Detecting LLM Hallucination Through Layer-wise Information Deficiency: Analysis of Ambiguous Prompts and Unanswerable Questions

Hazel Kim[†]**, Tom A. Lamb, Adel Bibi, Philip Torr**[‡]**, Yarin Gal**[‡]
University of Oxford
`{hazel.kim, yarin.gal}@cs.ox.ac.uk,`
`{thomas.lamb, adel.bibi, philip.torr}@eng.ox.ac.uk`

Abstract

Large language models (LLMs) frequently generate confident yet inaccurate responses, introducing significant risks for deployment in safety-critical domains. We present a novel, test-time approach to detecting model hallucination through systematic analysis of information flow across model layers. We target cases when LLMs process inputs with ambiguous or insufficient context. Our investigation reveals that hallucination manifests as usable information deficiencies in inter-layer transmissions. While existing approaches primarily focus on final-layer output analysis, we demonstrate that tracking cross-layer information dynamics ($\mathcal{L}$I) provides robust indicators of model reliability, accounting for both information gain and loss during computation. $\mathcal{L}$I integrates easily with pretrained LLMs without requiring additional training or architectural modifications.

1 Introduction

Large language models (LLMs) have achieved unprecedented success across diverse natural language tasks, particularly in complex reasoning ranging from commonsense to arithmetic knowledge (Achiam et al., 2023; Touvron et al., 2023; Abdin et al., 2024). However, these models face a critical challenge known as *hallucination*, a phenomenon where responses appear convincingly authoritative despite being inaccurate (Ji et al., 2023; Xu et al., 2024b; Liu et al., 2023). While numerous empirical studies have investigated potential sources of hallucination, recent theoretical work by Xu et al. (2024a) demonstrates the fundamental impossibility of eliminating this issue through any computable function.

Following Xu et al. (2024a), hallucination can be formally defined as the failure of LLMs to accurately reproduce the desired output of a computable function. This theoretical framework establishes that hallucination is an inherent characteristic of LLMs, persisting regardless of architectural choices, learning algorithms, prompting strategies, or training data composition. Building on this theoretical foundation, we hypothesize that hallucination emerges when LLMs lack sufficient information necessary for their computational functions to transmit messages across their internal processing systems effectively.

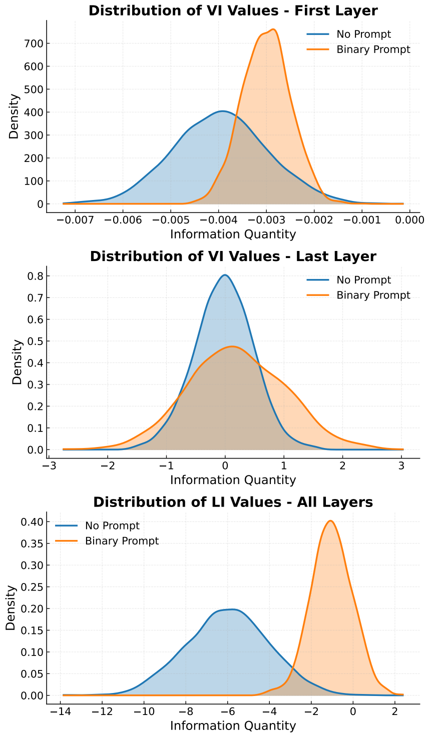


Figure 1: Distribution of $\mathcal{V}$-information ($\mathcal{V}$I) values in first and last layers, and $\mathcal{L}$-information ($\mathcal{L}$I) values (summation of $\mathcal{V}$I scores across all layers), as a function of prompt ambiguity. Results compare two prompt categories: (1) no instruction prompts and (2) binary instruction prompts ('Is this answerable?').

Prior research has focused primarily on analyzing final outputs to assess LLM confidence (Osband et al., 2023; Ahdritz et al., 2024; Lin et al., 2024) or identifying inherent data ambiguities that lead

[†] Correspondence to Hazel Kim <hazel.kim@cs.ox.ac.uk>.
[‡] Equal advising.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 32310–32322
November 4-9, 2025 ©2025 Association for Computational Linguistics

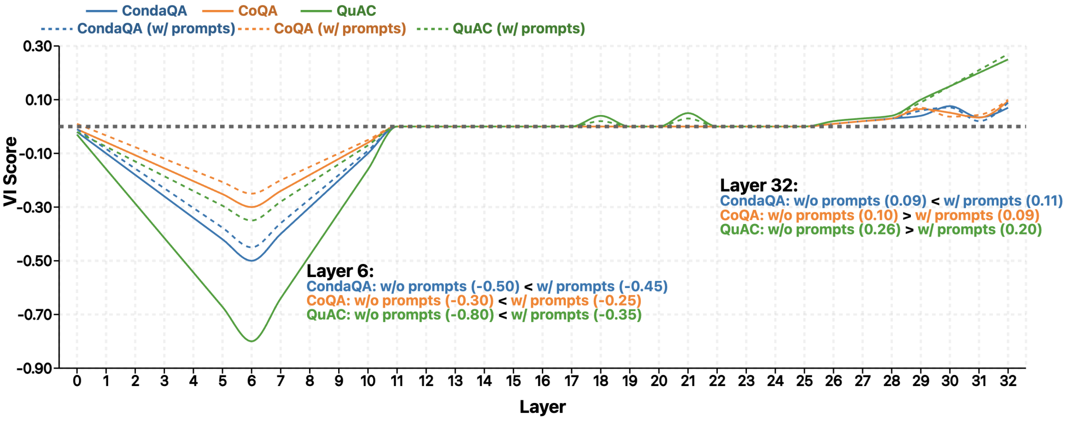


Figure 2: $\mathcal{V}$-usable information ($\mathcal{V}$I; Xu et al. (2020)) across layers for CondaQA, CoQA, and QuAC. Solid lines denote models without instruction prompts and dashed lines denote models with binary prompts ("Is this answerable?"). The quantity of $\mathcal{V}$I is not monotonically increased or decreased across depth. By the final layer (32), the relative effect of prompts becomes inconsistent across datasets, highlighting the value of analyzing all layers.

to predictive uncertainty (Cole et al., 2023; Kuhn et al., 2023a). However, the fundamental internal mechanisms of LLMs remain under-explored. Our analysis reveals that uncertainty estimation based solely on output layers or final computations overlooks critical insights into model *self-confidence*, thereby limiting our ability to detect hallucinatory behaviors.

To investigate LLM internal mechanisms, we build upon recent information theory frameworks proposed by Xu et al. (2020) and Ethayarajh et al. (2022). Their work introduces the concept of $\mathcal{V}$-usable information—the quantity of information a model family $\mathcal{V}$ can utilize to predict Y given X. This metric indicates prediction difficulty: lower $\mathcal{V}$-usable information corresponds to more challenging predictions for $\mathcal{V}$.

