Quantized Can Still Be Calibrated: Understanding and Improving Uncertainty Calibration for Quantized Large Language Models

Mingyu Zhong[1], **Guanchu Wang**[2], **Yu-Neng Chuang**[3], **Na Zou**[1]

[1]University of Houston; [2]University of North Carolina at Charlotte; [3]Rice University

mzhong5@uh.edu, gwang16@charlotte.edu
yc146@rice.edu, nzou2@central.uh.edu

Abstract

Although weight quantization helps large language models (LLMs) in resource-constrained environments, its influence on the uncertainty calibration remains unexplored. To bridge this gap, we present a comprehensive investigation of uncertainty calibration for quantized LLMs in this work. Specifically, we propose an analytic method to estimate the upper bound of calibration error (UBCE) for LLMs. Our method separately discusses the calibration error of the model's correct and incorrect predictions, indicating a theoretical improvement of calibration error caused by weight quantization. Our study demonstrates that quantized models consistently exhibit worse calibration performance than full-precision models, supported by consistent analysis across multiple LLMs and datasets. To address the calibration issues of quantized models, we propose a novel post-calibration method to recover the calibration performance of quantized models through soft-prompt tuning. Specifically, we inject soft tokens into quantized models after the embedding layers and optimize these tokens to recover the calibration error caused by weight quantization. Experimental results on multiple datasets demonstrate its effectiveness in improving the uncertainty calibration of quantized LLMs, facilitating more reliable weight quantization in resource-constrained environments.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across diverse tasks, including text generation, translation, question answering, and complex reasoning (Brown et al., 2020). A critical application of these models involves their deployment in resource-constrained environments, such as medical devices and mobile phones, where computational resources are limited. Weight quantization is a promising solution to reduce the memory requirements of LLMs, enabling their deployment on local machines with restricted computational capacity. This democratization of access to LLM capabilities has been facilitated by advanced techniques such as BitsandBytes (Dettmers et al., 2022) and GPTQ (Frantar et al., 2022), which utilize low-bit computation to improve generation efficiency.

While quantization effectively compresses these models, it introduces potential trade-offs in performance. Although quantized models generally maintain comparable accuracy to their full-precision counterparts, their uncertainty calibration, a crucial factor for trustworthy systems, requires careful examination (Guo et al., 2017). Well-calibrated uncertainty estimates are particularly vital in high-stakes applications like finance and healthcare (Bajwa et al., 2021). Additionally, accurate calibration enables broader advances in model development, such as leveraging high-confidence predictions on unlabeled data for semi-supervised learning (Sui et al., 2025; Jumper et al., 2021).

Despite the significance of uncertainty calibration, the relationship between model quantization and calibration performance remains understudied. Our research addresses this gap through both theoretical analysis and empirical investigation, focusing on two key research questions:

What is the Impact of Quantization on Model Calibration? Weight quantization significantly impairs model calibration, leading to less reliable uncertainty estimates. To better understand and quantify this effect, we propose a novel closed-form upper bound for calibration error. This metric decomposes the calibration error into components for correct and incorrect predictions. Our comprehensive benchmark on multiple LLMs and datasets reveals that quantized models consistently demonstrate higher calibration error compared with full-precision models. Weight quantization causes the model to behave both under-confidence on the correct answers and over-confidence on incorrect

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 30503–30517

cases. This degradation in calibration quality suggests that lower-bit representations introduce distortions that affect confidence estimates.

How Can We Address the Calibration Gap? We propose a soft-prompt tuning method for reducing the calibration gap of quantized models. This approach injects soft tokens to the embedding layers of quantized models, without changing the model weights. The optimization for the injected soft tokens aligns with the calibration error, ensuring that the tuning process naturally improves uncertainty calibration. Experimental results demonstrate that the calibration error of quantized models consistently reduces after the tuning, indicating its effectiveness in calibrating quantized models without significant computational overhead. The contributions of this work are summarized as follows:

- **Calibration Gap.** We discover the calibration gap between the quantized LLMs and full-precision ones. Our comprehensive analysis reveals that quantized LLMs consistently exhibit worse uncertainty calibration compared to full-precision ones.

- **Post Calibration.** We propose a novel post calibration method based on soft-prompt learning to recover the calibration gap of quantized models. The post calibration has a theoretical foundation towards reducing the calibration error.

- **Experimental Evaluation.** Experimental results on two datasets demonstrate that our proposed post calibration method can effectively reduce the calibration error of quantized models, reducing its gap with the full-precision models.

2 Preliminary

We introduce the notations and uncertainty metrics in this section.

2.1 Notations

Let f represent the Large Language Model (LLM) under consideration. We denote the instruction prompt as p; the question content as q; the golden answer a; and the model-generated answer $\hat{a} = f(p, q)$. Additionally, we define the correctness of $\hat{a}$ as $c \in \{0, 1\}$ where $c = 1$ if $\hat{a}$ is the same as a, otherwise $c = 0$.

2.2 Uncertainty Measurement

Uncertainty measurement aims to estimate the confidence of a model-generated answer, denoted as $\hat{c} \in \{0, 1\}$, where $\hat{c}$ should be well aligned with the answer's correctness. Specifically, a value of $\hat{c}$ approaching 1 indicates a higher likelihood of correctness, while a value approaching 0 suggests a higher likelihood of incorrectness. For this estimation, we consider two existing methods for uncertainty quantification: Correctness Classification Token Probability (CCTP) and Verbalized Confidence with Alternative Answers (VACC) (Kapoor et al., 2024; Tian et al., 2023).

Correctness Classification Token Probability (CCTP). CCTP quantifies the confidence of an answer by querying the model to verify the correctness of the answer (Kapoor et al., 2024). Given an answer a to a question q, the CCTP confidence is calculated as:

$$\hat{c}^{a}_{CCTP} = \frac{f(\text{`yes'}; p, q, a)}{f(\text{`yes'}; p, q, a) + f(\text{`no'}; p, q, a)}, \tag{1}$$

where p represents the prompt used to query the model about the correctness of a.

Verbalized Confidence with Alternative Answers (VCAA). VCAA quantifies the confidence of an answer by querying the model to state its confidence, given the alternative candidates for the answer (Tian et al., 2023). Specifically, for a given model-generated answer $\hat{a}$, VCAA collects n alternative candidates of answer $\hat{a}_1, \cdots, \hat{a}_n$ from the model. The model then provides a verbalized confidence score for $\hat{a}$ according to:

$$\hat{c}_{\text{vcaa}} = f(p, q, \hat{a}, \hat{a}_1, \cdots, \hat{a}_n), \tag{2}$$

where p denotes the prompt used to query the model's confidence in $\hat{a}$.

3 Uncertainty Calibration Gap

We explored the impact of model quantization on confidence calibration by comparing calibration errors between quantized and full-precision models. Specifically, we first propose a metric to quantify the calibration error of models; and then experimentally compare quantized and full-precision models based on this metric.

