# GrACE: A Generative Approach to Better Confidence Elicitation in Large Language Models

**Anonymous ACL submission**


## Abstract

Assessing the reliability of Large Language Models (LLMs) by confidence elicitation is a prominent approach to AI safety in high-stakes applications, such as healthcare and finance. Existing methods either require expensive computational overhead or suffer from poor calibration, making them impractical and unreliable for real-world deployment. In this work, we propose GrACE, a Generative Approach to Confidence Elicitation that enables scalable and reliable confidence elicitation for LLMs. GrACE adopts a novel mechanism in which the model expresses confidence by the similarity between the last hidden state and the embedding of a special token appended to the vocabulary, in real-time. We fine-tune the model for calibrating the confidence with targets associated with accuracy. Extensive experiments show that the confidence produced by GrACE achieves the best discriminative capacity and calibration on open-ended generation tasks without resorting to additional sampling or an auxiliary model. Moreover, we propose two confidence-based strategies for test-time scaling with GrACE, which not only improve the accuracy of the final decision but also significantly reduce the number of required samples, highlighting its potential as a practical solution for deploying LLMs with reliable, on-the-fly confidence estimation.


## 1 Introduction

Large Language Models (LLMs) have become popular solutions to knowledge-intensive and logically complex problems for their capability of utilizing and reasoning over a huge amount of knowledge they learn (Achiam et al., 2023; Dubey et al., 2024; Anthropic, 2024; Guo et al., 2025; Yang et al., 2025a). However, the reliance on LLMs raises broad concerns about the possibility that LLMs output seemingly correct but factually hallucinated responses, especially in high-stakes domains such as healthcare (Singhal et al., 2023), law (Magesh et al., 2024), and finance (Xie et al., 2024). Therefore, it is critical for LLMs to "know what they know" (Kadavath et al., 2022) to become more reliable and gain users' trust in their deployment.

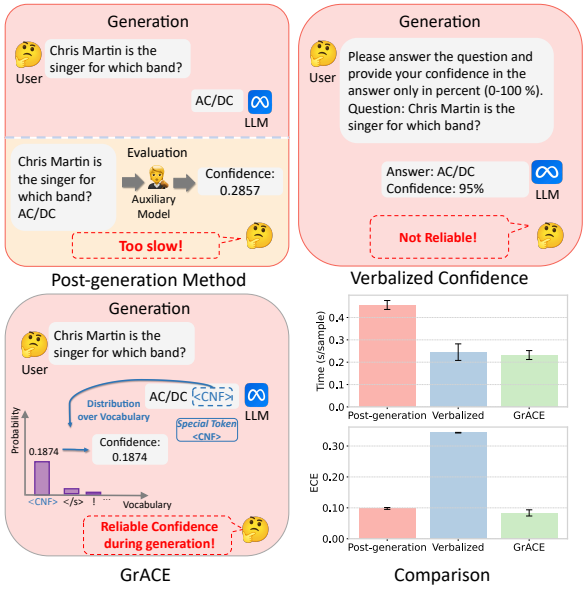


Figure 1: **Comparison among the workflow and performance of post-generation methods, verbalized confidence, and GrACE.** [1].

Estimating model confidence is a promising approach to assessing the reliability of model predictions. A desirable confidence score should exhibit strong separability between positive and negative samples (discrimination) and align closely with actual correctness (calibration). Although confidence elicitation and calibration have been extensively studied in classification tasks (Guo et al., 2017; Desai and Durrett, 2020; Li et al., 2025), recent attention has shifted to open-ended generation tasks

[1]We report time and Expected Calibration Error (ECE) using two representative methods in post-generation methods and verbalized confidence, namely $P(\text{True})$ (Kadavath et al., 2022) and verbalized confidence (Tian et al., 2023). The experiment is conducted with the Llama2-7B model on the TriviaQA dataset, using a single NVIDIA GeForce RTX 3090 GPU.

