Uncertainty as Feature Gaps: Epistemic Uncertainty Quantification of LLMs in Contextual Question-Answering

Yavuz Bakman[1]* **Zhiqi Huang**[2] **Chenyang Zhu**[2]
Anoop Kumar[2] **Alfy Samuel**[2] **Daben Liu**[2]

[1]University of Southern California [2]Capital One
ybakman@usc.edu

Abstract

Uncertainty Quantification (UQ) research has primarily focused on closed-book factual question answering (QA), while contextual QA remains unexplored, despite its importance in real-world applications. In this work, we focus on UQ for the contextual QA task and propose a theoretically grounded approach to quantify *epistemic uncertainty*. We begin by introducing a task-agnostic, token-level uncertainty measure defined as the cross-entropy between the predictive distribution of the given model and the unknown true distribution. By decomposing this measure, we isolate the epistemic component and approximate the true distribution by a perfectly prompted, idealized model. We then derive an upper bound for epistemic uncertainty and show that it can be interpreted as semantic feature gaps in the given model's hidden representations relative to the ideal model. We further apply this generic framework to the contextual QA task and hypothesize that three features approximate this gap: *context-reliance* (using the provided context rather than parametric knowledge), *context comprehension* (extracting relevant information from context), and *honesty* (avoiding intentional lies). Using a top-down interpretability approach, we extract these features by using only a small number of labeled samples and ensemble them to form a robust uncertainty score. Experiments on multiple QA benchmarks in both in-distribution and out-of-distribution settings show that our method substantially outperforms state-of-the-art unsupervised (sampling-free and sampling-based) and supervised UQ methods, achieving up to a 13-point PRR improvement while incurring a negligible inference overhead.

1 Introduction

Despite their impressive performance across a wide range of real-world tasks, Large Language Models (LLMs) still suffer from hallucinations and incorrect generations, which limit their deployment in high-stakes domains such as medicine and finance [7, 31]. Uncertainty Quantification (UQ) has emerged as a key tool for detecting such errors by only using the model itself, such as output consistency, log-probabilities, or internal activations. Recent works [6, 38] have demonstrated that UQ methods exhibit strong empirical performance across a variety of evaluation benchmarks.

While these results are encouraging, most existing works [39, 22, 21, 12] design and evaluate their methods primarily on closed-book factual question answering (QA) tasks, which test the success of UQ methods on the model's memory abilities. Although this direction is important, another critical ability of LLMs that deserves more attention from a UQ perspective is their *contextual* capabilities.

*Corresponding author & Work done during the internship at Capital One

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

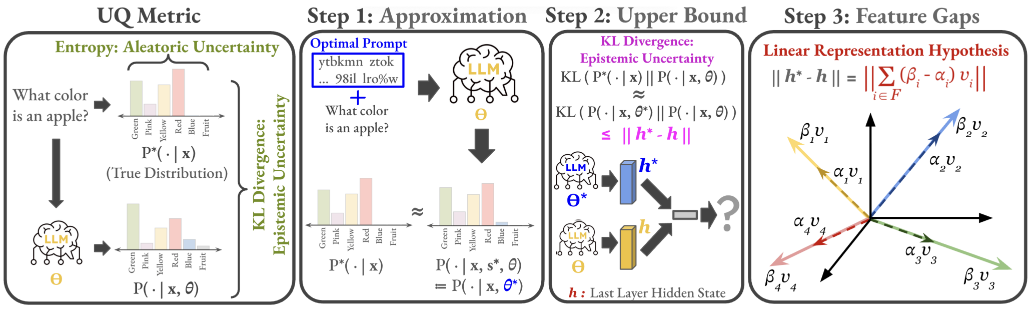


Figure 1: Derivation steps for epistemic uncertainty as feature gaps. Visualization of Section 3.

With the increasing popularity of Retrieval-Augmented Generation (RAG) in many LLM applications, detecting errors in model generations conditioned on retrieved context has become more important. However, relatively little effort has been devoted to this direction [34, 13, 29], where the proposed approaches often rely on heuristics rather than grounded theory.

Motivated by these observations, we focus on developing a theoretically grounded UQ method for contextual QA. In this setting, the relevant context is either already retrieved or directly provided by the user, and a context-relevant question is posed to the model. Our goal is to quantify the model's uncertainty for a given input as a means to assess whether its output is likely to be correct/reliable.

To quantify the uncertainty of an LLM, we first propose an uncertainty metric defined as the cross-entropy between the true predictive distribution and the given model's distribution, inspired by Schweighofer et al. [32]. Our approach introduces a key modification to their formulation by reversing the position of the true and given model distributions within the cross-entropy, and adapting it specifically for LLMs. We further decompose the total uncertainty into two components: *epistemic* and *aleatoric* uncertainty. In our problem setup, epistemic uncertainty, the model's lack of ability or knowledge to correctly and reliably answer a given question–context pair, is our main interest. After approximating the true predictive distribution with a perfectly prompted hypothetical ideal model, we show that epistemic uncertainty can be bounded by the distance between the last layer hidden state of the given model and the ideal hypothetical model. We further show that this distance can be expressed as the sum of distances over linearly independent model features. Importantly, this result generalizes to any LLM task and is visualized in Figure 1. To approximate this distance specifically in contextual question answering task, we hypothesize three desirable features that capture how far the given model is from the ideal model: 1) **Context reliancy**: the model should ground its answer in the provided context rather than relying on its parametric knowledge; 2) **Context comprehension**: the model should be able to extract and integrate relevant information from the context to answer the question accurately. 3) **Honesty**: the model should avoid intentionally outputting a wrong answer.

Following a top-down interpretability approach similar to Zou et al. [43], we extract the aforementioned high-level semantic features using a small set of labeled samples to identify the optimal layer for feature extraction. At test time, we combine the activation amount of three features to quantify epistemic uncertainty by computing only three dot products between the model's hidden state and the corresponding feature vectors, one per feature, without requiring any sampling. Our method is highly efficient and achieves substantial performance gains: it outperforms SOTA unsupervised, sampling-free, and sampling-based approaches by up to 16 PRR points. Furthermore, with the same amount of labeled data, it surpasses strong supervised baselines such as SAPLMA [4] and LookbackLens [9] by up to 13 PRR points, while exhibiting significantly better out-of-distribution generalization compared to SAPLMA, which is an important and desirable property for supervised UQ methods. The overview of our proposed method is visualized in Figure 1.

2 Preliminaries

2.1 Aleatoric and Epistemic Uncertainty

The total uncertainty of a model is typically decomposed into two components: *epistemic* and *aleatoric* uncertainty [17]. Epistemic uncertainty arises from a lack of knowledge. In the context of LLMs, when faced with a difficult question that the model does not know the answer to, its output distribution tends to be more uniform, which indicates uncertainty about which answer is correct. This uncertainty stems from the model's inability or lack of knowledge to provide the correct answer, and is therefore classified as *epistemic*. In contrast, *aleatoric* (or data) uncertainty captures variability inherent to the task or data, rather than limitations in the model's knowledge. For example, a model may be epistemically confident, knowing the answer, but still produce multiple valid responses due to ambiguity in the question or the presence of multiple equally correct phrasings. This variability arises from the nature of the query and the language itself, not from the model's lack of ability. In the next section, we discuss how existing works conceptualize UQ in LLMs, and how these concepts relate to epistemic and aleatoric uncertainty.

