Uncertainty Decomposition and Quantification for In-Context Learning of Large Language Models

Anonymous ACL submission

Abstract

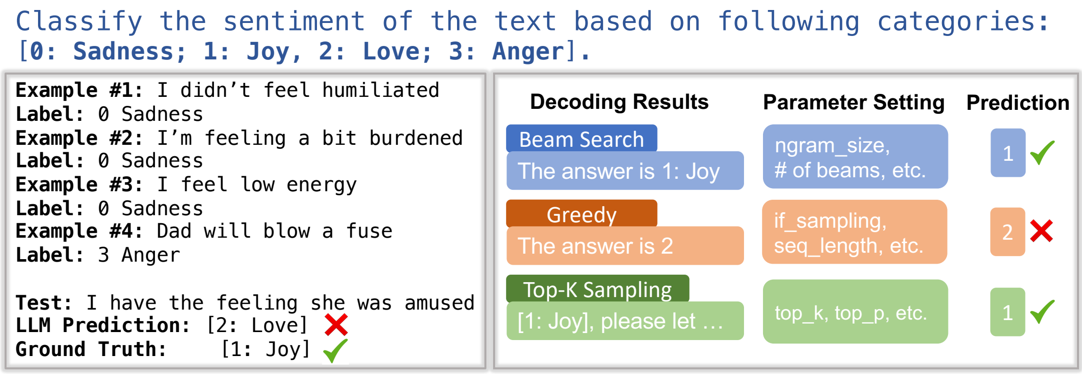
In-context learning has emerged as a groundbreaking ability of Large Language Models (LLMs) and revolutionized various fields by providing a few task-relevant demonstrations in the prompt. However, trustworthy issues with LLM's response, such as hallucination, have also been actively discussed. Existing works have been devoted to quantifying the uncertainty in LLM's response, but they often overlook the complex nature of LLMs and the uniqueness of in-context learning. In this work, we delve into the predictive uncertainty of LLMs associated with in-context learning, highlighting that such uncertainties may stem from both the provided demonstrations (aleatoric uncertainty) and ambiguities tied to the model's configurations (epistemic uncertainty). We propose a novel formulation and corresponding estimation method to quantify both types of uncertainties. The proposed method offers an unsupervised way to understand the prediction of in-context learning in a plug-and-play fashion. Extensive experiments are conducted to demonstrate the effectiveness of the decomposition. The code and data are available at: `anonymous.4open.science/r/UQ_ICL-6BF5`.

1 Introduction

Large Language Models (LLMs) have revolutionized diverse domains by serving as general task solvers, which can be largely attributed to the emerging capability: *in-context learning*. By providing demonstrations of the task to LLMs as part of the prompt, LLMs can quickly grasp the intention and make corresponding responses to the particular task (Min et al., 2022). In this paradigm, LLMs can quickly adapt to solve new tasks at inference time (without any changes to their weights). Advanced LLMs, e.g., GPT-4 and LLaMA, have achieved state-of-the-art results on LAMBADA (commonsense sentence completion), TriviaQA (question answering) (Xie et al., 2021), and many tasks in other domains (Ling et al., 2023a).

While in-context learning has achieved notable success, LLMs remain vulnerable to well-known reliability issues like hallucination (Rawte et al., 2023). Uncertainty quantification has emerged as a popular strategy to assess the reliability of LLM responses. In the past two years, numerous works (Xiao et al., 2022; Lin et al., 2023; Ling et al., 2023b; Amayuelas et al., 2023; Kuhn et al., 2023) have been proposed to quantify the uncertainty of LLM response. These approaches could return a confidence score or directly compute variance/entropy across multiple LLM responses; however, they often overlook the complex nature of LLMs and their reliance on provided demonstrations in in-context learning, so that existing methods may not provide insights into the underlying causes or the interactions among different factors contributing to uncertainty.

A natural question therefore arises: when LLM uses in-context learning to predict a wrong answer with high uncertainty, can we indicate if it is caused by the demonstration examples or by the model itself? Given LLM's responses to a particular task, it's essential to decompose the uncertainty into its primary sources to address the question. Specifically, *Aleatoric Uncertainty (AU)* refers to variations in the data, often linked to the demonstration examples. As shown in Figure 1 (a), LLM's output can easily be disturbed by inappropriate demonstrations since the provided demonstrations do not cover all possible labels. The noise and potential ambiguity of these demonstrations could bring uncertainty, which, in turn, may hinder the accuracy of the response. In contrast, *Epistemic Uncertainty (EU)* stems from ambiguities related to the model parameters or different configurations. As depicted in Figure 1 (b), different decoding strategies (e.g., beam search and greedy decoding) and their hyperparameter settings can have different decoding re-



(a) Inappropriate or insufficient few-shot demonstrations may cause uncertainty

(b) Various decoding strategies and parameter settings may cause uncertainty

Figure 1: Uncertainty in LLM's prediction can stem from two aspects: a) *Demonstration Quality*: LLMs are likely to make wrong predictions if the demonstrations are inappropriate; b) *Model Configuration*: different decoding strategies (e.g., beam_search and top_k sampling) and their parameter settings may return different predictions.

sults. Recognizing and quantifying the uncertainty from the model's perspective can also be critical in evaluating the generated responses, which allows us to understand the model's confidence level toward the task and make necessary adjustments (e.g., choosing a more powerful model or conducting an ensemble prediction).

Despite the strides made by existing works in understanding the total uncertainty, the decomposition of uncertainty in the realm of in-context learning remains under-explored. In this work, we propose a novel framework for decomposing uncertainty in in-context learning to aleatoric and epistemic components from the generated outputs. Our contributions are summarized as follows.

- **Problem.** We formulate the problem of uncertainty decomposition that extracts epistemic and aleatoric uncertainties from the predictive distribution of LLMs with in-context learning.
- **Method.** We propose quantifying both aleatoric and epistemic uncertainty from the mutual information perspective. A novel entropy-based estimation method is also designed to handle the free-form outputs of LLMs.
- **Experiment.** Extensive experiments are conducted to evaluate different aspects of uncertainty, followed by specific applications and case studies to show how two types of uncertainty influence the model's performance.

2 Uncertainty Decomposition of In-context Learning

We first formulate the process of in-context learning as Bayesian Neural Networks with latent variables. Based on the formulation, we propose to decompose the predictive uncertainty into its epistemic and aleatoric components from the mutual information perspective, followed by a novel way to estimate both uncertainties based on the entropy of the prediction's distribution.

2.1 Background

LLMs are typically trained using maximum likelihood estimation on a large corpus of text. The training goal is to maximize the likelihood of the observed data under the model: $\mathcal{L}(\Theta) = \prod_{i \leq N} p(\omega_i | \omega_1, \omega_2, \ldots, \omega_{i-1}; \Theta)$, where each $\omega_i \in \mathbf{x}$ is a token in a sentence $\mathbf{x} = [\omega_1, \ldots, \omega_N]$, and Θ denotes the set of parameters.

