Improving Language Model's Calibration with Reasoning-Aware Confidence Estimation

Cindy Zhang, Perry Jiang
Instructor: Jiaxin Huang
Washington University in St. Louis
St. Louis MO 63105, USA
`cindy.zhang, j.pengyu, jiaxinh@wustl.edu`

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of natural language processing tasks. However, their predictions often exhibit overconfidence or underconfidence, leading to a calibration gap—a mismatch between predicted confidence and actual correctness. While prior research primarily focuses on calibration in discrete settings such as multiple-choice question answering, less attention has been paid to open-ended, long-text generation tasks, where evaluating both accuracy and confidence is inherently more challenging. In this work, we address this gap by using a framework for calibrating LLMs in open-ended QA settings. We introduce a hybrid confidence estimation approach that combines similarity-based and consistency-based signals to extract meaningful confidence scores. These scores are then aligned with model accuracy metrics computed using semantic similarity and answer consistency across sampled generations. We also explore fine-tuning strategies to improve calibration, demonstrating that targeted updates can reduce calibration error without degrading task performance.

1 Introducton

Large Language Models (LLMs), such as GPT-3 and LLaMA, have demonstrated impressive performance across a wide range of natural language processing tasks. However, a persistent issue remains: overconfidence—the tendency of models to provide incorrect answers with unjustified certainty. This undermines the trustworthiness and reliability of LLMs, especially in applications that require dependable predictions. To address this, it is essential to minimize the calibration gap, defined as the discrepancy between a model's predicted confidence and its actual accuracy.

Calibration tuning refers to the process of calibrating a language model by directly fine-tuning its internal parameters using an external loss objective function measuring the calibration gap. In our framework, we propose a method that combines consistency-based and similarity-based approaches for extracting both confidence and accuracy to deal with the data with both a concrete answer and evaluated using the Expected Calibration Error (ECE).

2 Related Work

Defining and evaluating the calibration gap in LLMs is particularly challenging due to several factors: the lack of explicit, unified confidence outputs for entire responses, ambiguity in determining accuracy for open-ended or generative outputs, and the inherently distributional nature of calibration, which requires aggregating statistics across many samples rather than evaluating individual predictions(7).

Unlike traditional classification models, LLMs do not produce a single scalar confidence score for an entire answer. One intuitive approach explored in prior work is verbalized confidence extraction, where the model is prompted to self-report its confidence level. When given a question and its answer, then asked, *“How likely is the above answer to be correct"?* The procedure involves generating the answer in one chat session and obtaining its verbalized confidence in another independent chat session. While simple to implement, LLMs, when verbalizing their confidence, tend to be overconfident, potentially imitating human patterns of expressing confidence(7), which leads to poor calibration performance.

Another widely used approach is the token-probability method, which leverages the internal probabilities assigned to each generated token(6). This method works well in settings with a finite answer space, such as multiple-choice questions, where one can compute the likelihood of each candidate answer and normalize them to obtain confidence scores. However, for long-form, open-ended questions, this approach becomes impractical: token-level probabilities only measure the likelihood of a specific sequence, not the semantic correctness of an idea. Since the same concept can be expressed in countless ways with different wording and lengths, assigning confidence based on exact generation probabilities becomes meaningless(3).

To overcome these limitations, prior research has proposed the P(True) method(3): a secondary prompt is used to feed the model both the original question and its generated answer, asking whether the answer is "A. True" or "B. False." The probability assigned to "A" is then used as the confidence score. While clever, this method still depends on access to the model's internal probabilities, limiting its use in black-box settings.

In contrast, consistency-based methods are well-suited to black-box scenarios(7). The intuition is that a confident model should produce consistent answers across multiple generations(5). This method checks whether the first answer matches subsequent generations and assigns a binary score (1 for match, 0 for mismatch). Averaging these scores gives a proxy for model confidence(4).

Building on this idea, similarity-based methods introduce a more nuanced, continuous measure. Instead of binary consistency, an external model (such as a semantic similarity or entailment model) compares the generated answers to compute a similarity score between 0 and 1(2). This enables a more flexible comparison between diverse, unstructured answers—especially valuable when ground truth labels are vague or unavailable.