2.2 Uncertainty Quantification of LLMs

In the LLM literature, uncertainty quantification is typically used to identify incorrect or unreliable answers generated for a given query. Unlike the well-established frameworks in classification tasks [15], there is no widely accepted UQ framework for generative LLMs [6, 38]. With a few exceptions, most approaches rely on heuristic-based methods that estimate the correctness of a model's (greedy or sampled) generation. UQ methods only use the model itself to find such a score by using signals such as token probabilities [14], internal representations [8], or output consistencies [22]. Although rarely stated explicitly, the underlying objective in many of these works is to better quantify *epistemic* uncertainty, i.e., to produce an uncertainty score that quantifies the model's (lack of) certainty in the correctness of its own generation. The performance of UQ methods is typically evaluated using threshold-free metrics such as the Area Under the ROC Curve (AUROC) and the Prediction–Rejection Ratio (PRR) to measure which assess how well uncertainty scores distinguish between correct and incorrect outputs. A smaller number of studies [1, 2] take a more theoretically grounded approach, explicitly distinguishing between epistemic and aleatoric uncertainty. In this work, we also aim to separate epistemic and aleatoric uncertainty through our proposed UQ formulation. In the following section, we introduce our notation and describe the problem setup.

2.3 Problem Setup and Notation

We denote the context sequence as $\mathbf{c}$, and the question together with any relevant instructions as $\mathbf{x}$. The probability distribution over the token at position t produced by the model, conditioned on the context $\mathbf{c}$, query $\mathbf{x}$, and previously generated tokens, is given by: $P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \mathbf{c}, \theta)$, where $\mathbf{y}_{<t}$ denotes the sequence of tokens generated before timestep t, and θ represents the given model parameters. Our objective is to find an uncertainty quantification method $U(\mathbf{x}, \mathbf{c}, \mathbf{y}) \in \mathbb{R}$ that is *negatively correlated* with the correctness of the generated sequence $\mathbf{y}$. More formally, we aim to maximize $\mathbb{E}\left[\mathbb{1}_{U(\mathbf{x}_1,\mathbf{c}_1,\mathbf{y}_1)<U(\mathbf{x}_2,\mathbf{c}_2,\mathbf{y}_2)} \cdot \mathbb{1}_{\mathbf{y}_1 \in Y_1 \,\wedge\, \mathbf{y}_2 \notin Y_2}\right]$, where $(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2) \sim D_{\text{test}}$, with D_{test} denoting the evaluation dataset obtained by getting the most probable (greedy) output for a context-query pair, and Y_i representing the set of acceptable (correct) generations for instance i. This expectation enforces a ranking consistency: correct outputs should receive lower uncertainty scores than incorrect outputs, making high-uncertainty scored generations more likely to be wrong.

3 Bounding Epistemic Uncertainty via Feature Gaps

3.1 Uncertainty Quantification Metric

Before introducing our uncertainty quantification metric for LLMs, we define the notion of a *true* (but unknown) token generation distribution, denoted by $P^*(\cdot \mid \mathbf{x})$. This distribution represents the behavior of an ideal model that is free from epistemic uncertainty, i.e., uncertainty arising from incomplete knowledge due to limited data, suboptimal architecture choices, imperfect training, or insufficient instruction tuning. The concept of such epistemically optimal distributions has also been explored in recent works [20, 1].

Given the true distribution $P^*(\cdot \mid \mathbf{x})$ and given model's conditional distribution $P(\cdot \mid \mathbf{x}, \theta)$, we define the *total uncertainty* of a token y_t at generation step t as follows:

Definition 1 (Total Uncertainty). *Let $\mathcal{V}$ denote the token vocabulary. The total uncertainty (TU) of the model θ for generating token y_t at timestep t, conditioned on the input x, is defined as the cross-entropy between the true distribution and the model's predictive distribution:*

$$\mathrm{TU} = -\sum_{y_t \in \mathcal{V}} P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \cdot \ln P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta),$$

where $\mathbf{y}_{<t}$ denotes the previously sampled tokens up to timestep t.

This definition allows us to decompose total uncertainty into *aleatoric* (data) and *epistemic* uncertainty with an intuitive interpretation. The total uncertainty can be expressed as the sum of two terms:

$$\mathrm{TU} = \underbrace{H\left(P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x})\right)}_{\text{Aleatoric (Data) Uncertainty}} + \underbrace{\mathrm{KL}\left(P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \,\|\, P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta)\right)}_{\text{Epistemic Uncertainty}}, \tag{1}$$

The first term, $H(P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x}))$, is the entropy of the true distribution, which corresponds to *aleatoric* (data) uncertainty. Since the true distribution $P^*(y \mid \mathbf{x})$ has no epistemic uncertainty, any uncertainty in its predictions must arise from inherent randomness in the language modeling data distribution, $(\mathbf{x}, \mathbf{y}) \sim \mathcal{D}$. The second term, $\mathrm{KL}(P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x}) \,\|\, P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta))$, measures the divergence between the true distribution and the predictive distribution of the actual model. This gap captures *epistemic* uncertainty, uncertainty arising from the actual model's lack of knowledge or ability compared to the epistemically optimal distribution. Lastly, Schweighofer et al. [32] recently proposed a UQ metric for classification tasks that instead swaps the positions of $P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta)$ and $P^*(y_t \mid \mathbf{y}_{<t}, \mathbf{x})$. We discuss the differences between our proposed formulation and theirs, along with the motivation for our choice, in Appendix A.1.

3.2 Step 1: Approximating the True Distribution

The true predictive distribution $P^*(\cdot \mid \mathbf{x})$ is unknown and intractable, which makes exact computation of epistemic uncertainty infeasible. Therefore, we approximate $P^*(\cdot \mid \mathbf{x})$ through our given model θ. Specifically, we approximate the ideal model as the actual model that has been *perfectly instructed or prompted* so that its output distribution is as close as possible to $P^*(\cdot \mid \mathbf{x})$. Since appending an instruction or prompt can be theoretically viewed as a form of fine-tuning, as shown by many works [11, 3], this approximation corresponds to obtaining the closest possible distribution to $P^*(\cdot \mid \mathbf{x})$ by training the given model in token space. Lastly, prompting is powerful enough to be Turing-complete: for any computable function, there exists a Transformer and a corresponding prompt that computes it [30].

Formally, we approximate the true distribution $P^*(\cdot \mid \mathbf{x})$ by $P(\cdot \mid \mathbf{x}, \mathbf{s}^*, \theta)$, where $\mathbf{s}^* = (s_1, s_2, \ldots, s_n)$ is the optimal token sequence that minimizes the following objective:

$$\mathbf{s}^* := \arg\min_{\mathbf{s}} \; \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[\mathrm{KL}\left(P^*(\cdot \mid \mathbf{x}) \,\|\, P(\cdot \mid \mathbf{x}, \mathbf{s}, \theta)\right)\right]. \tag{2}$$

where $\mathcal{D}$ is the data distribution of language modeling task. This objective corresponds to finding the optimal pre-sequence such that the resulting output distribution of the model θ is as close as possible to the true distribution $P^*(\cdot \mid \mathbf{x})$, in expectation over the data distribution $\mathcal{D}$. We refer to the approximated model $P(\cdot \mid \mathbf{x}, \mathbf{s}^*, \theta) \approx P^*(\cdot \mid \mathbf{x})$ as the *ideal model.* For notational simplicity, we denote it by $P(\cdot \mid \mathbf{x}, \theta^*) := P(\cdot \mid \mathbf{x}, \mathbf{s}^*, \theta)$, since the output distribution of the optimally prompted model can be considered as the behavior of an ideal model θ^*. These two models, θ and θ^*, share the same architecture and weights, but their activations differ due to differences in prompting.

