Mind the Generation Process: Fine-grained Confidence Estimation Throughout the Generation of LLMs

Anonymous ACL submission

Abstract

Accurate confidence estimation of large language models (LLMs) is crucial for improving their generation reliability. While existing methods typically estimate confidence from limited perspectives and specific token positions, they fail to provide continuous confidence estimation throughout the generation process. In this paper, we introduce FineCE, a novel fine-grained confidence estimation method that provides the accurate and real-time confidence scores during the generation. Specifically, we develop a pipeline for construction training data to capture the inherent responses of LLMs, and design data formats for three different tasks to teach LLMs to express confidence. Additionally, we propose the Backward Confidence Integration (BCI) strategy, which integrates confidence scores from subsequent text sequences to provide a holistic confidence estimation for the current text sequence. Furthermore, we provide three strategies to identify the optimal positions to perform confidence estimation. Extensive experiments demonstrate that FineCE consistently outperforms existing baselines in various confidence estimation tasks. Our code and all baselines used in the paper are available in the GitHub https://anonymous.4open.science/r/FineCE/.

1 Introduction

Large language models (LLMs) have achieved remarkable capabilities across various tasks through extensive pre-training on text corpora followed by instruction fine-tuning on supervised datasets (Ouyang et al., 2022; Wei et al., 2021). Despite their impressive performance, LLMs still face problems with reliable generation, such as hallucination (Han et al., 2024). Confidence estimation has emerged as a crucial approach for estimating the probability of correctness in LLM outputs.

However, existing confidence estimation methods are limited by their coarse-grained confidence scores and a limited perspective, failing to provide a feasible confidence estimation. These works generally fall into question-oriented and outcome-oriented confidence estimation. The question-oriented confidence estimation task instructs LLMs to only respond to questions within their domain of knowledge scope and refuse to answer unknown questions (Zhang et al., 2023). When confronted with uncertain questions, LLMs refuse to answer the question (Kadavath et al., 2022) rather than attempting to deduce a potential answer from available information. This overly cautious strategy diminished the utility of LLMs. The outcome-oriented confidence estimation task requires LLMs to evaluate the quality of their entire generated answers (Zhang et al., 2024a; Zhao et al., 2024; Kuhn et al., 2023; Abbasi-Yadkori et al., 2024). Even if the final answer has a high confidence score, it does not represent that the generation process is completely accurate and reliable (Jiao et al., 2024). The difference between them is shown in Figure 1.

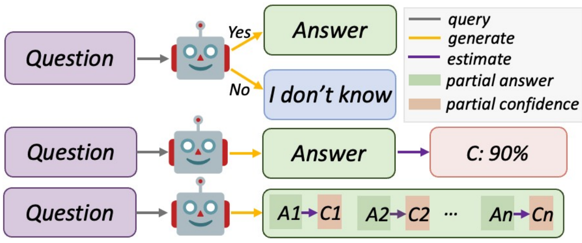

Figure 1: The difference between our proposed FineCE and existing confidence estimation method. **(Top):** LLMs either respond to queries within their knowledge scope or refuse queries beyond their capabilities. **(Middle):** The model provides a confidence score alongside an answer. **(Bottom):** Our proposed method FineCE provides the fine-grained confidence scores for any given text sequence during the generation process.

Therefore, it is necessary to develop fine-grained confidence estimation method, which provides accurate and real-time confidence scores for the intermediate generation steps. The direct benefit is to predict the likelihood of the LLM generating the correct answer in advance, without waiting for the entire answer generation to be completed. In addition, the confidence scores serve as supervisory signals for advanced LLMs, like O1[1], to guide their next generation action, whether to proceed or correct the previous errors. Furthermore, questions with consistently low confidence scores reveal deficiencies in LLM, which provides valuable insights for model improvements.

However, implementing fine-grained confidence estimation for LLMs presents three significant challenges. Firstly, ***(Task Learning:) How to teach LLMs to express their confidence?*** The inherent capabilities of LLM, including internal state representations (Su et al., 2024; Chen et al., 2024) and prompt-based instruction Branwen (2020), prove insufficient for reliable confidence estimation, necessitating dedicated training to enhance its confidence estimation abilities. But in practical scenarios, the LLM typically generates unstructured, free-form text sequences, making it difficult to assign the correct confidence scores to arbitrary text content. Secondly, ***(Effectiveness:) How to provide an accurate and unbiased confidence estimate for the current text?*** Even when provided with the same input text, LLMs generate highly variable subsequent outputs (Atil et al., 2024). Considering only local confidence estimate for the current text, while ignoring the confidence estimate of the subsequent texts, leads to biased confidence scores. Thirdly, ***(Efficiency:) Where are the optimal positions to perform confidence estimation?*** it is blind to output confidence score after each token, which is computationally redundant and unnecessary. Moreover, following the error propagation principle(Wang et al., 2024b; Liang et al., 2024), early errors in the generation sequence tend to amplify through subsequent steps, leading to deviations from the correct response. Therefore, it is essential to identify appropriate positions for confidence estimation during the generation process.

To address these challenges, in this paper, we introduce FinCE, a fine-grained confidence estimation method for LLMs. Specifically, we devise a complete pipeline for constructing training data to empower LLMs to estimate the fine-grained confidence score for any text during the generation process. Additionally, we introduce the Backward Confidence Integration (BCI) strategy for inference time, which provides more holistic confidence score by incorporating uncertainty information from subsequent text. Furthermore, to balance the trade-off between confidence estimation accuracy and computational efficiency, we propose three strategies for identifying optimal positions during the generation process.

Experiments demonstrate that FineCE significantly outperforms existing confidence estimation baselines across multiple metrics on two widely-used open-source LLMs. We further validated its performance in a downstream task where we implement a confidence score threshold filtering mechanism, accepting only responses above the setting thresholds. FineCE leads to a substantial 39.5% improvement in answer accuracy on the GSM8K dataset.

Our contributions are mainly four-fold: *1)* We introduce a fine-grained confidence estimation method FineCE. *2)*We provide a complete data construction pipeline and utilize Instruction Fine-tuning to enhance the capability of confidence estimation. *3)* We introduce BCI to generate a holistic confidence estimate for the current text by integrating the confidence of the subsequent text. *4)* We devise three strategies to find the optimal position to perform confidence estimation.