While these findings are significant, we empirically demonstrate that $\mathcal{V}$-usable information provides sub-optimal insight into model self-confidence because it only applies to the final layer. We propose layer-wise usable information ($\mathcal{L}$I), which quantifies the information changes within certain layers and aggregates those information dynamics across all model layers. As shown in Fig. 1, $\mathcal{L}$I provides more reliable indicators of LLM performance than final layer $\mathcal{V}$-usable information ($\mathcal{V}$I), particularly in detecting subtle variations in instruction prompt effectiveness.

Prior work on $\mathcal{V}$-usable information established that computation can *create* usable information during feature extraction, constituting a violation of the data processing inequality (DPI) in information theory (LeCun et al., 2015; Xu et al., 2020). Our research extends this understanding by demonstrating that LLMs both *create* and *lose* usable information during layer-wise updates. As illustrated in Fig.2, information flow is non-monotonic across layers, highlighting the limitation of analyzing only the final layer's computation.

To evaluate our framework, we focus on scenarios where LLMs must respond to queries with instruction prompts of different ambiguities (i.e., *ambiguous prompts*) and with constrained contextual information (i.e., *unanswerable* questions), settings particularly prone to hallucination. Our case study in Table 1 demonstrates that $\mathcal{L}$I strongly correlates with the difficulty of the question, influenced by the answerability and the clarity of the prompt, unlike $\mathcal{V}$I that shows no significant correlation with the model prediction.

Contributions Our primary contributions are:

- We propose $\mathcal{L}$I as a superior detector of unanswerable questions compared to existing baselines (Section 4.3), without requiring architectural modifications or additional training.
- We demonstrate that $\mathcal{L}$I effectively captures model confidence across varying levels of task difficulty induced by different instruction prompts (Section 4.2).
- We interpret that comprehensive layer tracking provides better insights into model internal confidence than single-layer analysis, using either initial or final layer (Sections 3 and 4.4).

2 Related Work

Contextual vs Factual Hallucinations Large language models (LLMs) often generate inaccurate outputs despite having access to correct information in their input context. This phenomenon is

Context

Once there was a beautiful fish named Asta. Asta lived in the ocean. There were lots of other fish in the ocean where Asta lived. They played all day long. One day, a bottle floated by over the heads of Asta and his friends. They looked up and saw the bottle. ... They took the note to Asta's papa. "What does it say?" they asked. Asta's papa read the note. He told Asta and Sharkie, "This note is from a little girl. She wants to be your friend. If you want to be her friend, we can write a note to her. **But you have to find another bottle so we can send it to her." And that is what they did.**

Instruction Prompt	Question	Ground-Truth Label	Prediction	*VI*	*LI*
Binary: "Is this answerable?"	What was the name of the fish?	Yes.	✓	0.3	0.2
	Were they excited?	No.	✓	0.3	-0.4
Open-ended Prompt: "Answer the question or say dont́ know"	What was the name of the fish?	Asta.	✓	0.3	-0.7
	Were they excited?	Don't know.	✗	-0.3	-1.4
No prompts	What was the name of the fish?	Asta.	✓	1.2	-6.8
	Were they excited?	Unknown.	✗	0.2	-7.7

Table 1: The $\mathcal{LI}$ scores provide a more comprehensive overview of prompt ambiguity compared to $\mathcal{VI}$. ***Prediction***: prediction generated by a language model — ✓: correct, ✗: incorrect. $\mathcal{VI}$: $\mathcal{V}$-usable information only applied to the final layer (Xu et al., 2020). $\mathcal{LI}$: Layer-wise usable information accumulated across layers (our proposed method).

known as contextual hallucination (Chuang et al., 2024). This issue is particularly concerning in high-stakes domains such as medicine and law, where acknowledging information gaps is preferable to making unfounded assumptions.

Most prior studies, however, focus on fact-based hallucination arising from parametric knowledge without input context. These hallucinations may result from inherent learning limitations or training data deficiencies, making their root causes difficult to isolate. Existing approaches have detected and mitigated such errors by using substantial annotated data to analyze various model components, including hidden states (Burns et al., 2022; Azaria and Mitchell, 2023), MLP and attention block outputs (Zhang et al., 2024; Simhi et al., 2024), and attention head outputs (Li et al., 2023; Simhi et al., 2024; Chen et al., 2024).

In contrast, contextual hallucination remains comparatively understudied. Existing work in this area has so far relied on annotated datapoints (Chuang et al., 2024) without probing the model internal mechanisms. Our research addresses this gap by examining contextual hallucination as an ideal setting to explore how LLMs behave when faced with insufficient information. We deliberately place models in situations where they must respond despite clearly inadequate input information, enabling us to study the fundamental nature of LLM hallucination behavior.

Unanswerable Questions Existing work has analyzed the model's capability to detect unanswerable questions from three main perspectives. One is self-evaluation which allows language models to generate probabilistic scores of how much the models believe their answers are trustworthy (Kadavath et al., 2022a; Yin et al., 2023) as we get access to advanced, well-calibrated models that can generate reliable results. The second perspective involves identifying the subspace of the model that is specifically responsible for answerability (Slobodkin et al., 2023). The third approach uses label information to train LLMs on whether questions are answerable, employing methods such as instruction-tuning or calibration (Jiang et al., 2021; Kapoor et al., 2024). While these studies suggest that LLMs can learn to express their confidence in responses when provided with additional information, they rely heavily on external calibration or fine-tuning. The outcome depends on the quality of the additional information or that of the annotation work. Unlike prior work, our investigation does not require label annotations to fine-tune classifiers or calibration tools to detect model confidence in their generated answers or ambiguous, unanswerable questions. We aim to obtain a computationally feasible method that applies to universal large language models.

Model Usable Information Our analysis builds upon the information-theoretic framework introduced by Xu et al. (2020) and expanded by Ethayarajh et al. (2022), focusing on quantifying the "usable information" accessible to models. Given a model family $\mathcal{V}$ that maps inputs X to outputs Y, the concept of $\mathcal{V}$-usable information measures how effectively $\mathcal{V}$ can leverage input data to predict outputs. Lower usable information correlates with increased prediction difficulty. For instance, encrypted or linguistically complex inputs reduce $\mathcal{V}$-usable information, increasing predictive challenges within the same model family.