3.1 Upper Bound Calibration Error (UBCE)

Traditional calibration metrics often provide aggregate measures that may obscure important patterns in model behavior. We propose the Upper Bound Calibration Error (UBCE) as a novel metric that enables more detailed analysis of calibration performance by separately considering correct and incorrect predictions. Specifically, UBCE decomposes the calibration error of a model into two components: the confidence error of correct and incorrect answers, and combines these components through a weighted sum, where weights correspond to the probability of answer correctness. The formal definition of UBCE is given in Definition 1.

Definition 1 (UBCE) *UBCE combines the confidence error of correct and incorrect answers using the probability of correctness. Specifically, its definition is given by:*

$$UBCE \triangleq \Pr(c=0)\mathbb{E}[\hat{c}|c=0] + \Pr(c=1)(1-\mathbb{E}[\hat{c}|c=1]) \quad (3)$$

where c is the ground truth correctness and $\hat{c}$ is predicted confidence scores.

Intuitively, $\mathbb{E}[\hat{c}|c=i]$ indicates the average confidence for the answers $i \in \{0,1\}$, where $c=1$ for correct answers and $c=0$ for incorrect answers. Thus, $\mathbb{E}[\hat{c}|c=0]$ quantifies the calibration error for incorrect answers, while $(1-\mathbb{E}[\hat{c}|c=1])$ measures the error for correct answers. These errors are weighted by their respective proportions $P(c=0)$ and $P(c=1)$ to compute the overall UBCE.

Relationship to Other Calibration Metrics. We demonstrate the relationship between our proposed UBCE metric with two established calibration error metrics: Brier Score (BS) and the Binned Expected Calibration Error (binnedECE) (Guo et al., 2017; Brier, 1950). Specifically, the Brier Score measures the mean squared error between confidence predictions and true labels:

$$\text{BS} \triangleq \mathbb{E}_{c\sim\{0,1\}}[(\hat{c}-c)^2].$$

The binnedECE provides a more granular assessment by examining calibration errors within confidence intervals:

$$\text{binnedECE}(\hat{c},\mathcal{I}) \triangleq \sum_{j\in[m]} |\mathbb{E}_{c\sim\{0,1\}}[(\hat{c}-c)I(\hat{c}\in\mathcal{I}_j)]|$$

where $\mathcal{I} = \mathcal{I}_1,\ldots,\mathcal{I}_m$ defines the partition set. The relationship between UBCE and these two metrics is formalized in Theorem 1.

Theorem 1 *UBCE provides a strict tight upper bound for both the Brier Score and binnedECE:*

$$UBCE \geq BS, binnedECE \quad (4)$$

The proof of Theorem 1 is given in Appendices A.

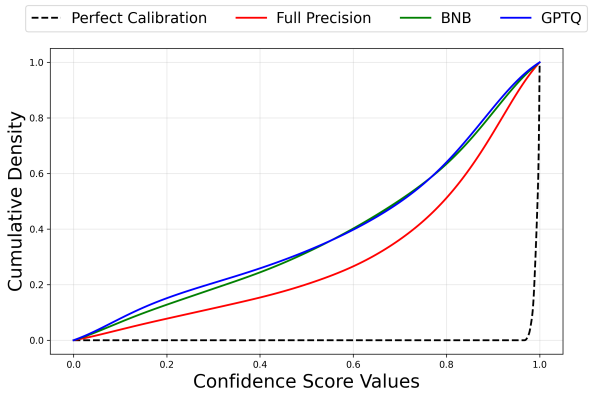
Theorem 1 indicates that UBCE provides an upper bound of the BS and binnedECE metrics. It is a more conservative estimate of calibration error, ensuring that estimation based on UBCE represents worst-case calibration performance.

3.2 Evaluation Experiment Setup

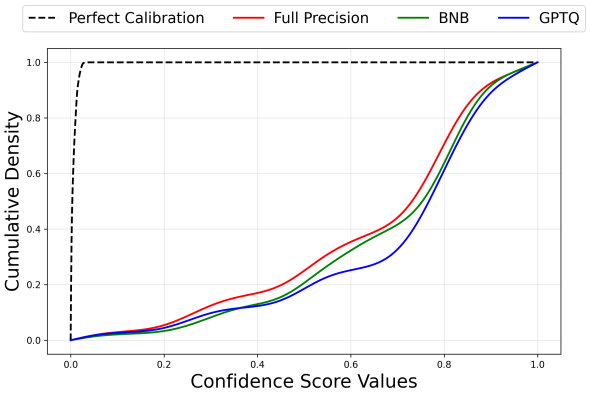
We benchmark the calibration error for three state-of-the-art LLMs at 7B scale of parameter: Llama-3.1-8B (Touvron et al., 2023), Mistral-v0.3-7B (Jiang et al., 2023), and Qwen-2.5-7B (Bai et al., 2023). Specifically, we compare the calibration error of their full-precision and quantized versions. For the quantized models, we consider the BNB (Dettmers et al., 2022) and GPTQ (Frantar et al., 2022) for these LLMs with 4-bit quantization. More details are given in Appendix J. The calibration performance of models was assessed through two metrics: the Brier Score (BS) and our proposed Upper Bound Calibration Error (UBCE).

Benchmark Datasets. The benchmark datasets are the CommonsenseQA and ARC-challenge datasets. While both datasets feature multiple-choice questions, we adapt them to an open-ended format where models must generate responses from the available options. Response correctness is determined through a comprehensive evaluation framework combining exact match and n-gram similarity metrics with ground truth answers.

Evaluation Pipeline. The evaluation process follows a systematic four-stage pipeline for assessing calibration performance across different models. First, we generate and collect responses through iterative prompting, where each iteration incorporates previous responses in the prompt and employs greedy decoding for reproducibility. Once we collect the complete set of responses across all models, we compute confidence scores using methods of CCTP and VCAA for each model. Finally,



(a) Cumulative Distributions of CCTP for Correct Answers in the ARC Dataset, Generated by the Llama-3.1-8B Models



(b) Cumulative Distributions of VCAA for Incorrect Answers in the ARC Dataset, Generated by the Qwen-2.5-7B Models

Figure 1: Cumulative Distributions of Confidence Scores

Table 1: Calibration difference (%) of Quantized LLMs: The percentage values represent the relative difference of calibration error with full-precision LLMs, using the full-precision model as the reference. Negative values indicate a reduction in calibration error for the quantized model, signifying improved calibration. Conversely, positive values indicate an increase in calibration error for the quantized model, meaning worse calibration performance.