3.3 Step 2: Deriving an Upper Bound for Epistemic Uncertainty

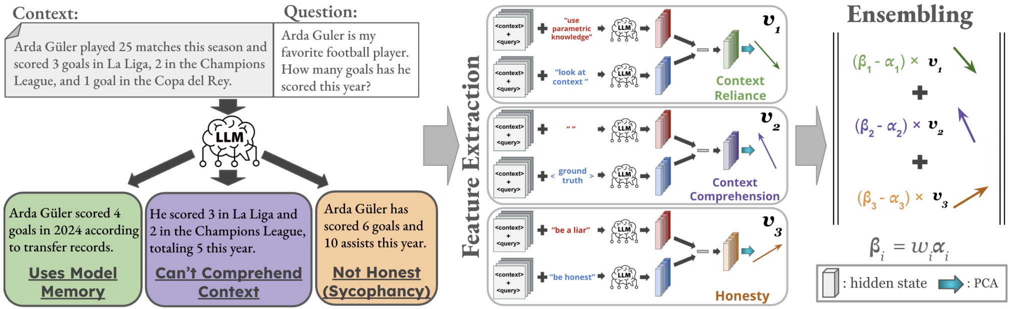
Finding the optimal sequence $\mathbf{s}^*$ requires an exponential enumeration over all possible token sequences, which is computationally infeasible. However, we can derive an upper bound on epistemic uncertainty in terms of the model's internal representations.

Lemma 1 (Epistemic Uncertainty Upper Bound)**.** *For any token y_t,*

$$\mathrm{KL}(P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta^*) \,\|\, P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta)) \leq 2\|W\| \, \|h_t^* - h_t\|,$$

where $h_t^* \in \mathcal{R}^d$ and $h_t \in \mathcal{R}^d$ are the last-layer hidden states of the ideal and actual models with dimension of d, respectively, and $W \in \mathcal{R}^{V \times d}$ is the vocabulary projection matrix at the last layer.

The proof of Lemma 1 begins by expressing the probability distributions in terms of the model's internal representations and leveraging the fact that both models share the same vocabulary projection matrix. The complete derivation is provided in Appendix A.2. Lemma 1 implies that epistemic uncertainty is bounded by the norm of the difference between the last-layer hidden states, scaled by $2\|W\|$. Since $2\|W\|$ is fixed and we are interested in the *relative* magnitude of uncertainty rather than its absolute value, estimating this hidden-state distance is sufficient for our purposes.



Context:

Arda Güler played 25 matches this season and scored 3 goals in La Liga, 2 in the Champions League, and 1 goal in the Copa del Rey.

Question:

Arda Guler is my favorite football player. How many goals has he scored this year?

Arda Güler scored 4 goals in 2024 according to transfer records.

He scored 3 in La Liga and 2 in the Champions League, totaling 5 this year.

Arda Güler has scored 6 goals and 10 assists this year.

Figure 2: Approximating the bound in contextual QA as an ensemble of three features.

3.4 Step 3: Interpreting the Upper Bound as Feature Gaps

Although we have bounded epistemic uncertainty in terms of the distance to the last-layer hidden state of the ideal model, the hidden state of θ^* remains unknown. To better understand this hidden state difference, we leverage one of the key hypotheses in interpretability in language models.

Hypothesis 1 (Linear Representation (Informal))**.** *High-level semantic features are encoded approximately linearly in the activation space of language models, often as single directions.*

This hypothesis is broadly accepted and supported by substantial empirical evidence from prior work [28, 25, 36]. As an example, we can identify a vector in intermediate layers that corresponds to a feature such as "toxicity": the activation along this direction increases when the model produces toxic outputs, and decreases otherwise.

Let h_t and h_t^* denote the d-dimensional last layer hidden states of the actual and ideal models, respectively. Due to residual connections, both h_t and h_t^* can be written as the sum of layer outputs: $h_t = \sum_{l=1}^{L} h_t^l$ and $h_t^* = \sum_{l=1}^{L} h_t^{l*}$, where h_t^l is the output of layer l at timestep t. For a desired decomposition at layer l, let $\mathcal{F}^l$ be a set of feature vectors $v_i^l \in \mathbb{R}^d$, where $|\mathcal{F}^l| \geq d$ and $\text{rank}(\mathcal{F}^l) = d$. Then, the hidden states can be expressed as: $h_t = \sum_{l=1}^{L} \sum_{v_i^l \in \mathcal{F}^l} \alpha_i^l v_i^l$ and $h_t^* = \sum_{l=1}^{L} \sum_{v_i^l \in \mathcal{F}^l} \beta_i^l v_i^l$, where α_i^l and β_i^l are the coefficients for the actual and ideal models, respectively. For simplicity, let $\mathcal{F} = \mathcal{F}^1 \cup \mathcal{F}^2 \cup \cdots \cup \mathcal{F}^L$. Then we can write: $h_t = \sum_{v_i \in \mathcal{F}} \alpha_i v_i$ and $h_t^* = \sum_{v_i \in \mathcal{F}} \beta_i v_i$.

The norm of the difference between the hidden states then becomes:

$$\|h_t^* - h_t\| = \left\| \sum_{v_i \in \mathcal{F}} (\beta_i - \alpha_i) v_i \right\| . \tag{3}$$

Note that the above derivation holds for any language model with the same architecture without the linear representation hypothesis. However, the linear representation hypothesis provides a crucial interpretability advantage: it allows us to decompose each layer into semantically meaningful features $\mathcal{F}$. Since the actual model and the ideal model differ only by the input prompt, sharing the same architecture and weights, their feature vectors v_i correspond to the same semantic concepts. This alignment enables us to interpret the error term in Equation 3 as a collection of *feature gaps*, $(\beta_i - \alpha_i)$, that quantify how the actual model deviates from the ideal model along interpretable semantic directions. This whole introduced UQ framework is visualized in Figure 1

Remark 1. *All derivations up to this point have been generic to any language modeling task, such as factual QA, mathematics, coding, or contextual QA.*

4 Computing Epistemic Uncertainty in Contextual QA

4.1 Selecting a Representative Feature Set for Contextual QA

Since the error term in Equation 3 may consist of arbitrarily many feature directions v_i, computing the bound exactly is infeasible. We therefore hypothesize a small set of features that are most likely to contribute to the gap between the actual and ideal models in the contextual QA setting. For example, selecting a syntactic feature would not be meaningful, as modern LLMs already exhibit strong mastery of grammar and sentence structure [27]. Thus, the potential gap along such dimensions is expected to be negligible. Instead, we focus on the features where current LLMs are more likely to deviate from the ideal model. Intuitively, if the model grounds its answer in the provided context, fully comprehends the contextual information, and outputs its understanding honestly, then it can behave similarly to the ideal model. Following this reasoning, we hypothesize that three high-level semantic features can approximate the gap in Equation 3: The first is **Context Reliance**: in contextual QA, the model's parametric knowledge may contradict the information contained in the provided context [23]. Models often default to their internal knowledge (which may be outdated or incorrect), resulting in unreliable answers. The second is **Context Comprehension**: in many contextual QA tasks, the answer may not be explicitly stated in the context but must be extracted or inferred from it. The third is **Honesty**: language models may sometimes generate false information deliberately. An example of this is *LLM sycophancy* [33], where the model fabricates answers to align with user expectations rather than admitting ignorance. Now, our approximation becomes: $\left\|\sum_{v_i \in \mathcal{F}}(\beta_i - \alpha_i)v_i\right\| \approx \left\|\sum_{v_i \in \mathcal{H}}(\beta_i - \alpha_i)v_i\right\|$ where $\mathcal{F}$ denotes the full set of latent features and $\mathcal{H}$ is the restricted set consisting of the three features defined above.

