Towards Harmonized Uncertainty Estimation for Large Language Models

Anonymous ACL submission

Abstract

To facilitate robust and trustworthy deployment of large language models (LLMs), it is essential to quantify the reliability of their generations through uncertainty estimation. While recent efforts have made significant advancements by leveraging the internal logic and linguistic features of LLMs to estimate uncertainty scores, our empirical analysis highlights the pitfalls of these methods to strike a harmonized estimation between *indication*, *balance*, and *calibration*, which hinders their broader capability for accurate uncertainty estimation. To address this challenge, we propose CUE💡 (Corrector for Uncertainty Estimation): A straightforward yet effective method that employs a lightweight model trained on data aligned with the target LLM's performance to adjust uncertainty scores. Comprehensive experiments across diverse models and tasks demonstrate its effectiveness, which achieves consistent improvements of up to 60% over existing methods.

1 Introduction

Uncertainty is the only certainty there is.
- by John Allen Paulos

Large Language Models (LLMs) have demonstrated exceptional capabilities in handling a wide range of downstream tasks (OpenAI, 2023; Touvron et al., 2023a,b; Dubey et al., 2024). They are gradually adopted as general-purpose API interfaces (e.g., ChatGPT[1]), providing valuable services and assistance in human life. Despite these impressive advancements, concerns persist regarding the tendency of LLMs to generate hallucinations and factual inaccuracies with confidence (Zhang et al., 2023; Wachter et al., 2024), which may mislead users to overestimate the reliability of the information provided by these models. To mitigate this issue, uncertainty estimation (Loquercio et al., 2020) proposed quantifying the reliability of model outputs so as to ensure the robustness and trustworthiness of AI-driven services.

Harmonized uncertainty estimation is expected to encompass three key aspects: **1) Indication**. The uncertainty score should clearly reflect the reliability of model responses, with higher scores signaling potential inaccuracies. This can be framed as a classification task, with "reliable" or "unreliable" as the classes. **2) Balance**. Within classification framework, it's critic to strike a balance between recall and precision, ensuring that challenging cases are appropriately flagged while minimizing the resources spent on false positives. **3) Calibration**. The uncertainty score should align with human intuition and probabilistic expectations, to facilitate effective calibration. By striking a harmonized balance between these three aspects, uncertainty estimation provides an ideal measure of the models reliability, offering both usability and interpretability.

There has been growing interest in developing uncertainty estimation methods tailored for LLMs. However, with a thorough analysis across diverse uncertainty estimation methods, we found that there still remains a large performance gap between existing methods to achieve the harmonized uncertainty estimation. Specifically, methods that excel in one aspect fall short in others. For instance, SAR (Duan et al., 2023), the outstanding and state-of-the-art method in the specific dataset SciQA (Auer et al., 2023), achieves the best performance in *indication* but performs poorly in the view of *calibration*. Furthermore, we found that the combination of uncertainty scores obtained by existing methods provides little improvement in uncertainty estimation performance, suggesting that these methods are quite homogeneous. These findings highlight considerable room for refinement in uncertainty estimation.

In this paper, we introduce CUE💡, a simple

[1] https://chat.openai.com

yet effective framework for adjusting uncertainty scores, which is orthogonal to existing uncertainty estimation methods. Specifically, we begin by curating dataset that are closely aligned with the target LLM's performance within a particular domain of knowledge. This dataset is then utilized to train an auxiliary lightweight model, which serves as a *Corrector* to adjust the uncertainty scores. By integrating the *Corrector* trained on global alignment information with those uncertainty estimation methods that rely solely on the intrinsic logic and linguistic features of LLMs, we can significantly refine the uncertainty scores.

Our main contributions are thus as follows:

- According to an empirical analysis of existing uncertainty estimation methods from both classification and calibration views, we found there is substantial room for improvement in their performance regarding classification indication, precision-recall balance, and calibration.
- We propose CUE 💡, an uncertainty score correction framework that employs a classifier, aligned with the model's task performance, as a *Corrector* to adjust uncertainty scores. This *Corrector* allows for seamless integration with existing uncertainty estimation methods.
- Extensive experiments demonstrate that our CUE 💡 consistently enhances various existing uncertainty estimation methods, showing significant improvements in a harmonized manner across diverse data domains and target models.

2 Related Work

2.1 Uncertainty Estimation for LLMs

As illustrated in Figure 1, uncertainty estimation methods for LLMs can be broadly categorized into logit-based methods, verbalized methods, consistency-based methods and internal state-based methods.

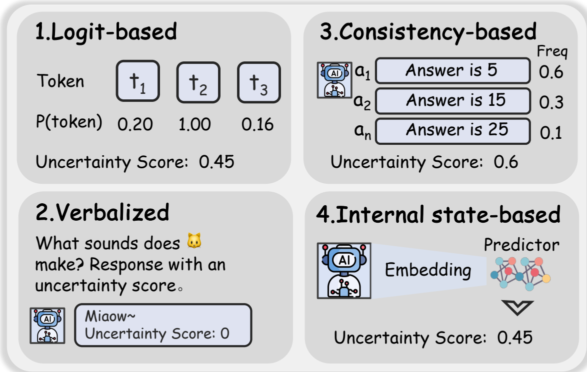

Figure 1: A concise overview figure of various uncertainty estimation method categories, including logit-based methods, verbalized methods, consistency-based methods, and internal state-based methods.

Logit-based methods are the most widely used and effective approaches in uncertainty estimation. Predictive Entropy (PE) (Malinin and Gales, 2020) defined uncertainty as the entropy of the output logits distribution, which is widely adopted and built upon in subsequent research. Follow that, Kuhn et al. (2023) introduced semantic entropy (SE) that estimates uncertainty by marginalizing over semantically-equivalent samples in NLG tasks. Duan et al. (2023) proposed Shifting Attention to Relevance (SAR), which focus on relevant information and assigns significance weights to tokens based on their contributions to the overall response. Yaldiz et al. (2024) introduced a Learnable Response Scoring Function (LARS), which utilizes supervised data to capture complex token-probability dependencies.

Verbalized methods (Xiong et al., 2023; Groot and Valdenegro-Toro, 2024) leverage LLMs' strong language and instruction-following abilities to express uncertainty, often by prompting the model to provide an uncertainty score. However, studies (Ni et al., 2024; Madhusudhan et al., 2024; Becker and Soatto, 2024) have shown that LLMs struggle with faithfully conveying their uncertainties, particularly due to overconfidence. **Consistency-based methods**, such as those proposed by Li et al. (2024b) and Becker and Soatto (2024), assess uncertainty through multiple generated answers, using techniques like perturbation and aggregation to improve reliability. Pedapati et al. (2024) further reduced overconfidence by guiding LLMs to justify their answers. **Internal state-based methods** (Azaria and Mitchell, 2023; Liu et al., 2024) analyze LLM activations to predict errors, with Kadavath et al. (2022) and Ji et al. (2024) exploring self-evaluation and probing estimators to enhance uncertainty estimation.