(e.g., question answering) for LLMs. Nevertheless, as illustrated in Figure 1, existing methods exhibit limitations in *when* a model expresses confidence and *how well* that confidence is calibrated, which constrains their practical utility in applications such as Test-Time Scaling (TTS) (Brown et al., 2024). Post-generation methods, which assign the precise confidence score after generation, dominate current confidence elicitation approaches. They elicit confidence either by prompting the original model again (Kadavath et al., 2022; Zhang et al., 2024) or using auxiliary models (Ulmer et al., 2024; Kapoor et al.). Such methods require an additional confidence evaluation stage after generation, increasing latency and degrading human-computer interaction. Verbalized confidence offers a generative alternative by producing numerical values (Tian et al., 2023; Xiong et al.) or linguistic expressions (e.g.,most likely, likely, not sure) (Band et al., 2024; Yang et al., 2024b, 2025b) during the generation stage. However, verbal expressions of confidence are inherently vague as the discrete vocabulary fails to capture nuance differences in confidence – for example, the confidence intervals for "most likely", "likely", and "unlikely" are neither standardized nor quantifiable. Moreover, LLMs tend to be overconfident when expressing verbal confidence, potentially due to too much focus on imitating human expressions (Xiong et al.).

In this paper, we enable LLMs to learn the true distribution of the prediction correctness and express the confidence with **G**ene**r**ative c**A**librated **C**onfidence **E**licitation (GrACE), making it an efficient and reliable alternative for existing methods. GrACE adds a special token <CNF> to the vocabulary and teaches the model to generate it at the end of the response for confidence elicitation. Specifically, we fine-tune the model to express confidence using the similarity between the last hidden state and the embedding of <CNF>, leveraging the inherent monotonicity and simplicity of the similarity function. To ensure the elicited confidence is discriminative and calibrated, we construct a calibration set where training samples are grouped according to the internal confidence of the model (Liu et al., 2024; Ji et al., 2025) and assign group accuracy as the calibration target. To reduce the cost of fine-tuning, we only fine-tunes LoRA (Hu et al.) and the embedding of <CNF> when implementing GrACE. The fine-tuning cost is amortised by the learned scalable confidence elicitation mechanism that generalizes across prompts and domains without an additional evaluation stage at test time. The on-the-fly confidence elicitation mechanism, together with discriminative and reliable confidence scores, makes GrACE naturally well-suited for applications such as TTS, which demand dependable decision-making and real-time responses.

Our contributions are summarized as follows:

- We propose a generative confidence elicitation method GrACE, which provides a novel and scalable mechanism for discriminative and calibrated confidence elicitation on-the-fly with negligible computation overhead.
- We design two strategies that integrate the real-time confidence scores generated by GrACE into the aggregation and adaptive sampling process of test-time scaling, improving both efficacy and efficiency.
- Extensive experiments on open-ended generation tasks and test-time scaling demonstrate that GrACE not only produces more accurate confidence estimates with strong generalization to unseen domains, surpassing the state-of-the-art method (Apricot (Ulmer et al., 2024)) by 2.6%, 1.8%, and 4.4% in terms of ECE, Brier score, and AUROC, respectively, but also provides reliable guidance that improves accuracy by an average of 1.54% while substantially reducing the number of samples required during test-time scaling.

## 2 Related Works

**Post-generation Confidence Elicitation.** Unlike model uncertainty, which captures the variability in a model's responses to questions (Kuhn et al.; Zhang et al., 2025), confidence elicitation directly assigns an explicit confidence score to the model's prediction. Post-generation methods follow the generate-then-evaluate paradigm, which first acquires model generations, and then evaluates confidence score with an auxilarity model. Kadavath et al. (2022) proposes $P$(True), which asks the model to self-evaluate its confidence on generation by converting the open-ended task to the multiple choice setting with careful prompting. Further works (Kapoor et al.) calibrate $P$(True) by fine-tuning the model with LoRA using binary labeled data. As for more efficient solutions (Azaria and Mitchell; Kossen et al., 2024; Liu et al., 2024), confidence can be detected from the model's hidden

state with a linear probe. Post-generation methods are also applicable to black / gray-box models. Ulmer et al. (2024) trains an auxiliary model with semantic-aware targets which predicts model confidence based on generation only. GrACE differs from post-generation methods by eliminating the separate evaluation stage and producing the confidence score in the generation stage, thereby reducing computational overhead and accelerating the confidence elicitation process.