2 Related Work

Verifier and Calibration Model Formally, the trained calibration model is very similar to the trained verifier. The function of these two models are distinct. However, the verifier model is employed to evaluate the generation quality, selecting the better answer with the highest evaluation score from multiple generated samples(McAleese et al., 2024; Ke et al., 2023; Huang et al., 2024). The verifier model provides a unique and consistent score for the same text, independent of the generation model used. In contrast, confidence estimation measures the probability of an LLM generates the correct answer. Different LLMs may generate different answers for the same input, with different probabilities of getting the correct answer (Atil et al., 2024; Song et al., 2024; Renze, 2024). Therefore, the calibration model assigns different confidence scores to the same text, which usually depends on the generative model used.

The similar to our work is to evaluate the reasoning steps (Wang et al., 2024a; Lightman et al., 2023) or the generation answers (Cobbe et al.,

[1] https://openai.com/openai-o1-contributions

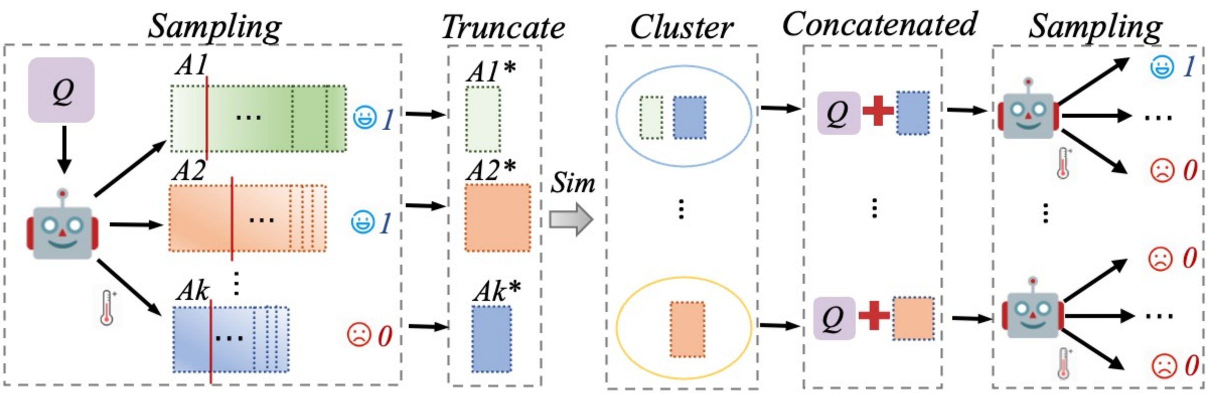

Figure 2: The process of constructing training data for confidence estimation. In the Sampling part, confidence scores for *Questions* and *Questions with Partial Answers* are calculated by Formula 2. Each sampling answer obtains a confidence score for the *Question with Answer* based on its correctness.

2021a) by training a reward model. These methods aimed to rank the multiple generated answers and select the best one or construct the step-wise data (Lai et al., 2024). However, they were designed for a particular task such as mathematical reasoning, and provided the discrete evaluation score for the reasoning steps to improve the final reasoning performance. Besides, they overlooked discussing the accuracy of evaluation. In contrast, we focus on exploring a universal method that can provide the fine-grained and accurate confidence estimates for any given text.

Confidence Expression in LLMs. In terms of confidence expression in LLMs, existing works have focused on evaluating the certainty or uncertainty of LLMs in generating correct answers to specific questions. One approach was to use carefully designed prompts to guide LLMs to express their confidence level in words along with the generated answers (Zhou et al., 2023; Xiong et al., 2023; Li and Nian, 2024; Zhang et al., 2024c). Branwen (2020) displayed GPT-3's capability to convey uncertainty on basic questions through few-shot prompts. Lin et al. (2022) introduced the concept of "verbalized confidence", which directly guided LLMs to output the confidence. Tian et al. (2023a) employed external annotations to instruct LLMs to express uncertainty in words during the answers generation processes. However, it was shown that LLMs exhibit high confidence when prompted to verbalize their confidence (Xiong et al., 2023), and they often struggle to follow complex instructions.

Another line of works focused on leveraging the logit values of specific tokens (e.g. A, B, C, etc) in the generated answer to measure the uncertainty of the entire answer sequence (Robinson et al., 2023). Kadavath et al. (2022) proposed probing the self-awareness of LLMs by incorporating a dedicated "Value Head". However, this method faced challenges when applied to general tasks due to its reliance on structured datasets, like multiple-choice questions. Moreover, there has been significant progress in developing metrics to measure the certainty of LLM responses. Kuhn et al. (2013) proposed utilizing semantic entropy among multiple sampled answers under the same questions to estimate model's uncertainty. The semantic similarity is quantified using a separated natural language inference classification system (NLI). Zhang et al. (2024b) decomposed LLMs' confidence into two dimensions, including the uncertainty about the question and the fidelity to the answer generated by the LLM.

Overall, current methods usually utilize the inherent capabilities or signals of LLMs to instruct their expression of confidence. These methods primarily rely on the capabilities of the model itself, targeting tasks with standardized answers. In this paper, we consider the ability to express confidence as a meta-capability that requires explicit training within LLMs.

3 Method

3.1 Task Formalization

Existing LLMs generally generate responses in an auto-regressive manner, sequentially predicting the next token based on the preceding sequence. Specifically, for a sequence of generated tokens $\{t_1, t_2, \cdots, t_n\}$, each token t_i $(i \in 1, 2, \cdots, n)$ is sampled from the probability distribution $P_i = \mathcal{P}(\cdot|x, t_{<i})$, where n represents the total number of

tokens generated, x represents the input text, and $t_{<i} = \{t_1, t_2, \cdots, t_{i-1}\}$ refers to the preceding tokens prior to t_i.