Due to space limitations, a more detailed discussion of related work is provided in the Appendix A.1.

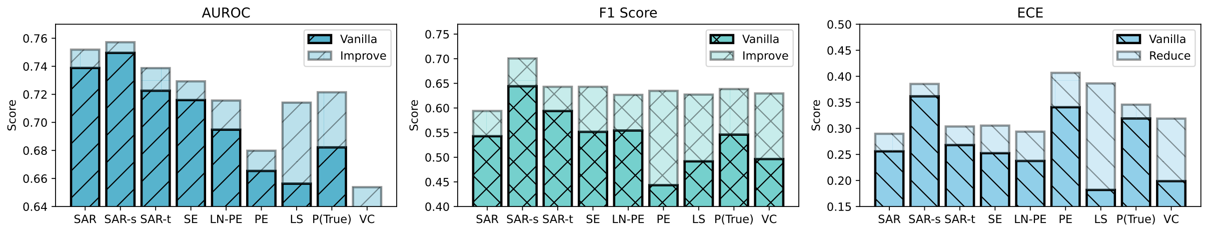

Figure 2: The performance of existing uncertainty estimation methods, evaluated on the SciQA dataset with the LLaMA-3-8B-Instruct model as the target, and the improvements after applying the *Corrector*. Note that a lower ECE score indicates better performance, so we report its reduction.

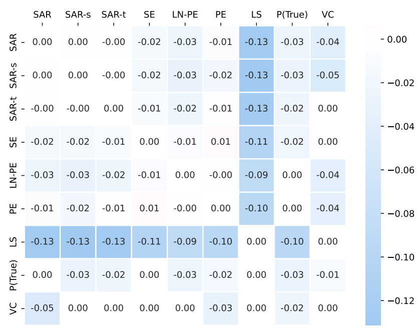

Figure 3: AUROC improvement across uncertainty scores combination from different existing methods.

3 Preliminary Study

3.1 Limitation of Existing UE Methods

We evaluate existing UE methods from both classification and calibration views, focusing on three key aspects of uncertainty scores: **indication, balance, and calibration**. From the classification view, uncertainty scores are utilized to guide the classification process. Instances with scores above a threshold are classified as c_1 (unreliable) and those below as c_0 (reliable). We employ AUROC to measure how well the scores indicate unreliability and F1 score to evaluate their balance between precision and recall. The calibration view involves a more rigorous assessment and interpretation of uncertainty scores. Well-calibrated scores should align with human probabilistic intuition and provide more precise instance rankings. We use ECE to assess calibration.

Basic methods exhibit poor indication performance. Firstly, we focus on representative but naive methods including Lexical Similarity (LS) (Fomicheva et al., 2020), Verbal Confidence (VC) (Xiong et al., 2023), P(true) (Kadavath et al., 2022), and Predictive Entropy (PE) (Malinin and Gales, 2020) that belong to four categories: consistency-based methods, verbal confidence methods, internal state-based methods, and logit-based methods, respectively. As shown in Figure 2 and Table 1, the AUROC scores for these methods across the target models and datasets exhibit general low performance, which is even close to random guessing.

Enhanced logit-based methods typically have low F1 scores. Some enhanced methods such as Length-normalized Predictive Entropy (LN-PE) (Malinin and Gales, 2020), SAR-t, SAR-s, SAR, and Semantic Entropy (SE) (Kuhn et al., 2023), make tailored adjustments to refine predictive entropy process, which show improvements over PE in terms of AUROC. However, no one is universally optimal for all target models and datasets. Moreover, as depicted in Figure 2 and Figure 5, the F1 scores of those methods are particularly low. This indicates that although those methods provided uncertainty scores with some potential to indicate the reliability of model response, they still fall short in strike a balance between precision and recall.

Most existing methods fall short in calibration. As shown in Table 1 and Figure 6, it appears that prior methods have overlooked the calibration aspect, resulting in relatively poor performance in terms of ECE scores.

3.2 Inter-method Cooperation

We examined whether the uncertainty scores derived from one uncertainty estimation method could refine the scores obtained from another method. Specifically, we integrated the uncertainty scores from each method using the weighted combination and compared its performance with the top-performing method in the pair. As illustrated in Figure 3, these integrations do not enhance overall performance and may lead to a decline. This underscores the limitations in the complementary

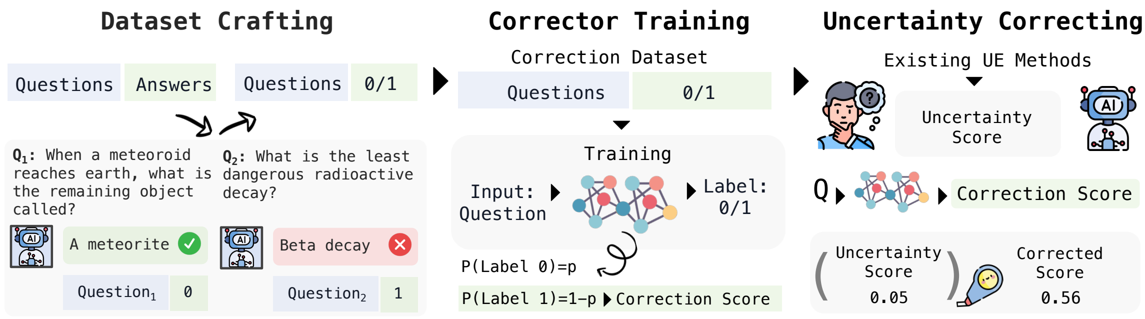

Figure 4: An overview of uncertainty score correction framework. Firstly, we construct a dataset that closely aligns with the target model's performance. This dataset is then utilized to train a lightweight auxiliary model that serves as a correction module, enabling seamless integration with existing uncertainty estimation methods to produce corrected uncertainty scores.

nature of existing methods.

Our analysis reveals a significant performance gap among existing methods in achieving harmonized uncertainty estimation, as individual methods excel in specific aspects but underperform in others. Furthermore, combining uncertainty scores from different methods yields minimal to no improvement, underscoring their homogeneous and non-complementary nature.

4 Method

In this section, we introduce CUE 💡, a correction framework featuring an intuitive approach to directly optimized for uncertainty estimation, where a *Corrector* is trained using a lightweight model to refine the uncertainty score. Through this method, we provide a more robust solution for uncertainty estimation. As shown in Figure 4, Our method comprises three main steps including *dataset crafting*, *corrector training* and *uncertainty correcting*.