**Verbalized Confidence Elicitation.** Another branch of confidence elicitation aims to teach the models to express their confidence in natural language. Tian et al. (2023) designs different prompting strategies to lead the model to output a calibrated confidence score. Some works (Band et al., 2024; Yang et al., 2024b; Hager et al., 2025) endow the model with the ability to use confidence-related phrases (e.g., not sure, uncertain) to express confidence as humans do. Cohen et al. (2024) asks the model to generate a special token `[idk]` when it feels uncertain about the responses. Although verbalized confidence is a natural and efficient approach to eliciting model confidence, LLMs are found to be overconfident when verbalizing their confidence (Xiong et al.). Moreover, the confidence in the form of phrases conveys vague and unquantifiable information, restricting its application to broad topics.

**Test-Time Scaling.** Test-Time Scaling (TTS) (Snell et al., 2024) improves the response quality of LLMs with additional computing during the inference stage. Repeated sampling (Brown et al., 2024) with decision-making rules such as Self-Consistency (Wang et al., 2022) is a common practice for selecting high-quality samples for given prompts. However, this process requires a fixed sample size during TTS, which is not always necessary (Chen et al., 2024). While some works investigate optimal computational resources allocation (Snell et al., 2024; Muennighoff et al., 2025), there is research focusing on dynamically adjusting the sample size. Some attempts (Li et al.; Aggarwal et al.) determine when to terminate sampling by pre-sampling and calculating statistical features within the pre-sample set. As a metric to evaluate the response quality, confidence is gradually applied to the decision-making process for improving TTS. Taubenfeld et al. (2025); Huang et al. (2025) optimize the sample size through the lens of confidence and calibration. However, these works are based on *P*(True) which requires an additional evaluation stage as previously noted. Concurrent work (Fu et al., 2025) leverages confidence to filter out low-quality reasoning traces but does not account for calibration, resulting in threshold selection without risk guarantees. Our work enables on-the-fly elicitation of discriminative and well-calibrated confidence, supporting efficient, effective, and risk-controlled TTS.

## 3 Methodology

In this section, we introduce the formulation of the generative confidence elicitation problem in §3.1 and the GrACE framework in §3.2, including the form of confidence elicitation in GrACE §3.2.1, the calibration target §3.2.2, training process §3.2.3, and GrACE's application on test-time scaling §3.2.4.

### 3.1 Problem Formulation

Given a prompt $\boldsymbol{x}$ and a model $\mathcal{M}$, we define the generative confidence (GenConf) $c$ as:

$$(\boldsymbol{y}_i, \boldsymbol{z}, c_i) = \mathcal{M}(\boldsymbol{x}),\ c_{\mathrm{i}} \in \mathbb{R}^+, \tag{1}$$

where $\boldsymbol{y}_i = \{y_i^1, ..., y_i^N\}$ is the $i$-th sample, $\boldsymbol{z} = \{z_1, ..., z_N\}$ is the model's hidden state, $c_i$ is the confidence score corresponding to the sample $\boldsymbol{y}_i$. Note that Eq.1 requires no auxiliary model and the confidence $c$ should be a real number extracted during generation rather than a textual string. To be effective and trustworthy in downstream applications, the GenConf $c$ should exhibit both discrimination and calibration. Formally:

$$\mathbb{P}(c_m > c_n | f(\boldsymbol{y}_m) = 1, f(\boldsymbol{y}_n) = 0) = 1, \tag{2}$$

$$\mathbb{P}(f(\boldsymbol{y}) = 1 | \boldsymbol{x}, \boldsymbol{y}, c) = c, \tag{3}$$

where $f(\cdot)$ is a function which assigns 1 to the correct response and 0 to the wrong response. Given these notions, discrimination requires $c$ to correctly rank positive samples above negative ones, whereas calibration requires $c$ to reflect the actual accuracy of the model's responses.

### 3.2 GrACE

We describe the GrACE framework (illustrated in Figure 2) for teaching the model to produce GenConf in this section.