Considering that the outputs generated by LLMs are often unstructured, free-form, it becomes challenging to evaluate the confidence score about these texts. Our goal is to provide confidence scores at any given position during the model's text generation process. In this paper, *we define confidence as the probability of the model generating the correct answer.* The confidence estimation task aims to enhance the model's calibration capabilities, ensuring better alignment between predicted probabilities and actual performance. Furthermore, different LLMs exhibit varying probabilities of generating correct responses even when presented with the same input text. We argue that the confidence estimation task is model-dependent, and formally define the confidence estimation task as follows:

$$Conf_s = p(y = \bar{Y} | s, M) \quad (1)$$

Here, M represents the generation model, $Conf_s$ is the confidence score of sequence s, which takes the value $[0, 1]$. The larger the value, the higher the probability that M generates the correct answer based on s. Besides, $y = \{t_1, t_2, \cdots, t_n\}$ represents the complete generated sequence, $\bar{Y}$ corresponds to the golden answer, and p denotes the probability.

Notably, when the input text s comprises solely a question, the task transforms into the question-oriented confidence estimation task; When the input contains a question and a partial answer, it offers confidence scores throughout the generation process; When s represents a complete answer, the task shifts to the outcome-oriented confidence estimation task. Here, we define*Partial Answer* as any intermediate output in the overall response generation process.

Above task formalization not only unifies existing confidence estimation tasks, also extends the scope of confidence estimation to cover the entire model generation process. Consequently, our method provides a comprehensive confidence estimation, capable of producing appropriate confidence estimation for any given text input at any stage of the generation process.

3.2 FineCE

3.2.1 Data Preparation

Preliminary. Traditional deep learning approaches for classification fail to capture the model uncertainty. The predictive probabilities provided by the softmax output are frequently misinterpreted as a measure of the model's confidence. However, the model may still be uncertain in its predictions despite producing a high softmax output (Gal and Ghahramani, 2016). Therefore, to obtain the LLM's inherent real responses based on the text s, we adopt the idea of Monte Carlo Sampling(Li et al., 2024) and employ the generative LLM M to repeatedly sample k answers $\{A_s^1, A_s^2, \cdots, A_s^k\}$ at hight temperature. In our work, the input text sequence s includes three distinct types: *Question*, *Question with Partial Answer* and *Question with Answer*. The confidence score $Conf_s$ is computed by evaluating the accuracy ration of these k generated answers with respect to a reference or golden answer $\bar{Y}$. Specifically, the confidence score is calculated as follows:

$$Conf_s = \frac{\sum_{i=1}^{k} \mathbf{I}(A_s^i) = \bar{y}_s}{k}, \quad (2)$$

where A_s^i represents the ith sampling answer generated based on sequence s. For closed-ended questions, $\bar{y}_s$ represents the predefined ground-truth answer. The indicator function $\mathbf{I}$ evaluates the degree of match between generated answer and standard answers, returning 1 for matches and 0 otherwise. For open-ended questions, the evaluation results can be derived either through human or advanced LLMs such as GPT-4.

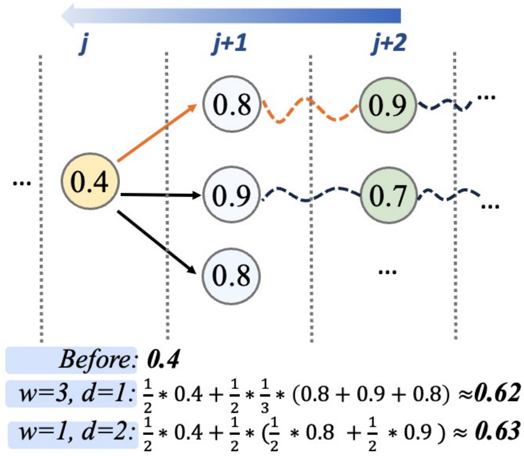

Figure 3: This is an example illustration of Backward Confidence Integration strategy.

Construction Data. The complete pipeline [2] of constructing the training data is shown in Figure

[2] *Note: Diagram notation may differ from main text notation for clarity and better visualization of the data preparation process.*

Datasets	Process	Metrics	Llama2-13B			Llama2-7B		
			Multi-Step	LECO	FineCE	Multi-Step	LECO	FineCE
GSM8K	$para(1)$	ECE	23.5	19.2	**9.3**	24.5	23.7	**12.9**
		AUROC	55.6	60.5	**73.8**	54.4	59.6	**75.3**
	$para(z-1)$	ECE	22.8	21.3	**8.4**	29.2	25.6	**13.8**
		AUROC	57.3	59.5	**77.7**	54.6	58.4	**76.8**
	avg	ECE	21.1	19.6	**6.7**	23.1	18.3	**7.2**
		AUROC	57.1	61.1	**78.1**	59.5	63.4	**78.6**
CSQA	$para(1)$	ECE	24.8	23.8	**18.3**	30.6	26.2	**15.9**
		AUROC	54.6	57.1	**66.2**	51.4	60.2	**69.5**
	$para(z-1)$	ECE	26.9	25.7	**16.2**	23.4	24.7	**16.7**
		AUROC	53.2	56.0	**69.3**	54.7	58.9	**69.8**
	avg	ECE	23.1	21.4	**11.7**	24.4	19.7	**12.8**
		AUROC	58.6	59.6	**71.3**	56	61.7	**72.5**
TriviaQA	para(1)	ECE	22.2	26.8	**14.5**	27.9	32.4	**20.1**
		AUROC	56.1	53.4	**70.8**	60.3	55.7	**73.6**
	$para(z-1)$	ECE	25.6	27.3	**15.0**	27.4	29.9	**21.0**
		AUROC	56.4	58.3	**74.2**	59.0	56.1	**73.3**
	avg	ECE	22.8	25.5	**11.3**	26.7	28.3	**16.1**
		AUROC	57.2	58.1	**76.1**	60.1	57.4	**77.2**

Table 1: Confidence estimation results throughout the generation process: the first paragraph, preceding $z-1$ paragraphs and overall average confidence scores.