Latent Concept. From the Bayesian point of view, LLM's in-context learning ability is obtained by mapping the training token sequence $\mathbf{x}$ to a latent *concept* z (Xie et al., 2021). The concept z is a latent variable sampled from a space of concepts $\mathcal{Z}$, which defines a distribution over observed tokens ω_i from a training context $\mathbf{x}$:

$$p(\omega_1, \ldots, \omega_N) = \int_{z \in \mathcal{Z}} p(\omega_1, \ldots, \omega_N | z) p(z) dz.$$

The concept can be interpreted as various document-level statistics, such as the general subject matter of the text, the structure/complexity of the text, the overall emotional tone of the text, etc.

In-context Learning. Given a list of independent and identically distributed (i.i.d.) in-context demonstrations (contain both question and answer) $[\mathbf{x}_1, \ldots, \mathbf{x}_{T-1}]$ concatenated with a test question (without the task answer) $\mathbf{x}_T$ as prompt. Each demonstration $\mathbf{x}_i$ in the prompt is drawn as a sequence conditioned on the same concept z and describes the task to be learned. LLMs generate a response $\mathbf{y}_T$ to the test question $\mathbf{x}_T$ based on the aggregated prompt $\mathbf{x}_{1:T}$:

$$p(\mathbf{y}_T|\mathbf{x}_{1:T}) = \int_{z \in \mathcal{Z}} p(\mathbf{y}_T|\mathbf{x}_{1:T}, z) p(z|\mathbf{x}_{1:T}) dz.$$

In-context learning can be interpreted as *locating* a pre-existing concept z based on the provided demonstrations $\mathbf{x}_{1:T-1}$, which is then employed to tackle a new task $\mathbf{x}_T$. Including more high-quality demonstrations within the prompt can help refine the focus on the relevant concept, enabling its selection through the marginalization term $p(z|\mathbf{x}_{1:T})$.

In this work, we focus on quantifying the predictive uncertainty of LLMs in deterministic NLP tasks, such as text classification. Specifically, we address tasks where the training dataset $\mathcal{D} = \{\mathcal{X}, \mathcal{Y}\}$ consists of token sequences $\mathcal{X} = \{\mathbf{x}\}$ and their corresponding target outputs $\mathcal{Y} = \{\mathbf{y}\}$. For LLMs, the generation process is defined by the function $\mathbf{y} = f(\mathbf{x}, z; \Theta)$, where $f : \mathcal{X} \times \mathcal{Z} \to \mathcal{Y}$ is a deterministic function. The output $\mathbf{y}$ exhibits stochastic behavior, influenced by the latent concept $z \sim p(z|\mathbf{x}_{1:T})$ and the model parameters/configurations Θ (e.g., temperature, etc.).

2.2 Predictive Uncertainty Formulation of In-context Learning

We formulate the predictive distribution of in-context learning for predicting $\mathbf{y}_T$ given few-shot demonstrations $\mathbf{x}_{1:T-1}$ and a test case $\mathbf{x}_T$ as:

$$p(\mathbf{y}_T|\mathbf{x}_{1:T}) \approx \int p(\mathbf{y}_T|\Theta, \mathbf{x}_{1:T}, z) \cdot p(z|\mathbf{x}_{1:T}) q(\Theta) dz \, d\Theta, \tag{1}$$

where $p(\mathbf{y}_T|\Theta, \mathbf{x}_{1:T}, z)$ is approximated by a Bayesian Neural Network-based likelihood function $\mathcal{N}(f(\mathbf{x}_{1:T}, z), \mathbf{\Sigma})$, and $\mathbf{\Sigma}$ is the covariance matrix contains the variances and covariances associated with LLM parameters. $q(\Theta)$ is the approximated posterior of the LLM's parameters Θ. Eq. (1) serves as an initial framework for generating predictions based on input data and accompanying demonstrations: $p(\mathbf{y}_T|\mathbf{x}_{1:T})$, which entangles different types of uncertainties. We first present the overall pipeline of our uncertainty quantification framework, followed by formulation on decomposing the total uncertainty based on mutual information (Sec. 2.3) and a novel way to estimate the uncertainty (Sec. 2.4). Note that LLMs can be categorized into white-box and black-box models (Ling et al., 2023a) based on their transparency.

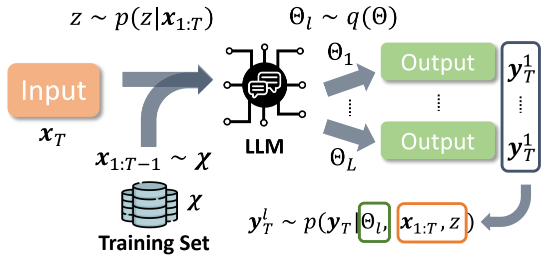


Figure 2: Uncertainty Quantification of In-context Learning Pipeline: we want to quantify the uncertainty that comes from 1) different in-context demonstrations $\mathbf{x}_{1:T}$; and 2) different model configurations Θ_l.

Quantifying mutual information involves accessing the probability of generated tokens, which is not applicable to black-box LLMs. In this study, we also provide a decomposition way from the variance perspective for black-box LLMs. Due to the space limit, the variance-based decomposition can be found in Appendix A.1.

Framework Pipeline. In this work, we employ a Bayesian framework to quantify the predictive uncertainty from LLMs, and the overall pipeline is visualized in Figure 2. Specifically, the input $\mathbf{x}_{1:T}$ is composed of a test query $\mathbf{x}_T$ and a set of demonstrations $\mathbf{x}_{1:T-1}$ sampled from $\mathcal{X}$. By sampling different model parameters/configurations $\Theta_l \sim q(\Theta)$, LLM can return different outputs $\mathbf{y}_T^l \in [\mathbf{y}_T^1, \cdots, \mathbf{y}_T^L]$ based on the conditional probability $p(\mathbf{y}_T|\Theta_l, \mathbf{x}_{1:T}, z)$. The collection of outputs $[\mathbf{y}_T^1, \cdots, \mathbf{y}_T^L]$ records the total uncertainty regarding Θ_l and demonstrations $\mathbf{x}_{1:T-1}$.

2.3 Entropy-based Decomposition

As a widely adopted measure of uncertainty, entropy provides a quantifiable and interpretable metric to assess the degree of confidence in the model's predictions (Malinin and Gales, 2020). Since white-box LLMs can return the probability of each token in the generated sequence, it naturally makes entropy-based uncertainty measures applicable uniformly across different types of white-box LLMs.