Dataset		ARC				CSQA			
Uncertainty Metrics		CCTP		VCAA		CCTP		VCAA	
Model	CE Metric	BNB	GPTQ	BNB	GPTQ	BNB	GPTQ	BNB	GPTQ
Llama- 3.1-8B	BS	21.05%	23.61%	3.14%	48.26%	17.06%	7.65%	1.56%	24.24%
	UBCE	15.29%	16.35%	-0.75%	24.65%	11.86%	3.44%	-0.56%	4.70%
Mistral-7B-v0.3	BS	1.71%	-0.72%	1.58%	-1.15%	2.96%	-2.52%	2.31%	0.08%
	UBCE	0.91%	1.14%	0.42%	0.36%	1.72%	-0.19%	0.97%	0.77%
Qwen-2.5-7B	BS	4.91%	24.00%	0.52%	3.63%	1.77%	36.42%	3.60%	6.57%
	UBCE	6.37%	10.09%	0.24%	-1.59%	1.92%	23.80%	3.31%	2.18%

we calculated calibration error metrics to measure the alignment between predicted confidence and answer correctness. For the convenience of comparison, the calibration performance gap between quantized and full-precision models is quantified as the relative percentage difference:

$$\text{Difference (\%)} = \frac{(CE_Q - CE_F) \times 100}{CE_F} \quad (5)$$

where CE_Q and CE_F denote the calibration errors of the quantized and full-precision models, respectively, which consider both the BS and UBCE metrics. This difference greater than 0 indicates a quantized model has a greater calibration error, indicating a negative impact of quantization on calibration performance. More details about the experiments are given in Appendix E.

3.3 Empirical Analysis

Our comprehensive analysis consistently reveals patterns of calibration degradation across various quantization methods and evaluation metrics. This is evidenced by a persistent calibration discrepancy between quantized and full-precision models, highlighting the existence of a calibration gap.

Calibration Difference Statistics. Our systematic analysis of calibration errors, presented in Table 1, provides a comprehensive comparison of quantized models with their full-precision counterparts across multiple metrics. The values represent the relative percentage change in calibration error following the quantization process. The data demonstrates that quantized models exhibit higher calibration errors in 85% of calibration error metric values, as evidenced by 41 out of 48 measurements. This consistent pattern of increased calibration errors following quantization establishes a significant calibration gap between the two model types. The observed gap between quantized and full-precision models clearly indicates the negative impact of quantization on the calibration performance of models. Therefore, it is necessary to measure and evaluate the calibration error after quantization for reliable model deployment.

Source of Calibration Gap. Understanding the sources of calibration errors requires a more nuanced analysis beyond aggregate metrics. To identify specific patterns of miscalibration, we examine the cumulative distribution of confidence scores for correct and incorrect predictions separately. In a perfectly calibrated model, confidence scores for correct predictions should approach 1, while scores for incorrect predictions should approach 0, as indicated by the dashed lines in Figure 1 (a) and (b). Specifically, Figure 1 (a) presents the cumulative distribution of confidence scores for correct answers by the Llama-3.1-8B model, while Figure 1 (b) shows the distribution for incorrect answers by the Qwen-2.5-7B model. It indicates that full precision model has better calibration performance than the quantized models, as evidenced by its confidence density curve closer to the perfectly calibration results.

This comprehensive analysis provides significant evidence of the quantization on calibration performance. While the weight quantization enhances computational efficiency, it unintentionally introduces systematic biases in the model's uncertainty estimates. Identifying and reducing this calibration error is crucial for developing reliable quantized models in practice.

4 Towards Calibration Recovery

In this section, we present an efficient method for reducing the calibration error of quantized LLMs. Specifically, we (i) establish an upper bound on the calibration error in Theorem 2, (ii) propose a soft-prompt-learning framework to reduce this error, (iii) empirically validate the framework's effectiveness, (iv) demonstrate its superiority to existing calibration techniques, and (v) show that it improves accuracy on relevant downstream tasks while preserving generalization on others.

4.1 Post Calibration

We establish the upper bound of calibration error as the perplexity of quantized models.

Theorem 2 (Upper bound of UBCE) *The perplexity establishes an upper bound for the UBCE. Specifically, for a golden answer $a = [t_1, \cdots, t_n]$, the upper bound of UBCE is given by*

$$UBCE_{\hat{a}} \leq \sum_{i=1}^{n-1} -\log \Pr[t_{i+1} \mid p, q, t_1, \cdots, t_i] \tag{6}$$

We given the proof of Theorem 2 in Appendix B. Theorem 2 suggests that minimizing for perplexity of quantized models can naturally reduce the calibration error. This provides a theoretical foundation for optimizing the quantized models for reducing their calibration error.

4.2 Soft-prompt Learning for Post Calibration

We propose an efficient way to the quantized models to reduce their calibration error. Specifically, LLMs typically have embedding layers to map the prompts to the embedding space: $[t_1, \cdots, t_n] \rightarrow \mathbb{R}^{n \times d}$. To reduce the calibration error of quantized models, we inject k learnable soft tokens after the embedding layer's mapping (Xu et al.). Specifically, these soft tokens are vectors in the embedding space, defined as $e_1, e_2, \cdots, e_k \in \mathbb{R}^d$. We note that the soft prompt learning process is data-driven, and we utilize a text dataset $\mathcal{D}$ for learning. For every sequence $[t_1, \cdots, t_n] \in \mathcal{D}$, we minimize the following loss function $\mathcal{L}$,

$$\mathcal{L} = \sum_{i=1}^{n-1} -\log \Pr[t_{i+1} \mid q, e_1, \cdots, e_k, t_1, \cdots, t_i]. \tag{7}$$

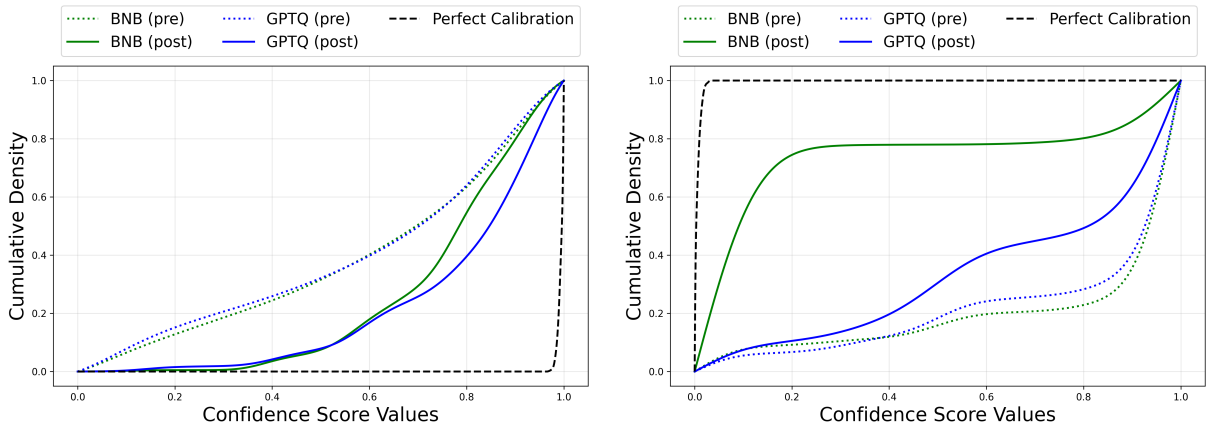
Different from the prompt tokens $t_1, \cdots, t_n$ which are processed by the embedding layer, the soft tokens $e_1, e_2, \cdots, e_k$ are injected into the model after the embedding layer. For minimizing the loss function in Equation (7), the model parameters remain frozen, and only the soft tokens $e_1, e_2, \cdots, e_k$ are updated. Formally, the optimization is given by $e_1^*, \cdots, e_k^* = \min_{e_1, \cdots, e_k} \mathcal{L}$. Algorithm 1 in Appendix C provides the step-by-step procedure of post calibration.