This approach extends traditional information theory, particularly Shannon's mutual information (Shannon, 1948) and the data processing inequality (DPI; Pippenger, 1988). Mutual information quantifies the theoretical information shared between inputs and outputs, while DPI states that this quantity cannot increase as data passes through further transformations. Although these measures describe theoretical information flow, they do not capture how much of that information is practically usable by a given model family. In practice, usable information can diverge from the classical measures for two reasons: (1) computational constraints limit the extent to which models can realize the ideal mapping from X to Y (Xu et al., 2020); and (2) deep representation learning not only restructures inputs across layers to extract features from incomplete or noisy data, but also leverages prior knowledge stored in pretrained weights (LeCun et al., 2015; Goldfeld and Greenewald, 2021). These considerations motivate the notion of $\mathcal{V}$-usable information, which explicitly quantifies the information that is practically available to a given model family. The two frameworks leveraging $\mathcal{V}$-usable information are:

Predictive $\mathcal{V}$-information quantifies aggregate informativeness or dataset difficulty given model family constraints, expressed as $I_\mathcal{V}(X \to Y)$ (Xu et al., 2020).

Pointwise $\mathcal{V}$-information evaluates the information usability of individual instances relative to a specific dataset distribution, denoted as $\text{PVI}(x \to y)$ (Ethayarajh et al., 2022).

Formally, predictive $\mathcal{V}$-information is defined as:

Definition 2.1 (Predictive $\mathcal{V}$-information, Xu et al. 2020)**.** *Given predictive conditional entropy $H_\mathcal{V}(Y|X)$:*

$$I_\mathcal{V}(X \to Y) = H_\mathcal{V}(Y|\varnothing) - H_\mathcal{V}(Y|X). \quad (1)$$

Traditionally, the model family $\mathcal{V}$ has been instantiated as supervised models such as BERT (Devlin et al., 2019), trained to minimize expected log-loss risk on labeled datasets $(x, y) \sim p_\mathcal{D}$. This yields the following definitions of conditional predictive entropy:

$$H_\mathcal{V}(Y|X) = \mathbb{E}_{(x,y)\sim p_\mathcal{D}}\left[-\log_2 p(y|x)\right],$$
$$H_\mathcal{V}(Y|\varnothing) = \mathbb{E}_{(x,y)\sim p_\mathcal{D}}\left[-\log_2 p(y|\varnothing)\right].$$

In contrast to these supervised approaches, our proposed $\mathcal{L}I$ does not require training. In the next section, we will explain our proposed methodology. Detailed background on $\mathcal{V}$-usable information is provided in Appendix A.

Algorithm 1 Computing layer-wise usable information ($\mathcal{L}I$) without fine-tuning

Input: dataset $\mathcal{D} = \{(c_i, q_i)\}_{i=1}^m$, pretrained model with layers $\mathcal{L}$
Output: $\mathcal{L}I(C \to Q)$

```
$\varnothing \leftarrow$ empty context (null string)
for each example $(c_i, q_i) \in \mathcal{D}$ do
    for each layer $\ell \in \mathcal{L}$ do
        Compute token log-probs $p_\ell(q_t \mid q_{<t}, \varnothing)$
        Compute token log-probs $p_\ell(q_t \mid q_{<t}, c_i)$
        $H_\ell^{(i)}(Q \mid \varnothing) \leftarrow \frac{1}{T_i}\sum_{t=1}^{T_i} -\log_2 p_\ell(q_t \mid q_{<t}, \varnothing)$
        $H_\ell^{(i)}(Q \mid C) \leftarrow \frac{1}{T_i}\sum_{t=1}^{T_i} -\log_2 p_\ell(q_t \mid q_{<t}, c_i)$
        $I_\ell^{(i)} \leftarrow H_\ell^{(i)}(Q \mid \varnothing) - H_\ell^{(i)}(Q \mid C)$
    end for
end for
$\mathcal{L}I(C \to Q) \leftarrow \frac{1}{m}\sum_{i=1}^m \sum_{\ell \in \mathcal{L}} I_\ell^{(i)}$
```

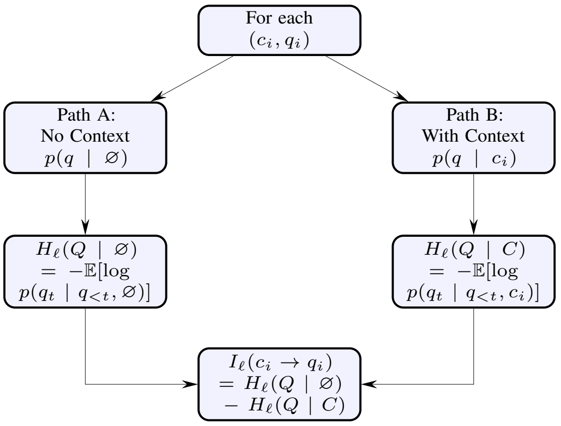

For each (c_i, q_i)
Path A: No Context $p(q \mid \varnothing)$
Path B: With Context $p(q \mid c_i)$
$H_\ell(Q \mid \varnothing) = -\mathbb{E}[\log p(q_t \mid q_{<t}, \varnothing)]$
$H_\ell(Q \mid C) = -\mathbb{E}[\log p(q_t \mid q_{<t}, c_i)]$
$I_\ell(c_i \to q_i) = H_\ell(Q \mid \varnothing) - H_\ell(Q \mid C)$

Figure 3: Illustration of computing layer-wise usable information for an example (c_i, q_i) at a *single layer* ℓ.

3 Layer-Wise Usable Information

We extend the $\mathcal{V}$-usable information framework to quantify information at the layer level in generative language models. *Layer-wise usable information* ($\mathcal{L}$I) measures how much a context C changes the predictive entropy of a question Q at each layer ℓ. The per-layer contribution is I_ℓ, and the total $\mathcal{L}I = \sum_{\ell \in \mathcal{L}} I_\ell$ aggregates these differences across all layers.

Concretely, given a context C, the model generates a free-form answer to a question Q. Let $\mathcal{L}$ denote the set of layers in a pre-trained language model. Each layer $\ell \in \mathcal{L}$ produces hidden representations. Projected through the pretrained language model head, the hidden states at layer ℓ induce a conditional distribution

$$f^{(\ell)} : \mathcal{C} \cup \{\varnothing\} \to P(\mathcal{Q}),$$

where $\mathcal{Q}$ is the token vocabulary. For a given question prefix $q_{<t}$ and context $c \in \mathcal{C}$ (or $\varnothing$), this distribution specifies probabilities $p_\ell(q_t \mid q_{<t}, c)$ over the next token $q_t \in \mathcal{Q}$.