4.1 Dataset Crafting

We begin by extracting data from existing datasets to create an evaluation set for assessing the target model M's performance in a specific domain. This set consists of a collection of question-answer pairs, denoted as $\mathcal{D} = \{(q_i, a_i) \mid i = 1, \ldots, n\}$. We then prompt M to generate responses r_i for each question q_i, forming a response set $\mathcal{R} = \{r_i \mid i = 1, \ldots, n\}$. Subsequently, each response r_i is subjected to a rigorous evaluation against the ground truth a_i, employing a hybrid approach that combines both rule-based and LLM-based methods. The rule-based method compares response r_i to the ground truth a_i using the longest common subsequence (LCS). A response r_i is considered equivalent to a_i only if its ROUGE-L score, computed as ROUGE-L$(r_i, a_i) = \frac{\text{LCS}(r_i, a_i)}{\min(\text{len}(r_i), \text{len}(a_i))}$, is greater than threshold value, formalized as $\mathcal{M}_{Rule}(r_i, a_i) = \mathbb{I}_{\text{RougeL}(r_i,a_i)>0.7}$. Additionally, we utilize GPT-turbo-3.5-0613 (Ouyang et al., 2022) to assess the equivalence between r_i and a_i by directly prompting, formalized as $\mathcal{M}_{LLM}(r_i, a_i) = \mathbb{I}_{\text{True in LLM}(r_i,a_i)}$.

To mitigate false positives, we apply rigorous thresholds and strict prompting rules. The final judgment is determined using an "OR" logic: $\mathcal{M}(r_i, a_i) = \mathcal{M}_{\text{Rule}}(r_i, a_i) \vee \mathcal{M}_{\text{LLM}}(r_i, a_i)$, preventing the omission of positive instances.

After that, a binary label c_i is assigned to each sample, defined as

$$c_i = \begin{cases} 1 & \text{if } \mathcal{M}(r_i, a_i) = 1 \\ 0 & \text{if } \mathcal{M}(r_i, a_i) = 0 \end{cases} \tag{1}$$

By pairing question q_i with the label c_i, we form a correction dataset $\mathcal{D}_{\text{cor}} = \{(q_i, c_i) \mid i = 1, \ldots, n\}$, which serves as a representation of the target model's performance in generating correct responses across a particular knowledge domain. To directly associate the questions with uncertainty, we transform the dataset form into $\mathcal{D}^*{}_{\text{cor}} = \{(q_i, 1 - c_i) \mid i = 1, \ldots, n\}$.

4.2 *Corrector* Training

Employing the correction dataset $\mathcal{D}^*{}_{\text{cor}}$, we train a classifier to align with the performance of the target model. Specifically, the classifier integrates a fully connected layer following a lightweight encoder model, such as RoBERTa (Liu, 2019) and Deberta (He et al., 2021a), with the representation of the special token $[CLS]$ as its input, denote as $\text{h}_{[CLS]} \in \mathbb{R}^d$. The output of the classifier is given by $\hat{y}_i = \sigma\left(\mathrm{W} \cdot \mathrm{h}_{[CLS]} + b\right)$, where $\sigma(z)$ is the sigmoid function, used to compute the likelihood y_i that a data point belongs to label c_1. During training, we minimize the binary cross-entropy loss function $\mathcal{L} = -\sum i = 1^N \left[y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)\right]$ across the correction dataset.

This results in a *Corrector*, an auxiliary component that can be integrated with existing uncertainty estimation methods to enhance their reliability.

4.3 Uncertainty Correcting

We derive the probability that an instance x belongs to category c_1 from the *Corrector*. This probability, denoted as the correction score $C(x)$, can be utilized to adjust the uncertainty scores to align with the target model's performance, thereby refining the uncertainty estimation process.

In the refinement process, we first normalize the uncertainty scores generated from existing UE methods to match human probabilistic intuition, ensuring they fall within the range $[0, 1]$. Normalization is achieved via $U_{\text{norm}}(x) = \frac{U(x) - \min(U)}{\max(U) - \min(U)}$, where $U(x)$ represents the uncertainty score for a specific instance x, computed by a chosen UE method. The terms $\min(U)$ and $\max(U)$ denote the minimum and maximum uncertainty scores across the entire dataset, respectively. Following normalization, we apply our correcting by combining the normalized score $U_{\text{norm}}(x)$ with the correction score $C(x)$ generated by the *Corrector*. The combination employs a weighted approach, where the corrected uncertainty score $U_{\text{cor}}(x)$ is computed as:

$$U_{\text{cor}}(x) = w^* \cdot U_{\text{norm}}(x) + (1 - w^*) \cdot C(x) \quad (2)$$

The optimal weight w^* is determined through a grid search on the development dataset. This weighted method ensures that the corrected uncertainty scores balance the contributions of both the original and correction scores, thereby enhancing the reliability of the uncertainty estimation.

5 Experiments

5.1 Experiments Setup

5.1.1 Models

Target models We selected the OPT-6.7B[2] (Zhang et al., 2022), a model widely utilized in previous studies (Kuhn et al., 2023; Duan et al., 2023), and the advanced open-source model LLaMA-3-8B-Instruct[3] (Dubey et al., 2024) as the target models for our main experiments.

Base Models We employed many lightweight encoder model as the base model to train the *Corrector*, including models from the RoBERTa series (Liu, 2019) and DeBERTa series (He et al., 2021a,b).

5.1.2 Metrics

AUROC We use the area under the receiver operating characteristic curve (AUROC) to evaluate uncertainty estimation methods from a classification view. In our setting, an AUROC of 1 signifies perfect indicative performance to distinguish between samples the target model can answer reliably and those it cannot, while an AUROC of 0.5 indicates that the estimation is no better than random guessing.

F1 Score F1 score is used to evaluate the balance between precision and recall in classification tasks. It is the harmonic mean of precision and recall, where both are equally important. The F1 score ranges from 0 to 1, with 1 indicating perfect precision and recall.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

ECE We use Expected Calibration Error (ECE) to evaluate the performance of calibration, which is calculated by partitioning predicted confidence scores into bins and comparing the average confidence in each bin to the actual fraction of correct predictions, formalized as

$$\text{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{n} \left|\text{acc}(B_m) - \text{conf}(B_m)\right| \quad (4)$$

In the computing of ECE, we treat the confidence score as 1 minus uncertainty score.

5.1.3 Datasets

We focus on the question-answering task using two representative datasets in the main experiments: **TriviaQA** (Joshi et al., 2017), and **SciQA** (Auer et al., 2023). TriviaQA comprises 95,000 question-answer pairs created by trivia enthusiasts, supplemented with independently sourced evidence documents. SciQA contains 2,565 question-answer

[2] huggingface.co/facebook/opt-6.7b.

[3] huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct.

pairs fetched from the open research knowledge graph, covering several research fields ranging from science and technology like Computer Science, Engineering, Chemistry, and Geology, life sciences like Immunology and Genetics to social sciences like Economics and Urban Studies.

5.1.4 Baselines

We select a variety of representative uncertainty estimation methods as baselines, with a particular focus on logits-based methods.