4.3 Training Dataset

The training dataset integrates data from ARC-challenge (Clark et al., 2018) and CommonsenseQA (Talmor et al., 2019) into a unified structure designed for effective calibration learning. Each training instance is built from four essential components: questions from the source

Table 2: **Reduced Calibration Gap (↓ lower is better):** The percentage values indicate the relative difference in calibration error between the pre-calibrated and post-calibrated models, using the pre-calibrated quantized model as the reference. A negative percentage means the calibration error decreased by that amount, indicating successful calibration improvement. A positive percentage means the calibration error increased, indicating a decline in calibration quality. Differences are highlighted using colored cells: green for improvement and red for deterioration.

		ARC				CSQA			
		CCTP		VCAA		CCTP		VCAA	
Model	CE Metric	BNB	GPTQ	BNB	GPTQ	BNB	GPTQ	BNB	GPTQ
Llama-3.1-8B	BS	-11.42%	-16.30%	12.84%	-31.50%	-6.34%	16.36%	16.09%	-13.95%
	UBCE	-4.15%	-13.86%	-19.06%	-35.07%	-10.10%	9.48%	-22.17%	-26.22%
Mistral-v0.3-7B	BS	-42.99%	-40.60%	-27.11%	-27.87%	-42.26%	-35.14%	-44.04%	-26.16%
	UBCE	-6.98%	-7.17%	-29.69%	-22.81%	-11.82%	-9.78%	-46.32%	-22.68%
Qwen-2.5-7B	BS	2.57%	-9.60%	-53.38%	-49.87%	5.39%	-14.64%	-44.25%	-40.36%
	UBCE	43.76%	47.43%	-41.11%	-52.10%	52.56%	40.07%	-35.32%	-47.39%



(a) Distributions of CCTP for Correct Answers

(b) Distributions of VCAA for Incorrect Answers

Figure 2: Distributions of Confidence Scores for Calibrated Quantized Models on the ARC-Challenge Dataset

datasets, corresponding ground truth answers, model-generated responses, and associated correctness scores derived from our evaluation function. We construct these instances by combining soft prompt tokens with instruction prompts and question-answer pairs, using the correctness evaluation function's output as ground truth labels. To ensure dataset quality while maintaining computational efficiency, we implement a 4,000-sample cap per source dataset, using complete datasets when they fall below this threshold. Model responses and correctness scores are systematically collected following the Response Generation procedure outlined in our evaluation pipeline (detailed in Appendix E). This comprehensive approach yields a training dataset that maintains consistency across model variants while supporting effective calibration fine-tuning. The training resources usage is reported in Table 8.

4.4 Post Calibration Empirical Analysis

We evaluate the effectiveness of our proposed post-calibration method in terms of the calibration error of quantized LLMs. To be specific, the calibration error difference between post-calibrated and pre-calibrated models is given in Table 2, where a difference smaller than 0 indicates a post-calibrated model has a lower calibration error, indicating the effectiveness of the post-calibration process. On different models and datasets, almost 80% of the cases (38 out of 48) show a significantly lower calibration error after the post-calibration. These systematic improvements indicate that our proposed post-calibration process effectively addresses the discrepancy between confidence and correctness introduced by quantization.

We further give detailed analysis of confidence score distributions, conditioned on answer correctness to show this improvement. Specifically,

Figure 2 demonstrates that post-calibrated models achieve better alignment with the ideal calibration distribution. This improvement is quantitatively verified through calibration error measurements for both correct and incorrect answers. For the confidence scores conditioned on correct answers Figure 2 (a), the GPTQ quantized model's calibration error decreased from 0.39 to 0.19, while the BNB quantized model's error reduced from 0.37 to 0.24. In addition, incorrect answers in Figure 2 (b), the GPTQ model's calibration error improved from 0.85 to 0.74, and the BNB model showed a substantial reduction from 0.86 to 0.21. This improvement is consistent with the overall improvements in calibration performance in Table 2, indicating the effectiveness of our proposed post-calibration in reducing calibration error in quantized models.

Additionally, to evaluate the scalability of our approach, we examined the effect of both model size and the number of trainable soft-prompt tokens. For different scaled models, we conducted experiments on the ARC dataset using VCAA as confidence-score measures measured by UBCE across Qwen-2.5 models ranging from 0.5B to 32B parameters with both BNB and GPTQ quantization techniques. Results in Table 10 demonstrate that the problem of quantization-induced calibration degradation exists across all model sizes, indicating that larger models cannot inherently compensate for this effect. Importantly, our proposed calibration approach effectively reduced calibration error for both quantization methods. We further investigated the impact of soft-prompt token quantity using Mistral-v0.3-7B on the CommonsenseQA dataset, varying the count from 128 to 512 tokens. As shown in Table 9, increasing the number of soft-prompt tokens generally improved calibration performance, confirming that our approach scales effectively with both model size and prompt length.

4.5 Reliability Diagram of Calibration

To compare pre- and post-calibrated models, we adopt the reliability diagram to show their performance of binECE (Guo et al., 2017). Specifically, Figure 3 shows the binECE performance in terms of accuracy versus confidence, where the dashed black line represents perfect calibration—the ideal scenario where nominal predictions of correctness align exactly with empirical accuracy. Across Figure 3 (a)-(c), it is observed a consistent pattern of calibration error reduction from the post-calibrated models. This improvement can be implied by the decreased calibration gap between the top of the bins and the perfect calibration reference line when comparing the post-calibrated model to both the full precision and pre-calibrated models.

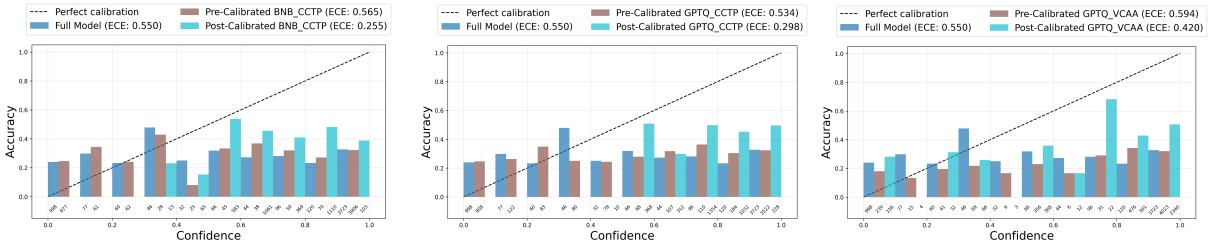
The post-calibrated model exhibits several empty bins in the range of low confidence. This pattern demonstrates improved calibration behavior. The absence of predictions in these ranges reflects the model's ability to consistently produce high confidence scores for the dataset that contains its own generated response, which predominantly contains correct answers. The concentration of confidence scores in high-accuracy bins aligns with the expected behavior for a well-calibrated model.