#### 3.2.1 Confidence Elicitation in GrACE

Ideally, the confidence should be expressed in continuous space during generation as proposed in

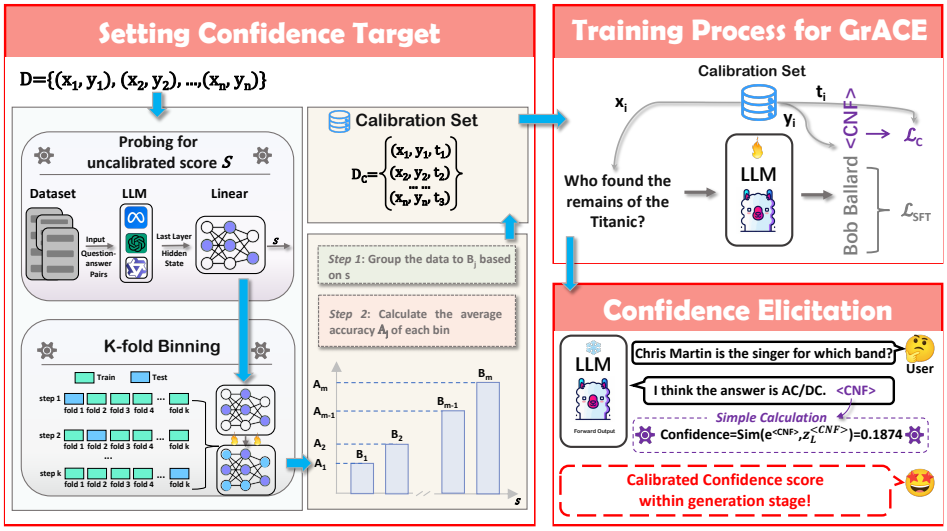


Figure 2: **The illustration of GrACE framework.** We group the calibration data using the model's self-awareness and use the accuracy of each group as the calibration target (§3.2.2). The model is trained with the combination of calibration loss and supervised fine-tuning loss (§3.2.3). The confidence is elicited by the distance between the hidden state and <CNF> embedding (§3.2.1).

§3.1. Inspired by the pause token mechanism (Goyal et al.; Wang et al.) which prompts the model to "think" by generating a sequence of special, non-semantic tokens before reasoning, we introduce a new token, <CNF>, into the vocabulary to encourage the reflective behavior of the model in the latent space. Specifically, we append <CNF> to the end of each question-answer pair. We train the model to generate <CNF> at the end of its response, thereby enabling an additional forward pass in which the model can reflect on its output and estimate confidence in the latent space. In what follows, we describe how the model expresses its confidence in the latent space.

Let $\boldsymbol{e}^{\texttt{<CNF>}}$ be the embedding of the new token <CNF>, and $\boldsymbol{z}_L^{\texttt{<CNF>}}$ be hidden states in the last layer $L$ corresponding to the <CNF>. Then, the confidence is estimated as the similarity between $\boldsymbol{e}^{\texttt{<CNF>}}$ and $\boldsymbol{z}_L^{\texttt{<CNF>}}$, that is:

$$c = \text{sim}(\boldsymbol{z}_L^{\texttt{<CNF>}}, \boldsymbol{e}^{\texttt{<CNF>}}). \tag{4}$$

In causal language models, the language modeling head (typically a linear projection followed by a softmax) serves as a mechanism to project the last hidden state to the vocabulary. This projection implicitly evaluates the similarity between the last hidden state and each token embedding in the vocabulary. We leverage this built-in capability as the foundation for defining function $\text{sim}(\cdot)$, aligning it with the model's native representation geometry:

$$c = \text{softmax}(\boldsymbol{E}\boldsymbol{z}_L^{\texttt{<CNF>}})_{\texttt{<CNF>}}, \tag{5}$$

where $\boldsymbol{E} \in \mathbb{R}^{(|\mathcal{V}|+1)\times d}$ is the embedding matrix of the vocabulary, $d$ is the dimension of token embeddings, the subscript <CNF> is the position associated with <CNF>. Note that the position of <CNF> is flexible depending on the segment that the users care about, we provide the experiment results with <CNF> prepending the response in Ablation Study in Appendix B.4.