Epistemic Uncertainty (EU). Let $H(\cdot)$ be the differential entropy of a probability distribution, the total uncertainty in Eq. (1) can be quantified as $H(\mathbf{y}_T|\mathbf{x}_{1:T})$, which entangles both aleatoric (i.e., demonstration $\mathbf{x}_{1:T-1}$) and epistemic (i.e., model parameter Θ) uncertainties. To estimate the EU, we condition Eq. (1) on a specific realization of the model parameter Θ, yielding $p(\mathbf{y}_T|\mathbf{x}_{1:T}, \Theta) = \int p(\mathbf{y}_T|\mathbf{x}_{1:T}, z, \Theta) p(z|\mathbf{x}_{1:T}) dz$ with an associated entropy $H(\mathbf{y}_T|\mathbf{x}_{1:T}, z, \Theta)$. The expected value of this entropy under different demonstration sets

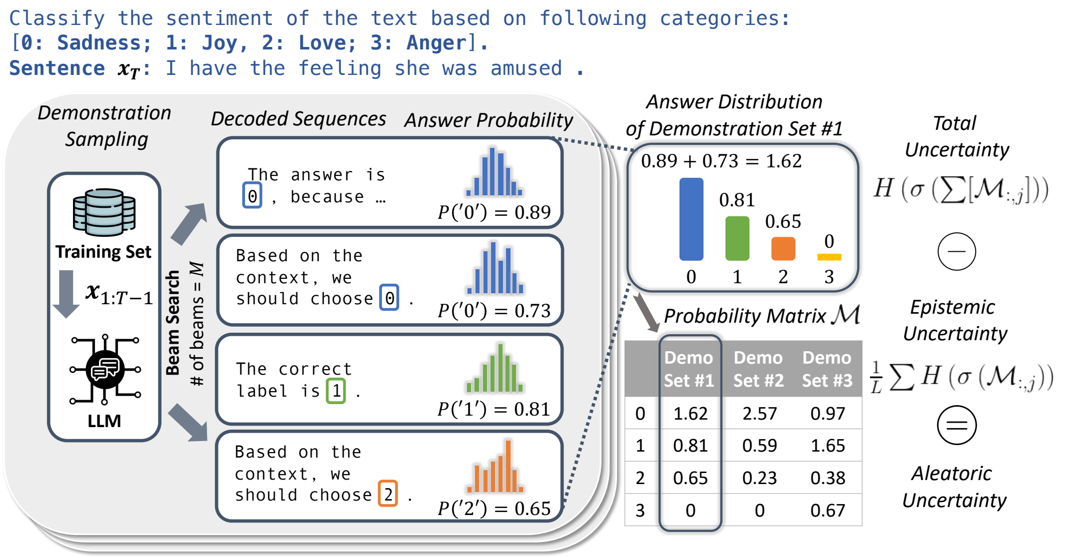


Figure 3: Framework of entropy-based uncertainty estimation, which consists of 1) generating M sequences based on a set of $\mathbf{x}_{1:T-1}$; 2) selecting token(s) that is relevant to the answer and extract the probabilities; 3) aggregating the token probabilities of M sequences into a distribution of predicted labels; 4) iterating the process L times corresponding to L different demonstration sets and form a probability matrix $\mathcal{M}$, where the column denotes different demonstration sets and the row denotes labels of the dataset.

can be expressed as $\mathbb{E}_z\left[H(\mathbf{y}_T|\mathbf{x}_{1:T}, z, \Theta)\right]$, which serves as a metric to quantify the EU in Eq. (1).

Aleatoric Uncertainty (AU). In terms of AU, the randomness comes from different sets of demonstration $\mathbf{x}_{1:T-1}$ and their corresponding latent concept z. To estimate AU, we can quantify the mutual information between $\mathbf{y}_T$ and latent concept z, which can often be leveraged as an evaluation metric of AU (Wimmer et al., 2023). As we have already obtained the EU, AU can subsequently be calculated as the discrepancy between the total uncertainty and the epistemic uncertainty:

$$\begin{aligned} I(\mathbf{y}_T, z|\Theta) = & H\left(\mathbf{y}_T|\mathbf{x}_{1:T}, \Theta\right) \\ & - \mathbb{E}_z\left[H(\mathbf{y}_T|\mathbf{x}_{1:T}, z, \Theta)\right]. \end{aligned} \tag{2}$$

The entropy $H\left(\mathbf{y}_T|\mathbf{x}_{1:T}, \Theta\right)$ can be approximately calculated as $-\sum_t \left[p(\omega_t^{\mathbf{y}_T}) \cdot \log p\left(\omega_t^{\mathbf{y}_T}\right)\right]$, where $p(\omega_t^{\mathbf{y}_T})$ represents the probability of each possible next token $\omega_t^{\mathbf{y}_T}$ given the input prompt $\mathbf{x}_{1:T}$. Therefore, the AU in Eq. (2) can be approximated by sampling many z (by sampling different sets of demonstrations) to obtain different $\mathbf{y}_T$ conditioning on one set of parameters Θ. The group of $\mathbf{y}_T$ can then be used to approximate the respective entropies for each group of demonstrations $\mathbf{x}_{1:T-1}$:

$$\begin{aligned} & I(\mathbf{y}_T, z|\Theta) \\ & = H\left(\mathbf{y}_T|\mathbf{x}_{1:T}, \Theta\right) - \mathbb{E}_z\left[H(\mathbf{y}_T|\mathbf{x}_{1:T}, z, \Theta)\right] \\ & \approx \sum^{M\times L} H(\mathbf{y}_T) - \frac{1}{M}\sum_{m=1}^{M}\sum_{l=1}^{L}\left[H(\mathbf{y}_T^{\Theta_{m,l}})\right], \end{aligned} \tag{3}$$

where $[\mathbf{y}_T^{\Theta_{m,l}}]$ are obtained corresponding to different demonstrations $[\mathbf{x}_{1:T-1}^1, \ldots, \mathbf{x}_{1:T-1}^L]$, and $[\Theta_1, \ldots, \Theta_M]$ are sampled from $q(\Theta)$. For some LLMs that do not allow sampling different sets of parameters from the learned $q(\Theta)$ as a standard Bayesian Neural Network, we can instead leverage different decoding strategies (e.g., Beam-search or Multinomial sampling) in order to enable stochastic output from LLMs. In addition, since calculating the entropy $H\left(\mathbf{y}_T\right)$ entails to obtain the joint probability of the generated tokens $p(\mathbf{y}_T) = (\omega_1^{\mathbf{y}_T}, \ldots, \omega_T^{\mathbf{y}_T})$, entropy-based method may only be applicable to white-box LLMs.

2.4 Entropy Approximation

In some cases, the generation of LLMs is free-form, which makes the uncertainty estimation for in-context learning is still different from well-studied classification models that have specific labels. Specifically, not only may the LLM not always be able to return an expected answer, but the generated sequence may also consist of placeholder tokens. Calculating the entropy of the whole generated sequence would involve redundant tokens. Therefore, in this work, we propose to approximate the entropy of the output $H(\mathbf{y}_T)$, and the process is summarized in Figure 3.