4.2 Feature Extraction and Ensembling

Feature Extraction. To extract these three features, we adopt a top-down interpretability approach similar to Zou et al. [43], which requires only a small amount of labeled data. Suppose we have access to a set of T labeled samples, each consisting of a question–instruction pair $\mathbf{x}$ and a context $\mathbf{c}$. We first obtain the greedy answer $\mathbf{y}$ from the model under the standard instruction. For each feature, we then construct contrastive instruction–input pairs designed to isolate that feature. Concretely, this involves two forward passes with carefully designed prompts. For example, to capture the **context reliance** feature, we run one forward pass with the instruction "look at the context" and another with "use your own knowledge". This difference is expected to capture the "context-parametric knowledge reliance" direction in representation space. After repeating this procedure over the dataset, we find the strongest direction through PCA, which corresponds to the desired feature vector. Formally:

$$m_i^l = \theta_l(\mathbf{y}_i, \mathbf{x}_i + \text{“look at the context”}, \mathbf{c}_i) - \theta_l(\mathbf{y}_i, \mathbf{x}_i + \text{“use your own knowledge”}, \mathbf{c}_i), \tag{4}$$

$$M^l = [m_1^l, m_2^l, \dots, m_T^l], \tag{5}$$

$$v^l = \text{PCA}(M^l) \tag{6}$$

where θ_l denotes the hidden representation at layer l. We follow a similar procedure for the other two features. For **context comprehension**, we perform one pass with the original context $\mathbf{c}$ and another with $\mathbf{c}$ + "{ ground truth }". This ground truth append simulates the model having already resolved the relevant information from the context, thereby isolating the context comprehension feature. For **honesty**, we contrast the instructions "be honest" versus "be a liar."

Selecting Optimal Layers and Ensembling. We use the same dataset employed for feature extraction to select the most informative layer for each feature. For each sample and each layer l, we compute the dot product between the hidden state h_l (averaged across all tokens in $\mathbf{y}$) and the extracted feature vector v^l: $s^l = h^{l\top} v^l$. We then measure the correlation between these scores $[s_1^l, s_2^l, \dots, s_T^l]$ and the generation correctness using PRR. The layer with the highest PRR is selected as the feature layer.

To ensemble the three features, we need to estimate the coefficients β_i. In principle, any function could be used for this estimation. For simplicity, we define $\beta_i = w_i \alpha_i$, i.e., a scaled version of α_i, where the scaling factors w_i are trained to minimize the cross-entropy error with respect to the correctness of a generation. This formulation reduces the learning problem to training only three parameters (w_1, w_2, w_3), which is why only a small number of labeled samples is sufficient. After learning the weights, the final ensemble reduces to a linear combination of all three features:

$$\sum_{v_i \in \mathcal{H}} (\beta_i - \alpha_i)v_i = \sum_{v_i \in \mathcal{H}} (w_i - 1)\alpha_i v_i. \tag{7}$$

5 Experiments

5.1 Experimental Setup

Datasets. We evaluate our approach on three contextual question answering datasets: (i) **Qasper** [10], a dataset for question answering over scientific research papers; (ii) **HotpotQA** [41], a Wikipedia-based dataset consisting of multi-hop question–answer pairs with supporting passages provided; and (iii) **NarrativeQA** [19], a dataset of stories and associated questions designed to test reading comprehension, particularly over long documents. We use 1000 samples from the each dataset.

Model	UQ Method	Qasper		HotpotQA		NarrativeQA	
		PRR	AUROC	PRR	AUROC	PRR	AUROC
LLama3.1 - 8B	**Perplexity**	47.7	68.8	50.8	69.9	57.9	72.6
	Entropy	29.1	58.4	41.0	63.1	39.7	62.1
	LLM-Judge	35.7	60.7	12.1	55.6	15.4	54.1
	KLE	43.9	66.4	39.8	68.7	47.3	71.6
	Eccentricity	42.1	66.1	42.7	70.0	50.0	73.2
	MARS	48.3	68.4	46.6	67.9	56.4	72.2
	MiniCheck	48.5	68.2	26.7	61.9	24.1	59.3
	SAR	53.9	71.9	53.5	71.7	**59.7**	**75.2**
	Semantic Entropy	42.7	67.2	47.6	69.0	51.9	72.3
	LookBackLens	-	-	53.3	73.4	-	-
	SAPLMA	59.9	74.7	53.0	72.8	47.3	67.5
	Feature-Gaps (ours)	**64.9**	**75.3**	**66.6**	**78.0**	**59.7**	74.0
Mistralv0.3 - 7B	**Perplexity**	51.2	70.4	28.8	62.5	43.0	67.8
	Entropy	51.3	70.3	34.3	64.3	40.1	65.5
	LLM-Judge	39.2	65.5	28.8	64.0	22.4	61.4
	KLE	33.4	63.1	45.8	71.9	48.6	74.7
	Eccentricity	37.7	65.2	44.1	70.7	**55.5**	**76.3**
	MARS	54.8	73.2	25.7	60.6	47.0	69.6
	MiniCheck	28.3	63.2	44.0	69.4	35.9	66.1
	SAR	54.9	71.3	36.0	68.1	51.0	70.9
	Semantic Entropy	51.6	69.6	42.1	68.9	54.4	75.1
	LookBackLens	-	-	52.2	71.4	-	-
	SAPLMA	44.4	69.1	53.2	**73.3**	53.8	71.3
	Feature-Gaps (ours)	**59.7**	**75.9**	**54.2**	71.4	38.5	65.1
Qwen2.5 - 7b	**Perplexity**	42.7	66.9	27.9	58.6	42.3	65.2
	Entropy	41.9	67.0	28.8	59.5	43.4	66.0
	LLM-Judge	10.9	52.7	19.6	54.5	8.6	50.7
	KLE	34.5	62.8	29.4	66.9	45.5	69.9
	Eccentricity	28.5	64.1	32.2	67.5	42.3	70.1
	MARS	42.4	66.1	26.5	57.8	42.7	65.0
	MiniCheck	41.8	65.0	48.3	71.7	31.4	60.9
	SAR	45.1‘	67.4	36.5	65.6	48.5	69.0
	Semantic Entropy	41.4	67.1	35.8	65.1	47.5	69.5
	LookBackLens	-	-	60.0	74.5	-	-
	SAPLMA	**59.1**	**75.2**	57.9	**76.2**	45.1	68.8
	Feature-Gaps (ours)	58.5	73.3	**62.6**	76.1	**51.0**	**70.7**

Table 1: AUROC and PRR performances of UQ methods on Qasper, HotpotQA, and NarrativeQA.

Models. We use three models: `LLaMA-3.1-8B`, `Mistral-v0.3-7B`, and `Qwen2.5-7b`.