### 3.2.2 Setting Target Confidence for GrACE

To guide the model to generate a discriminative and well-calibrated confidence score $c$, it is important to set target confidences for responses $\boldsymbol{y}$ that are distributionally sharp and numerically aligned with their actual probability of correctness (Kuleshov and Liang, 2015). Traditional methods use binary labels as confidence targets, leading to only a sharp distribution with poor calibration (Niculescu-Mizil and Caruana, 2005). We propose to smooth the confidence target and ensure calibration by grouping the training data and assigning the empirical accuracy within each group as the target confidence. The grouping is determined by the outputs of a linear probe trained with binary labels, which helps ensure a sharper distribution. Specifically, given training set $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, f(\boldsymbol{y}_i))\}_{i=1}^{|\mathcal{D}|}$, we probe an uncalibrated confidence score $s$ from the last hidden layer $L$ of the model for generation in sequence length $N$ with a linear probe parameterized with $\theta$, formulated as:

$$s = \boldsymbol{W}_\theta^T \bar{\boldsymbol{z}}_L + \boldsymbol{b}_\theta, \quad \bar{\boldsymbol{z}}_L = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{z}_L^i \tag{6}$$

We perform k-fold binning (Liu et al., 2024) to align $s$ from probing with response accuracy for every training sample. Specifically, we partition the dataset into $k$ folds, iteratively train the linear probe on $k-1$ folds with mean squared error loss, and test on the remaining fold until all the samples in the training data $(\boldsymbol{x}, \boldsymbol{y})$ are annotated with $s$. To convert $s$ to response accuracy, we group the data into $M$ bins with equal intervals according to $s$ and calculate the accuracy $\mathrm{A}_m$ in each bin $m$. Let the target $t_i = A_m$ if the data point $i \in \mathcal{B}_m$, where $\mathcal{B}_m$ denotes the set of indices belonging to bin $m$. The resulting calibration set is defined as $\mathcal{D}_c = \{(\boldsymbol{x}_i, \boldsymbol{y}_i, t_i)\}_{i=1}^{|\mathcal{D}_c|}$. In this way, we obtain targets for GrACE that are numerically equal to the accuracies within each group and sharply distributed, as the probe is trained to assess the factuality of the question–answer pairs. In Appendix B.1 and B.2, we plot the kernel density estimation of the target confidence and demonstrate that aligning the predicted confidence with the target confidence $t$ improves the calibration by reducing the expected calibration error.

#### 3.2.3 Training Process

We optimize the model to align its self-reflective confidence $c$ with the calibration target $t$ using the Mean Squared Error (MSE) as calibration loss. However, fine-tuning the model towards only the calibration objective will sacrifice the model's overall performance. To maintain the model utility, we adopt the supervised fine-tuning (SFT) loss on the original generation of the model as a regularizer. Note that we exclude the <CNF> from the calculation of SFT loss to avoid the conflict between the two objectives. The total loss is formulated as:

$$\begin{aligned}\mathcal{L}_T &= \mathcal{L}_{\mathrm{C}} + \gamma \mathcal{L}_{\mathrm{SFT}} \\ &= \frac{1}{|\mathcal{D}_c|} \sum_{i=1}^{|\mathcal{D}_c|} \left(t_i - \mathrm{sim}(\boldsymbol{z}_L^{\texttt{<CNF>}}, e^{\texttt{<CNF>}})\right)^2 - \gamma \log p_\theta(\boldsymbol{y}_i \mid \boldsymbol{x}_i)\end{aligned} \tag{7}$$

where $\gamma$ is a hyperparameter for balancing the loss terms $\mathcal{L}_{\mathrm{C}}$ and $\mathcal{L}_{\mathrm{SFT}}$. This objective prioritizes the token <CNF> at the target position by assigning it a higher probability while suppressing competing tokens, thereby training the model to consistently produce <CNF> at the end of its response.

#### 3.2.4 Test-Time Scaling with GrACE

As mentioned in §3.1, GrACE helps the model to output discriminative and well-calibrated confidence with negligible inference cost. Building upon this, we incorporate GrACE into the common aggregation and adaptive sampling practices, namely self-consistency (Wang et al., 2022) and early-stopping (Li et al.), in TTS for improving the scaling law at inference time without much additional computation for confidence estimation.