Among the baselines, we cover multiple categories, including logit-based, verbalized, internal state-based, and consistency-based methods, including: **Lexical Similarity (LS)** (Fomicheva et al., 2020), which computes the similarity between multiple sentences as a measure of consistency; **Verbal Confidence (VC)** (Xiong et al., 2023), which requires the target model to respond and provide a confidence score; **P(True)** (Kadavath et al., 2022), which first asks the target model to propose an answer and then evaluates it using an internal probability mechanism; and **Predictive Entropy (PE)** (Malinin and Gales, 2020), which calculates uncertainty by measuring the entropy of the predictive posterior.

We also explore a series of advanced logit-based methods including: **Length-normalized Predictive Entropy (LN-PE)** (Malinin and Gales, 2020), which adjusts PE by normalizing it according to sentence length; **Semantic Entropy (SE)** (Kuhn et al., 2023), which clusters sentences with equivalent meanings and calculating cluster-wise entropy; and **Shifting Attention to Relevance (SAR)** (Duan et al., 2023), which encompasses **SAR-t**, **SAR-s** and **SAR**, donated as the token-shifted predictive entropy, sentence-shifted predictive entropy, and both token- and sentence-shifted predictive entropy respectively.

5.1.5 Implementation Details

Dataset Splitting For the TriviaQA dataset, we randomly selected 5,000 samples from the training set for data crafting and corrector training. For datasets with limited data, SciQA, we utilized the entire training set. We then used half of the test set to search for the optimal hyperparameter w, while the other half was employed to evaluate the method's effectiveness.

Hyperparameter For each dataset and model pair, we train a corresponding *Corrector*, which is universally applicable across various methods. Additionally, for every method, dataset, and model combination, we derive the weight using the development set respectively.

5.2 Main Result

We has evaluated existing methods in Section 3 and found that there still remains a large performance gap between existing methods to achieve the harmonized uncertainty estimation. In this part, we present the performance of CUE 💡 from both classification and calibration views, demonstrating that integrating a *Corrector* with existing UE methods significantly enhances uncertainty estimation across multiple dimensions, including classification indication, precision-recall balance, and calibration.

Classification View As illustrated in Table 1, the *Corrector* has brought in significant improvements, with an average AUROC score increase of 0.27 for TriviaQA and 0.09 for SciQA. Even when applied to challenging methods such as SE and SAR, the *Corrector* boosts AUROC scores by 0.01 to 0.03. Since AUROC reflects the UE methods ability to assign higher scores to instances where the target model responds unreliably compared to those it responds to reliably, these improvements indicate that the deployment of the *Corrector* **enhances the overall indicative capacity of the uncertainty scores, making it more effective for users in determining whether to trust the models responses.**

Furthermore, as illustrated in Figure 5, the F1 score is also boosted by the *Corrector*, achieving an average increase of 38.97%. This notable improvement demonstrates the *Corrector*'s ability to help balance precision and recall, effectively mitigating the polarization tendency in the uncertainty scores observed in previous methods.

Calibration View **Although calibration is not the direct training objective of our *Corrector*, its application yields favorable calibration results.** When employing the OPT-6.7B model as the target, we observed average ECE reductions of 0.34 on TriviaQA and 0.21 on SciQA. With the LLaMA-3-8B-Instruct model as the target, the reductions are 0.11 and 0.07, respectivelystill considerable. To further illustrate the calibration performance facilitated by the *Corrector*, we provide calibration plots in Figure 6.

In summary, integrating the *Corrector* helps achieve harmonized uncertainty estimation. With

Method	TriviaQA						SciQA					
	AUROC(↑)			ECE(↓)			AUROC(↑)			ECE(↓)		
	Vanilla	+Corrector	Improv	Vanilla	+Corrector	Improv	Vanilla	+Corrector	Improv	Vanilla	+Corrector	Improv
OPT-6.7B												
LS	46.49	**65.11**	+18.62	72.71	**41.76**	-30.94	44.12	**49.40**	+5.29	76.38	**32.78**	-43.60
VC	60.41	**70.55**	+10.15	49.13	**27.61**	-21.52	51.69	**56.55**	+4.86	62.65	**38.99**	-23.66
P(True)	66.74	**72.29**	+5.84	45.00	**32.63**	-12.80	56.12	**59.49**	+3.37	58.79	**34.52**	-24.27
PE	56.36	**66.62**	+10.25	42.39	**20.28**	-22.12	50.07	**56.02**	+5.95	62.05	**36.92**	-25.13
LN-PE	78.37	**79.93**	+1.57	32.29	**20.80**	-11.49	60.88	**64.23**	+3.35	49.52	**34.68**	-14.84
SE	80.66	**81.00**	+0.34	36.64	**27.05**	-9.59	64.52	**66.15**	+1.63	52.66	**42.23**	-10.43
SAR-t	78.24	**80.21**	+1.97	40.14	**37.85**	-2.30	60.00	**63.75**	+3.74	45.33	**44.19**	-1.14
SAR-s	51.77	**55.83**	+4.06	53.78	**49.65**	-4.13	53.20	**54.15**	+0.95	76.21	**34.83**	-41.38
SAR	75.32	**78.67**	+3.35	40.61	**31.02**	-9.59	60.04	**62.72**	+2.68	49.40	**38.99**	-10.41
LLaMA-3-8B-Instruct												
LS	19.57	**69.82**	+50.25	70.25	**7.41**	-62.84	53.67	**65.38**	+11.71	38.64	**18.19**	-20.45
VC	62.34	**74.89**	+12.55	23.41	**16.78**	-6.63	68.22	**72.15**	+3.93	31.88	**19.47**	-12.36
P(True)	57.14	**72.29**	+15.15	24.67	**19.84**	-4.83	65.63	**71.41**	+5.78	34.56	**31.92**	-2.64
PE	64.52	**69.76**	+5.25	21.38	**17.24**	-4.13	66.54	**67.98**	+1.44	40.67	**34.07**	-6.60
LN-PE	72.55	**74.79**	+2.24	14.31	**11.53**	-2.79	69.48	**71.56**	+2.08	29.38	**23.76**	-5.62
SE	80.92	**82.12**	+1.20	13.07	**12.76**	-0.31	71.59	**72.93**	+1.34	30.54	**25.23**	-5.30
SAR-t	79.55	**79.93**	+0.38	16.40	**13.70**	-2.70	72.26	**73.87**	+1.61	30.37	**26.81**	-3.56
SAR-s	69.87	**77.09**	+2.95	23.17	**20.00**	-3.17	74.96	**75.72**	+0.76	38.54	**36.18**	-2.37
SAR	80.92	**81.90**	+0.98	16.17	**13.76**	-2.41	73.88	**75.19**	+1.31	28.97	**25.60**	-3.37

Table 1: AUROC and ECE scores on the TriviaQA and SciQA datasets obtained by applying the *Corrector* to existing uncertainty estimation methods. LS denotes the Lexical Similarity method. VC denotes the Verbal Confidence method. PE denote the Predictive Entropy method. LN-PE denotes the Length-normalized Predictive Entropy method. SE denote the Semantic Entropy. SAR-t refers to the token-level version of the SAR method, while SAR-s denotes the sentence-level version.