Given the probability distributions of the generated tokens $p(\mathbf{y}_T)$ for one set of demonstrations, we only select token(s) $\omega_t^{\mathbf{y}_T}$ that directly answer the provided question. Taking the text classification task as an example, LLM is asked to directly output a numerical value standing for a predefined category (e.g., 0: Sadness, 1: Joy, etc.). The probability of the token $\omega_t^{\mathbf{y}_T}$ that represents the numerical value is then leveraged to denote the overall distribution of $p(\mathbf{y}_T)$. We aggregate the answer probabilities from all M decoded sequences and transform them as an answer distribution (as shown in the top right corner in Figure 3). After repeating the process L times, where L corresponds to L different sets of demonstrations, we have a matrix $\mathcal{M}$ recording the answer distributions of choosing different demonstrations and model configurations (as shown in the lower right corner in Figure 3). The EU and AU can then be approximated as:

$$EU = \frac{1}{L}\sum H\left(\sigma(\mathcal{M}_{:,j})\right),$$

$$AU = H\left(\sigma\left(\sum[\mathcal{M}_{:,j}]\right)\right) - \frac{1}{L}\sum H\left(\sigma\left(\mathcal{M}_{:,j}\right)\right),$$

where $\sigma(\cdot)$ normalizes the column $\mathcal{M}_{:,j}$ into a probability distribution, and entropy $H(\cdot)$ can be calculated as $-\sum_{k=1}^{K}\left(p(\mathcal{M}_{k,j}) * \log(p(\mathcal{M}_{k,j}))\right)$ if the number of labels is K.

3 Related Works

Uncertainty Quantification and Decomposition. Uncertainty quantification aims to measure the confidence of models' predictions, which has drawn attention from various domains (Zhao et al., 2020; Ling et al., 2022; Malo et al., 2014). Measuring uncertainty is essential in many real-world NLP applications where making a wrong prediction with high confidence can be disastrous (e.g., assessing the confidence in a translation or a generated piece of information). This is especially important in foundation models since we do not have enough resources to finetune the model (Abdar et al., 2021). To better understand the uncertainty, the primary focus is on understanding and categorizing the sources of uncertainty for interpreting the models' outputs more effectively. The output uncertainty can typically be categorized into *Aleatoric Uncertainty* that arises from the inherent noise in the data, and *Epistemic Uncertainty* that arises due to inappropriate model architecture or overfitted/underfitted parameters. Existing methods (Chowdhary and Dupuis, 2013; Depeweg et al., 2017; Malinin and Gales, 2020) have come up with various methods (e.g., Bayesian neural network, Deep Ensembles, and Monte Carlo Dropout) to decompose the uncertainty.

Uncertainty in Language Models. Earlier works (Xiao and Wang, 2019; Desai and Durrett, 2020; Jiang et al., 2021) on uncertainty in language models have focused on the calibration of classifiers (e.g., applying dropout to the model parameters or leveraging ensemble voting) to better assess the confidence of the generated output. When it comes to the era of LLMs, multiple works (Xiao and Wang, 2021; Xiao et al., 2022; Lin et al., 2022; Yu et al., 2022; Lin et al., 2023; Kuhn et al., 2023; Fadeeva et al., 2023) have been proposed to measure the uncertainty of LLM's prediction in multiple aspects (e.g., lexical uncertainty, text uncertainty, and semantic uncertainty) for multiple NLP tasks. Another line of works (Kadavath et al., 2022; Zhou et al., 2023; Amayuelas et al., 2023) instead tries to analyze how to extract knowledge from a language model correctly and self-evaluate the correctness with a confidence score. However, despite these commendable efforts, existing methods still lack an effective way to directly quantify and decompose the uncertainty inherent in the outputs of LLMs with in-context learning.

4 Experiments

We evaluate the uncertainty decomposition procedure on realistic natural language understanding problems. By comparing state-of-the-art uncertainty quantification methods, we aim to examine what type of uncertainty is the most promising indicator of high confidence for LLMs. In addition, we also provide generalization analysis and two specific out-of-distribution detection applications. Due to the space limit, extra experiments and experiment settings are provided in the Appendix.

4.1 Experiment Setup

We evaluate the decomposed uncertainties on open-source LLMs with different model sizes. We leverage LLAMA-2 (Touvron et al., 2023), which is the most widely applied open LLM, with its 7B, 13B, and 70B model sizes. The primary experiments are conducted with LLAMA-2 models. In order to further demonstrate the generalization ability of our method, we apply our uncertainty quantification method on OPT-13B (Zhang et al., 2022).

Data. We consider different Natural Language Understanding tasks. *1) Sentiment Analysis*: EMOTION (Saravia et al., 2018) contains $2,000$ test cases and six classes; Financial Phrasebank (Financial) (Malo et al., 2014) contains 850 financial news and three sentiment classes; Stanford Sentiment Treebank v2 (SST2) (Socher et al., 2013) consists of 872 sentences from movie reviews and two classes. *2) Linguistic Acceptability.* The Corpus of Linguistic Acceptability (COLA) (Warstadt et al., 2019) is about English acceptability judgments, which has $1,040$ test cases and two classes. *3) Topic Classification.* AG_News (Zhang et al., 2015) contains $1,160$ test cases and four classes.

Demonstration & Model Configuration Sampling. We evaluate each method on the testing set of each dataset and choose two strategies to randomly sample in-context learning demonstrations. 1) *Random*: we randomly sample demonstrations (training instances with labels) from the training set regardless their labels. 2) *Class*: we randomly sample demonstrations but ensure there is at least one demonstration per label class. To generate various sequences based on one set of demonstrations, we adopt Beam Search with beam width $= 10$ to approximate the sampling process of $\Theta \sim q(\Theta)$.

Comparison Methods. Our study also evaluates the following baseline uncertainty estimation methods: 1) *Likelihood-based Uncertainty* (Likelihood) (Malinin and Gales, 2020) calculates the sum of log probabilities of all tokens generated from language models and normalizes it by the sequence length. 2) *Entropy-based Uncertainty* (Entropy) (Xiao and Wang, 2019) calculates the entropy of the probability distributions of the generated tokens. 3) *Semantic Uncertainty* (Semantic) (Kuhn et al., 2023) is the most advanced entropy-based uncertainty estimation method, which groups generated sequences into clusters according to their semantic embeddings. The average entropy across all groups is viewed as the uncertainty score.

Evaluation Metrics. We show the prediction accuracy of each dataset. In addition, we leverage two standard metrics: the Area under Precision-Recall Curve (AUPR) and AUROC (ROC) to evaluate the uncertainty. AUPR calculates the area under the Precision-Recall curve. AP is high when both precision and recall are high, and low when either of them is low across a range of confidence thresholds. ROC represents the likelihood that a correct answer is selected. An ideal ROC rating is 1, whereas a random uncertainty estimate would yield ROC $= 0.5$.

4.2 Quantitative Analysis

We compare the performance of different methods in assessing the misclassification samples based on their perspective uncertainty scores. Intuitively, misclassified samples should have larger uncertainty scores. The results are shown in Table 1. Note that our proposed method can decompose the uncertainty into epistemic uncertainty (EU) and aleatoric uncertainty (AU), we thus show the performance of EU and AU separately.