For long-text generative tasks, accuracy extraction also requires more than direct string comparison(10). We follow the idea of comparing model-generated responses with reference answers using strong external models capable of semantic or reasoning-level evaluation(1).

In this work, we evaluate different confidence extraction approach using the GSM8K dataset, a benchmark consisting of grade-school-level math word problems with step-by-step reasoning and final answers. This dataset is particularly suitable for our setting, as it provides both structured final answers and detailed intermediate reasoning steps. By combining consistency- and similarity-based methods, we develop a hybrid confidence and accuracy estimation pipeline tailored for long-form open-ended tasks, and show that it effectively reduces the calibration gap.

3 Background Knowledge

Autoregressive Language Models. LLMs perform next-token prediction over sequences. The model parameters, θ, are trained with cross-entropy loss, and parameterize a conditional distribution

$$p_\theta(w_{t+1} \mid w_{0:t}), \tag{1}$$

where the prompt $w_{0:t}$ is the input tokens, and w_{t+1} is the next token. In this paper we consider using LLMs for question answering, which involves the following inputs

$\boldsymbol{P}$: the text prompt used to contextualize the question.

$\boldsymbol{Q}$: the question, in text.

$\boldsymbol{A}$: the ground-truth answer in text.

$\hat{\boldsymbol{A}}$: the language model's answer.

Calibration. A model is well-calibrated when an outcome predicted with probability p does occur p fraction of the time in reality(4). This alignment between predictions and reality is measured using the expected calibration error (ECE) via empirical binning , such that an ECE of 0 corresponds to perfect calibration, i.e., a model knows when it's wrong(9). Having well-calibrated probabilities is crucial for effective downstream decision-making.

Verbalized Confidence After the model generates the answer, it further prompts the model to generate the confidence score by using the suffix *"Confidence (0–1):"*.

Expected Calibration Error (ECE). A model's uncertainties are well calibrated if they align with the empirical probabilities—i.e. an event assigned probability p occurs at rate p in reality. Following (Naeini et al., 2015), we estimate ECE by binning the maximum output probability of each of n samples into b equally-spaced bins $\mathcal{B} = \{B_j\}_{j=1}^{b}$ w.r.t. the prediction confidence estimated for each sample. The empirical ECE estimator is given by,

$$\widehat{\mathrm{ECE}} = \sum_{j=1}^{b} \frac{|B_j|}{n} \left|\mathrm{conf}(B_j) - \mathrm{acc}(B_j)\right|, \tag{2}$$

where $\mathrm{conf}(B_j)$ is the average confidence of samples in bin B_j and $\mathrm{acc}(B_j)$ is the corresponding accuracy within the bin. As is typical in literature, we use $b = 10$ bins. An ECE of 0 corresponds to a perfectly calibrated model, i.e. in each bin, the predicted confidence perfectly aligns with the proportion of the correct predictions of the model.

CoT prompting Chain-of-Thought prompting is a prompting technique used with large language models (LLMs) to encourage step-by-step reasoning when solving complex problems, especially in tasks like math, logic, and commonsense reasoning.

Verbalized Confidence Give the model a question and its answer, then add *"How likely is the above answer to be correct"?* at the end of the prompt.

Top-K. Another way to alleviate overconfidence is to realize the existence of multiple possible solutions or answers, which acts as a normalization for the confidence distribution. Motivated by this, Top-K prompts LLMs to generate the top K guesses and their corresponding confidence for a given question.

Method	**Prompt**
Vanilla	Read the question, provide your answer, and your confidence in this answer.
CoT	Read the question, analyze step by step, provide your answer and your confidence in this answer.
Self-Probing	Question: [...] Possible Answer: [...] Q: How likely is the above answer to be correct? Analyze the possible answer, provide your reasoning concisely, and give your confidence in this answer.
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.
Top-K	Provide your *K* best guesses and the probability that each is correct (0% to 100%) for the following question.

Table 1: Prompt templates for different uncertainty estimation methods

4 Framework

4.1 Model

In our experiment, we utilized TinyLlama-1.1B-Chat-v1.0, a compact language model with approximately 1.1 billion parameters. This model was pre-trained on 3 trillion tokens over a period of 90 days using 16 A100-40G GPUs. TinyLlama adopts the same architecture and tokenizer as LLaMA 2, ensuring compatibility with existing LLaMA-based tools and frameworks. The chat version used in our study was fine-tuned using Hugging Face's Zephyr training recipe, beginning with a synthetic dialogue dataset called UltraChat, and further refined with TRL's DPOTrainer on the UltraFeedback dataset, which includes 64k prompts with human and model rankings. This combination results in a lightweight yet capable conversational model suitable for resource-efficient deployment and experimentation.