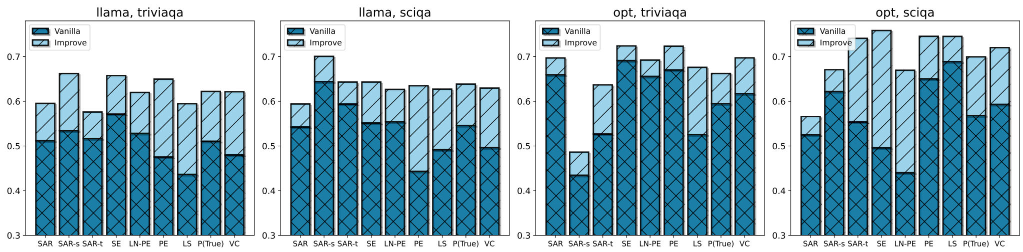

Figure 5: The performance gains of using the *Corrector* to adjust the uncertainty scores for various methods on the datasets of TriviaQA and SciQA, and the target models of LLaMA-3-8B-Instruct and OPT-6.7B, are evaluated in terms of F1 score.

the Corrector, we can improve the reliability of uncertainty scores and alignment with the actual performance of the model.

5.3 Ablation Study

We conducted ablation studies to scrutinize the impact of the base model, the correction score formats and its acquisition methods.

Formats We compared the efficacy of probabilistic values versus label values for correction. As shown in Table 2, probabilistic correction scores demonstrate clear superiority, as they allow finer-grained adjustments by leveraging a broader spectrum for integration. Conversely, discrete values, such as 0 and 1, tend to introduce significant biases in the corrected uncertainty scores.

Base Model We utilized various encoder models as base models to train the *Corrector* and assess the impact on correction performance. Specifically, we employed models from the RoBERTa series, including RoBERTa-base[4] and RoBERTa-large[5], as well as models from the DeBERTa series, such as DeBERTa-base[6], DeBERTa-v3-large[7], DeBERTa-v3-base[8], DeBERTa-v3-small[9], and DeBERTa-v3-xsmall[10]. These models represent different types, series, and sizes. As illustrated in Figure 7, more advanced, later-generation, and larger models yield

[4] huggingface.co/FacebookAI/roberta-base
[5] huggingface.co/FacebookAI/roberta-large
[6] huggingface.co/microsoft/deberta-base
[7] huggingface.co/microsoft/deberta-v3-large
[8] huggingface.co/microsoft/deberta-v3-base
[9] huggingface.co/microsoft/deberta-v3-small
[10] huggingface.co/microsoft/deberta-v3-xsmall

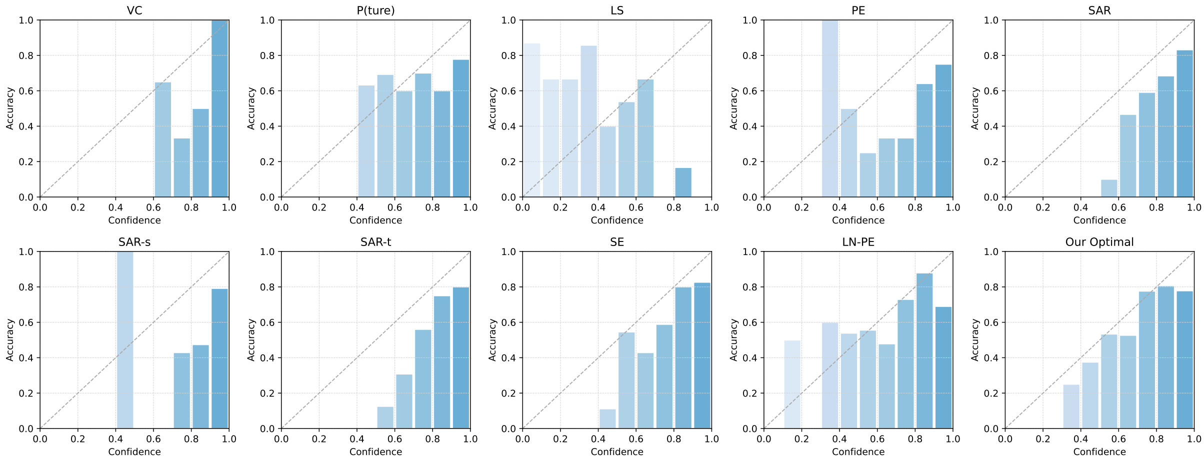

Figure 6: Calibration Plots. These plots depict the relationship between predicted confidence and observed frequencies. The diagonal line represents perfect calibration, where predicted confidence aligns precisely with actual outcomes. Bars extending above the diagonal indicate underestimation of confidence, while bars below the diagonal reflect overestimation. The final plot highlights the optimal calibration performance achieved through our *Corrector*.

superior results.

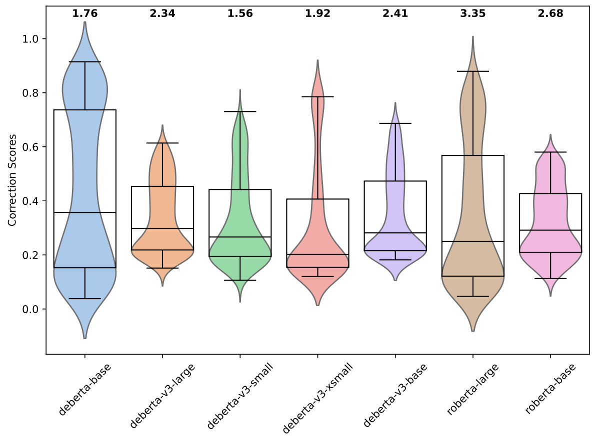

Figure 7: The overall AUROC gains achieved by *Correctors* trained on different base models across various UE methods on the SciQA dataset and Llama-3-8B-Instruct target model.

Acquisition We compare correction scores from a lightweight classifier with those estimated using GPT-4o. We attempt not to rigorously assess the target models answers but to predict its reliability. Despite GPT-4's strong performance in question answering, our results show it is less effective than the classifier in directly predicting reliability of target models when faced with questions. Additionally, as detailed in Section **??**, combining uncertainty scores from different UE methods does not improve and may even degrade performance. This highlights the *Corrector*'s unique role as a complement to existing methods.