This formulation allows us to measure usable information both at the level of individual layers, I_ℓ, and in aggregate across the model, $\mathcal{L}I$. Unlike prior work, we do not fine-tune $f^{(\ell)}$ on labeled data, but instead directly use the pretrained model outputs.

Definition 3.1 (Predictive conditional ℓ-entropy). Let q_t denote the t-th token in a question sequence $q \in Q$. The predictive conditional entropy at layer ℓ is

$$H_\ell(Q|C) = \mathbb{E}_{q \sim Q}\Big[- \log_2 p_\ell(q_t \mid q_{<t}, C)\Big], \quad (2)$$

and, similarly for the null context,

$$H_\ell(Q|\varnothing) = \mathbb{E}_{q \sim Q}\Big[- \log_2 p_\ell(q_t \mid q_{<t}, \varnothing)\Big].$$

These quantities represent the predictive uncertainty of the distributions derived from layer ℓ, either conditioned on the context C or without it. In practice, we report per-token entropies by averaging these values across all positions t in the question sequence.

Definition 3.2 (Predictive $\mathcal{L}$-information). The layer-wise usable information from C to Q is defined as

$$\begin{aligned} \mathcal{L}I(c \to q) &= \sum_{\ell \in \mathcal{L}} I_\ell(c \to q), \\ I_\ell(c \to q) &= H_\ell(Q|\varnothing) - H_\ell(Q|C). \end{aligned} \quad (3)$$

Here, $I_\ell(c \to q)$ measures the change in entropy due to the presence of context at layer ℓ, and $\mathcal{L}I(c \to q)$ aggregates these contributions across all layers. Algorithm 1 and Figure 3 illustrate this computation.

3.1 Implications

Using layer-wise usable information ($\mathcal{L}$I), we contribute to the following accomplishments:

- Detection for unanswerable questions by computing $\mathcal{L}I(c \to q)$ in datasets $\{c \in C, q \in Q\}$ for the same $\mathcal{L}$: we classify questions that lack sufficient usable information as unanswerable, likely to be inaccurate responses (Fig. 5 and 6).

- Evaluation on different prompts with Q for $\mathcal{L}$ by estimating $\mathcal{L}I(C \to Q')$. We quantify how different instruction prompt Q' influences usable information (Figs. 4 and Table 2).

- Analysis of importance of all layer information estimating $\mathcal{L}I(C \to Q)$. Aggregating across layers shows that full $\mathcal{L}I$ provides stronger separation between answerable and unanswerable questions than any single layer alone (Figs. 4 and 7 and Table 4).

4 Experiments

We demonstrate that layer-wise usable information ($\mathcal{L}$I) is an effective way to capture the ambiguity of prompts and detect unanswerable questions for large language models.

4.1 Experimental Setup

Evaluation Metric. We classify unanswerable questions based on uncertainty scores. The ground-truth labels of the unanswerability based on the contextual information are provided by the original benchmark datasets (Reddy et al., 2019; Choi et al., 2018; Ravichander et al., 2022). We evaluate uncertainty under the assumption that we assess whether to trust a model's generated response in a given context, i.e., deciding whether to accept an answer to a question. Our primary metric for this assessment is the area under the receiver operating characteristic curve (AUROC). The AUROC measures the *discrimination* ability of a scoring function—how well it separates correct from incorrect predictions. In our setting, this corresponds to distinguishing answerable from unanswerable questions using the uncertainty score provided by $\mathcal{L}I$. Higher AUROC scores indicate better performance, with a perfect score of 1 representing optimal uncertainty estimation, while a score of 0.5 represents random uncertainty.

We choose to use the AUROC as it suits well for evaluating uncertainty in free-form text responses, as opposed to calibration measures like the Brier score, frequently used in classification tasks or multiple-choice question answering. The Brier score requires calculating the total probability mass assigned to all possible tokens of a correct answer sequences. This causes the task to be intractable in free-form text settings where probabilities with respect to meanings are unavailable. Therefore we use the AUROC to capture the uncertainty associated with the model's outputs more accurately, and classify the unanswerable questions. One exception is model answers that we simply match the lexical words by instructing models to generate "Yes" or "No" to additional question prompts

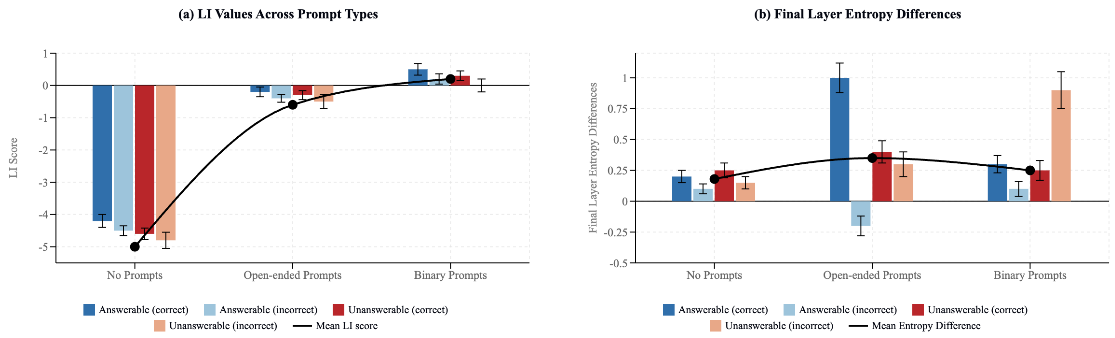


Figure 4: Impact of instruction prompts on layer information in QuAC. (a) $\mathcal{L}$I: scores increase systematically as prompts become more explicit (no prompt → open-ended → binary). Within each prompt type, correct answers have higher scores than incorrect ones (answerable-correct > answerable-incorrect; unanswerable-correct > unanswerable-incorrect). (b) Final-layer $\mathcal{V}$I: scores show no consistent progression and correct–incorrect separation.

asking if they can answer the question based on the context.

Baselines. We evaluate our method against several benchmarks, including model-generated answers, P(True) (Kadavath et al., 2022b), predictive token entropy and normalized entropy (Malinin and Gales, 2020), semantic entropy (Farquhar et al., 2024), and pointwise $\mathcal{V}$-information (PVI) (Ethayarajh et al., 2022) on the first and the last layers respectively. Model-generated answers are raw responses by models. P(True) measures the probability that a LLMs predict the next token as 'True' when provided with few-shot prompts that compare a primary answer to various alternative answers. Predictive entropy is calculated by conditional entropy over the output distribution. Predictive normalized entropy is obtained by dividing the total sequence-level entropy, computed as the negative log-likelihood, by the sequence length. We use a single model to meausre the normalized entropy, following the setups by (Kuhn et al., 2023b). Semantic entropy follows the confabulation mechanism to classify unanswerable questions. PVI is to measure difficult datapoints. We assume that difficult instances for language models are likely to be unanswerable questions.