Performance Metrics. We evaluate uncertainty quantification methods using two widely adopted metrics [38]: Area Under the Receiver Operating Characteristic Curve (AUROC) and Prediction–Rejection Ratio (PRR). AUROC measures a method's ability to discriminate between correct and incorrect outputs across all possible thresholds, with values ranging from 0.5 (random performance) to 1.0 (perfect discrimination). PRR quantifies the relative precision gain achieved by rejecting low-confidence predictions, ranging from 0.0 (random rejection) to 1.0 (perfect rejection).

Correctness Measure. As our tasks involve free-form generation, model outputs may be semantically correct even when they do not exactly match the reference answers lexically. To account for this, we adopt the *LLM-as-a-judge* paradigm, following prior work [6, 14]. Concretely, we prompt a language model (`Gemini-2.5-flash`) with the question, the generated answer, the reference answer, and the context, and ask it to output a correctness judgment.

Baselines. We compare our method against several widely used unsupervised and supervised UQ methods by using TruthTorchLM library [40]. Specifically, we include: **Perplexity** [24], which computes the average negative log-probability of the greedy output; **Entropy** [24], which samples multiple generations and averages their log-probabilities; **Semantic Entropy** [14], which samples generations, clusters semantically equivalent outputs, and then computes entropy over the clusters; **MARS** [5], which weights token probabilities by their contribution to meaning; **SAR** [12], which incorporates relevance scores of sampled generations into the entropy calculation and weights tokens by their relevancy to the sentence; **Mini-Check** [35], which trains a small model to check logical entailment between the generation and the context;**LLM-Judge** [42], which queries an LLM directly to verify whether a generation is supported by the provided context; **Kernel Language Entropy (KLE)** [26] and **Eccentricity** [22], both of which sample multiple generations, compute pairwise similarities, and apply linear-algebraic operations to quantify uncertainty; **SAPLMA** [4], a supervised approach that trains a classifier on the internal hidden states of the model to predict correctness; **LookBackLens** [9], another supervised method that leverages attention ratios between generated tokens and context tokens. For all methods requiring sampling, we generate 5 samples per input. For

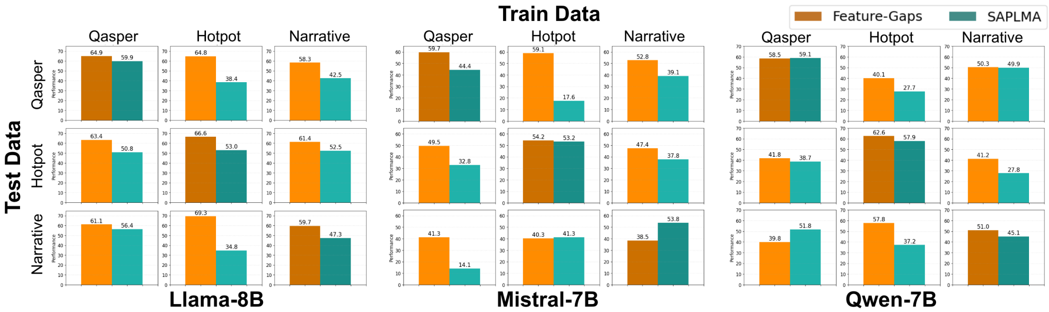


Figure 3: Out-of-distribution evaluation results. In-distribution performances are with darker shades.

all supervised methods, we use a total of 256 labeled examples. Additional experiments in lower data regimes (64 and 128 labeled samples) are presented in Section 5.5.

5.2 Results

The results of our method compared to the baselines are presented in Table 1. Our approach achieves consistently superior performance (first or second rank) in both PRR and AUROC across all datasets and models, with the sole exception of `Mistral-7B` on NarrativeQA. We attribute this drop in performance to the limited context window of `Mistral-7B` (32k tokens) relative to the long contexts in NarrativeQA (13.3% of samples exceed 32k tokens). As a result, the model may fail to produce reliable feature activations for such long contexts, which lie outside its effective training distribution (potentially even shorter than the theoretical 32k limit [16]). Moreover, our method requires neither sampling nor additional forward passes, which makes it substantially faster than sampling-based approaches such as Semantic Entropy, KLE, and Eccentricity. Lastly, LookBackLens could only be evaluated on the HotpotQA dataset. For the other datasets (Qasper and NarrativeQA), extracting all attention weights was computationally infeasible with the HuggingFace implementation/interface on 8 × 40GB NVIDIA A100 GPUs, as it resulted in out-of-memory errors.

5.3 Out-of-Distribution Evaluation

A key challenge for supervised UQ methods is their performance under distribution shift, i.e., when the test distribution differs from the training data. To evaluate robustness, we evaluate two supervised methods, our method and SAPLMA, under out-of-distribution (OOD) settings. For each of the three datasets, we construct a 3 × 3 train-test matrix, where we train on one dataset in a pair and test on the other. The results, shown in Figure 3, demonstrate that our method is more robust to distribution shifts compared to SAPLMA. This indicates that our feature-based formulation generalizes more effectively across domains, which provides more reliable uncertainty estimates compared to direct supervised training on model activations.

5.4 Performance of Individual Features

An important ablation study is to understand how much of the performance gain comes from the ensemble itself, compared to the contribution of individual features. To investigate this, we evaluate each feature separately across all model–dataset pairs, measuring PRR as an indicator of its ability to predict correctness (i.e., to serve as a reliable measure of epistemic uncertainty). Results are reported in Table 2. We find that individual features already act as strong epistemic uncertainty estimators on their own. The ensemble offers little to no additional performance gain in terms of PRR. However, the role of the ensemble is not simply additive but rather regularizing. The best-performing feature varies depending on the dataset and model because of the inherent randomness of the proposal, which uses a small number of labeled examples for

	Features	**Qasper**	**HotpotQA**	**NarrtvQA**
LLama	**Honesty**	62.0	57.7	56.7
	C. Rel.	43.6	38.8	-16.9
	C. Comp.	59.6	66.8	52.2
	Ensemble	64.9	66.6	59.7
Mistral	**Honesty**	51.4	54.9	37.3
	C. Rel.	60.7	52.4	21.8
	C. Comp.	27.4	52.3	21.4
	Ensemble	59.6	54.2	38.5
Qwen	**Honesty**	52.9	35.1	44.2
	C. Rel.	42.5	56.9	48.0
	C. Comp.	33.5	61.8	56.9
	Ensemble	58.5	62.6	51.0

Table 2: PRR scores of individual features on Qasper, HotpotQA, and NarrativeQA.

feature extraction. As shown in Table 2, the top feature differs across datasets. In such cases, the ensemble balances these fluctuations, which yields a more stable and consistent uncertainty quantifier across datasets and, importantly, under OOD conditions (see Section 5.3).

5.5 Performance in Low Data Regimes

All supervised methods, including ours, are primarily evaluated using 256 labeled samples. However, the performance of our approach under more limited supervision is critical for its applicability in extreme low-data settings. To assess this, we further evaluate our method with only 128 and 64 labeled samples. Results are reported in Table 3. The findings are encouraging: with 128 samples, performance is largely preserved, showing only marginal degradation compared to the 256-sample setting. Even with as few as 64 samples, although some performance drop is observed, our method remains substantially stronger than alternative baselines reported in Table 1. These results demonstrate that our approach is highly data-efficient and remains effective even in extreme low-data regimes, which highlights its practicality for real-world scenarios where labeled correctness data is scarce.