Methods	AUROC (↑)	ECE (↓)
Corrector	69.87	6.73
Original Best	80.92	11.53
+Corrector Probability	**82.12**	**10.46**
+Corrector Label	80.92	11.53
+GPT-4o Score	80.92	11.53

Table 2: Ablation Study. LLaMA-3-8B-Instruct as the target model and TriviaQA as the test dataset. **Original Best** refers to the peak performance achieved by various baseline when the *Corrector* is not incorporated.

6 Conclusion

Our study highlights the limitations of current uncertainty estimation methods in terms of classification accuracy, precision-recall balance, and calibration. We introduce an innovative uncertainty score correction framework that utilizes a classifier as a *Corrector* to refine these scores, ensuring alignment with the model's true task performance. This *Corrector* integrates seamlessly with existing methods, enhancing their effectiveness. Extensive experiments validate that the *Corrector* consistently improves performance across various metrics, data domains, and target models. Furthermore, our ablation study underscores the *Corrector*'s capacity to provide substantial and heterogeneous improvements to existing uncertainty estimation techniques.

Limitations

Although the CUE method proposed in the paper demonstrates good performance, its generalization ability across different data domains and target models may be limited. We only compared our method with works that have open-source code, which are often designed for white-box models. Therefore, the effectiveness of our method on black-box models has not been demonstrated through experiments. However, our method does not necessitate access to the inner states of target models, making it a general enhancement strategy for both black-box and white-box uncertainty estimation.

Ethics Statement

In this study, we introduce a method for improving uncertainty estimation in the context of LLMs, which presents no immediate ethical concerns, but certain considerations must be addressed. Uncertainty estimation has significant potential to evaluate the reliability and safety of LLM outputs. However, this potential benefit comes with the risk that systematic mistakes in the uncertainty assessment could foster unfounded and misplaced confidence. Consequently, even re-calibrated uncertainty estimates should be interpreted cautiously, particularly in critical decision-making scenarios where the consequences of inaccuracies can be profound.

The datasets used in our experiment are publicly released and labeled through interaction with humans in English. In this process, user privacy is protected, and no personal information is contained in the dataset. The scientific artifacts that we used are available for research with permissive licenses. And the use of these artifacts in this paper is consistent with their intended use. Therefore, we believe that our research work meets the ethics of ACL.

References

Mari-Liis Allikivi, Joonas J¨arve, and Meelis Kull. 2024. Cautious calibration in binary classification. *arXiv preprint arXiv:2408.05120*.

Sören Auer, Dante A. C. Barone, Cassiano Bartz, Eduardo G. Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, Ivan Shilin, Markus Stocker, and Eleni Tsalapati. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports*, 13(1):7240.

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

Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, Chenyang Tao, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Mars: Meaning-aware response scoring for uncertainty estimation in generative llms. *arXiv preprint arXiv:2402.11756*.

Evan Becker and Stefano Soatto. 2024. Cycles of thought: Measuring llm confidence through stable explanations. *arXiv preprint arXiv:2406.03441*.

Jinhao Duan, Hao Cheng, Shiqi Wang, Chenan Wang, Alex Zavalny, Renjing Xu, Bhavya Kailkhura, and Kaidi Xu. 2023. Shifting attention to relevance: Towards the uncertainty estimation of large language models. *arXiv preprint arXiv:2307.01379*.

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

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

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. 2014. Deep autoregressive networks. In *International Conference on Machine Learning*, pages 1242–1250. PMLR.

Tobias Groot and Matias Valdenegro-Toro. 2024. Overconfidence is key: Verbalized uncertainty evaluation in large language and vision-language models. *arXiv preprint arXiv:2405.02917*.

Haixia Han, Tingyun Li, Shisong Chen, Jie Shi, Chengyu Du, Yanghua Xiao, Jiaqing Liang, and Xin Lin. 2024. Enhancing confidence expression in large language models through learning from past experience. *arXiv preprint arXiv:2404.10315*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *Preprint*, arXiv:2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024. Llm internal states reveal hallucination risk faced with a query. *arXiv preprint arXiv:2407.03282*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

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*.

Amita Kamath, Robin Jia, and Percy Liang. 2020. Selective question answering under domain shift. *arXiv preprint arXiv:2006.09462*.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in llms. *arXiv preprint arXiv:2406.15927*.

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

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2024a. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36.

Moxin Li, Wenjie Wang, Fuli Feng, Fengbin Zhu, Qifan Wang, and Tat-Seng Chua. 2024b. Think twice before assure: Confidence estimation for large language models through reflection on multiple answers. *arXiv preprint arXiv:2403.09972*.

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

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

Yinhan Liu. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Antonio Loquercio, Mattia Segu, and Davide Scaramuzza. 2020. A general framework for uncertainty estimation in deep learning. *IEEE Robotics and Automation Letters*, 5(2):3153–3160.

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

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

Shiyu Ni, Keping Bi, Lulu Yu, and Jiafeng Guo. 2024. Are large language models more honest in their probabilistic or verbalized confidence? *arXiv preprint arXiv:2408.09773*.

Alexander Nikitin, Jannik Kossen, Yarin Gal, and Pekka Marttinen. 2024. Kernel language entropy: Fine-grained uncertainty quantification for llms from semantic similarities. *arXiv preprint arXiv:2405.20003*.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Christos E Papadopoulos and Hoi Yeung. 2001. Uncertainty estimation and monte carlo simulation method. *Flow Measurement and Instrumentation*, 12(4):291–298.

Tejaswini Pedapati, Amit Dhurandhar, Soumya Ghosh, Soham Dan, and Prasanna Sattigeri. 2024. Large language model confidence estimation via black-box access. *arXiv preprint arXiv:2406.04370*.

Linwei Tao, Minjing Dong, and Chang Xu. 2023. Dual focal loss for calibration. In *International Conference on Machine Learning*, pages 33833–33849. PMLR.

Shuchang Tao, Liuyi Yao, Hanxing Ding, Yuexiang Xie, Qi Cao, Fei Sun, Jinyang Gao, Huawei Shen, and Bolin Ding. 2024. When to trust llms: Aligning confidence with response quality. *arXiv preprint arXiv:2404.17287*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Artem Vazhentsev, Gleb Kuzmin, Akim Tsvigun, Alexander Panchenko, Maxim Panov, Mikhail Burtsev, and Artem Shelmanov. 2023. Hybrid uncertainty quantification for selective text classification in ambiguous tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11659–11681.

Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2024. Do large language models have a legal duty to tell the truth? *Royal Society Open Science*, 11(8):240197.

Zhiyuan Wang, Jinhao Duan, Chenxi Yuan, Qingyu Chen, Tianlong Chen, Huaxiu Yao, Yue Zhang, Ren Wang, Kaidi Xu, and Xiaoshuang Shi. 2024. Word-sequence entropy: Towards uncertainty estimation in free-form medical question answering applications and beyond. *arXiv preprint arXiv:2402.14259*.