Model	TinyLlama-1.1B-Chat-v1.0	GPT-3
Accuracy (final answer-only)	2.4%	52.31%
ECE	93.7%	47.49%

Table 2: Performance of TinyLlama and GPT-3 on the GSM8K dataset

Based on the table comparing the performance of TinyLlama-1.1B-Chat-v1.0 and GPT-3 on the GSM8K dataset, we can observe a significant gap in both accuracy and calibration, indicating severe overconfidence or miscalibration.. This suggests that while TinyLlama is compact and efficient, its current performance on math reasoning tasks like GSM8K lags significantly behind larger, more mature models like GPT-3 in both correctness and confidence reliability.

4.2 Calibration Method

In this work, we use the GSM8K dataset, which consists of math word problems paired with step-by-step solutions annotated with intermediate reasoning and a final numeric answer. To perform calibration tuning, we require ground truth question–answer pairs (Q, A), the language model's generated answer $\hat{A_0}$ and candidate answers $\hat{A_1}, ..., \hat{A_m}$, its self-assessed confidence in the correctness of that answer $\hat{C}$, and the actual correctness label C. We generate $\hat{A_0}, ..., \hat{A_m}$ using Chain-of-Thought (CoT) prompting, which encourages the model to explicitly articulate intermediate reasoning steps.

4.2.1 Confidence Extraction Methods

Consistency. A natural idea of aggregating different answers is to measure the degree of agreement among the candidate outputs and integrate the inherent uncertainty in the model's output.(8)
For any given question and an associated answer $\tilde{Y}$, we sample a set of *candidate answers* $\hat{Y}_i$, where $i \in \{1, \ldots, M\}$. The agreement between these candidate responses and the original answer then serves as a measure of confidence, computed as follows:

$$C_{\text{consistency}} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\{\hat{Y}_i = \tilde{Y}\}. \tag{3}$$

Similarity The structure is similar to the consistency except changing consistency score to similarity score determined by an external language model.

P(True) Let the model determine whether it's correct by giving the question and its answer back to the model and use prompt in Figure 2 to let it determine whether the answer generated by itself is correct.

We evaluate different confidence extraction methods and tune the model using the method with best performance. We first applied similarity based method. To obtain this $\hat{C}$, we use the language model itself to generate the answer $\hat{A_0}$ and candidate answers $\hat{A_1}, ..., \hat{A_m}$ (shown in Figure 2). Then we use GPT-4o as an external powerful language model to determine the similarity scores $s_1, ...s_m$ between $\hat{A_0}$ and the candidate answers using prompt in Figure 2. Then we calculate $\hat{C}$ by averaging $s_1, ..., s_m$.

The other method is consistency based method by determining the consistency scores based on the final answer only. Using P(True) method, we put Q and $\hat{A}$ as the input to get confidence score.

4.2.2 Accuracy Extranction Method

We get C by comparing the final answer's correctness and the similarity between the reasoning steps in $\hat{A}_0$ and A. Since both A and $\hat{A}$ contain structured multi-step reasoning along with the final solution, we use an external language model—GPT-4o—along with an auxiliary grading prompt to evaluate whether the generated reasoning is logically sound. To determine the final accuracy label C, we assign a weighted score: 60% based on the correctness of the final numeric answer and 40% based on the validity of the intermediate reasoning as judged by the external grader.