2. First, starting from a question x, the model M generates k diverse answers $A_x^1, A_x^2, \cdots, A_x^k$ using high temperature sampling. Here, A_x^i represents the ith response conditioned on input x. The confidence score for x is calculated according to Formula 2. Subsequently, to generate partial answer $A_x^{1*}, A_x^{2*}, \cdots, A_x^{k*}$, we randomly truncate each of k responses at selected positions (The red vertical line indicates the truncation position for the current text). These partial answers are then grouped into $m(1 \leq m \leq k)$ clusters based on their semantic similarity. We randomly sample cluster centroids as representatives and concatenate the original question with the selected partial answers as the model input for continue sampling, and thus obtain the confidence scores of the generation trajectory.

It is worth to note that the truncation of answer A_x^i to obtain partial answers can be implemented through various human-defined rules such as steps-, paragraphs-, or fixed lengths based partitioning. To enhance the robustness and diversity of the training dataset, we also apply multiple truncation strategies simultaneously and perform truncations multiple times.

Upon completion of the aforementioned process, we obtain a diverse set of candidate responses for a question, responses that align with the ground truth are assigned a confidence score of 1, while those that deviate from the expected output receive a confidence score of 0.

Therefore, we construct a training dataset comprising tuples in the form of $\langle s, Conf_s \rangle$. The training data format is shown in the Appendix.

Training Technique To optimize the confidence estimation capability, we investigate two distinct training technique, including the Additional Value Head and Instruction Fine-Tuning (IFT) (Ouyang et al., 2022). The additional value head, reformulates confidence estimation as a multi-classification task, enabling token-level confidence predictions throughout the generation sequence. In contrast, the IFT leverages natural language generation capabilities to produce confidence estimates in a more interpretable format. In the Appendix (Figure 8) provides a comprehensive comparison of these two technique in our proposed task. In this paper, FineCE adopts the IFT training paradigm.

3.2.2 Identify the Calibration Position

While existing confidence estimation methods typically perform at a coarse-grained level, FineCE introduces fine-grained confidence estimation for LLMs. However, it is unnecessary to perform confidence calibration after each token generation. Therefore, we propose three strategies to identify optimal positions for confidence estimation during the generation process.

Paragraph-End Calibration. This strategy per-

Base Models	Baselines	GSM8K			CSQA			TriviaQA		
		ACC↑	ECE↓	AUROC↑	ACC↑	ECE↓	AUROC↑	ACC↑	ECE↓	AUROC↑
Llama2-13B	*P(IK)*	30.4	14.5	64.8	**69.9**	29.9	59.5	**66.2**	18.7	65.0
	FineCE	**33.6**	**8.9**	**67.3**	65.6	**16.2**	**69.3**	64.8	**15.5**	**68.4**
	First-Prob	30.4	23.3	59.7	62.5	22.3	60.1	63.1	27.6	57.1
	SuC	31.0	28.8	57.3	60.1	27.2	56.7	62.8	23.5	58.2
	Verb	31.0	29.3	56.2	64.3	21.7	58.3	**65.1**	27.1	53.7
	Fidelity	-	-	-	54.5	18.3	67.1	-	-	-
	FineCE	**33.6**	**5.1**	**77.8**	**65.6**	**11.5**	**70.5**	64.8	**12.0**	**76.9**
Llama2-7B	*P(IK)*	**30.7**	16.3	62.8	**64.8**	24.7	57.4	**57.4**	20.9	68.3
	FineCE	30.3	**13.1**	**72.9**	63.7	**15.9**	**69.5**	53.9	**19.1**	**68.9**
	First-Prob	29.7	25.4	58.1	62.1	25.3	57.7	52.8	25.7	55.1
	SuC	29.1	28.7	57.3	63.4	22.7	55.8	52.1	29.3	57.4
	Verb	**30.3**	28.10	56.2	62.5	26.4	55.4	**54.2**	28.6	55.8
	Fidelity	-	-	-	**40.6**	14.1	68.9	-	-	-
	FineCE	**30.3**	**6.5**	**78.9**	63.7	**11.7**	**72.3**	53.9	**15.4**	**76.8**

Table 2: The confidence estimation results across baselines for question-oriented and outcome-oriented tasks.

forms confidence estimation at natural sentence boundaries, leveraging linguistic breaks in the generation process. By calibrating at paragraph endpoints, it minimizes the disruption to the generation flow while preserving semantic coherence and contextual integrity.

Periodic Calibration. It implements confidence estimation at fixed tokens intervals throughout the generation process, such as each 50 tokens This regular, interval-based strategy offers a deterministic mechanism for confidence monitoring, ensuring consistent quality assessment across the entire generated sequence.

Entropy-based Calibration. We can set a entropy threshold to decide whether to start the confidence estimation. Though entropy is also a signal to measure model uncertainty during generation, it alone is insufficient to accurately predict the probability of generating the correct answer. The calibration is more meaningful and reliable when entropy values are higher.

We aim to identify an effective strategy and establish basic guidelines for selecting appropriate confidence estimation positions in different generation scenarios.

3.2.3 Backward Confidence Integration (BCI)

For the same LLM, it may generate diverse answers even if the input is the same. To revise either excessively high or low confidence level and mitigate output confidence bias, we introduce the Backward Confidence Integration strategy. This strategy not only considers the confidence score of the current text, also incorporates the confidence of its subsequent text, thereby deriving a more holistic confidence score for the current text sequence. Specifically, for a text sequence, $Conf_{s_j}$ denotes confidence estimation at the jth calibration position, and w represents the number of sampled answers. The adjusted confidence score $Conf'_{s_j}$ is calculated as follows:

$$Conf'_{s_h} = \begin{cases} \alpha Conf_{s_h} + (1-\alpha)\frac{1}{w}\sum_{b=1}^{w} Conf'_{s^b_{h+1}}, h \in (j, j+d) \\ Conf_{s_h}, h = j+d \end{cases}$$

where α controls the revision ratio, which determines the degree to which the subsequent context is integrated into the current confidence calculation. A smaller α A smaller places greater emphasis on the confidence scores of subsequent text generations. Parameters w and d represents the depth and width of fusion respectively. This back-to-forward inference strategy enables a global and accurate confidence estimation for s_j. $Conf_{s^b_h}$ represents the confidence score of the text at the hth calibration position in the bth sampled answer. An illustrative example is provided in Figure 3.