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

Duygu Nur Yaldiz, Yavuz Faruk Bakman, Baturalp Buyukates, Chenyang Tao, Anil Ramakrishna, Dimitrios Dimitriadis, and Salman Avestimehr. 2024. Do not design, learn: A trainable scoring function for uncertainty estimation in generative llms. *arXiv preprint arXiv:2406.11278*.

Adam Yang, Chen Chen, and Konstantinos Pitas. 2024. Just rephrase it! uncertainty estimation in closed-source language models via multiple rephrased queries. *arXiv preprint arXiv:2405.13907*.

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

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. 2023. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*.

A Appendix

A.1 Related Work

Uncertainty estimation methods for LLMs have gained significant attention, with approaches can be broadly categorized into logit- based methods, verbalized methods, consistency-based methods, and internal state-based methods..

Logit-based methods Logit-based methods are the most widely used and effective approaches in uncertainty estimation. As a foundational method, Predictive Entropy (PE) (Malinin and Gales, 2020), defines total uncertainty as the entropy of the output logits distribution. After that, researchers proposed a series of methods based on the inherent characteristics of natural language generation to improve upon PE methods. Kuhn et al. (2023) introduced semantic entropy (SE) that estimates uncertainty by marginalizing over semantically-equivalent samples in NLG tasks. In the similar framework, Nikitin et al. (2024) employed positive semi-definite kernels and von Neumann entropy to capture semantic similarities. Furthermore, Wang et al. (2024) proposed Word-Sequence Entropy (WSE) to adjust uncertainty proportions at both the word and sequence levels based on semantic relevance, ensuring that uncertainty is aligned with the semantic importance of words within a response. In addition to measuring the similarity between generated responses, Wang et al. (2024) proposed to judge the similarity between the target response and the generations. Duan et al. (2023) proposed Shifting Attention to Relevance (SAR), which focus on relevant components and assigns significance weights to tokens based on their contributions to the overall response. Unlike these carefully designed methods, Yaldiz et al. (2024) introduced a Learnable Response Scoring Function (LARS), which utilizes supervised data to capture complex token-probability dependencies. While effective, the above methods are computationally expensive. To alleviate these computational cost, Kossen et al. (2024) proposed Semantic Entropy Probes (SEPs) to approximate semantic entropy by leveraging hidden states from a single generation.

Verbal confidence methods Due to LLMs' strong language abilities and adherence to instructions, Verbal confidence methods are proposed. For instance, one may attach the question with a prompt like "Please respond and provide your confidence score ranging from 0 to 100.". Xiong et al. (2023) constructed a prompting, sampling, and aggregation framework to systematically evaluate various strategies and their integration, enabling LLMs to express their confidence in response. Groot and Valdenegro-Toro (2024) proposed FaR prompting strategy, which improves the confidence calibration of LLMs by separating the fact retrieval and reflective reasoning steps. However, verbal confidence methods face significant challenges with over-confidence. Ni et al. (2024) found that LLMs cannot convey their uncertainties faithfully in natural language. Becker and Soatto (2024) found that combining language confidence and proxy model probability estimation can improve the estimation of uncertainty. Madhusudhan et al. (2024) noted LLMs' language perception accuracy often lags behind probability perception, especially in specific domains Furthermore, Tao et al. (2024) found that LLMs often exhibit a high degree of overconfidence when expressing their own confidence by comparing language-based methods, consistency-based methods, and their hybrid benchmark testing methods. Their research indicates that some prompt strategies can improve the calibration of verbal confidence.

Internal state-based method Internal state-based methods suggest that the activation of the target model can be analyzed to predict the model errors. Azaria and Mitchell (2023) proposed SAPLMA by training a classifier on the hidden layer activations of an LLM to assess statement truthfulness. Similarly, Liu et al. (2024) also introduced a supervised method by training a model on labeled datasets that analyze hidden layer activations and probability-related features. Focusing on the self-assessment capabilities of LLMs, Kadavath et al. (2022) trained models to explore the LLMs' ability to evaluate the accuracy of their responses through calibration on multiple-choice and true/false questions. Ji et al. (2024) employed a probing estimator to analyze the internal mechanisms of LLMs across various NLG tasks, assessing uncertainty before response generation. Additionally, some works introduced novel interventions to refine the uncertainty estimation performance during inference. Han et al. (2024) proposed to learn from past experience (LePe) method by leveraging historical performance records and fine-tuning instructions. Li et al. (2024a) presented Inference-Time Intervention (ITI) to adjust model activations selectively during inference across a limited number of attention heads, guided by a predefined set of directions.

Consistency-based method The consistency-based method is to evaluate the uncertainty of the large model through multiple generated answers. Recently, Li et al. (2024b) employed UQ sampling with perturbation and an aggregation module to quantify sampling uncertainty in text generation tasks. Pedapati et al. (2024) proposed a paradigm to reduce overconfidence in incorrect answers by having LLMs reflect on and justify each candidate answer, then aggregating these justifications to calibrate confidence estimates. Becker and Soatto (2024) proposed extracting semantic diversity and syntactic similarity from perturbed prompts, training a model on these features to estimate confidence. Yang et al. (2024) explored the stability of explanations generated by LLMs to estimate the model's confidence in its answers. Lin et al. (2023) discussed combining observed consistency and self-reflection to assess language model uncertainty

A.2 Preliminary

In this section, we commence by clarifying the two scales of uncertainty: *relative uncertainty* and *absolute uncertainty*. We then formalize the relative uncertainty estimation as a classification task to determine whether the target model can correctly respond to a given question. Subsequently, we delve into the theoretical foundations of widely-used logit-based uncertainty estimation methods, and critically examine the inherent limitations shared by those approaches that rely exclusively on target model outputs.

A.2.1 Relative Uncertainty and Absolute Uncertainty

Research on uncertainty estimation has led to two key concepts (Kamath et al., 2020; Vazhentsev et al., 2023): *relative uncertainty* and *absolute uncertainty*, each providing distinct methods for assessing and interpreting levels of uncertainty. Given an input x, a ground truth answer y, and the predictive distribution of Y, the predictive uncertainty for the target model regarding the input x is denoted as $\mathrm{UE}(x, \theta)$. Relative uncertainty scores emphasize the accuracy of sample ranking, especially in discerning questions that the target model can correctly respond to from those it struggles with. Ideally, for every pair (x_i, y_i) and (x_j, y_j) with their predictive distributions Y_i and Y_j, we should have

$$\begin{aligned} &\mathrm{UE}(x_i, \theta) \leq \mathrm{UE}(x_j, \theta) \iff \\ &P(Y_i = y_i | x_i, \theta) \geq P(Y_j = y_j | x_j, \theta). \end{aligned} \tag{5}$$

Stricter than relative uncertainty scores, absolute uncertainty scores support to represent the model's accuracy. In cases where there is an 80% uncertainty prediction, it implies that the question is expected to be answered correctly only 20% of the time under similar conditions. This relationship can be mathematically expressed as

$$P(Y = y | \mathrm{UE}(x, \theta) = q) = 1 - q. \tag{6}$$

As relative uncertainty concerns solely with the relative rankings of $h(x) = \mathrm{UE}(x, \theta)$, it can be framed as a classification problem aimed at finding a function h that minimizes the expected loss of misclassification (Allikivi et al., 2024; Tao et al., 2023). Consider two class labels, $\mathcal{C} = \{c_0, c_1\}$, indicating whether the targrt model can correctly answer the question or not, respectively. This leads to the formulation of a decision rule

$$g(h;\tau) = \begin{cases} c_0 & \text{if } h(x) \leq \tau \text{ (confident)} \\ c_1 & \text{if } h(x) > \tau \text{ (uncertain)} \end{cases}, \quad (7)$$

where $h(x)$ is a scalar measure of uncertainty and τ is the threshold.