4.6 Confidence Score Adaptation

To examine the effectiveness of calibration on fine-tuned quantized models at the sample level, we conducted a case study using the ARC-Challenge dataset, as illustrated in Figure 4. While our previous analysis demonstrated improved calibration through summary statistics and confidence score distributions, this case study allows us to verify these improvements at the most granular level. Specifically, we analyzed how confidence scores changed for individual question-answer pairs, examining whether these scores better aligned with the ground truth correctness after fine-tuning. The results, shown in Figures 4 (a) and (b), demonstrate improved calibration performance for both BNB and GPTQ quantized Llama models for two cases which are the correct answer and incorrect answer, with updated confidence scores more accurately reflecting answer correctness comparing to the original confidence score generated by the uncalibrated quantized models.

4.7 Baseline Calibration Techniques Comparison

To contextualize the effectiveness of our approach, we benchmarked it against two widely used tuning-free calibration techniques—*temperature scaling* and *isotonic regression*. Both baselines learn a monotonic mapping from the model's raw confidence scores to calibrated probabilities: temperature scaling optimizes a single scalar temperature via negative log-likelihood, while isotonic regression fits a piece-wise constant non-decreasing



(a) CCTP of the Mistral BNB Quantized Model on the CSQA Dataset

(b) CCTP of the Mistral GPTQ Quantized Model on the CSQA Dataset

(c) VCAA of the Mistral GPTQ Quantized Model on the CSQA Dataset

Figure 3: Reliability Diagram for Evaluating the Calibration of Confidence Scores

Table 3: Brier Score (left block) and Upper-Bound Calibration Error (right block) for three calibration methods (↓ lower is better). Best value within each *(Calibration Error Metric, Quantization, Confidence Score)* column is **bold**.

Calib. Method	**Brier Score**				**UBCE**			
	GPTQ		**BNB**		**GPTQ**		**BNB**	
	CCTP	**VCAA**	**CCTP**	**VCAA**	**CCTP**	**VCAA**	**CCTP**	**VCAA**
Temperature Scaling	0.249	0.251	0.249	0.251	0.499	0.499	0.499	0.499
Isotonic Regression	0.261	0.248	0.298	**0.248**	0.472	0.492	0.468	0.492
Ours	**0.181**	**0.244**	**0.176**	0.286	**0.456**	**0.387**	**0.455**	**0.381**

function. Experiments were conducted with a quantized Mistral-v0.3-7B model on the ARC dataset. The comparison of calibration errors for the calibration approaches is summarized in Table 3. Our post-training calibration consistently outperforms both baselines on every metric, demonstrating superior alignment between predicted confidences and empirical accuracies.

4.8 Impact on Accuracy of Downstream Task and Generalization

Since calibrating confidence scores can inadvertently alter model outputs, we investigated the change in accuracy of the quantized model before and after calibration on the downstream tasks and cross-domain generalization tasks.

Downstream tasks. After calibration, we assessed answer accuracy on ARC and CommonsenseQA with the Mistral-v0.3-7B model (Table 4). Calibration increased average accuracy by 7%, indicating that our method simultaneously improves probabilistic reliability and task performance.

Generalization. To evaluate the impact of our calibration on the generalization tasks, we measured the accuracy of the pre- and post-calibrated Llama-3.1-8B-bnb model on the MMLU benchmark dataset (Table 11). The average accuracy decreased by less than 2%, indicating that our proposed post-calibration keeps the generalization of quantized models on other tasks.

Table 4: Down-stream accuracy on ARC and CSQA before and after our post-calibration of Mistral-7B-v0.3 model (↑ higher is better). Best accuracy is in **bold**.

Quant.	**Model**	**ARC**	**CSQA**
GPTQ	Baseline (Mistral)	0.648	0.455
	+ Post-Calibration	**0.746**	0.448
BNB	Baseline (Mistral)	0.660	0.555
	+ Post-Calibration	**0.683**	**0.717**

5 Related Work

Uncertainty quantification in large language models (LLMs) has emerged as a critical research area, leading to numerous approaches for estimating confidence scores. However, while uncertainty quantification in standard LLMs has received considerable attention, its application to quantized models remains understudied, particularly for generative tasks that characterize modern LLM applications.

Previous research in this domain has been limited in scope. Proskurina et al. (2024) investigated uncertainty calibration of quantized LLMs, but their work primarily addressed single-token classification in multiple-choice scenarios. This narrow focus leaves critical questions unanswered about calibration in more complex generative tasks, such as

Question: Beavers build their homes in ponds and streams. Which characteristic is least critical to building homes in an aquatic environment? **Choices:** ['waterproof fur', 'webbed hind feet', 'large, sharp teeth', 'flat, wide tail'].

Ground Truth Answer: large, sharp teeth

Model Generated Answer: large, sharp teeth
Answer Correctness: 1
Full Model VCAA: 0.75
BNB Model VCAA: 0.75 → 0.95
GPTQ Model VCAA: 0.25 → 0.5

(a) Correct answers.

Question: An atom of which element contains one more proton than one atom of chlorine (Cl)? **Choices:** ['sulfur (S)', 'argon (Ar)', 'fluorine (F)', 'bromine (Br)'].

Ground Truth Answer: argon (Ar)

Model Generated Answer: bromine (Br)
Answer Correctness: 0
Full Model VCAA: 0.75
BNB Model VCAA: 0.5 → 0
GPTQ Model VCAA: 0.5 → 0

(b) Incorrect answers.

Figure 4: Confidence of quantized LLMs between pre-calibration and post-calibration on correct answers (a) and incorrect answers (b). The question is from the ARC-Challenge dataset. The change in calibration is represented as (pre-calibrated confidence scores → post-calibrated confidence scores).

open-ended question answering.

Recent promising approaches to uncertainty quantification, while innovative, face substantial methodological challenges. Huang et al. (2023) introduced the Variation Ratio for Original Prediction (VRO), which employs a sentence transformer model to evaluate response consistency as a proxy for confidence (Reimers, 2019). However, this method encounters significant limitations in real-world applications. First, its effectiveness is compromised in scenarios with multiple valid responses, as it assumes the sentence transformer can accurately comprehend context and assess consistency. Furthermore, maintaining the effectiveness of these auxiliary models presents an ongoing challenge, as they must be continuously updated to keep pace with rapidly evolving LLM capabilities.

Another notable approach, proposed by Yadkori et al. (2024), measures response independence for confidence scoring. This method operates on the assumption that an LLM's generation probabilities for confident responses remain stable regardless of additional input information, such as hints or previous answers. However, this approach assumes that LLMs can robustly discriminate relevant prompt information—a capability that current open-source LLMs have not yet demonstrated, as they typically process all prompt information indiscriminately.