**Self-Consistency with GrACE (GrACE-SC).** Self-consistency determines the accepted response by taking a majority vote over all samples generated through TTS. Intuitively, given the variance in sample quality, the voting process should account for this heterogeneity by assigning greater weight to samples with higher confidence. Thus, we leverage the discrimination of the confidence and reformulate the self-consistency. Given the sample answer set $\mathcal{A} = \{a_1, ..., a_T\}$ where the answer $a_i$ is the final answer extracted from model response $\boldsymbol{y}_i$ and $T$ is the sampling budget, GrACE-SC determines the final answer $a_{sc}^*$:

$$a_{sc}^* = \arg\max_{a \in \mathcal{A}} \sum_{i=1}^{T} \mathbb{I}(a_i = a) \cdot c_i, \tag{8}$$

where $\mathbb{I}(\cdot)$ is an indicator function which returns 1 if the answer extracted from the $i$-th sample $a_i$ matches $a$ else returns 0.

**Early Stopping with GrACE (GrACE-ES).** Early stopping terminates the sampling process once the generated samples meet predefined criteria, thereby reducing the number of samples required by the TTS algorithm. Recent approaches (Li et al.; Wan et al., 2025; Aggarwal et al.) build upon the concept of self-consistency. They partition the sampling trajectory into smaller windows and halt further sampling once the consistency within a window exceeds a specified threshold. However, these methods typically rely on pre-sampling to compute the stopping criterion. As a sampling-free method, GrACE produces discriminative and calibrated confidence scores in real-time for each sample during inference, which serves as an effective and reliable criterion for early-stopping. We keep sampling until we obtain a sample with confidence larger than the threshold $\tau$ and use this sample as the final prediction. If the sampling budget runs out and no sample's confidence reach the threshold, we pick the final answer using GrACE-SC. The pseudocode for GrACE-ES is in Appendix A.4.

## 4 Experiments

We introduce the experimental settings and the results in this section to demonstrate the effectiveness

| *Backbone* | *Methods* | TriviaQA | | | | SciQ | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | ECE ↓ | Brier ↓ | AUROC ↑ | Acc↑ | ECE ↓ | Brier ↓ | AUROC ↑ | Acc↑ |
| Phi3-3.8B | Seq. Likelihood | 14.57 | 20.62 | 81.45 | 64.40 | 14.47 | 23.81 | 73.69 | 58.50 |
| | Platt Scaling | 10.75 | 22.43 | 81.45 | - | 10.83 | 24.57 | 73.69 | - |
| | P(True) | 12.65 | 17.85 | 81.69 | - | 31.82 | 30.09 | 79.66 | - |
| | Verbal | 27.34 | 28.96 | 63.01 | - | 36.52 | 36.55 | 56.44 | - |
| | Apricot | 9.69 | 17.85 | 78.36 | - | 7.09 | 18.80 | 76.45 | - |
| | ActCab | 8.47 | 22.46 | 64.27 | - | 8.72 | 22.20 | 70.24 | - |
| | GrACE | **7.63** | **16.64** | **81.85** | 64.70 | **5.21** | **16.93** | **82.37** | 58.10 |
| Llama2-7B | Seq. Likelihood | 12.56 | 21.88 | 73.27 | 59.90 | 16.52 | 26.74 | 61.41 | 57.20 |
| | Platt Scaling | 13.95 | 23.47 | 73.27 | - | 10.12 | 24.84 | 61.41 | - |
| | P(True) | 9.82 | 23.12 | 66.02 | - | 7.46 | 24.94 | 49.41 | - |
| | Verbal | 34.31 | 37.02 | 50.89 | - | 56.05 | 37.94 | 53.07 | - |
| | Apricot | 11.88 | 18.55 | 81.61 | - | 8.42 | 16.69 | 82.04 | - |
| | ActCab | **5.18** | 23.75 | 56.21 | - | 9.34 | 24.51 | 60.00 | - |
| | GrACE | 8.41 | **16.78** | **84.22** | 59.10 | **5.45** | **14.33** | **87.72** | 57.40 |
| Llama3.1-8B | Seq. Likelihood | 15.71 | 17.32 | 79.44 | 78.00 | 24.63 | 27.26 | 69.46 | 62.80 |
| | Platt Scaling | 13.26 | 16.96 | 79.44 | - | 9.03 | 23.37 | 69.46 | - |
| | P(True) | 17.33 | 22.24 | 53.73 | - | 17.96 | 27.90 | 52.83 | - |
| | Verbal | 11.45 | 17.36 | 69.96 | - | 21.11 | 26.20 | 64.09 | - |
| | Apricot | 9.80 | 15.56 | 74.44 | - | 17.06 | 24.17 | 69.94 | - |
| | ActCab | 7.84 | 16.01 | 72.74 | - | 10.99 | 21.65 | 69.12 | - |
| | GrACE | **5.94** | **12.95** | **81.03** | 78.40 | **8.16** | **19.13** | **75.62** | 63.50 |