Drawing from decision theory, we derive the expected loss as *conditional risk* for the sample x:

$$\text{Risk}(x) = \lambda_{c_i,c_{1-i}} h_{c_{1-i}}(x), \quad (8)$$

where $c_i, i \in \{0,1\}$ denotes the true label of the sample x, and $h_{c_{1-i}}(x) = P(c_{1-i} \mid x)$ is the posterior probability of misclassifying the sample x as class c_{1-i}. $\lambda_{c_i,c_{1-i}}$ represents the loss associated with this misclassificationspecifically, a penalty incurred when the sample with the label c_i is classified as c_{1-i}. Our task is to find h^* that minimizes the overall risk

$$\text{Risk}(h) = \mathbb{E}_x \left[\text{Risk}(h(x)) \mid x\right]. \quad (9)$$

A.2.2 Theoretical Foundations of Uncertainty Estimation for LLM

LLMs typically generate outputs in an autoregressive manner, which iteratively predict the probability distribution of the subsequent token based on the evolving context (Gregor et al., 2014). Given an input sequence x with the objective of generating an output sequence $y = \{y_1, y_2, \ldots, y_L\}$, the conditional probability of the l-th token y_l is denoted as $P(y_l|y_{<l}, x;\theta)$. This probability depends on all previously generated tokens $y_{<l} = \{y_1, y_2, \ldots, y_{l-1}\}$ as well as the input x. The probability of generating the entire sequence y can be expressed as the product of the conditional probabilities of each individual token:

$$P(y|x;\theta) = \prod_{l=1}^{L} P(y_l|y_{<l}, x;\theta), \quad (10)$$

where $P(y_l|y_{<l}, x;\theta) = \frac{e^{z_l/T}}{\sum_j e^{z_j/T}}$, z is the raw logit, and T is the temperature that controls the smoothness of the probability distribution. This posterior probability provides a probabilistic framework for sequence generation. Moreover, according to prior research (Malinin and Gales, 2020), the total uncertainty for the generation of y is given by the entropy of the predictive posterior:

$$\begin{aligned} \text{PE}(x) &= \mathcal{H}[P(y \mid x,\theta)] \\ &= \mathbb{E}_{P(y|x,\theta)}[-\ln P(y \mid x,\theta)] \\ &= -\sum_{y\in Y} P(y \mid x,\theta) \ln P(y \mid x,\theta). \end{aligned} \quad (11)$$

In practice, due to the exponential computational complexity of traversing the entire response set, Monte Carlo approximation method (Papadopoulos and Yeung, 2001) is employed via beam search with a single target model for generation. The approximate entropy is defined as

$$PE(x) \approx -\frac{1}{B}\sum_{b=1}^{B} \ln P(y_b|x,\theta), \quad (12)$$

where $P(y_b|x,\theta)$ denotes the posterior probability of the b-th beam search candidate. Base on these, Kuhn et al. (2023) proposed to cluster generations with similar meanings and compute entropy using the probabilities associated with each semantic cluster. This approach is formulated as

$$SE(x,\theta) = -\frac{1}{C}\sum_{i=1}^{C} \ln P(c_i|x,\theta), \quad (13)$$

where c_i denotes each semantic cluster and C represents the set of all clusters.

Another form of improvement is to assign weights to each token in the generation when calculating posterior probabilities (Duan et al., 2023; Bakman et al., 2024), either through a manually designed algorithm or a training way, which can be formulated as

$$\tilde{P}(y \mid x;\theta) = \prod_{l=1}^{L} P(y_l \mid y_{<l}, x;\theta) \cdot w_l, \quad (14)$$

where w_l represents the weight assigned to the l-th token.

A.3 Generalization

The above results indicate that the *Corrector* performs effectively on the evaluation set comprising in-distribution data. However, our analysis highlights two primary variables that can lead to out-of-distribution scenarios: **domain of data** and **target model**. The generalization performance of the *Corrector* is evaluated through the average improvement of AUROC scores across all baselines from both RBS and CBS.

	TriviaQA	SciQA
TriviaQA	19.59	4.05
SciQA	6.03	10.20

(a) Generalization for Domain of Data

	OPT-2.7B	OPT-6.7B	LLaMA3-8B
OPT-2.7B	19.59	11.80	3.23
OPT-6.7B	6.08	11.21	3.43

(b) Generalization for Target Model

Table 3: Average AUROC scores improvement of after appling our method to baselines. (a) The leftmost column indicates the domains of data used in training, while the topmost row represents the domains of data used for evaluating, with OPT-2.7B serving as the target model. (b) The leftmost column denotes the target model during training, whereas the topmost row signifies the target model during evaluating, with TriviaQA utilized as the target domain of data.

Domain of Data To evaluate the generalization capability of our *Corrector* across different data domains, we conduct experiments by training the *Corrector* on the dataset $\mathcal{D}^*_{\text{cor}}$, crafted from either TriviaQA or SciQA, and then evaluating it on the alternate one. As illustrated in Table 3, the *Corrector* achieves optimal performance when both training and evaluating occur within the same data domain. Remarkably, even when training and evaluating on different domains, the *Corrector* still demonstrates a enhancement, yielding an average improvement of approximately 0.05. One possibility is that the target model exhibits comparable knowledge proficiency across both data domains.

Target Model We investigate the generalization for target model by training *Corrector* on $\mathcal{D}^*_{\text{cor}}$ sourced from a different target model than the one used for evaluating. As shown in Table 3, in cases where models exhibit relatively comparable knowledge capabilities, such as OPT-2.7B and OPT-6.7B, the *Correcter* exhibits generalization ability, yielding average AUROC improvements of 0.11 and 0.06, respectively. Conversely, when a substantial performance gap exists between models, such as OPT-2.7B and LLaMA-3-8B-Instruct, we achieve an average AUROC improvement of 0.03. When focusing solely on the challenging baselines from CBS, the improvement drops to 0.01.