Kapoor et al. (2024) has shown promising results in reducing model calibration error through direct training on answer correctness evaluation. Their approach achieved notable improvements in Expected Calibration Error using a confidence score function similar to our proposed $CCTP$ approach. However, their work lacks the theoretical foundations necessary to explain these improvements, limiting our understanding of why these methods succeed and how they might be further enhanced.

Our work addresses these limitations by focusing on three key aspects: identifying the calibration gap in quantized LLMs, providing a theoretical foundation for understanding this gap, and introducing a post-calibration method with theoretical guarantees. This comprehensive approach advances the field beyond existing empirical studies and provides a more robust framework for uncertainty calibration in quantized LLMs.

6 Conclusion

In this work, we presented a systematic investigation of uncertainty calibration for quantized LLMs. First, we propose a upper bound calibration error (UBCE) that provides deeper insights on calibration performance. Second, our empirical analysis definitively establishes that quantized models consistently demonstrate higher calibration errors than full-precision ones. Finally, we propose an efficient method to recover the calibration gap of quantized models based on soft-prompt tuning. The tuning process can explicitly reduce the upper bound calibration error with a theoretical foundations. This facilitates more reliable quantization of LLMs in resource-constrained environments.

Acknowledgments

The authors gratefully acknowledge support from the National Science Foundation under grant numbers IIS-2450662 and IIS-2450671. We are also grateful to the anonymous reviewers for their insightful questions and constructive feedback, which significantly contributed to the enhancement of this paper.

Limitations

As the demand for more sophisticated LLMs continues to grow, so does their environmental impact. This limitation underscores the urgent need to explore and adopt more sustainable practices and technologies in the development and learning to calibrate with fewer optimization steps to mitigate their ecological footprint.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Junaid Bajwa, Usman Munir, Aditya Nori, and Bryan Williams. 2021. Artificial intelligence in healthcare: transforming the practice of medicine. *Future healthcare journal*, 8(2):e188–e194.

Glenn W Brier. 1950. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Tim Dettmers, Elias Frantar, Saleh Ashkboos, Alexander Deji, Zachary DeVito, Rishi Rajani, Zachary Lubich, Thomas Bettels, Rodrigo Yulo, and Max Bileschi. 2022. bitsandbytes: State-of-the-art 8-bit optimizers for PyTorch.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Yuheng Huang, Jiayang Song, Zhijie Wang, Shengming Zhao, Huaming Chen, Felix Juefei-Xu, and Lei Ma. 2023. Look before you leap: An exploratory study of uncertainty measurement for large language models. *arXiv preprint arXiv:2307.10236*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 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)*, pages 1–14.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Irina Proskurina, Luc Brun, Guillaume Metzler, and Julien Velcin. 2024. When quantization affects confidence of large language models? *arXiv preprint arXiv:2405.00632*.

N Reimers. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, et al. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

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

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing.

Zhaozhuo Xu, Zirui Liu, Beidi Chen, Shaochen Zhong, Yuxin Tang, WANG Jue, Kaixiong Zhou, Xia Hu, and Anshumali Shrivastava. Soft prompt recovers compressed llms, transferably. In *Forty-first International Conference on Machine Learning*.

Yasin Abbasi Yadkori, Ilja Kuzborskij, András György, and Csaba Szepesvári. 2024. To believe or not to believe your llm. *arXiv preprint arXiv:2406.02543*.

Appendix

A Derivation of UBCE Definition and Inequalities

The Upper Bound Calibration Error (UBCE) could be derived as an upper bound for both Brier Score and binnedECE.

Recall the definition of binnedECE,

$$\text{binECE}(\hat{c}, \mathcal{I}) \triangleq \sum_{j\in[m]} |\mathbb{E}_{c\sim\{0,1\}}[(\hat{c}-c)I(\hat{c}\in\mathcal{I}_j)]|$$

where $\mathcal{I} = \mathcal{I}_1, \ldots, \mathcal{I}_m$ defines the partition set. Then, we will show that UBCE is an upper bound for binnedECE.

$$\begin{aligned}
\text{binnedECE} &= \sum_{j\in[m]} |\mathbb{E}[(\hat{c}-c)I(\hat{c}\in I_j)]| \\
&\le \sum_{j\in[m]} \mathbb{E}[|(\hat{c}-c)I(\hat{c}\in I_j)|] \\
&\quad \text{by Jensen's inequality} \\
&= \sum_{j\in[m]} \mathbb{E}[\mathbb{E}[|(\hat{c}-c)\cdot I(\hat{c}\in I_j)|\big|c]] \\
&\quad \text{by tower rule} \\
&= P(c=0) \\
&\quad \times \sum_{j\in[m]} \mathbb{E}[|(\hat{c}-0)\cdot I(\hat{c}\in I_j)|\big|c=0] \\
&\quad + P(c=1) \\
&\quad \times \sum_{j\in[m]} \mathbb{E}[|(\hat{c}-1)\cdot I(\hat{c}\in I_j)|\big|c=1] \\
&= P(c=0) \\
&\quad \times \sum_{j\in[m]} \mathbb{E}[\hat{c}\cdot I(\hat{c}\in I_j)|c=0] \\
&\quad + P(c=1) \\
&\quad \times \sum_{j\in[m]} \mathbb{E}[(1-\hat{c})\cdot I(\hat{c}\in I_j)|c=1] \\
&\quad \text{since } \hat{c}\in[0,1] \\
&= P(c=0) \\
&\quad \times \mathbb{E}[\sum_{j\in[m]} \hat{c}\cdot I(\hat{c}\in I_j)|c=0] \\
&\quad + P(c=1) \\
&\quad \times \mathbb{E}[\sum_{j\in[m]} (1-\hat{c})\cdot I(\hat{c}\in I_j)|c=1] \\
&= P(c=0)\mathbb{E}[\hat{c}|c=0] \\
&\quad + P(c=1)(1-\mathbb{E}[\hat{c}|c=1]) \\
&= UBCE
\end{aligned}$$

Similarly, recall the definition of Brier Score,

$$\text{BS} \triangleq \mathbb{E}_{c\sim\{0,1\}}[(\hat{c}-c)^2].$$

Then, we will show that UBCE is an upper bound for the Brier Score:

$$\begin{aligned}
\widehat{BS} &= \frac{1}{n}\sum_i (\hat{c}_i - c_i)^2 \\
&= \frac{1}{n}\sum_i \hat{c}_i^2 + c_i^2 - 2\hat{c}_i c_i \\
&\text{Since } \hat{c}_i^2 \le \hat{c}_i \text{ for } \hat{c}_i \in [0,1] \\
&\text{and } c_i^2 = c_i \text{ for } c_i \in \{0,1\} \\
&\le \frac{1}{n}\sum_i \hat{c}_i + c_i - 2\hat{c}_i c_i \\
&= \frac{1}{n}\sum_i c_i \cdot (1-\hat{c}) + (1-c_i)\cdot \hat{c}_i \\
&= \widehat{UBCE}
\end{aligned}$$