4 Experiments

4.1 Experiment Setting

Dataset. We evaluate the performance of confidence estimation across three datasets including *GSM8K* (Cobbe et al., 2021b), *TriviaQA*(Joshi et al., 2017) and *CommonsenseQA*(CSQA; Talmor et al., 2018).

Models and Baselines. We employ two widely-used open-source models, including Llama2-7B and Llama2-13B (Touvron et al., 2023). And the baselines we compared include the following three

Strategy	Dataset	ACC	ACC_δ	ECE_1	ECE_{avg}	Ratio
Paragraph	GSM8K	33.6	73.1 (+39.5)	9.8	7.7	30.4
	CSQA	65.6	73.5 (+7.9)	26.8	13.0	22.0
	TriviaQA	64.8	80.0 (+15.2)	17.2	14.5	28.5
Entropy	GSM8K	33.6	72.5 (+38.9)	13.2	7.7	10.0
	CSQA	65.6	81.1 (+15.5)	27.1	18.8	7.0
	TriviaQA	64.8	80.2 (+15.4)	18.5	15.4	13.4
Fixed-token	GSM8K	33.6	71.6 (+38.0)	13.1	10.8	23.5
	CSQA	65.6	78.9 (+13.3)	24.2	20.7	34.7
	TriviaQA	64.8	78.8 (+14.0)	20.0	18.0	34.1

Table 3: Performance comparison of three strategies for identifying optimal calibration positions in Llama2-13B. Ration(%) denotes the proportion of tokens preceding the calibration position relative to token count.

types: 1) **Question-oriented:** *P(IK)*(Kadavath et al., 2022); 2) **Outcome-oriented:** *First-Prob* (Santurkar et al., 2023), *SuC*(Lin et al., 2022), *Verbalized Porb* (Verb Tian et al., 2023b), *Fidelity (Zhang et al., 2024a)* ; 3) **Step-wise estimation:** *Multi-Step* (MP; Xiong et al., 2023), *LECO*(Yao et al., 2024)

Evaluation Metrics. We adopt several widely used metrics including *Expected Calibration Error (ECE)*, *Receiver Operating Characteristic Curve (AUROC)* and *Accuracy (ACC)*.

Further details about datasets, baselines, implementations (including all prompts used in this paper, important parameters, and platforms) can be found in Appendix A.1.

4.2 Main Results and Analysis

RQ1: How does FineCE perform compared with baselines? We demonstrate that ***base models provide the accurate confidence estimates for any given text sequence on three datasets after using FineCE.*** The overall results are shown in Table 1 and Table 2. The results in the two tables are the average values.

From Table 1, we observe that ***FineCE delivers the accurate confidence estimates during the generation process.*** Notably, the AUROC values obtained by our method are greater than 70% in most cases, showing a strong performance for accurate identification. In contrast, the AUROC for the other two baselines are always around 60% across these datasets, which is almost close to random guessing. Besides, the outstanding performance on process-oriented confidence estimation task shows that our proposed method FineCE can provide the accurate estimates for any given text sequence, which is significantly different from other methods. In the table, $para(1)$ and $para(z-1)$ respectively represent the first paragraph and the $z-1$ paragraphs of the generated answer. avg represents the average confidence estimates for the entire generation process.

From Table 2, ***our method consistently outperforms all baselines in terms of ECE and AUROC, and shows excellent calibration capability.*** Taking the GSM8K dataset as an example, on the answer-oriented confidence estimation task, Llama2-13B achieves a lower ECE 5.1%, and the AUROC is as high as 78.9%. At the same time, we observe that although FineCE improves the confidence calibration ability through fine-tuning, it does not lead to a decrease in accuracy, showing close accuracy of the outcomes achieved through the prompt engineering method. This is because we conduct the replaying strategy during fine-tuning and mix some general IFT datasets.

4.3 Ablation Analysis

RQ2: Where does FineCE perform the confidence estimation? We conduct a comparative analysis of three calibration position strategies in FineCE using the Llama2-13B model. The results are shown in Table 3. In this experiment, we set the entropy threshold to *1e-10* for the Entropy-based strategy and fixed the token length to 30 for the Prediodic Calibration strategy. We find all three strategies demonstrate comparable performance in terms of ECE, with Paragraph-end Calibration strategy showing slightly superior results. This can be attributed to preserve the complete semantic information truncated by paragraph. And the Entropy-based strategy tends to trigger calibration earlier in the generation process (indicated by smaller ratio values). It represents that entropy-based strategy is likely to frequently perform confidence estimation.

We provide some basic principles. For general

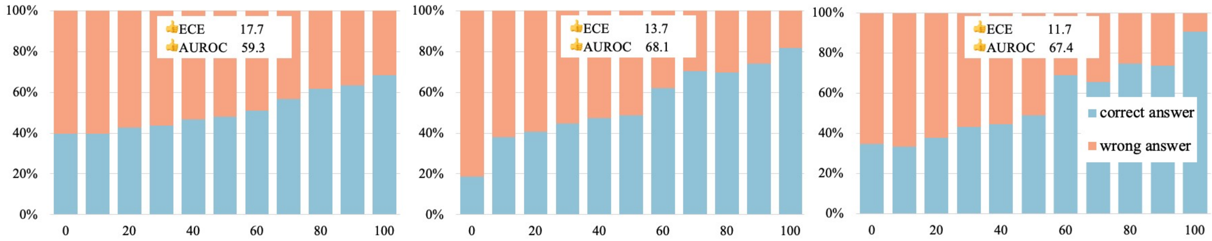

Figure 4: The Zero-shot performance on OpenBookQA dataset. From left to right, the figures show the confidence estimation performance of FineCE for the question, partial answer, and complete answer. The x-axis represents the confidence scores (%), and the y-axis represents the ratio of quantities. The top area contains the detailed values of ECE and AUROC.

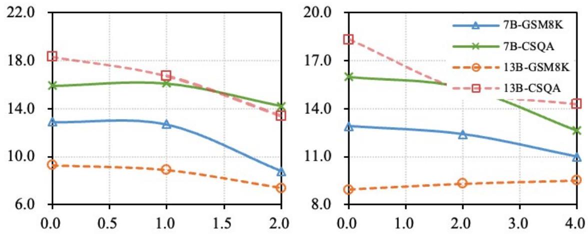

Figure 5: The impact of fusion depth (left) and width (right) on confidence estimation.

tasks, it is sufficient to estimate at the end of paragraph, which alleviate token consumption. For more complex tasks, employing entropy-based strategies for dual verification may be better.