As shown in the table, in most cases, our proposed methods (EU and AU) consistently show higher AUPR and ROC scores across all datasets, which indicates a better performance in assessing misclassification samples based on uncertainty scores. Moreover, we also draw some observations from the table. *1. Class Sampling Strategy Proves Superior*: The class sampling strategy generally yields higher AUPR and ROC scores across datasets, which proves it is more effective than random demonstration sampling. Class sampling ensures that each class is represented in the sample and reduces sampling bias, which is crucial in scenarios where the dataset might be imbalanced or where certain classes are underrepresented. *2) Increasing Model Size Enhances Performance*: Larger models (moving from 7B to 70B) tend to have better performance in terms of AUPR and ROC. Specifically, there's a general trend of increasing AUPR and ROC scores as model size increases from 7B to 13B to 70B for all comparison methods. Some datasets and metrics do not strictly follow this trend. For instance, in the EMOTION dataset, the 70B model sometimes shows a slight decrease in performance compared to the 13B model. The inconsistencies in performance improvement with larger models, especially for EU, hint at the complexity of uncertainty assessment in different contexts and datasets. *3. Treating all tokens equally can be harmful in uncertainty quantification*: both Likelihood and Entropy Uncertainty treat all tokens equally. However, some tokens carry greater relevance and representativeness than others, owing to the phenomenon of "linguistic redundancy". However, most uncertainty estimation methods treat all tokens with equal importance when estimating uncertainty, disregarding these inherent generative inequalities.

4.3 Generalization Capability

In this work, we also show how our method performs when applied to different LLMs. We compare the performance of misclassification rate when using OPT-13B and LLAMA-2-13B. We com-

	Inference Model	ACC	**Likelihood**		**Entropy**		**Semantic**		**Ours (EU)**		**Ours (AU)**	
			AUPR	ROC	AUPR	ROC	AUPR	ROC	AUPR	ROC	AUPR	ROC
Emotion	LLaMA-7B-Random	0.407	0.423	0.426	0.448	0.501	0.598	0.607	**0.688**	**0.667**	0.625	0.579
	LLaMA-7B-Class	0.411	0.562	0.423	0.657	0.538	0.697	0.653	**0.745**	**0.696**	0.691	0.601
	LLaMA-13B-Random	0.501	0.597	0.613	0.584	0.503	0.612	0.625	**0.645**	**0.681**	0.559	0.585
	LLaMA-13B-Class	0.533	0.641	0.578	0.593	0.554	0.652	0.701	**0.622**	**0.686**	0.526	0.599
	LLaMA-70B-Random	0.584	0.512	0.462	0.491	0.452	0.657	0.696	**0.667**	**0.713**	0.531	0.663
	LLaMA-70B-Class	0.592	0.537	0.484	0.469	0.442	0.622	0.689	**0.659**	**0.721**	0.612	0.693
Financial	LLaMA-7B-Random	0.379	0.821	0.532	0.728	0.438	0.715	0.624	**0.731**	**0.672**	0.669	0.582
	LLaMA-7B-Class	0.397	0.593	0.505	0.548	0.362	0.732	0.699	**0.803**	**0.711**	0.753	0.589
	LLaMA-13B-Random	0.476	0.894	0.571	0.652	0.463	0.705	0.545	0.718	0.512	**0.729**	**0.573**
	LLaMA-13B-Class	0.477	0.752	0.594	0.692	0.531	0.694	0.543	**0.765**	**0.610**	0.758	0.592
	LLaMA-70B-Random	0.530	0.816	0.509	0.754	0.493	0.679	0.688	**0.779**	**0.754**	0.734	0.642
	LLaMA-70B-Class	0.537	0.668	0.469	0.623	0.439	0.774	0.649	**0.893**	**0.804**	0.739	0.659
SST-2	LLaMA-7B-Random	0.856	0.149	0.636	0.135	0.587	0.244	0.593	**0.286**	0.683	0.205	**0.702**
	LLaMA-7B-Class	0.897	0.230	0.666	0.196	0.579	0.253	0.577	0.248	**0.701**	**0.302**	0.673
	LLaMA-13B-Random	0.866	0.268	0.472	0.204	0.467	**0.355**	0.712	0.314	0.677	0.326	**0.816**
	LLaMA-13B-Class	0.928	0.178	0.425	0.113	0.439	0.343	0.631	**0.397**	**0.836**	0.367	0.639
	LLaMA-70B-Random	0.932	0.091	0.597	0.137	0.475	0.258	0.565	**0.318**	**0.764**	0.298	0.571
	LLaMA-70B-Class	0.938	0.132	0.552	0.185	0.531	0.312	0.679	0.331	**0.851**	**0.362**	0.697
COLA	LLaMA-7B-Random	0.599	0.388	0.557	0.329	0.443	0.358	0.502	**0.416**	**0.562**	0.377	0.517
	LLaMA-7B-Class	0.639	0.392	0.523	0.381	0.478	0.425	0.526	**0.473**	**0.587**	0.401	0.506
	LLaMA-13B-Random	0.652	0.389	0.498	0.287	0.512	0.433	0.562	0.469	**0.572**	**0.488**	0.565
	LLaMA-13B-Class	0.649	0.412	0.418	0.342	0.517	0.426	0.548	0.456	0.568	**0.523**	**0.641**
	LLaMA-70B-Random	0.826	0.481	0.599	0.312	0.471	0.372	0.625	0.317	**0.716**	**0.329**	0.676
	LLaMA-70B-Class	0.852	0.357	0.612	0.397	0.588	0.397	0.613	0.389	**0.727**	**0.425**	0.682
AG_News	LLaMA-7B-Random	0.646	0.238	0.472	0.265	0.463	0.312	0.612	**0.448**	**0.634**	0.361	0.537
	LLaMA-7B-Class	0.679	0.267	0.505	0.372	0.523	0.378	0.562	**0.384**	**0.627**	0.326	0.538
	LLaMA-13B-Random	0.685	0.365	0.517	0.364	0.522	0.374	0.548	**0.395**	**0.648**	0.378	0.552
	LLaMA-13B-Class	0.685	0.378	0.528	0.359	0.413	0.411	0.566	**0.429**	**0.654**	0.401	0.569
	LLaMA-70B-Random	0.792	0.311	0.478	0.316	0.498	**0.401**	0.552	0.309	**0.635**	0.319	0.543
	LLaMA-70B-Class	0.838	**0.302**	0.511	0.271	0.528	0.354	0.532	0.274	**0.662**	0.283	0.571

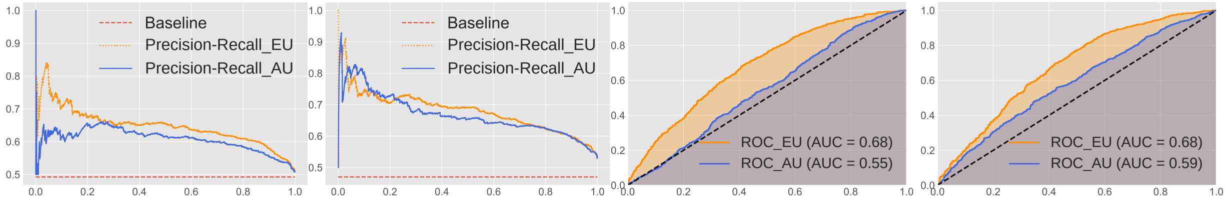
Table 1: The performance comparison on the misclassification rate based on the uncertainty score from each approach. For each dataset, correct predictions are labeled as 0 and incorrect ones are labeled as 1. We show the AUPR and ROC (the higher the better) based on the uncertainty score and misclassification rate with two types of demonstration selection strategy: Random and Class as well as three LLaMA model sizes: 7B, 13B, and 70B.

pute the precision-recall (PR) curve and ROC curve using two backbone LLMs on the Emotion dataset, and the results are shown in Figure 4.