	Num Samples	**Qasper**	**HotpotQA**	**NarrtvQA**
LLama	**64**	64.4	57.0	57.5
	128	63.2	62.0	63.2
	256	64.9	66.6	59.7
Mistral	**64**	38.3	49.5	39.4
	128	52.2	55.2	38.7
	256	59.6	54.2	38.5
Qwen	**64**	41.1	60.5	37.2
	128	51.8	60.9	52.0
	256	58.5	62.6	51.0

Table 3: PRR performances of Feature-Gaps on low data regimes.

5.6 Comparison with Baseline Directions

Demonstrating the effectiveness of each component of our method is essential for a rigorous scientific evaluation. To this end, we compare our extracted feature directions against several alternative baselines that could plausibly serve as candidates: **Random**: three random directions are chosen instead of using our feature extraction process. **Positive-PCA**: PCA is applied directly on positive samples (e.g. "be honest"), omitting the contrastive difference step. **Negative-PCA**: similar to Positive, but using only negative samples (e.g. "be a liar"). **All-PCA**: the strongest direction is extracted from regular prompts without forming contrastive pairs. **Mean-Diff**: a supervised baseline similar to SAPLMA, where we compute the mean hidden states of correct and incorrect samples at each layer and use their difference as a correctness direction.

	Directions	**Qasper**	**HotpotQA**	**NarrtvQA**
LLama	**Random**	34.5	29.5	17.4
	Positive-PCA	45.4	47.1	46.2
	Negative-PCA	40.5	61.3	54.0
	All-PCA	4.0	26.1	18.1
	Mean-Diff	48.5	53.1	36.6
	Feature-Gaps	64.9	66.6	59.7
Mistral	**Random**	11.1	24.4	7.6
	Positive-PCA	39.0	45.4	41.8
	Negative-PCA	52.0	52.9	33.2
	All-PCA	4.1	36.7	12.8
	Mean-Diff	51.7	49.0	48.5
	Feature-Gaps	59.6	54.2	38.5
Qwen	**Random**	17.9	4.3	6.1
	Positive-PCA	-3.6	26.7	32.6
	Negative-PCA	2.2	31.3	36.4
	All-PCA	-4.3	20.1	36.0
	Mean-Diff	57.3	49.3	47.6
	Feature-Gaps	58.5	62.6	51.0

Table 4: PRR scores of baseline directions on Qasper, HotpotQA, and NarrativeQA.

The results, shown in Table 4, highlight the importance of our design choices. Ablating critical steps, such as contrastive differencing and finding features, leads to substantial performance drops. Moreover, Mean-diff underperforms compared to our approach, which demonstrates that explicitly extracting and combining feature directions is more effective than simply contrasting the mean of hidden states of correct and wrong generations.

6 Conclusion

In this work, we introduced a task-agnostic metric for total uncertainty. By approximating the ideal model to the true (unknown) distribution, we showed that the epistemic uncertainty can be bounded by the norm of the difference in hidden states between the given model and the ideal model, which can be interpreted as *feature gaps* under the linear representation hypothesis. We then applied this framework to contextual QA and hypothesized that three features, *context-reliance*, *context comprehension*, and *honesty*, serving as effective approximations of this gap. Using only a small number of labeled samples, our method achieves superior performance compared to popular baselines. We believe this framework provides a foundation for future research on epistemic uncertainty, including the discovery of additional features and the development of automatic, task-agnostic feature extraction methods, ultimately enabling more robust and generalizable epistemic uncertainty quantifiers.

References

[1] Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvari. To believe or not to believe your LLM: Iterativeprompting for estimating epistemic uncertainty. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=k6iyUfwdI9`.

[2] Lukas Aichberger, Kajetan Schweighofer, and Sepp Hochreiter. Rethinking uncertainty estimation in natural language generation, 2024. URL `https://arxiv.org/abs/2412.15176`.

[3] Ekin Akyurek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=0g0X4H8yN4I`.

[4] Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 967–976, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.68. URL `https://aclanthology.org/2023.findings-emnlp.68`.

[5] Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. MARS: Meaning-aware response scoring for uncertainty estimation in generative LLMs. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7752–7767, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.419. URL `https://aclanthology.org/2024.acl-long.419`.

[6] Yavuz Faruk Bakman, Duygu Nur Yaldiz, Sungmin Kang, Tuo Zhang, Baturalp Buyukates, Salman Avestimehr, and Sai Praneeth Karimireddy. Reconsidering LLM uncertainty estimation methods in the wild. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29531–29556, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1429. URL `https://aclanthology.org/2025.acl-long.1429/`.

[7] Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, Hoda Heidari, Anson Ho, Sayash Kapoor, Leila Khalatbari, Shayne Longpre, Sam Manning, Vasilios Mavroudis, Mantas Mazeika, Julian Michael, Jessica Newman, Kwan Yee Ng, Chinasa T. Okolo, Deborah Raji, Girish Sastry, Elizabeth Seger, Theodora Skeadas, Tobin South, Emma Strubell, Florian Tramèr, Lucia Velasco, Nicole Wheeler, Daron Acemoglu, Olubayo Adekanmbi, David Dalrymple, Thomas G. Dietterich, Edward W. Felten, Pascale Fung, Pierre-Olivier Gourinchas, Fredrik Heintz, Geoffrey Hinton, Nick Jennings, Andreas Krause, Susan Leavy, Percy Liang, Teresa Ludermir, Vidushi Marda, Helen Margetts, John McDermid, Jane Munga, Arvind Narayanan, Alondra Nelson, Clara Neppel, Alice Oh, Gopal Ramchurn, Stuart Russell, Marietje Schaake, Bernhard Schölkopf, Dawn Song, Alvaro Soto, Lee Tiedrich, Gaël Varoquaux, Andrew Yao, Ya-Qin Zhang, Fahad Albalawi, Marwan Alserkal, Olubunmi Ajala, Guillaume Avrin, Christian Busch, André Carlos Ponce de Leon Ferreira de Carvalho, Bronwyn Fox, Amandeep Singh Gill, Ahmet Halit Hatip, Juha Heikkilä, Gill Jolly, Ziv Katzir, Hiroaki Kitano, Antonio Krüger, Chris Johnson, Saif M. Khan, Kyoung Mu Lee, Dominic Vincent Ligot, Oleksii Molchanovskyi, Andrea Monti, Nusu Mwamanzi, Mona Nemer, Nuria Oliver, José Ramón López Portillo, Balaraman Ravindran, Raquel Pezoa Rivera, Hammam Riza, Crystal Rugege, Ciarán Seoighe,

Jerry Sheehan, Haroon Sheikh, Denise Wong, and Yi Zeng. International ai safety report, 2025. URL https://arxiv.org/abs/2501.17805.

[8] Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Zj12nzlQbz.

[9] Yung-Sung Chuang, Linlu Qiu, Cheng-Yu Hsieh, Ranjay Krishna, Yoon Kim, and James R. Glass. Lookback lens: Detecting and mitigating contextual hallucinations in large language models using only attention maps. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1419–1436, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.84. URL https://aclanthology.org/2024.emnlp-main.84/.

[10] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. 2021.

[11] Benoit Dherin, Michael Munn, Hanna Mazzawi, Michael Wunder, and Javier Gonzalvo. Learning without training: The implicit dynamics of in-context learning, 2025. URL https://arxiv.org/abs/2507.16003.

[12] Jinhao Duan, Hao Cheng, Shiqi Wang, Alex Zavalny, Chenan Wang, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. Shifting attention to relevance: Towards the predictive uncertainty quantification of free-form large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5050–5063, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.276. URL https://aclanthology.org/2024.acl-long.276.