Models. We use Llama3 (Dubey et al., 2024) and Phi3 models (Abdin et al., 2024). We vary the size of the models between 3.8B, 8B, and 14B parameters. We report our headline results using the most computationally efficient model, with 3.8B parameters unless we notify otherwise. In all cases we use only a single unmodified model since recent foundation models are not practical to modify the architectures and are often too costly to fine-tune them on datasets. Above all, we are interested in investigating internal language model behaviors than simply achieving optimal performance results. Hence we use them in their pre-trained form.

Prompt	Ans.	Unans.	△ (Ans.–Unans.)
Binary ("Is this question answerable?")	0.322	0.321	0.001
Always answer YES.	0.329	0.295	0.033
Always answer NO.	0.316	0.287	0.029
Is this question interesting?	0.031	-0.008	0.039
Did your family like cappuccino?	0.180	0.155	0.025
Can you give the wrong answer?	0.134	0.089	0.044
Can you give the correct answer?	0.180	0.117	0.064
Do you like your answer?	0.161	-0.023	0.184

Table 2: $\mathcal{L}$I scores on QuAC (100 examples averaged). Task-relevant binary prompts yield the highest scores on question examples, while irrelevant prompts reduce them. Larger deltas (Δ) indicate stronger separation of (un)answerability, which remains detectable even under irrelevant prompts.

Datasets. We use Conversational Question Answering Challenge dataset (CoQA) (Reddy et al., 2019) and Question Answering In Context (QuAC) (Choi et al., 2018) as question-answering tasks, where the model responds to questions using information from a supporting context paragraph. Our experiments are conducted on the development set, which contains approximately 8,000 questions. We also use CondaQA (Ravichander et al., 2022) which features 14,182 question-answer pairs with over 200 unique negation cues in addition to CoQA and QuAC to evaluate how trustworthy the $\mathcal{L}$I is to detect unanswerable questions. Given that goal, we employ a 1-to-1 ratio of answerable to unanswerable questions for a clear performance evaluation.

4.2 Do $\mathcal{L}$I scores indicate the ambiguity of prompts?

Figure 4 compares how instruction prompts influence $\mathcal{L}$I and final-layer $\mathcal{V}$I. Without prompts, $\mathcal{L}$I scores remain strongly negative (-4 to -5), indicating high uncertainty. As prompts become more explicit, scores increase systematically: open-ended prompts yield intermediate values (-0.5 to 0), while binary prompts produce the highest (slightly positive) scores. This progression shows that $\mathcal{L}$I is sensitive to the specificity of instructions, reliably reflecting prompt ambiguity. In contrast, final-layer $\mathcal{V}$I (Figure 4b) shows no consistent progression across prompt types. Because $\mathcal{V}$I was originally defined only for the final layer, such analyses obscure the consistent benefits of explicit prompts that are visible in intermediate representations.

Table 2 further illustrates how prompt relevance affects $\mathcal{L}$I. With binary prompts, scores are around 0.32, higher than most random prompts. Answer-forcing prompts such as *"Always answer YES/NO"* also yield relatively high $\mathcal{L}$I values, since these responses are easy for the model to generate. However, their scores remain below those of task-aligned prompts, reflecting that outputs are produced mechanically rather than through correct reasoning. Irrelevant prompts (*"Is this question interesting?"*, *"Did your family like cappuccino?"*) push scores substantially lower, showing how off-task instructions increase ambiguity. Misdirecting prompts (*"Can you give the wrong answer?"*) reduce scores even further, consistent with the uncertainty introduced by conflicting instructions. By contrast, task-aligned prompts (*"Can you give the correct answer?"*) partially restore $\mathcal{L}$I, while meta-reflective prompts (*"Do you like your answer?"*) cause the strongest shifts, showing that self-assessment language accentuates the effect of prompt relevance.

Overall, these patterns demonstrate that $\mathcal{L}$I is sensitive not only to the presence of an instruction but also to its relevance and quality. Irrelevant or adversarial prompts depress $\mathcal{L}$I, while task-relevant or self-reflective prompts elevate it, confirming that $\mathcal{L}$I provides a robust signal of prompt ambiguity.

4.3 Do $\mathcal{L}$I scores capture unanswerable questions?

Beyond prompt ambiguity, $\mathcal{L}$I also serves as a reliable signal for unanswerable questions. Across CoQA, QuAC, and CondaQA, $\mathcal{L}$I consistently outperforms baseline methods in distinguishing answerable from unanswerable questions (Figure 5). The advantage holds across different models and parameter sizes (Figure 6), whereas semantic entropy (SE) performs poorly on this task despite strong results elsewhere. This highlights that $\mathcal{L}$I is especially suited for (un)answerability detection, while other existing methods are not. Rejection analysis strengthens this conclusion. Filtering out predictions with the lowest $\mathcal{L}$I values steadily improves AUROC (Figure 5), confirming that low $\mathcal{L}$I reliably flags unanswerable cases.

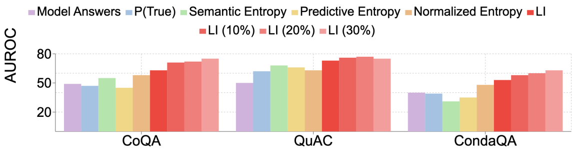
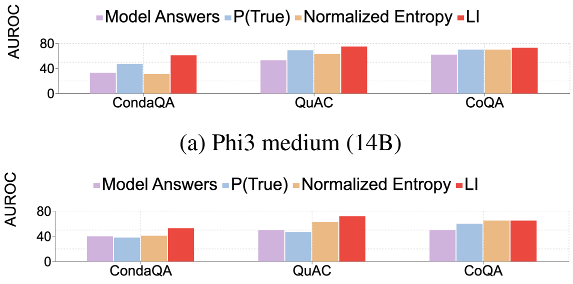


Figure 5: Performance of (un)answerability detection across datasets, comparing $\mathcal{L}$I with other baselines. $\mathcal{L}$I(10%), $\mathcal{L}$I(20%), and $\mathcal{L}$I(30%) shows the rejection rate based on low scores.



(a) Phi3 medium (14B)

(b) Llama3 (8B)

Figure 6: Performance of (un)answerability detection, compared to selected competitive baselines.

	Answerable	Unanswerable
Correct	0.4668 (e.g., Yes, there were clues...)	0.5088 (e.g., I cannot provide ...)
Incorrect	0.0189 (e.g., In 1983,...)	-0.0578 (e.g., Yes, she had...)
Unsure	0.0932 (e.g., I cannot provide...)	- (LLMs are correct in this case.)