Task	**Prompt**
Generating $\hat{A}$ using CoT	Instruction:<*question*> Explain your reasoning step-by-step. Response
Similarity	The following are two answers from different students for the same question. Please evaluate the similarity between these answers as a percentage from 0% to 100%. Give relatively higher weight to the final answer and lower weight to the reasoning part. Focus less on wording differences but more on general reasoning similarity. Answer 1:<*Answer1*> Answer 2:<*Answer2*> Similarity Score (only return a number, without symbol:
P(True)	Question: <*question*> Proposed Answer: <*output*> Is the proposed answer true or false? Choose one from the following: (A)True (B)False The proposed answer is:
Accuracy	The following are two answers from different students for the same question. Please evaluate the similarity of reasoning between these answers as a percentage from 0% to 100%. Focus less on wording differences but more on general reasoning similarity. Part 1:<*Answer1*> Part 2:<*Answer2*> Similarity Score (only return a number, without % symbol):

Table 3: Prompts for different tasks

4.3 Calibration Tuning

Overview

To improve the confidence calibration of a language model, we fine-tune the TinyLlama-1.1B-Chat model using a supervised regression objective. The model learns to output a scalar confidence estimate between 0 and 1, indicating how likely its own answer is to be correct. Unlike conventional accuracy-oriented tuning, our objective is to minimize calibration error—particularly the Expected Calibration Error (ECE)—without modifying the model's answer-generation capability.

Data Preparation

Both training and validation datasets are constructed from previously generated model outputs. Each example contains a question, the model's answer, and a soft accuracy label. The label combines exact match correctness (1 or 0) and GPT-4o-evaluated reasoning similarity into a single score. And the input prompt for calibration takes the following format:

```
Question: <...>
Answer: <...>
How confident are you that this answer is correct? Respond
with a number between 0 and 1.
```

This setup defines a supervised regression task: mapping a prompt to a calibrated confidence score.

Training Process

We apply LoRA (Low-Rank Adaptation) to fine-tune the model. Adaptation is limited to the attention projection layers (`q_proj` and `v_proj`) of the TinyLlama-1.1B-Chat model. For LoRA, we keep the default hyperparameters - rank r = 8, α = 32, and dropout probability 0.1. The model is optimized to minimize the Mean Squared Error (MSE) between the predicted confidence and the soft accuracy label. Formally, the loss function used during training is:

$$\mathcal{L} = \sum_{i=1}^{n} (C_i - Y_i)^2$$

where C_i is the model-predicted confidence score and Y_i is the ground-truth soft accuracy score for the i-th sample. The predicted confidence C_i is obtained by applying a lightweight regression head on top of the final hidden state of the language model. Specifically, we extract the embedding of the final token and pass it through a linear layer followed by a sigmoid activation to produce a scalar confidence score in the range $[0, 1]$.

Validation and Evaluation

After each epoch, we evaluate calibration performance using the predicted confidences and ground-truth accuracy labels. The Expected Calibration Error (ECE) is computed by binning the confidence–accuracy pairs, calculating the absolute difference between mean confidence and mean accuracy within each bin, and aggregating these into a weighted error metric.

5 Results

To evaluate the effectiveness of different confidence extraction methods and their impact on calibration quality, we compute both the model's accuracy and Expected Calibration Error (ECE) under several configurations.

For accuracy evaluation, we consider two criteria: one based solely on exact answer match, and another that incorporates reasoning similarity between the model's and ground-truth answers using GPT-4o. For calibration assessment, we measure the ECE for each confidence estimation method, including consistency-based confidence, similarity-based scoring, pTrue, and an uncalibrated baseline, which is the verbalized confidence.

The results are summarized in Table 4 and Table 5.

Method	**Accuracy**
Final Answer-Only	0.0240
Similarity-Based	0.138

Table 4: Accuracy under two evaluation schemes

Method	**ECE Score**
Baseline ECE	0.8228
pTrue ECE	0.6570
Similarity ECE	0.6534
Consistency ECE	0.3783

Table 5: ECE scores under different confidence estimation methods

Table 4 reports the accuracy of the model under two evaluation schemes. The *Final Answer-Only* approach, which evaluates correctness based solely on the exact answer match, results in a low accuracy of 2.4%, reflecting the strictness of this criterion. In contrast, the *Similarity-Based* metric incorporates reasoning similarity—computed using GPT-4o—and yields a more informative measure of correctness, resulting in a notably higher accuracy of 13.8%. The similarity-based accuracy actually provides a more realistic evaluation of model performance compared to the 2.4% exact match score. While exact match is overly strict, the similarity metric accounts for reasoning quality and partial correctness, better reflecting the model's actual understanding and reasoning ability in real-world applications. Therefore, when calculating ECE scores, we use the similarity-based accuracy instead of the final answer-only one.