RQ3: How effective is the BCI strategy? To evaluate the effectiveness of the BCI strategy, we conduct ablation experiments on the GSM8K and CSQA datasets using two base models. We evaluate the ECE of $para(1)$, and the results are shown in Figure 5. When $d = 0$ and $w = 0$, it represents FineCE without using the BCI. We find that ***using the BCI method significantly enhances the confidence estimation performance***. Moreover, we observe that the performance enhancement becomes more pronounced as the fusion width w and d increases.

4.4 Generalization Analysis

RQ4: How does FineCE perform with zero-shot prompt on new task? To evaluate the generalizability of the FineCE method, we test the confidence estimation performance of FineCE on OpenBookQA dataset (Mihaylov et al., 2018) using Llama2-13B, and the results are shown in Figure 4. We find that FineCE exhibits outstanding performance across both the ECE and AUROC confidence metrics. Additionally, there is a robust positive correlation between the model's confidence estimates and the actual accuracy of the answers. Specifically, we observe that higher confidence levels correlated with higher accuracy. It indicates that ***our method possesses noteworthy generalization capabilities and is capable to offer reliable confidence estimates when applied to new tasks.*** Besides, we investigate how different training datasets from different models affect model performance in Appendix A.2.

4.5 Downstream Application

RQ5: How does FineCE perform on downstream application? We set a confidence threshold δ to filter the answers. Only when the confidence estimates exceeds the threshold, we accept the generation answer. The results are shown in Table 3. We leverage the first confidence estimates. δ is set to 80%, and ACC_δ represents the accuracy rate among responses that surpass the confidence threshold. We find FineCE enables early performance prediction and provides a reliable mechanism for filtering model outputs. **Compared with unconditionally accepting the output results of the LLM, the accuracy of the model has been significantly improved after introducing output confidence.**

5 Conclusion

In this paper, we propose a fine-grained confidence estimation method FineCE to provide accurate confidence scores throughout the generation process. We first introduce the difference between FineCE and existing popular related works, and describe the dataset construction process. We introduce the BCI to generate a holistic confidence estimate for the current text and three strategies for identifying the optimal estimation position. Extensive experiments demonstrate our proposed method's superior performance across various confidence estimation task and downstream task.

6 Limitations

Although FineCE demonstrates effectiveness in providing accurate confidence scores across various confidence estimation task, it still faces challenges with highly open-ended problems as all existing confidence estimation methods. For example, questions like "*How to stay healthy?*" lack explicit response constraints (e.g., perspective, scope or response length). The inherent ambiguity and vast solution space of such queries pose significant challenges for this task. Our future work will explore more robust confidence estimation methods specifically for such highly open-ended questions.

References

Yasin Abbasi-Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesv'ari. 2024. To believe or not to believe your llm. *ArXiv*, abs/2406.02543.

Berk Atil, Alexa Chittams, Liseng Fu, Ferhan Ture, Lixinyu Xu, and Breck Baldwin. 2024. Llm stability: A detailed analysis with some surprises. *arXiv preprint arXiv:2408.04667*.

Gwern Branwen. 2020. Gpt-3 nonfiction- calibration. Technical report, The institution that published. Last accessed on 2022-04-24.

Haozhe Chen, Carl Vondrick, and Chengzhi Mao. 2024. Selfie: Self-interpretation of large language model embeddings. *ArXiv*, abs/2403.10949.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. *ArXiv*, abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021b. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Haixia Han, Jiaqing Liang, Jie Shi, Qi He, and Yanghua Xiao. 2024. Small language model can self-correct. In *AAAI Conference on Artificial Intelligence*.

Hui Huang, Yingqi Qu, Hongli Zhou, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge model is not a general substitute for gpt-4.

Fangkai Jiao, Chengwei Qin, Zhengyuan Liu, Nancy F. Chen, and Shafiq Joty. 2024. Learning planning-based reasoning by trajectories collection and process reward synthesizing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 334–350, Miami, Florida, USA. Association for Computational Linguistics.

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

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

Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Sheng-Ping Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, Jie Tang, and Minlie Huang. 2023. Critiquellm: Towards an informative critique generation model for evaluation of large language model generation. In *Annual Meeting of the Association for Computational Linguistics*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2013. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

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

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *ArXiv*, abs/2406.18629.

Meng Li and Heng Nian. 2024. Perturbation amplitudes design method based on confidence interval evaluation for impedance measurement. *IEEE Transactions on Industrial Electronics*, 71(10):12323–12337.

Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. 2024. Neuro-symbolic data generation for math reasoning.

Yuhang Liang, Xinyi Li, Jie Ren, Ang Li, Bo Fang, and Jieyang Chen. 2024. Light-weight fault tolerant attention for large language model training. *ArXiv*, abs/2410.11720.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *arXiv preprint arXiv:2205.14334*.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Cer'on Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *ArXiv*, abs/2407.00215.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. 2022. Training language models to follow instructions with human feedback. *ArXiv*, abs/2203.02155.

Matthew Renze. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356, Miami, Florida, USA. Association for Computational Linguistics.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2023. Leveraging large language models for multiple choice question answering. *Preprint*, arXiv:2210.12353.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? *ArXiv*, abs/2303.17548.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2024. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. *ArXiv*, abs/2407.10457.

Weihang Su, Changyue Wang, Qingyao Ai, Hu Yiran, Zhijing Wu, Yujia Zhou, and Yiqun Liu. 2024. Unsupervised real-time hallucination detection based on the internal states of large language models. *ArXiv*, abs/2403.06448.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023a. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *arXiv preprint arXiv:2305.14975*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D. Manning. 2023b. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. *ArXiv*, abs/2305.14975.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *Preprint*, arXiv:2312.08935.

Xiaochen Wang, Junqing He, Liang Chen, Reza Haf Zhe Yang, Yiru Wang, Xiangdi Meng, Kunhao Pan, and Zhifang Sui. 2024b. Sg-fsm: A self-guiding zero-shot prompting paradigm for multi-hop question answering based on finite state machine. *ArXiv*, abs/2410.17021.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652.