Table 3: $\mathcal{L}$I scores for answerable and unanswerable questions with an *instruction prompt: Are you certain about the answer?* on QuAC (100 examples averaged). Unsure indicates that models express the uncertainty.

This trend extends more broadly across different prompt types in Table 2. With task-relevant binary prompts, unanswerable questions tend to have slightly lower scores, indicating sensitivity to unanswerability. This pattern persists under random, irrelevant, or misleading prompts. Though the absolute values vary, unanswerable questions remain associated with relatively small $\mathcal{L}$I scores. This consistency demonstrates that $\mathcal{L}$I distinguishes answerable from unanswerable questions not only under optimal instructions, but also when the prompting conditions are weak or noisy.

Another noteworthy role of $\mathcal{L}$I appears in Ta-

	SQuAD		NQ	
	NA	Binary	NA	Binary
$\mathcal{L}I$ (i.e., all layers)	74.78	76.42	55.93	57.26
Slobodkin et al. (2023)				
(1st layer)	26.14	31.01	17.04	21.12
(last layer)	51.78	59.42	55.23	56.26

Table 4: Fact-based hallucination detection results (classification accuracy, %). NA: no instruction prompt. Binary: binary instruction prompt "Is this answerable?".

ble 3, where the explicit certainty prompt (*"Are you certain about the answer?"*) elicits a different pattern from the binary prompt (*"Is this question answerable?"*). Correct unanswerable responses achieve the highest $\mathcal{L}$I scores (0.509), even higher than correct answerable ones (0.467), reflecting that the model is appropriately certain about its own uncertainty. Incorrect answers, by contrast, yield the lowest values (0.019 for answerable, –0.058 for unanswerable), as expected from clear mismatches. Answerable questions where the model expressed uncertainty ("unsure") obtain a modest score (0.093), higher than incorrect responses but far lower than correct ones. This gradient indicates that $\mathcal{L}$I distinguishes not only correctness but also the appropriateness of expressed uncertainty: it assigns high values to justified abstentions while assigning low values to hallucinations and unwarranted hesitation. These overall results show that $\mathcal{L}$I serves as a consistent and practical indicator of how models handle ambiguous or unanswerable questions, remaining robust across datasets, model families, and diverse prompt formulations.

4.4 Do we really need to consider all layers instead of the final layer?

A critical question is whether it is sufficient to probe a single layer or a subset of layers, or whether information must be accumulated across the entire model depth. Our experiments suggest that this choice is non-trivial. For example, we observed that intermediate layers such as layer 6 already exhibit strong separation between answerable and unanswerable questions, raising the possibility that probing one such layer—or even aggregating only the first k layers—might capture the relevant information without requiring the full $\mathcal{L}$I computation. However, Figure 7 shows that signals at individual layers are not stable: while some intermediate layers appear informative, others lose or distort information before it reaches the final layer. As a result, relying on any single layer or partial accumulation risks missing critical dynamics. By contrast, $\mathcal{L}$I accumulated across all layers consistently provides a clearer and more reliable distinction, demonstrating that usable information must be tracked throughout the full network depth.

To further test whether aggregating across all layers is necessary, we evaluate fact-based hallucination detection using the probing framework of Slobodkin et al. (2023). As shown in Table 4, $\mathcal{L}$I, which accumulates information across the full model depth, consistently outperforms single-layer methods. On SQuAD (Rajpurkar et al., 2018), it achieves 74.78% and 76.42% accuracy under no-prompt and binary-prompt settings, respectively, compared to 51.78% and 59.42% for last-layer probing and only 26.14% and 31.01% for first-layer probing. On NQ (Kwiatkowski et al., 2019), the advantage of all-layer accumulation remains, though the margins are narrower (55.93% and 57.26% vs. 55.23% and 56.26% for the last layer). Binary prompts improve performance across all methods, confirming the value of explicit guidance, but the gains are most pronounced for the all-layers $\mathcal{L}$I approach. This reinforces that aggregating usable information across all layers provides a more robust signal for hallucination detection than relying on any single layer.

4.5 Are $\mathcal{L}$I scores computationally inexpensive?

$\mathcal{L}$I scores are computationally inexpensive compared to other baseline methods. The approach requires two forward passes, one with context and one without. However, because the second pass involves only the short question sequence, the marginal cost is negligible (Table 5). According to dataset statistics, CoQA averages 271 context words and 5.5 question words (Reddy et al., 2019), QuAC averages 401 context tokens and 6.5 question tokens (Choi et al., 2018), and CondaQA averages 131 context tokens and 24.4 question tokens (Ravichander et al., 2022). As a result, the actual overhead is close to a single forward pass: 1.02× on CoQA, 1.01× on QuAC, and 1.16× on CondaQA.

In contrast, competing methods are far more computationally demanding. P(TRUE) incurs about 11× cost because each test query is paired with k demonstrations plus the target input, yielding $(k+1)\times$ forward passes (with $k = 10$, reduced from 20 in Kadavath et al. (2022b)). Semantic Entropy (SE) is even more expensive. It estimates

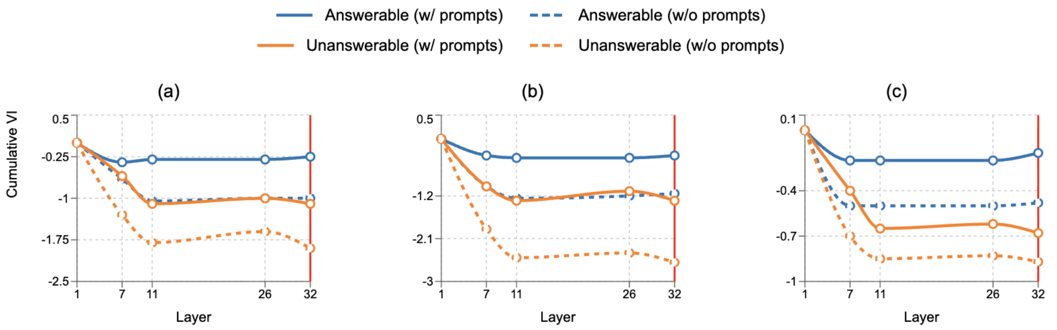


Figure 7: Cumulative $\mathcal{V}$I, accumulating from the beginning to the layer i. (a) CondaQA, (b) CoQA, (c) QuAC. The $\mathcal{L}$I scores accumulated from the beginning to the last layers (red vertical line) show the apparent, accurate differences between answerable and unanswerable questions in both settings with and without instruction prompts.