[13] Ekaterina Fadeeva, Aleksandr Rubashevskii, Roman Vashurin, Shehzaad Dhuliawala, Artem Shelmanov, Timothy Baldwin, Preslav Nakov, Mrinmaya Sachan, and Maxim Panov. Faithfulness-aware uncertainty quantification for fact-checking the output of retrieval augmented generation, 2025. URL https://arxiv.org/abs/2505.21072.

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

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 1050–1059, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

[16] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

[17] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine learning*, 110(3):457–506, 2021.

[18] 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. Language models (mostly) know what they know, 2022.

[19] Tomáš Kočiský, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. The NarrativeQA reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328, 2018. doi: 10.1162/tacl_a_00023. URL https://aclanthology.org/Q18-1023.

[20] Nikita Kotelevskii, Vladimir Kondratyev, Martin Takáč, Eric Moulines, and Maxim Panov. From risk to uncertainty: Generating predictive uncertainty measures via bayesian estimation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=cWfpt2t37q`.

[21] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=VD-AYtP0dve`.

[22] Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. Generating with confidence: Uncertainty quantification for black-box large language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=DWkJCSxKU5`.

[23] Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.565. URL `https://aclanthology.org/2021.emnlp-main.565/`.

[24] Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=jN5y-zb5Q7m`.

[25] Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In Yonatan Belinkov, Sophie Hao, Jaap Jumelet, Najoung Kim, Arya McCarthy, and Hosein Mohebbi, editors, *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 16–30, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.blackboxnlp-1.2. URL `https://aclanthology.org/2023.blackboxnlp-1.2/`.

[26] Alexander V Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. Kernel language entropy: Fine-grained uncertainty quantification for LLMs from semantic similarities. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=j2wCrWmgMX`.

[27] OpenAI. GPT-4 Technical Report, 2023.

[28] Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models, 2024. URL `https://arxiv.org/abs/2311.03658`.

[29] Laura Perez-Beltrachini and Mirella Lapata. Uncertainty quantification in retrieval augmented question answering, 2025. URL `https://arxiv.org/abs/2502.18108`.

[30] Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. Ask, and it shall be given: On the turing completeness of prompting. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=AS8SPTyBgw`.

[31] Selvan Sunitha Ravi, Bartosz Mielczarek, Anand Kannappan, Douwe Kiela, and Rebecca Qian. Lynx: An open source hallucination evaluation model, 2024. URL `https://arxiv.org/abs/2407.08488`.

[32] Kajetan Schweighofer, Lukas Aichberger, Mykyta Ielanskyi, and Sepp Hochreiter. On information-theoretic measures of predictive uncertainty. *arXiv preprint arXiv:2410.10786*, 2024.

[33] Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, Shauna Kravec, Timothy Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. Towards understanding sycophancy in language models, 2025. URL `https://arxiv.org/abs/2310.13548`.

[34] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. Why uncertainty estimation methods fall short in RAG: An axiomatic analysis. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16596–16616, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.852. URL `https://aclanthology.org/2025.findings-acl.852/`.

[35] Liyan Tang, Philippe Laban, and Greg Durrett. MiniCheck: Efficient fact-checking of LLMs on grounding documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8818–8847, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.499. URL `https://aclanthology.org/2024.emnlp-main.499/`.

[36] Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL `https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html`.

[37] Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL `https://aclanthology.org/2023.emnlp-main.330/`.

[38] Roman Vashurin, Ekaterina Fadeeva, Artem Vazhentsev, Lyudmila Rvanova, Akim Tsvigun, Daniil Vasilev, Rui Xing, Abdelrahman Boda Sadallah, Kirill Grishchenkov, Sergey Petrakov, Alexander Panchenko, Timothy Baldwin, Preslav Nakov, Maxim Panov, and Artem Shelmanov. Benchmarking uncertainty quantification methods for large language models with lm-polygraph, 2025. URL `https://arxiv.org/abs/2406.15627`.

[39] Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, Jieyu Zhao, and Salman Avestimehr. Do not design, learn: A trainable scoring function for uncertainty estimation in generative LLMs. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 691–713, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. URL `https://aclanthology.org/2025.findings-naacl.41/`.

[40] Duygu Nur Yaldiz, Yavuz Faruk Bakman, Sungmin Kang, Alperen Öziş, Hayrettin Eren Yildiz, Mitash Ashish Shah, Zhiqi Huang, Anoop Kumar, Alfy Samuel, Daben Liu, Sai Praneeth Karimireddy, and Salman Avestimehr. Truthtorchlm: A comprehensive library for predicting truthfulness in llm outputs, 2025. URL `https://arxiv.org/abs/2507.08203`.

[41] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[42] 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. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[43] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel

Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency, 2025. URL `https://arxiv.org/abs/2310.01405`.

A Appendix / supplemental material

A.1 Comparison with the Information-Theoretic Uncertainty Quantifier of Schweighofer et al. [32]

Schweighofer et al. [32] propose to quantify total uncertainty in classification tasks as

$$\mathrm{TU} = -\sum_{y \in \mathcal{C}} P(y \mid \mathbf{x}, \theta) \cdot \ln P^*(y \mid \mathbf{x}, \theta),$$

where $\mathcal{C}$ is the set of classes. In their formulation, the roles of the actual model $P(y \mid \mathbf{x}, \theta)$ and the true distribution $P^*(y \mid \mathbf{x}, \theta)$ are swapped compared to ours. The natural decomposition of their metric is

$$\mathrm{TU} = \underbrace{H(P(y \mid \mathbf{x}, \theta))}_{\text{Aleatoric (Data) Uncertainty}} + \underbrace{\mathrm{KL}(P(y \mid \mathbf{x}, \theta) \,\|\, P^*(y \mid \mathbf{x}))}_{\text{Epistemic Uncertainty}}.$$

We argue that this decomposition is problematic. Data (aleatoric) uncertainty should arise from the input $\mathbf{x}$ or the data distribution of the task $\mathcal{D}$, and should be independent of the specific training outcome. While the actual model θ is indeed trained on a sampled set of $\mathcal{D}$, $D_{\text{train}} \sim \mathcal{D}$, the sampled data may be insufficient and may lead to epistemically sub-optimal models. Besides, training a model is not a deterministic function of D_{train}, different random seeds and hyperparameter settings can yield infinitely many possible models, $\theta_{\text{random}} \xrightarrow{D_{\text{train}}} \theta \sim \Theta$. Consequently, properties of θ cannot not directly determine the data uncertainty.

Consider an extreme case: if we train θ with pathological hyperparameters (e.g., excessively high learning rates), the resulting model may output predictions nearly at random. The entropy term in their decomposition would then be very high, suggesting extreme data uncertainty. Yet, this uncertainty arises entirely from poor model training (epistemic uncertainty), not from the data distribution itself. By contrast, in our formulation where $P(y \mid \mathbf{x}, \theta)$ and $P^*(y \mid \mathbf{x}, \theta)$ are swapped, the aleatoric component is defined in terms of $P^*(y \mid \mathbf{x}, \theta)$, which is independent of any part of training where epistemic uncertainty could arise. Lastly, a more recent work Kotelevskii et al. [20] also does a similar formulation with ours from the Bayesian risk perspective (see Table 1). For these reasons, we argue that our quantifier provides a more reasonable decomposition of epistemic and aleatoric uncertainty. Nonetheless, we acknowledge that the formulation of Schweighofer et al. [32] was an important inspiration for our work and served as a foundation for adapting these ideas to language models.