As shown in Figure 4, our method exhibits consistent trends across different LLMs. The precision-recall curves of both uncertainties (Figure 4 (a) and 4 (b)) between the two methods are almost identical, and the model's capability between two LLMs is also reflected in the PR curves of EU. Furthermore, by comparing Figure 4 (c) and 4 (d), the ROC curves of both LLMs show a similar pattern, with the AUC scores not deviating significantly. Specifically, both OPT-13B and LLaMA-2-13B exhibit the same Area Under ROC (AUROC) curve $= 0.68$ for AU. Since LLaMA-2-13B is a more powerful LLM than OPT-13B, our method can quantify that EU of LLaMA-2-13B (AUROC $= 0.59$) is better than OPT-13B (AUROC $= 0.55$). This finding further supports our method maintains its performance irrespective of the underlying model and its robust generalization capability.

4.4 Out-of-domain Demonstration Detection

Out-of-domain (OOD) demonstration refers to coupling a test instance with less relevant or OOD demonstrations, potentially leading the model to be misled and handle the test instance unreliably. In this study, we investigate whether uncertainty scores can effectively distinguish between in-domain and OOD demonstrations. In our labeling scheme, in-domain demonstrations are labeled as 0, while OOD demonstrations are labeled as



(a) PR by OPT-13B (b) PR by LLAMA-2-13B (c) ROC by OPT-13B (d) ROC by LLAMA-2-13B

Figure 4: The performance of misclassification rate using two backbone LLMs: OPT-13B and LLAMA-2-13B on EMOTION dataset. (a) and (b) demonstrate the precision-recall curves (x-axis is the recall and y-axis is the precision) for OPT-13B and LLAMA-2-13B; (c) and (d) demonstrate the ROC curve (x-axis is the false positive rate and y-axis is the true positive rate) for OPT-13B and LLAMA-2-13B, respectively.

	Semantic		Ours (EU)		Ours (AU)	
	AUPR	ROC	AUPR	ROC	AUPR	ROC
Relevant Demo	0.702	0.644	**0.742**	**0.935**	0.657	0.682
OOD Demo	0.698	0.712	**0.784**	**0.941**	0.773	0.607

Table 2: Out-of-domain demonstration detection conducted with LLAMA-2-13B on EMOTION Dataset.

1. AUPR and ROC analyses are performed based on the labels and uncertainty scores, with results summarized in Table 2. Specifically, we conduct experiments on the EMOTION dataset, involving two scenarios: in-domain demonstrations (sampled from its training set) and relevant demonstrations (sampled from Finance Phrasebank, a three-class sentiment analysis task). Additionally, we compare in-domain demonstrations with complete OOD demonstrations (sampled from COLA, a binary linguistic acceptability task).

As shown in Table 2, compared to the state-of-the-art Semantic Uncertainty and the AU, the EU demonstrates the best indicator to detect both less relevant and OOD demonstrations. Intuitively, the model's predictions would be impacted by irrelevant and OOD demonstrations and exhibit large variance. AU is less effective than EU in detecting OOD demonstrations since the demonstrations already have large inherent variability. Semantic Uncertainty instead cannot really distinguish what is the root cause of the predictive uncertainty.

4.5 Semantic Out-of-distribution Detection

Semantic out-of-distribution (SOOD) detection refers to distinguishing test samples with semantic shifts from the given demonstrations and the prompt. In this study, we mask out a few classes and ask LLMs to classify test samples into the rest of the classes. The method is expected to return a higher uncertainty score of SOOD test samples.

	Semantic		Ours (EU)		Ours (AU)	
	AUPR	ROC	AUPR	ROC	AUPR	ROC
7B	0.477	0.532	**0.548**	**0.658**	0.461	0.570
13B	0.417	0.468	**0.525**	**0.592**	0.414	0.437

Table 3: Semantic out-of-distribution detection using LLAMA-2 7B and 13B on EMOTION Dataset.

Specifically, we mask two classes *1: sadness* and *2: anger* out of six classes from the EMOTION dataset and ask LLM to categorize a given test sample only into the rest four classes. The SOOD samples are labeled as 1 and in-distribution samples are labeled as 0. Results of AUPR and ROC are recorded in Table 3 in terms of different model sizes.

As shown in the table, EU still performs the best as a better indicator to recognize SOOD samples across different model sizes. SOOD samples are semantically different from the provided demonstrations, and the task description also masks out the correct class of these SOOD samples, leading to higher uncertainty in the model's predictions. Given the inappropriate task description and demonstrations, AU may not necessarily perform better in the presence of SOOD samples.

5 Conclusion

To better understand and quantify the inherent uncertainties associated with LLM's in-context learning, we provide a novel approach to decompose the predictive uncertainty into its aleatoric and epistemic perspectives from the Bayesian perspective. We also provide novel approximation methods to quantify different uncertainties based on the decomposition. Extensive experiments are conducted to verify the effectiveness and better performance of the proposed method than others. We believe this research stands as a significant stride toward harnessing the full potential of LLMs while being acutely aware of their performance boundaries.

Broader Impact

The broader impact of this work can be considered from several perspectives, particularly in building a trustworthy LLM environment. Specifically, by decomposing uncertainty into aleatoric and epistemic components, this work can significantly contribute to increasing the reliability and trustworthiness of LLMs. Users and developers can better understand when and why an LLM might fail or provide inaccurate responses. This understanding is crucial for critical applications where reliability is paramount, such as in healthcare, legal advice, or educational tools.

Limitations

The proposed work aims at quantifying predictive uncertainty and decomposing the value into its aleatoric and epistemic components. While we can achieve the best result compared to other methods, the proposed framework may only be applied in natural language understanding tasks (e.g., multiple-choice QA, text classification, linguistics acceptability, etc.). The proposed uncertainty estimation algorithm may have limited usage in quantifying uncertainties of generation tasks since we cannot tell which part of the generated sequence is semantically important.

References

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

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

Kamaljit Chowdhary and Paul Dupuis. 2013. Distinguishing and integrating aleatoric and epistemic variation in uncertainty quantification. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 47(3):635–662.