Dataset	Context	Question	$\mathcal{LI}_{\text{context + question}}$	$\mathcal{LI}_{\text{question}}$	$\mathcal{LI}_{\text{total}}$ †	P(TRUE)	Semantic Entropy
CoQA	271 words	5.5 words	1.00	0.02	1.02×	11×	100×
QuAC	401 tokens	6.5 tokens	1.00	0.01	1.01×	11×	100×
CondaQA	131 tokens	24.4 tokens	1.00	0.16	1.16×	11×	100×

Table 5: Computational overhead of $\mathcal{L}$I compared to P(TRUE) and Semantic Entropy (SE). Unlike P(TRUE) (11× overhead) or SE (100× overhead), $\mathcal{L}$I requires a lightweight question-only pass, 1.01–1.18× overhead in practice.

uncertainty by generating 50 samples for each input, each conditioned on a 20-shot prompt (Farquhar et al., 2024), which results in roughly 100× overhead.

4.6 Can calibration metrics such as ECE apply to $\mathcal{L}$I?

While AUROC is our primary evaluation metric, one may ask whether calibration metrics such as Expected Calibration Error (ECE) are also applicable and provide meaningful insights in this setting. Since $\mathcal{L}$I produces scalar confidence values, they fit a logistic regression model to map them into $[0, 1]$, following standard practice for ECE. Calibration is performed on a small held-out subset of the training data, separate from the evaluation set. We examine this on QuAC dataset with binary classification in two settings: question (un)answerability (Table 6) and instruction-prompt ambiguity (Table 7).

The results show consistent trends with AUROC. First, $\mathcal{L}$I achieves lower ECE than verbalized baselines across all conditions, indicating that its scores are inherently better calibrated. One exception is under the binary instruction prompt with 10 calibration examples (LI 0.343 vs. Verbalized 0.312) to capture uanswerability, but it goes back to the consistent when it is trained with 100 examples. As expected, LI shows stronger with binary than without prompt. In ambiguity detection, $\mathcal{L}$I maintains an advantage, reaching an ECE as low as 0.039. Although AUROC remains the primary evaluation metric given the advantage of $\mathcal{L}$I as parameter-free answerability signal, the ECE results highlight its reliable calibration across tasks and prompt types.

Prompt	#Trainset	Method	ECE ↓
No instruction Prompt	10	$\mathcal{L}$I	0.365
		Verbalized	0.451
	100	$\mathcal{L}$I	0.187
		Verbalized	0.398
Binary (yes/no)	10	$\mathcal{L}$I	0.343
		Verbalized	0.312
	100	$\mathcal{L}$I	0.177
		Verbalized	0.276

Table 6: ECE for question (un)answerability.

#Trainset	Method	ECE ↓
10	$\mathcal{L}$I	0.052
	Verbalized	0.062
100	$\mathcal{L}$I	0.039
	Verbalized	0.054

Table 7: ECE for instruction-prompt ambiguity (binary vs. no instruction prompt).

5 Conclusion

We propose layer-wise usable information ($\mathcal{L}$I) to detect ambiguous or unanswerable questions. Because prior methods exclusively rely on final layers or output spaces to estimate model confidence, we argue that tracking usable information all across the layers is critical to comprehensively understand model behaviors.

Limitations

One limitation of our method may come from its unsupervised nature. When comparing our approach to supervised methods, it may be less optimal. While supervised techniques benefit from labeled data, enabling them to learn from specific examples, our approach targets to understand large pre-trained language models in in-context question-answering tasks. Depending on their use cases and specific purposes, some may prefer supervised methods despite the associated computational costs.

Acknowledgments

We thank Kawin Ethayarajh for his helpful feedback on $\mathcal{V}$-usable information theory. We thank the Oxford Applied and Theoretical Machine Learning (OATML) Group and Torr Vision Group (TVG) at the University of Oxford for their interesting discussions that have inspired this work. YG proposed the topic and provided the early-stage feedback. PT reviewed the work and provided later-stage feedback. HK led the work by proposing the methodology, designing the experiments, and writing the paper. TL reviewed the draft and helped clarify the theoretical aspects. AB contributed to the initial discussions of the work.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

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

Gustaf Ahdritz, Tian Qin, Nikhil Vyas, Boaz Barak, and Benjamin L. Edelman. 2024. Distinguishing the knowable from the unknowable with language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Amos Azaria and Tom Mitchell. 2023. The internal state of an llm knows when it's lying. *arXiv preprint arXiv:2304.13734*.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. 2022. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.

Zhongzhi Chen, Xingwu Sun, Xianfeng Jiao, Fengzong Lian, Zhanhui Kang, Di Wang, and Chengzhong Xu. 2024. Truth forest: Toward multi-scale truthfulness in large language models through intervention without tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20967–20974.

Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. 2018. Quac: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2174–2184. Association for Computational Linguistics.

Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. 2024. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA. Association for Computational Linguistics.

Jeremy R. Cole, Michael J. Q. Zhang, Daniel Gillick, Julian Eisenschlos, Bhuwan Dhingra, and Jacob Eisenstein. 2023. Selectively answering ambiguous questions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 530–543. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Ziv Goldfeld and Kristjan Greenewald. 2021. Sliced mutual information: A scalable measure of statistical dependence. *Advances in Neural Information Processing Systems*, 34:17567–17578.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

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

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022a. Language models (mostly) know what they know. *CoRR*, abs/2207.05221.

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

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 2024. Calibration-tuning: Teaching large language models to know what they don't know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14, St Julians, Malta. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023a. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023b. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2024. Generating with confidence: Uncertainty quantification for black-box large language models. *Trans. Mach. Learn. Res.*, 2024.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. *CoRR*, abs/2308.05374.

Andrey Malinin and Mark Gales. 2020. Uncertainty estimation in autoregressive structured prediction. *arXiv preprint arXiv:2002.07650*.

Ian Osband, Zheng Wen, Seyed Mohammad Asghari, Vikranth Dwaracherla, Morteza Ibrahimi, Xiuyuan Lu, and Benjamin Van Roy. 2023. Epistemic neural networks. *Advances in Neural Information Processing Systems*, 36:2795–2823.

Nicholas Pippenger. 1988. Reliable computation by formulas in the presence of noise. *IEEE Transactions on Information Theory*, 34(2):194–197.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for squad. *arXiv preprint arXiv:1806.03822*.

Abhilasha Ravichander, Matt Gardner, and Ana Marasović. 2022. CONDAQA: A contrastive reading comprehension dataset for reasoning about negation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 8729–8755. Association for Computational Linguistics.

Siva Reddy, Danqi Chen, and Christopher D. Manning. 2019. Coqa: A conversational question answering challenge. *Trans. Assoc. Comput. Linguistics*, 7:249–266.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. 2024. Constructing benchmarks and interventions for combating hallucinations in llms. *arXiv preprint arXiv:2404.09971*.