B Derivation of UBCE $\le$ BCE

Denote

- n_0 as the number of correct answers
- n_1 as the number of incorrect answers
- $n = n_0 + n_1$ as the total number of answers

The sample estimator of $UBCE$ could be obtained naturally from its formal definition:

$$\begin{aligned}
UBCE &= P(c=0)\mathbb{E}[\hat{c}|c=0] \\
&\quad + P(c=1)(1-\mathbb{E}[\hat{c}|c=1]) \\
\widehat{UBCE} &= \frac{n_0}{n_0+n_1}\frac{1}{n_0}\sum_i \hat{c}_i \cdot I(c_i=0) \\
&\quad + \frac{n_1}{n_0+n_1}\frac{1}{n_1}\sum_i (1-\hat{c}_i)\cdot I(c_i=1) \\
&= \frac{1}{n_0+n_1}\sum_i \hat{c}_i \cdot I(c_i=0) \\
&\quad + (1-\hat{c}_i)\cdot I(c_i=1) \\
&= \frac{1}{n_0+n_1}\sum_i \hat{c}_i \cdot (1-c_i) \\
&\quad + (1-\hat{c}_i)\cdot c_i \\
&= \frac{1}{n_0+n_1}\sum_i c_i \cdot (1-\hat{c}_i) \\
&\quad + (1-c_i)\cdot \hat{c}_i \\
&= \frac{1}{n}\Big[\sum_i c_i\cdot(1-\hat{c}_i) + (1-c_i)\cdot\hat{c}_i\Big]
\end{aligned}$$

Similarly, the sample estimator of Binary Cross Entropy (BCE) could be obtained as:

$$\begin{aligned}
\text{BCE} &= c\cdot -\log(\hat{c}) + (1-c)\cdot -\log(1-\hat{c}) \\
\widehat{\text{BCE}} &= \frac{1}{n}\sum_i \Big[c_i \cdot -\log(\hat{c}_i) \\
&\qquad + (1-c_i)\cdot -\log(1-\hat{c}_i)\Big]
\end{aligned}$$

Putting the sample estimator of BCE and UBCE together and compare every term,

$$\begin{aligned}
\widehat{UBCE} &= \frac{1}{n}\sum_i \Big[c_i\cdot(1-\hat{c}_i) \\
&\qquad + (1-c_i)\cdot\hat{c}_i\Big] \\
\widehat{\text{BCE}} &= \frac{1}{n}\sum_i \Big[c_i \cdot -\log(\hat{c}_i) \\
&\qquad + (1-c_i)\cdot -\log(1-\hat{c}_i)\Big]
\end{aligned}$$

Note that $(1-\hat{c}_i) \le -\log(\hat{c}_i)$ and $\hat{c}_i \le -\log(1-\hat{c}_i)$ for $\hat{c}_i \in [0,1]$ with log base 2 and e. Therefore, $\widehat{UBCE} \le \widehat{BCE}$ for log with base 2 and e. Strict equality occurs only at optimality.

C Soft Prompt Tuning Details

The calibration process employs Supervised Fine-Tuning (SFT) with Cross Entropy loss, implemented through PyTorch. The initial soft prompt text is included in Appendix D. The pseudo code of soft prompt tuning is provided below in Algorithm 1. Specifically, soft prompt tuning begins by initializing a sequence of learnable embeddings $\mathbf{e}_1, ...\mathbf{e}_k$ that act as 'virtual tokens' with text prompt detailed in Appendix D (line 2). We then freeze all the weights of the quantized LLM and train only these new embeddings (line 3). For each training example $(\mathbf{x}, y)$ from the dataset, we first tokenize the input $\mathbf{x}$ into numerical tokens $\tilde{\mathbf{x}}$ (line 6). Next, we prepend the learnable embeddings $[\mathbf{e}_1, ...\mathbf{e}_k]$ to these tokens, forming the combined input $\tilde{\mathbf{E}} = [\mathbf{e}_1, ...\mathbf{e}_k, \tilde{\mathbf{x}}]$ (line 7). We feed this augmented sequence into the model and compute the perplexity loss by comparing the model's output to the target y (line 8). Finally, we adjust the soft prompt embeddings to minimize that loss (line 9), while leaving the LLM itself untouched. This iterative process continues across the entire training set, yielding a tailored prompt that effectively directs the frozen model toward producing the correct output. Please see Figure 5 for a diagram that provides a holistic overview of the soft prompt tuning

Algorithm 1: Soft-Prompt Learning for a Quantized LLM

Input: Quantized LLM f, training data $\mathcal{D}$
Output: Soft-prompt embeddings $\mathbf{e}_1, \ldots, \mathbf{e}_k$

```
###-- Initialization --###
```

Initialize $\mathbf{e}_1, \ldots, \mathbf{e}_k$ with textual tokens or random vectors

Freeze all parameters of f; **Only** $\mathbf{e}_1, \ldots, \mathbf{e}_k$ are trainable

```
###-- Optimization loop --###
```

foreach $(\mathbf{x}, y) \sim \mathcal{D}$ **do**

  $\tilde{\mathbf{x}} \leftarrow \text{Tokenize}(\mathbf{x})$

  $\tilde{\mathbf{E}} \leftarrow [\mathbf{e}_1, \ldots, \mathbf{e}_k, \tilde{\mathbf{x}}]$

  $\mathcal{L} \leftarrow \text{PPL}\big(f(\tilde{\mathbf{E}}), y\big)$ from Eq. (7)

  Update $\mathbf{e}_1, \ldots, \mathbf{e}_k$ to minimize $\mathcal{L}$

end

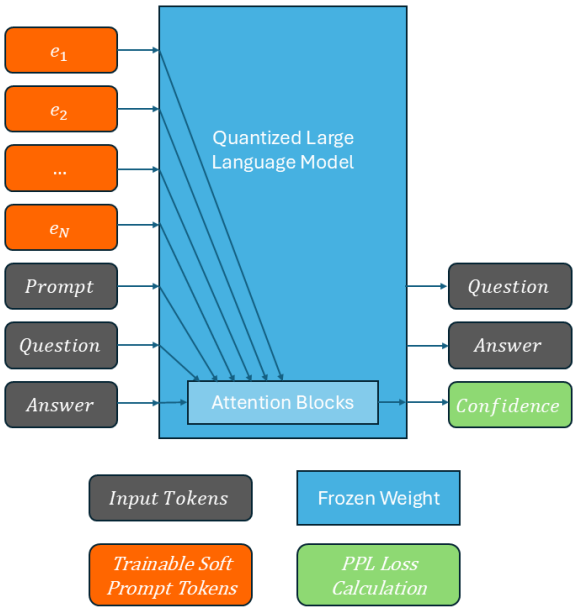


Figure 5: Visual representation of the Soft Prompt Tuning process for a Quantized Large Language Model. Trainable soft prompt tokens are prepended to input tokens and processed by a frozen LLM. The PPL loss on the confidence prediction is used to update only the soft prompts.

process used for the uncertainty calibration of a quantized LLM.

