determine the factuality of the remaining sequence generated, and if the LLM is consistent with its first tokens predicted and these tokens are a hallucination then the remaining predictions will continue the hallucination
Short-range bias	Prediction consistency	influence	The prediction path set by the first predicted tokens may be influenced by or reinforced due to the effect of short-range bias
Over-confidence	Prediction consistency	influence	The prediction of tokens with overconfidence may influence the LLM to keep in the thread started from the first predicted tokens
Teacher forcing	Exposure bias	causal	The mechanism of <i>feeding</i> the LLM with a golden input sequence to predict the next token is precisely what gives rise to the exposure bias issue due to the difference between receiving such golden input with respect to receiving the previously generated sequence as input at inference time

Table 11: Edges in our theoretical model of hallucinations (arising at inference time) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Exposure bias	Over-confidence	influence	The phenomenon of exposure bias may influence the overconfidence of LLMs predictions by incrementing the confidence of the current prediction when given as input its own previous predictions matching an already known, familiar, or similar sequence
Exposure bias	Prediction consistency	influence	The fact that LLMs are trained to predict based on a gold input sequence but at inference time they predict based on their own generations may influence the prediction consistency by not being able to control and self-correct the current generation
Knowledge overshadow	Hallucination	causal	The presence of more than one entity in an input query can induce hallucinations if the query's most frequent entity in pretraining data is not the main entity in the query, thus becoming a dominant entity overshadowing the main entity due to a possible <i>misdirection</i> of the LLM focus
Needle-in-a-haystack	Hallucination	causal	The problem of finding specific key information within a long input context can cause a hallucination depending on the complexity of the problem. Varying factors such as context length or position of key information can impact on the likelihood of inducing the hallucination
Prompt features	Hallucination	causal	Several features such as the specificity and precision of the query and context, whether the instruction was unseen at training time, the input query length, among others, can trigger hallucinations. Notably, in-context examples can reduce the likelihood of hallucination by providing key cues to better ground the expected answer

Table 12: Continuation of edges in our theoretical model of hallucinations (arising at inference time) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Number of pretraining epochs	Amount of pretraining data	interaction	The strength of the effect of the amount of data on hallucinations depends on the number of epochs: While huge amounts of data are beneficial, to some extent, for LLMs to guard against hallucinations, it depends on the right number of pretraining cycles to ensure an optimal learning of the information and rich contextualization of embeddings
Amount of pretraining data	Multilingual data	interaction	Depending on the amount of data in languages with significant variations from English is the magnitude of the effect on altering English patterns and thus causing hallucinations
Number of pretraining epochs	Multilingual data	interaction	Depending on how many epochs the parameters are trained using data in languages with significant variations from English is the likelihood of the effect on altering English patterns and thus causing hallucinations
Loss function	Teacher forcing	interaction	Depending on the loss function used is the possibility of using teacher forcing for pretraining, i.e. with cross-entropy is possible to apply the teacher-forcing mechanism for pretraining
LLM size	Imitative falsehoods	interaction	The accuracy in modeling an imitative behavior of falsehoods from pretraining data may depend on the number of parameters of the LLM: Bigger LLMs may better learn to model such falsehoods than smaller LLMs
LLM size	Data quality	interaction	Depending on the size of the LLM is the effect of the quality of the data on the hallucinations: Optimal data quality with large or small LLMs will lead to an optimal scenario reducing hallucinations, but deficient data quality will have a bigger impact on hallucinations with large LLMs since they can better capture noise and biases than smaller LLMs
Number of fine-tuning epochs	Amount of SFT data	interaction	The appropriate fine-tuning to avoid hallucinations depends on the right number of epochs and right amount of data
Incorrect labels	Amount of SFT data	interaction	The more the number of instances with incorrect labels, the higher the change of inducing hallucinations due to more exposure to incorrect input-output mappings at training time

Table 13: Edges in our theoretical model of hallucinations (interactions between factors) and their corresponding relation type and propositional statement.