Stefan Depeweg, José Miguel Hernández-Lobato, Finale Doshi-Velez, and Steffen Udluft. 2017. Uncertainty decomposition in bayesian neural networks with latent variables. *arXiv preprint arXiv:1706.08495*.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. *arXiv preprint arXiv:2003.07892*.

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

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

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

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

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Zi Lin, Jeremiah Zhe Liu, and Jingbo Shang. 2022. Towards collaborative neural-symbolic graph semantic parsing via uncertainty. *Findings of the Association for Computational Linguistics: ACL 2022*.

Chen Ling, Junji Jiang, Junxiang Wang, and Zhao Liang. 2022. Source localization of graph diffusion via variational autoencoders for graph inverse problems. In *Proceedings of the 28th ACM SIGKDD*, pages 1010–1020.

Chen Ling, Xujiang Zhao, Jiaying Lu, Chengyuan Deng, Can Zheng, Junxiang Wang, Tanmoy Chowdhury, Yun Li, Hejie Cui, Xuchao Zhang, Tianjiao Zhao, et al. 2023a. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv preprint arXiv:2305.18703*.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Yanchi Liu, Wei Cheng, Haoyu Wang, Zhengzhang Chen, Takao Osaki, Katsushi Matsuda, Haifeng Chen, et al. 2023b. Improving open information extraction with large language models: A study on demonstration uncertainty. *arXiv preprint arXiv:2309.03433*.

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

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. 2014. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the*

Association for Information Science and Technology, 65.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*.

Vipula Rawte, Amit Sheth, and Amitava Das. 2023. A survey of hallucination in large foundation models. *arXiv preprint arXiv:2309.05922*.

Elvis Saravia, Hsien-Chi Toby Liu, Yen-Hao Huang, Junlin Wu, and Yi-Shin Chen. 2018. CARER: Contextualized affect representations for emotion recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3687–3697.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

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

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7322–7329.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. *arXiv preprint arXiv:2103.15025*.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. *arXiv preprint arXiv:2210.04714*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Jialin Yu, Alexandra I Cristea, Anoushka Harit, Zhongtian Sun, Olanrewaju Tahir Aduragba, Lei Shi, and Noura Al Moubayed. 2022. Efficient uncertainty quantification for multilabel text classification. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Xujiang Zhao, Feng Chen, Shu Hu, and Jin-Hee Cho. 2020. Uncertainty aware semi-supervised learning on graph data. *Advances in Neural Information Processing Systems*, 33:12827–12836.

Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. 2023. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv preprint arXiv:2302.13439*.

A Appendix

A.1 Variance-based Decomposition

Alternatively, we can use the variance as a measure of uncertainty. Let $\sigma^2(\cdot)$ compute the variance of a probability distribution, and the total uncertainty present in Eq. (1) is then $\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T})$. This quantity can then be decomposed using the law of total variance:

$$\begin{aligned}\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T}) =&\sigma^2_{q(\Theta)}\left(\mathbb{E}[\mathbf{y}_T|\mathbf{x}_{1:T},\Theta]\right) \\ &+\mathbb{E}_{q(\Theta)}\left[\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T},\Theta)\right].\end{aligned} \tag{4}$$

where $\mathbb{E}[\mathbf{y}_T|\mathbf{x}_{1:T},\Theta]$ and $\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T},\Theta)$ are mean and variance of $\mathbf{y}_T$ given $p\left(\mathbf{y}_T|\mathbf{x}_{1:T},\Theta\right)$. $\sigma^2_{q(\Theta)}\left(\mathbb{E}[\mathbf{y}_T|\mathbf{x}_{1:T},\Theta]\right)$ represents the variance of $\mathbb{E}[\mathbf{y}_T|\mathbf{x}_{1:T},\Theta]$ when $\Theta\sim q(\Theta)$, which indicates the epistemic uncertainty since it ignores the contribution of z. In contrast, $\mathbb{E}_{q(\Theta)}\left[\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T},\Theta)\right]$ in Eq. (4) represents the aleatoric uncertainty since it denotes the average value of $\sigma^2(\mathbf{y}_T|\mathbf{x}_{1:T},\Theta)$ with $\Theta\sim p(\Theta)$ and ingores the contribution of Θ to $\mathbf{y}_T$. Note that variance-based uncertainty decomposition does not involve the probability of the generated tokens, which is applicable to black-box LLMs (e.g., GPT models).

Variance Approximation. In practice, when we are dealing with black-box LLMs (e.g., ChatGPT), there are multiple hyperparameters (e.g., `temperature` and `top_p`) allowing to return different responses. Specifically, $[\mathbf{y}_T^1, \dots, \mathbf{y}_T^L]$ can be obtained through querying the LLM with different demonstrations $[\mathbf{x}_{1:T-1}^1, \dots, \mathbf{x}_{1:T-1}^L]$ L times. The different set of parameter configurations are denoted as $[\Theta_1, \dots, \Theta_M]$. The $\mathbb{E}[\mathbf{y}_T|\mathbf{x}_{1:T}, \Theta]$ can then be calculated as the expected model output given the input data and the model parameters Θ. Calculate the variance of this expectation with respect to a set of model configurations over all sets of demonstrations gives the epistemic uncertainty. The variance $\sigma^2(\mathbf{y}_T)$ can also be obtained given a set of few-shot demonstrations over all model parameters. Finally, average this variance over the certain model configuration to obtain the aleatoric uncertainty.

A.2 Dataset Description

Sentiment Analysis. 1) EMOTION (Saravia et al., 2018) contains 2,000 test cases, where LLMs are asked to classify a given sentence with six categories: *sadness*, *joy*, *love*, *anger*, *fear*, *surprise*. 2) Financial Phrasebank (Financial) (Malo et al., 2014) contains 850 test cases, where LLMs are asked to classify a given financial news with three categories: *negative*, *neutral*, *positive*. 3) Stanford Sentiment Treebank v2 (SST2) (Socher et al., 2013) consists of 872 sentences from movie reviews and human annotations of their sentiment, where the language model is asked to predict the sentiment from two classes: *positive* and *negative*.

Linguistic Acceptability. 1) The Corpus of Linguistic Acceptability (COLA) (Warstadt et al., 2019) is about English acceptability judgments drawn from books and journal articles on linguistic theory. Each example is a sequence of words annotated with whether it is a grammatical English sentence, and there are 1,040 test cases in total.

Topic Classification. TC aims at categorizing the given sentence into predefined topics. We adopt AG_News (Zhang et al., 2015) is a dataset that collects more than 1 million news articles, where LLMs are asked to classify a given news into four categories: *World*, *Sports*, *Business*, and *Sci/Tech*. There are 1,160 test cases in total.