Table 5 shows the Expected Calibration Error (ECE) obtained using different confidence estimation strategies. Among these, the *Consistency-Based* method achieves the lowest ECE of 0.3783, indicating that its predicted confidences are best aligned with actual correctness. The *Similarity-Based* and *pTrue* methods both produce moderate ECEs (around 0.65), while the *Baseline* method yields the highest ECE of 0.8228, revealing poor calibration quality and the fact that the model is overconfident towards its answers. These findings suggest that consistency-based confidence extraction provides the most effective calibration. Therefore, we use the consistency-based confidence as our soft label for calibration tuning.

After applying our calibration tuning procedure, the model's Expected Calibration Error (ECE) was significantly reduced from 0.3783 (consistency-based confidence) to 0.2819. This demonstrates that the model learned to better align its predicted confidence with actual correctness, improving reliability in downstream decision-making tasks.

6 Discussion

6.1 Limitations

While our calibration tuning approach effectively reduces Expected Calibration Error (ECE) and improves the reliability of the model's confidence estimates, there are several limitations to the current methodology. First, the supervision signal (soft accuracy) is derived from a heuristic combination of reasoning similarity and exact match correctness, which, while informative, may introduce noise or bias that affects the quality of calibration. Second, our current tuning is performed on top of a frozen language model with LoRA adapters; while this reduces compute cost, it may also limit the model's capacity to fully adapt to calibration objectives. Additionally, confidence is predicted based only on the final token embedding, which may not fully capture the richness of reasoning present in the entire generated answer.

6.2 Future Directions

Several avenues exist for improving this work. One direction is to enhance the soft accuracy labels by incorporating human annotations or more sophisticated agreement metrics beyond GPT-based similarity. Another is to explore multi-token or sequence-level regression heads that consider the full output trajectory, potentially yielding better confidence prediction. Furthermore, joint training of answer generation and calibration within a multi-task or reinforcement learning framework may allow the model to better align answer quality and confidence. Finally, evaluating calibration across diverse task types (e.g., commonsense reasoning, summarization) would help assess the generality and robustness of the proposed tuning strategy.

7 References

References

[1] Detommaso, G., Bertran, M., Fogliato, R., & Roth, A. (2024). Multicalibration for confidence scoring in llms. arXiv preprint arXiv:2404.04689.

[2] Loka Li, Guangyi Chen, Yusheng Su, Zhenhao Chen, Yixuan Zhang, Eric Xing, and Kun Zhang. (2024). Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. arXiv preprint arXiv:2402.12563.

[3] Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., & Kaplan, J. (2022). Language models (mostly) know what they know. arXiv preprint arXiv:2207.05221.

[4] Kapoor, S., Gruver, N., Roberts, M., Pal, A., Dooley, S., Goldblum, M., & Wilson, A. (2024). Calibration-tuning: Teaching large language models to know what they don't know. In Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024) (pp. 1–14). St. Julians, Malta: Association for Computational Linguistics.

[5] Stengel-Eskin, E., Hase, P., & Bansal, M. (2024). LACIE: Listener-Aware Finetuning for Confidence Calibration in Large Language Models. arXiv preprint arXiv:2405.21028.

[6] Ulmer, D., Gubri, M., Lee, H., Yun, S., & Oh, S. J. (2024). Calibrating Large Language Models Using Their Generations Only. arXiv preprint arXiv:2403.05973.

[7] Xie, L., Liu, H., Zeng, J., Tang, X., Han, Y., Luo, C., ... & He, Q. (2024). A Survey of Calibration Process for Black-Box LLMs. arXiv preprint arXiv:2412.12767.

[8] Xiong, M., Hu, Z., Lu, X., Li, Y., Fu, J., He, J., & Hooi, B. (2023). Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. arXiv preprint arXiv:2306.13063.

[9] Zhao, X., Zhang, H., Pan, X., Yao, W., Yu, D., Wu, T., & Chen, J. (2024). Fact-and-Reflection (FaR) Improves Confidence Calibration of Large Language Models. arXiv preprint arXiv:2402.17124.

[10] Zhang, C., Liu, F., Basaldella, M., & Collier, N. (2024). Luq: Long-text uncertainty quantification for llms. arXiv preprint arXiv:2403.20279.