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*.

Yuxuan Yao, Han Wu, Zhijiang Guo, Biyan Zhou, Jiahui Gao, Sichun Luo, Hanxu Hou, Xiaojin Fu, and

Linqi Song. 2024. Learning from correctness without prompting makes llm efficient reasoner. *ArXiv*, abs/2403.19094.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi Ren Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Instructing large language models to say 'i don't know'. In *North American Chapter of the Association for Computational Linguistics*.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024a. Calibrating the confidence of large language models by eliciting fidelity. *ArXiv*, abs/2404.02655.

Mozhi Zhang, Mianqiu Huang, Rundong Shi, Linsen Guo, Chong Peng, Peng Yan, Yaqian Zhou, and Xipeng Qiu. 2024b. Calibrating the confidence of large language models by eliciting fidelity. *Preprint*, arXiv:2404.02655.

Yuhang Zhang, Yue Yao, Xuannan Liu, Lixiong Qin, Wenjing Wang, and Weihong Deng. 2024c. Open-set facial expression recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(1):646–654.

Xinran Zhao, Hongming Zhang, Xiaoman Pan, Wenlin Yao, Dong Yu, Tongshuang Wu, and Jianshu Chen. 2024. Fact-and-reflection (far) improves confidence calibration of large language models. *ArXiv*, abs/2402.17124.

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

A Appendix

A.1 Additional Experiments Details

Baselines. We introduce each method in the baseline, and the prompts used are shown in the Table 6.

- **P(IK).** It trains a logistic regression with the additional value "head" added to the model to output the confidence estimated.
- **First-Prob.** It uses the logits of the first token of LLM's generated answer as the confidence estimate.
- **SuC.** It first clusters the sub-questions, and use the same confidence estimate for questions in the same cluster.
- **Verb.** It is a prompt-based method. It designs the prompts to guide the model to output its confidence score alongside with the generated answer.
- **Fidelity.** For MCQA, it decomposes the LLM confidence into the *Uncertainty* about the question and the *Fidelity* to the answer generated by LLMs.
- **LECO.** It also proposes leveraging logits to estimate step confidence. Besides, it further designs three logit-based scores that comprehensively evaluate confidence from both intra- and inter-step perspectives.
- **Multi-Step.** It also uses prompts to guide the model to output the process confidence and takes the average as the final result.

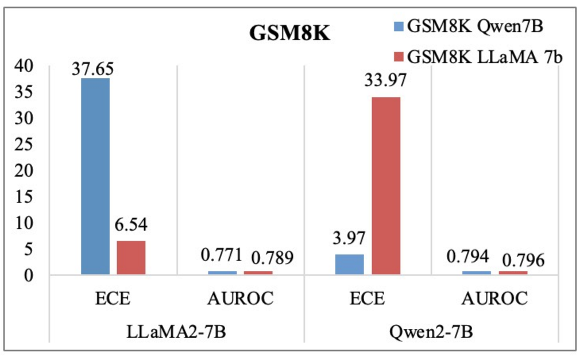

Figure 6: On GSM8K dataset, the performance confidence estimation for the two different families models using datasets from different sources.The horizontal axis represents the base models.

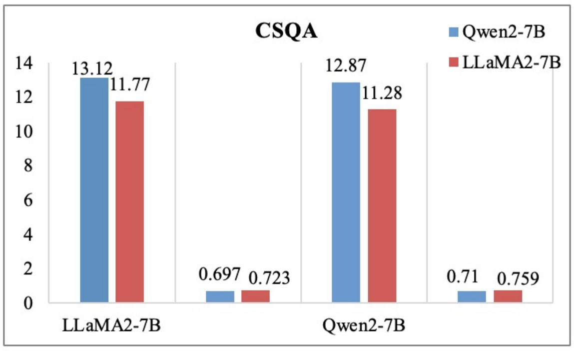

Figure 7: On CSQA dataset, the performance confidence estimation for the two different families models using datasets from different sources.The horizontal axis represents the base models.

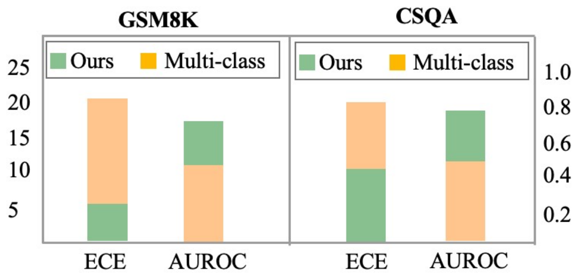

Figure 8: The performance comparison using different training technical. The left side of the vertical axis indicates the value of ECE, and the right side indicates the value of AUROC.

Important Parameters Settings. During fine-tuning, we employ the AdamW optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.5$. The initial learning rate is set to 1e-4, with the warmup phase of 300 steps. All experiments are conducted on the workstations of NVIDIA A800 PCIe with 80GB memory and the environment of Ubuntu 20.04.6 LTS and torch 2.0.1.

Training Data We provide three types of training data format in Table 5. All the prompts used in this paper are shown in Table 6.

A.2 Discussions

RQ6: How does FineCE perform when trained using datasets from different model? First, for the LLaMA2-13B and LLaMA2-7B two base models, we employ two distinct models to construct the training datasets: the model itself or an alternative model. The results are shown in Figure 9. Training with datasets generated from the alternative model achieves confidence calibration performance very close to the obtained using the dataset constructed by the model itself, especially on the GSM8K and

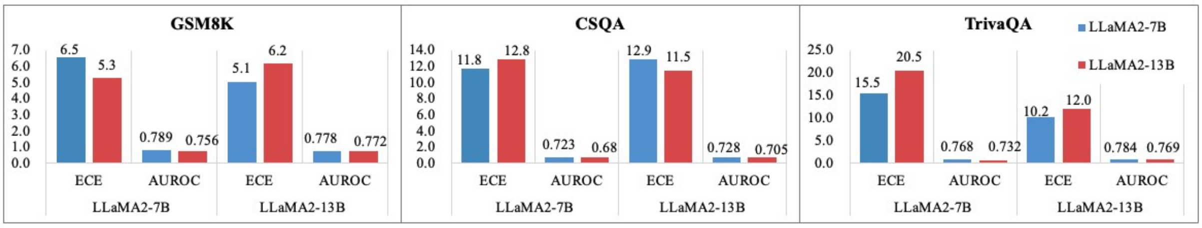

Figure 9: The performance confidence estimation for two base models using training datasets from different sources. The horizontal axis represents the base models