Specifically, the training hyperparameter used in the fine-tuning is in the Table 5.

Table 5: Hyper-parameters used for soft-prompt tuning

Hyper-parameter	Value
Task type	CAUSAL-LM
Prompt-tuning initialization	Text
Number of virtual tokens	256
Optimizer	AdamW
LR scheduler	Linear
Warm-up steps (%)	3 %
Learning rate	10^{-5}
Batch size	1
Fine-tuning epochs	50

Complete training performance metrics are documented in Table 6.

D Initialization Prompt for Soft Prompt Tuning

"Before formulating your response, take the necessary time to thoroughly read and analyze the provided instructions and the question. Ensure that you fully comprehend all requirements, constraints, and nuances of the request. Carefully consider how to structure your response to address each component effectively while adhering to any specified formatting or style guidelines. Once you have achieved a complete understanding of the question, construct a comprehensive and well-structured response that reflects your careful consideration of all aspects of the task. Pay close attention to the context of the question, recall all relevant information, and use it appropriately into your answer.Both the answer to the question and your response should be accurate, concise, and directly address the question without unnecessary elaboration. If any of these criteria are not met-such as incorrect information, excessive verbosity, or failure to directly answer the question-the response should be considered incorrect or inadequate and revised accordingly."

E Evaluation Pipeline

The evaluation pipeline systematically assesses calibration performance through a four-stage process that generates model responses, computes confidence scores, evaluates calibration metrics, and quantifies calibration gaps between full-precision and quantized models.

1. **Response Generation.** Model responses are collected through iterative prompting, where each subsequent generation incorporates all previous responses in the input prompt. The

process employs greedy decoding for reproducibility and continues for seven iterations, yielding between one and seven unique responses per model (allowing for repetition). Each response is then evaluated for correctness using dataset-specific evaluation functions.

2. **Confidence Score Computation.** Confidence scores are generated for the complete set of responses collected across all models. This comprehensive approach requires each model, whether quantized or full-precision, to evaluate both its own responses and those generated by other models, ensuring consistent comparison across the unified response set.

3. **Calibration Metric Assessment.** Calibration errors are computed by measuring the alignment between predicted confidence scores and actual correctness values. These metrics quantify how well each model's confidence predictions correspond to its true performance.

4. **Comparative Analysis.** The calibration performance gap between quantized and full-precision models is expressed as a relative percentage difference:

$$\text{Difference (\%)} = \frac{(CE_Q - CE_F) \times 100}{CE_F} \tag{8}$$

where CE_Q and CE_F represent the calibration errors of the quantized and full-precision models, respectively. This metric provides a direct measure of how quantization affects calibration performance relative to the full-precision baseline.

F Training Result

Table 6: Training and validation loss for each 7B quantized model.

Model	Quantization	Train loss	Valid loss
Llama	BNB	0.11	0.09
Llama	GPTQ	0.11	0.09
Mistral	BNB	0.19	0.21
Mistral	GPTQ	0.19	0.25
Qwen	BNB	0.15	0.05
Qwen	GPTQ	0.14	0.06

G Computational Resource

Table 7: Configuration of Computational Resource

Name	Value
Computing Infrastructure	GPU
GPU Model	NVIDIA-A100
GPU Memory	80GB
GPU Number	1
CUDA Version	12.1
CPU Memory	128GB

H Training Resource Usage

Table 8: Training throughput and peak GPU memory for different quantization schemes.

Quantization	Training speed (samples/s)	Memory usage (MB)
None	10	23 257
GPTQ	4	24 643
AWQ	3	13 243
BNB	6	13 661

I Calibration Error for Different Number of Soft Prompt Tokens

Table 9: Calibration error for different number of soft prompt tokens after fine-tuning.

Measure	Prompt Tokens	UBCE	BS
CCTP	128	0.504	0.299
	256	0.505	0.240
	512	0.447	0.276
VCAA	128	0.431	0.313
	256	0.380	0.303
	512	0.351	0.285

J Model and Packages

All models are sourced from the Huggingface platform (Wolf et al., 2020). The list of models used are:

1. Qwen/Qwen2.5-7B-Instruct-GPTQ-Int4
2. unsloth/Qwen2.5-7B-bnb-4bit
3. Qwen/Qwen2.5-7B-Instruct
4. parasail-ai/Mistral-7B-Instruct-v0.3-GPTQ-4bit

Table 10: Upper Bound Calibration Error (UBCE) for varying sizes of Qwen-2.5 models evaluated on the ARC dataset using VCAA confidence scores (↓ Lower is better). Model size is denoted in billions of parameters (B).

Model Type	**0.5B**	**1.5B**	**3B**	**7B**	**14B**	**32B**
Full Model	0.666	0.460	0.394	0.357	0.436	0.187
GPTQ Model	0.482	0.473	0.408	0.441	0.443	0.184
Calibrated GPTQ Model	0.253	0.454	0.321	0.216	0.471	0.145
BNB Model	0.603	0.482	0.358	0.449	0.457	0.189
Calibrated BNB Model	0.259	0.469	0.319	0.270	0.415	0.165

5. unsloth/mistral-7b-instruct-v0.3-bnb-4bit
6. hugging-quants/Meta-Llama-3.1-8B-Instruct-GPTQ-INT4
7. unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit
8. meta-llama/Llama-3.1-8B-Instruct
9. mistralai/Mistral-7B-Instruct-v0.3

The experiments were conducted using the vLLM (Kwon et al., 2023) framework and the Huggingface Transformers library.

K MMLU Accuracy

Table 11: MMLU accuracy of `Llama-3.1-8B-BNB` before and after calibration. Δ denotes the accuracy gap. The overall accuracy drop is $< 2\%$, indicating generalization is preserved.

Category	**Pre-Calib.**	**Post-Calib.**	Δ
subcat:math	0.472	0.444	−0.028
subcat:health	0.717	0.681	−0.036
subcat:physics	0.577	0.559	−0.017
subcat:business	0.682	0.668	−0.014
subcat:biology	0.782	0.784	0.002
subcat:chemistry	0.564	0.528	−0.036
subcat:computer science	0.638	0.590	−0.049
subcat:economics	0.685	0.658	−0.027
subcat:engineering	0.634	0.607	−0.028
subcat:philosophy	0.631	0.606	−0.025
subcat:other	0.768	0.768	0.000
subcat:history	0.767	0.763	−0.003
subcat:geography	0.843	0.823	−0.020
subcat:politics	0.781	0.779	−0.002
subcat:psychology	0.786	0.762	−0.024
subcat:law	0.532	0.527	−0.005
subcat:sociology	0.866	0.846	−0.020
cat:STEM	0.581	0.556	−0.025
cat:humanities	0.620	0.607	−0.013
cat:social sciences	0.769	0.749	−0.019
cat:other	0.723	0.701	−0.022
overall	0.667	0.649	−0.019

L Code

The source code can be found on GitHub.