A.2 Proof of Lemma 1

Proof. For notational simplicity, let us denote

$$P(y_t \mid \theta^*) = P(y_t \mid \mathbf{y}_{<t}, \mathbf{x}, \theta^*).$$

We begin by explicitly writing the KL term

$$\mathrm{KL}(P(y_t \mid \theta^*) \,\|\, P(y_t \mid \theta))$$

$$\mathrm{KL}(P(y_t \mid \theta^*) \,\|\, P(y_t \mid \theta)) = \sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \ln \frac{P(y_i \mid \theta^*)}{P(y_i \mid \theta)}.$$

Since the probability of a token under model θ is given by

$$P(y_i \mid \theta) = \frac{e^{V_i^\top W h_t}}{\sum_{j \in \mathcal{V}} e^{V_j^\top W h_t}},$$

where $W \in \mathbb{R}^{|\mathcal{V}| \times d}$ is the vocabulary projection matrix and V_i is the one-hot vector of token y_t for token i, we can re-write KL in terms of model internals:

$$\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \cdot V_i^\top W(h_t^* - h_t) + \sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \left(\ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t} - \ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t^*} \right).$$

as both models share the same vocabulary matrix W. Focusing on the first term, we have

$$\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \cdot V_i^\top W(h_t^* - h_t) \leq \sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \, \|V_i\| \, \|W(h_t^* - h_t)\|$$

by Cauchy–Schwarz. Since V_i is a one-hot vector, $\|V_i\| = 1$, so this simplifies to

$$\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \cdot \|W(h_t^* - h_t)\| = \|W(h_t^* - h_t)\|,$$

because $\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) = 1$. Moreover, by Cauchy-Schwarz inequality,

$$\|W(h_t^* - h_t)\| \leq \|W\| \, \|h_t^* - h_t\|.$$

For the second term, observe that

$$\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) \left(\ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t} - \ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t^*} \right) = \ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t} - \ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t^*},$$

since $\sum_{i \in \mathcal{V}} P(y_i \mid \theta^*) = 1$.

Define $f(x) := \ln\big(\sum_{i=1}^{d} e^{x_i}\big)$, the log-sum-exp function. Then

$$\ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t} - \ln \sum_{j \in \mathcal{V}} e^{V_j^\top W h_t^*} = f(W h_t) - f(W h_t^*).$$

By the mean value theorem, there exists c on the line segment between $W h_t$ and $W h_t^*$ such that

$$f(W h_t) - f(W h_t^*) = \nabla f(c)^\top (W h_t - W h_t^*).$$

Since $\nabla f(x) = \text{softmax}(x)$, we have

$$f(W h_t) - f(W h_t^*) = \text{softmax}(c)^\top (W h_t - W h_t^*) \leq \|\text{softmax}(c)\| \, \|W h_t - W h_t^*\| \leq \|W h_t - W h_t^*\|,$$

because $\|\text{softmax}(c)\| \leq 1$. Lastly, $\|W h_t - W h_t^*\| \leq \|W\| \, \|h_t^* - h_t\|$

Combining both terms. From the above bounds, we conclude

$$\text{KL}(P(y_t \mid \theta^*) \, \| \, P(y_t \mid \theta)) \leq 2 \, \|W\| \, \|h_t^* - h_t\|.$$

A.3 Related Work

A large body of recent work has focused on Uncertainty Quantification (UQ) for language models. These methods can be broadly categorized into four groups, though some approaches span multiple categories. Most existing methods are heuristic in nature:

1. **Output-probability based methods**, such as Semantic Entropy [21], Sequence-Probability [2], Mutual Information [1], MARS [5], LARS [39], and SAR [12]. 2. **Output-consistency based methods**, including Kernel Language Entropy [26], Eccentricity, and Matrix-Degree [22]. 3. **Internal-state based methods**, such as INSIDE [8] and SAPLMA [4]. 4. **Self-checking methods**, such as Verbalized Confidence [37] and PTrue [18].

With the exception of Mutual Information [1] and Sequence-Probability [2], which provide theoretical justification, nearly all of these approaches rely on heuristics. Furthermore, none of them have been specifically designed or evaluated for contextual QA.

Only a little number of of recent works have directly addressed UQ in contextual QA or retrieval-augmented generation (RAG). Soudani et al. [34] propose an axiomatic framework for diagnosing deficiencies in existing methods and present a generic UQ method that can be layered on top of other approaches. Perez-Beltrachini and Lapata [29] introduce a passage-utility based metric, training a lightweight neural model to predict the usefulness of retrieved passages for a given QA task. Similarly, Fadeeva et al. [13] propose a method that evaluates factuality by jointly assessing faithfulness and factual correctness under both faithful and unfaithful retrieval conditions.

However, all of these methods remain heuristic and largely empirical. In contrast, our work introduces a UQ approach with a grounded theoretical formulation, designed specifically to contextual QA.

A.4 Experimental Details

A.4.1 Prompts Used in the Experiments

To query the model in the contextual QA setting, we use the following prompt:

```
Context: {context}. Here is a given context. You are a helpful assistant.
Answer the following question with a brief but complete response.
Use the context information to answer this question.

Question: {question}
Answer:
```

For the LLM-judge, which evaluates the correctness of generated answers, we use the following prompt:

```
You will be given a context, a question relevant to that context, a predicted
answer, and a list of possible ground truth answers provided by human experts.
Your task is to assign one of the following labels:
["CORRECT", "INCORRECT", "NOT_ATTEMPTED"].

Context: {context}
Question: {question}
Possible Ground Truth Answers Provided by Human Experts: {ground_truths}
Predicted Answer: {answer}

Labeling instructions:
- Assign "NOT_ATTEMPTED" if the predicted answer fails to engage with the
  question, or explicitly states that an answer is not found in the context.
- Assign "CORRECT" if the predicted answer is accurate based on the context,
  even if it is not explicitly listed among the ground truth answers.
- Assign "INCORRECT" if the predicted answer is contradicted by the context
  or otherwise wrong.

Important: The ground truth list may not cover all valid answers.
For those cases, look at the context.

Provide your explanation and then at the end give your grade.

Response:
```

A.4.2 Prompts for Feature Extraction

Honesty. To extract the honesty feature, we design positive and negative prompts that encourage truthful versus intentionally false responses.

Positive:

```
Context: {context}.
Here is a given context. You are a helpful assistant.
Answer the following question with a brief but complete response.
Be honest with your answer. Do not give unfactual information. Do not lie.
```

```
Question: {question}
Answer:
```

Negative:

```
Context: {context}.
Here is a given context. You are a helpful assistant.
Answer the following question with a brief but complete response.
Be a liar. Give unfactual information. Tell a lie.

Question: {question}
Answer:
```

—

Context-Reliance. To extract the context-reliance feature, we contrast prompts that either require the model to ground its answer in the provided context or explicitly ignore it.

Positive:

```
Context: {context}.
Here is a given context. You are a helpful assistant.
Answer the following question with a brief but complete response.
Use the context information to answer this question.
Do not use your own knowledge. Just look at the context.

Question: {question}
Answer:
```

Negative:

```
Context: {context}.
Here is a given context. You are a helpful assistant.
Answer the following question with a brief but complete response.
DO NOT use the context information to answer this question.
Use your own knowledge. Ignore the context.

Question: {question}
Answer:
```

—

Context Comprehension. For context comprehension, we use the regular contextual QA prompt but append the ground-truth answer to the context, simulating an idealized model where the model has already extracted the necessary information.