Dataset	Base Models	ACC-before	ACC-after
GSM8K	LLaMA2-7B	30.3	58.8 (+28.5)
	LLaMA2-13B	33.6	78.3 (+44.7)
CSQA	LLaMA2-7B	63.7	79.9 (+16.2)
	LLaMA2-13B	65.6	81.8 (+16.2)
TrivalQA	LLaMA2-7B	53.9	70.3 (+16.4)
	LLaMA2-13B	64.8	80.7 (+15.9)

Table 4: Comparison of the model's accuracy performance across three datasets with a set confidence threshold of 80%.

CAQA datasets. We guess that it may be related to the used models being from the same family and exhibit significant similarities in their knowledge capabilities. ***It suggests that larger models could effectively instruct smaller models to learn to express the confidence. In addition, leveraging smaller models to construct training datasets may be a cost-efficient alternative.***

We also use two models from different families to explore this phenomenon further, including Qwen2-7B and LLaMA2-7B, which are from different model families. The results are show in Figure 6 and Figur 7. We find that there are two different phenomena on different datasets. On the GSM8K dataset, compared with using the model itself to construct training data, the confidence training data constructed with the help of other models performed poorly, especially in the ECE value, where the difference was particularly significant. On the CSQA dataset, the performance difference between the two methods is small. This may be because there is a large difference in the accuracy of Qwen2-7B and LLaMA2-7B on the GSM8K dataset, which makes it impossible to effectively migrate the confidence training data constructed by these two models to each other.

We can conclude that **if the performance of two models on a task is close, the confidence training data constructed using one of the models can be effectively used in the training stage of the other model.**

RQ7: Which training skill is more suitable? On the GSM8K training dataset, we employ two distinct training techniques using the LLaMA2-13B model. One is to add a multi-classification head at the end of the model to output the confidence estimates through classification. The other is the instruction fine-tuning method as we used in the experiment. The outcome confidence estimates results are shown in Figure 8, it suggests that ***under the same data scale, the multi-classification techniques exhibited poor performance in confidence estimation task.***

RQ8: How does our method perform on highly open questions? We randomly select 300 single-round English open question-answering data on Sharegpt [3], and use LLaMA2-7B to provide confidence estimates, and compared the output confidence with the evaluation score of the generated answers using GPT4 to calculate ECE. We find that for highly open questions, our proposed method achieved a higher ECE value of 65.66. This is also in line with our expectations. This is because we did not use GPT4's evaluation to assist in constructing training data, resulting in a large difference between the confidence provided by the model and the GPT4 scoring results.

[3] https://huggingface.co/datasets/OpenGVLab/ShareGPT-4o

< Question, Conf >
Input: If a vehicle is driven 12 miles on Monday, 18 miles on Tuesday, and 21 miles on Wednesday. What is the average distance traveled per day?
Output: Conf:0.7

< Question + Partial Answer, Conf >
Input: If a vehicle is driven 12 miles on Monday, 18 miles on Tuesday, and 21 miles on Wednesday. What is the average distance traveled per day? The total number of miles driven is
Output: Conf:0.9

< Question + Answer, Conf >
Input: If a vehicle is driven 12 miles on Monday, 18 miles on Tuesday, and 21 miles on Wednesday. What is the average distance traveled per day? The total number of miles driven is 12 + 18 + 21 = «12+18+21=51»51 miles.The average distance traveled per day is 51 miles / 3 days = «51/3=17»17 miles.
Output: Conf:1.0

Table 5: Three training data formats of FineCE.

Method	**Prompt**
Verb	Read the question, analyze step by step, provide your answer and your confidence in this answer. Use the following format to answer: "Explanation: [insert step-by-step analysis here] Answer: [ONLY the option letter; not a complete sentence], Confidence (0-100):[Your confidence level, please only include the numerical number in the range of 0-100]%” Please refer to the example I have given: <example> {few-shot} </example> Question: {question} Now, please answer this question and provide your confidence level. Let’s think it step by step.
Multi-step	Read the question, break down the problem into K steps, think step by step, give your confidence in each step, and then derive your final answer and your confidence in this answer. Note: The confidence indicates how likely you think your answer is true. Use the following format to answer: Step 1: [Your reasoning], Confidence: [ONLY the confidence value that this step is correct]% Step K: [Your reasoning], Confidence: [ONLY the confidence value that this step is correct]% Final Answer: [ONLY the answertype; not a complete sentence] Overall Confidence(0-100): [Your confidence value]% Please refer to the example I have given: <example> {few-shot} </example> Question: {question} Now, please answer this question and provide your confidence level. Let’s think it step by step.
FineCE(ours)	Below is a question and some steps: Question: {question} {steps} Please give your confidence.

Table 6: The prompts used in the baselines.

Strategy	Dataset	ACC	ACC_δ	ECE_1	ECE_{avg}	Ratio
Paragraph	GSM8K	30.3	62.6	12.5	8.8	28.6
	CSQA	63.7	79.6	19.8	13.2	53.2
	TriviaQA	53.9	66.2	24.5	20.7	42.0
Entropy	GSM8K	30.3	57.5	11.4	9.5	9.3
	CSQA	63.7	84.1	21.2	16.4	8.9
	TriviaQA	53.9	71.1	24.1	20.2	13.2
Fixed-token	GSM8K	30.3	62.3	12.3	8.3	22.1
	CSQA	63.7	82.9	20.2	19.0	32.0
	TriviaQA	53.9	72.0	23.8	19.5	33.4

Table 7: Performance comparison of three strategies for optimal calibration position detection in Llama-7B. Ration(%) denotes the proportion of ttokens preceding the calibration position relative to token count.