Factors		Relation		Proposition
Antecedent	Consequent			
New knowl- edge	Amount of SFT data	of	interaction	The more the number of instances containing new knowledge the higher the change of inducing hallucinations due to a bigger likelihood of destroying or altering learned patterns from pretraining or inducing catastrophic forgetting
Instruction types	Amount of SFT data	of	interaction	The more the number of different types of instances from different tasks or domains the higher the risk of inducing hallucinations due to requiring a bigger amount of gradient signal for each task type to accurately learn them
Creative data (SFT)	Amount of SFT data	of	interaction	The more the number of instances containing creative content that diverges from factuality, such as open-ended dialogues, the higher the chance of inducing hallucinations due to promoting diverging from factuality
Number of alignment epochs	Amount of alignment data	of	interaction	Dependent on the number of epochs and amount of alignment data is the correct modeling of the alignment to human preferences and thus the impact on generating hallucinations (we note this interaction further depends on the alignment method used such as DPO, RLHF, or other)
Creative data (human alignment)	Amount of alignment data	of	interaction	The more the number of instances containing creative content, the higher the chance of aligning to factors such as creativity or politeness at the risk of overlooking factuality in order to please humans
Incorrect annotation or reward	Amount of alignment data	of	interaction	The more the number of incorrectly annotated instances or incorrectly modeled rewards, the higher the chance of emerging misleading alignments or reward hacking, and thus the higher the risk of hallucinating
Recency bias	Attention sink		interaction	The synergistic interaction of these two factors can lead to the phenomenon of lost-in-the-middle
Positional bias	Domain data	of	interaction	Features from the pretraining documents such as writing style, among others, can further increase the effect of positional bias by, for example, pre-training with documents in which the first sentence always summarize each document, or where the most important key information is placed at the middle of the documents

Table 14: Continuation of edges in our theoretical model of hallucinations (interactions between factors) and their corresponding relation type and propositional statement.

Factors		Relation	Proposition
Antecedent	Consequent		
Long-range dependencies	Short-range bias	interaction	The localized effect from the short-range bias may aggravate the small attention weight between two far away but semantically related tokens happening due to their distance, which further increases the chance of hallucination
Vocabulary size	Amount of pretraining data	interaction	Depending on the amount of data is the quality of each token (embedding) in the vocabulary (a large vocabulary with a small corpus will lead to poorly contextualized representations) and thus it may influence the choice of the vocabulary size
Vocabulary size	Number of pretraining epochs	interaction	The larger the vocabulary size, the more the number of epochs that may be required to accurately encode information in the embeddings of entities and concepts
LLM size	Knowledge overshadow	interaction	The larger the LLM the more the tendency of dominant entities to overshadow infrequent entities in input prompts due to the diminishing ability of the LLMs to retain semantical knowledge of unpopular entities, leading to more hallucinations
Needle-in-a-haystack	Prompt features	interaction	Features from input prompts such as the length of the context, or the query type, can increase the difficulty of finding the key information required to answer the input query, leading to a bigger likelihood of hallucination
Needle-in-a-haystack	Positional bias	interaction	The difficulty of retrieving key information from the input context not only depends on its position in the context, but it also depends on the most prominent type of positional bias embodied by an LLM, which can further aggravate the difficulty of retrieving this information leading to a higher chance of hallucination
Needle-in-a-haystack	Long-range dependencies	interaction	If the key information to be retrieved encompasses two subsequences which are separated by a long distance inside the input context, then due to the long-range dependencies problem the likelihood of recovering the complete information decreases and as such the chance of hallucination increases

Table 15: Continuation of edges in our theoretical model of hallucinations (interactions between factors) and their corresponding relation type and propositional statement.

Author Index

Abubakar, Abdulhamid, 48
Achakulvisut, Titipat, 141
Adam, Faisal Muhammad, 48
Angamo, Abenezzer, 20
Anika, Israt Jahan, 77
Arsomngern, Pattaramanee, 141
Azime, Israel Abebe, 20

Bahmani, Soufia, 105
Bohra, Abhinav, 121
Bohra, Anuj, 121
Boonnag, Chiraphat, 141

Chuthong, Watcharakorn, 141

Dong, Bin, 181

Gebbru, Israel D., 68

Islam, Md. Adnanul, 128
Islam, Md. Rakibul, 77

Jana, Abhik, 61
Jiang, Shanshan, 181

Kalavantavanich, Tarm, 141
Kale, Sahil, 10
Khalid, Abubakar Auwal, 48
Klakow, Dietrich, 20

Manoshi, Farzana Zahid, 77
Marilign, Dagnachew Mekonnen, 20
Muhammad, Shamsuddeen Hassan, 48

Nawzad, Akam, 173

Patil, Prashant W, 37
Pipatsrisawat, Knot, 141
Ponarchar, Teerawut, 141
Putri, Divi Galih Prasetyo, 87

Rahman, Mizanur, 96
Rashid, Sadman Bin, 77

Saha, Kowshik, 96
Saha, Priyam, 61
Sahoo, Debiprasanna, 61
Sanchez Carmona, Vicente Ivan, 181
Sani, Babangida, 48
Sewunetie, Walelign Tewabe, 68
Sharma, Akshit, 37
Shuvo, Md. Asaduzzaman, 128
Slusallek, Philipp, 20

Tamiru, Hana Mekonen, 20
Tamiru, Rahel Mekonen, 68
Turad, Md Tanvirul Islam, 96

Uddin, Moyeen, 96

Vertanen, Keith, 105

Wonglertsakul, Jenta, 141

Yadav, Tanushree Ravindra Pratap, 1
Yimam, Seid Muhie, 20

Zhang, Yuming, 181
Zhou, Junyu, 181