A.3 Experiment Setup

We conduct experiments primarily on LLAMA-2-7B-CHAT-HF, LLAMA-2-13B-CHAT-HF, and LLAMA-2-70B-CHAT-HF, where the model weights are downloaded from the website[1]. Since we cannot actually "sample" model weights as Bayesian Neural Networks, in order to let LLMs return different outputs, we leverage Beam Search since it considers multiple best options based on beam width using conditional probability, which is better than the sub-optimal Greedy search. The beam search is conducted with the beam size 10 and the max number of new tokens is set to be 16 uniformly across all datasets. We choose a different number of demonstrations (details are recorded in Table 4) to allow the LLM to achieve the best performance on each dataset, and we sample demonstrations four times uniformly across all datasets.

	Random	Class
Emotion	6	1 per class
Financial	6	2 per class
SST2	4	2 per class
COLA	2	1 per class
AG_News	4	1 per class

Table 4: The number of demonstrations selected in each dataset.

A.4 Prompt Template

In this work, we uniformly apply the following prompt template for all datasets. Take the EMOTION dataset as an example, we summarize the prompt in Table 5. Note that all datasets use the same template, small modifications are made on the actual label information and different demonstration numbers of different datasets.

A.5 Misclassification Rate with Out of Domain Demonstration

Out-of-domain in-context Demonstration refers to the test instance being coupled with less relevant or out-of-domain demonstrations, which the model may be misled and not handle the test instance reliably. In this work, we analyze the misclassification rate of out-of-domain Demonstration in the EMOTION dataset (six-class sentiment analysis task) by providing LLMs with relevant demonstrations (sampled from Finance Phrasebank which is a three-class sentiment analysis task) and complete out-of-domain demonstrations (sampled from

[1]https://ai.meta.com/resources/models-and-libraries/llama-downloads/

System Prompt	### Below is an instruction that describes a task. Clearly follow the instruction and write a short response to answer it.
Task Description	### Instruction: Classify the sentiment in the following text based on the six categories: [0: Sadness; 1: Joy, 2: Love; 3: Anger; 4: Fear, 5: Surprise]. Provide the information in a structured format WITHOUT additional comments, I just want the numerical label for each text.
Demonstrations	### Here are some examples: Example 1: Sentence: {i didnt feel humiliated} Category: {0: Sadness} Example 2: Sentence: {im grabbing a minute to post i feel greedy wrong} Category: {3: anger} Example 3: Sentence: {i have the feeling she was amused and delighted} Category: {1: joy} Example 4: Sentence: {i feel more superior dead chicken or grieving child} Category: {1: joy} Example 5: Sentence: {i get giddy over feeling elegant in a pencil skirt} Category: {1: joy} ...
Test Query	### Test Sentence: { } Category:

Table 5: Prompt Template consists of four parts: 1) *System Prompt* aims at providing a basic hint of the task; 2) *Task Description* provides some details of the task, e.g., if it is a sentiment analysis task or how many labels are there; 3) *Few-shot Demonstrations* are leveraged to give LLMs some basic formats of how the returned responses can be constructed; and 4) *Test Query* is the final test query that we want LLMs to classify/categorize, and the LLM is only expected to return an exact answer to solve the given question.

COLA which is a binary linguistic acceptability task). We conduct the task with two demonstration selection strategies, and the results are provided in Table 6.

	LLaMA-13B-Random		LLaMA-13B-Class	
	EU	AU	EU	AU
Original Demo	0.681	0.585	0.686	0.599
Relevant Demo	0.688 (+1.0%)	0.541 (−7.5%)	0.671 (−2.2%)	0.524 (−12.5%)
OOD Demo	0.671 (−1.4%)	0.501 (−13.3%)	0.673 (−1.8%)	0.497 (−17.0%)

Table 6: Comparison of AUROC in misclassificatin rate on EMOTION dataset, where "Original Demo" indicates we sample demonstrations from its original training set, "Relevant Demo" indicates we sample demonstrations from Finance Phrasebank Dataset (a relevant sentiment analysis task, and "OOD Demo" indicates we sample demonstrations from an irrelevant dataset: **COLA**.

As shown in the table, changes in the performance of the EU are relatively minor under all conditions, suggesting that the model is more stable or less sensitive to the changes in demonstration data within this metric. In contrast, the AU shows more significant fluctuations, which implies that the AU is more sensitive to the quality and relevance of demonstration data. When relevant demonstrations from the Finance Phrasebank sentiment analysis dataset are used, there's a slight improvement or a minor decrease in EU, but a notable decrease in AU. This suggests that even relevant but not identical data can confuse the model, especially for the AU. With out-of-domain demonstrations from COLA, the model's performance drops more significantly, with the AU metric showing a dramatic decrease of up to 17.0%, which indicates that the model struggles significantly when the demonstrations are not relevant to the task it's being tested on.

A.6 Case Study

Table 7 demonstrates the actual changes in AU and EU when presenting LLMs with different sizes and different demonstrations. Given the test query is: *I had stated to her the reason I feel so fearful is because I feel unsafe* with the ground truth label is *(4: fear)*, which is a sentence with negative feeling. For LLAMA-2-7B, by presenting LLMs with more diverse demonstrations (contain both positive and negative sentences), the results would be more diverse between different beam search returned sequences, leading to a relatively higher AU than EU. For LLAMA-2-70B with a lerger parameter space and model capability,

Testing Query: I had stated to her the reason I feel so fearful is because I feel unsafe (4: fear)		Extracted Predictions	EU	AU
LLaMA-2-7B	1. i felt anger when at the end of a telephone call (3: anger) 2. i feel a little mellow today (1: joy) 3. i don t feel particularly agitated (4: fear) 4. i hate it when i feel fearful for absolutely no reason (4: fear) 5. im updating my blog because i feel shitty (0: sadness)	0, 0, 0, 1, 3 4, 3, 4, 4, 4	0.171	0.372
	1. i am feeling outraged it shows everywhere (4: fear) 2. i do feel insecure sometimes but who doesnt (4: fear) 3. i start to feel emotional (0: sadness) 4. i feel so cold a href http irish (3: anger) 5. i feel i have to agree with her even though i can imagine some rather unpleasant possible cases (0: sadness)	4, 4, 1, 3, 4 4, 4, 4, 5, 4	0.163	0.189
LLaMA-2-70B	1. i felt anger when at the end of a telephone call (3: anger) 2. i feel a little mellow today (1: joy) 3. i don t feel particularly agitated (4: fear) 4. i hate it when i feel fearful for absolutely no reason (4: fear) 5. im updating my blog because i feel shitty (0: sadness)	4, 3, 4, 3, 4 4, 4, 2, 4, 4	0.012	0.079
	1. i am feeling outraged it shows everywhere (4: fear) 2. i do feel insecure sometimes but who doesnt (4: fear) 3. i start to feel emotional (0: sadness) 4. i feel so cold a href http irish (3: anger) 5. i feel i have to agree with her even though i can imagine some rather unpleasant possible cases (0: sadness)	4, 4, 4, 4, 4 4, 4, 4, 4, 4	0.004	0.009

Table 7: Case study on the actual EU and AU decomposed from the predictive uncertainty