Aviv Slobodkin, Omer Goldman, Avi Caciularu, Ido Dagan, and Shauli Ravfogel. 2023. The curious case of hallucinatory (un)answerability: Finding truths in the hidden states of over-confident large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 3607–3625. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. 2020. A theory of usable information under computational constraints. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024a. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.

Ziwei Xu, Sanjay Jain, and Mohan S. Kankanhalli. 2024b. Hallucination is inevitable: An innate limitation of large language models. *CoRR*, abs/2401.11817.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8653–8665. Association for Computational Linguistics.

Shaolei Zhang, Tian Yu, and Yang Feng. 2024. Truthx: Alleviating hallucinations by editing large language models in truthful space. *arXiv preprint arXiv:2402.17811*.

A Background in Usable Information

In this section, we explain the information-theoretic foundations for measuring model-usable information, established by Xu et al. (2020) and Ethayarajh et al. (2022). Consider a model family $\mathcal{V}$ that maps text input X to output Y. The $\mathcal{V}$-usable information quantifies the amount of information a model family can extract to predict Y given X. This metric inversely correlates with prediction difficulty. The lower the $\mathcal{V}$-usable information, the harder the dataset is for $\mathcal{V}$. Consider text that is encrypted or translated into a language with more complex grammatical structures. Such transformations decrease $\mathcal{V}$-usable information, making the prediction of Y given X more challenging within the same model family $\mathcal{V}$.

This concept challenges traditional information theory principles, notably Shannon's mutual information (Shannon, 1948) and the data processing inequality (DPI) (Pippenger, 1988). Shannon's theory fails to account for scenarios where X contains less usable information than the mutual information $I(X;Y)$ due to encryption. Similarly, DPI cannot explain how model family $\mathcal{V}$ acquires additional information through computational constraints or advanced representation learning. Two key factors demonstrate this limitation; (1) computational constraints prevent input data from fully representing ideal world knowledge (Xu et al., 2020); (2) advanced language models can extract meaningful features from incomplete representations, achieving progressive information gains during computation (LeCun et al., 2015; Goldfeld and Greenewald, 2021).

Recent work has introduced two frameworks adopting $\mathcal{V}$-usable information to capture these phenomena. The first framework to capture the $\mathcal{V}$-usable information is called **predictive $\mathcal{V}$-information** (Xu et al., 2020). The predictive $\mathcal{V}$-information measures how much information can be extracted from X about Y when constrained to model family $\mathcal{V}$, written as $I_{\mathcal{V}}(X \to Y)$. The greater the $I_{\mathcal{V}}(X \to Y)$, the easier the dataset is for $\mathcal{V}$. While predictive $\mathcal{V}$-information provides an aggregate measure of informativeness of computational functions or dataset difficulty, **pointwise $\mathcal{V}$-information** (Ethayarajh et al., 2022) measures usable information in individual instances with respect to a given dataset distribution, written as PVI$(x \to y)$. The higher the PVI, the easier the instance is for $\mathcal{V}$, under the given distribution.

We first define predictive conditional $\mathcal{V}$-entropy to introduce the predictive $\mathcal{V}$-information. We follow the formal notations, defined in Xu et al. (2020):

Definition 2.1 (Xu et al., 2020) *Let predictive family $\mathcal{V} \subseteq \Omega = \{f : \mathcal{X} \cup \varnothing \to P(\mathcal{Y})\}$, where X and Y are random variables with sample space $\mathcal{X}$ and $\mathcal{Y}$, and $P(\mathcal{Y})$ be the set of all probability measures on $\mathcal{Y}$ over the Borel algebra on $\mathcal{X}$. The* ***predictive conditional $\mathcal{V}$-entropy*** *is defined as*

$$H_{\mathcal{V}}(Y|X) = \inf_{f \in \mathcal{V}} \mathbb{E}_{x,y \sim X,Y}[-\log_2 f[X](Y)]. \tag{4}$$

The conditional $\mathcal{V}$-entropy is given a random variable X as side information, so the function $f[X](Y)$ produces probability distributions over the output Y based on the side information X. Suppose $\varnothing$ denote a null input that provides no information about Y. Note that $\varnothing \notin X$. The predictive family $\mathcal{V}$ is a subset of all possible mappings from

X to $P(Y)$ that satisfies *optional ignorance*; whenever the P predicts the outcome of Y, it has the option to ignore side information, X. That is,

$$H_{\mathcal{V}}(Y|\varnothing) = \inf_{f\in\mathcal{V}}\mathbb{E}_{y\sim Y}[-\log_2 f[\varnothing](Y)].$$

This is identical to the classic $\mathcal{V}$-entropy, denoted as $H_{\mathcal{V}}(Y)$. We additionally specify the notation $\varnothing$ because the conditional $\mathcal{V}$-information given null input is crucial to measure how the existence of X affect $\mathcal{V}$ to obtain the relevant information. The entropy estimation specifies the infinite functions f are in $\mathcal{V}$ as Xu et al. (2020) illustrate that the predictive family in theory does not take into account the computational constraints.

Definition 2.2 (Xu et al., 2020) *Let X, Y denote random variables with sample space $\mathcal{X}\times\mathcal{Y}$, and $\mathcal{V}$ be a predictive model or function family. Then the* ***predictive $\mathcal{V}$-information*** *from X to Y is defined as*

$$I_{\mathcal{V}}(X\to Y) = H_{\mathcal{V}}(Y|\varnothing) - H_{\mathcal{V}}(Y|X). \quad (5)$$

Definition 2.3 (Ethayarajh et al., 2022) *Given random variables X, Y and a predictive family $\mathcal{V}$, the* ***pointwise $\mathcal{V}$-information (PVI)*** *of an instance (x, y) is*

$$I_{\mathcal{V}}(x\to y) = -\log_2 f'[\varnothing](y) + \log_2 f'[x](y). \quad (6)$$

Ethayarajh et al. (2022) have extended the Equation A to estimate the difficulty of point-wise instances for the predictive family $\mathcal{V}$. Most neural networks fine-tuned to fit label information Y meet the definition of the predictive family $\mathcal{V}$ here. If $\mathcal{V}$ were, for instance, the BERT function family, $f'[X]$ and $f'[\varnothing]$ would be the models after finetuning BERT with and without the input X respectively. Higher PVI means that the instance is easy for $\mathcal{V}$ while lower PVI means difficult among the given distribution. This comes from the intuition that predicting minority instances expects $\mathcal{V}$ to require more side information of X to understand the instances.