Table 1: **Performance comparison for open-end generation calibration task.** Note that the competitors do not fine-tune the backbone model so the accuracy is the same among the competitors. We report all the results in percentage (%). The best results are **bolded**.

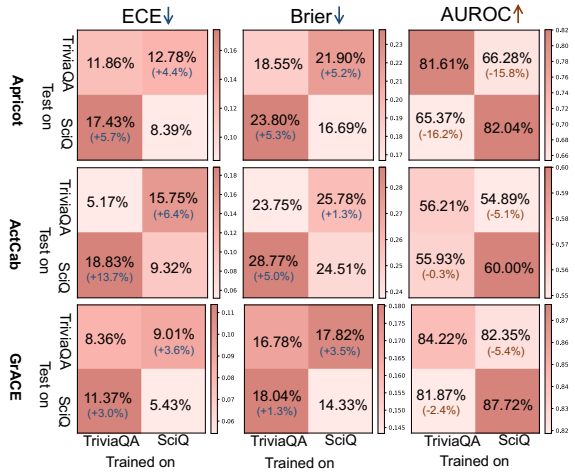


Figure 3: **Generalization ability of different confidence elicitation methods.** The values in the brackets represent how much the model's performance changes when moving from its training domain to an unseen domain.

of GrACE on both 1) open-ended generation and 2) test-time scaling.

### 4.1 GrACE for Open-ended Generation

**Experiment Settings.** We use two popular question-answering datasets, TriviaQA (Joshi et al., 2017) and SciQ (Welbl et al., 2017), for evaluating the quality of confidence produced by GrACE in the open-ended generation task. Note that in the open-ended generation setting, the model is not provided with candidate answers, unlike in the multiple-choice setting. Instead, we ask the model to generate answers with greedy decoding. The correctness of model answers is determined by ROUGE score (Lin, 2004) following the setting in Ulmer et al. (2024). We select three backbone models with different architectures, scales, and training strategies: Phi-3-3.8B (Abdin et al., 2024), Llama-2-7B (Touvron et al., 2023), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024), to comprehensively validate the effectiveness of our method.

We compare GrACE with six strong confidence elicitation and calibration methods as competitors with GrACE: **Seq. Likelihood** (Malinin and Gales), **Platt Scaling** (Platt et al., 1999), ***P*(True)** (Kadavath et al., 2022), **Verbal** (Tian et al., 2023), **Apricot** (Ulmer et al., 2024), **ActCab** (Liu et al., 2024). We report the commonly used calibration metrics, including expected calibration error (ECE) (Naeini et al., 2015) and Brier score (Brier, 1950), to comprehensively examine the calibration performances of different methods. ECE measures the average discrepancy between a model's predicted confidence and its true accuracy across discrete confidence bins. Brier score is the mean squared difference between the predicted confidence scores and the actual correctness. Additionally, we report

| Backbone | Method | MathQA | | ARC_C | |
|---|---|---|---|---|---|
| | | Acc ↑ | $\hat{T}$ ↓ | Acc ↑ | $\hat{T}$ ↓ |
| Llama-3.1 | SC | 79.60 | 8.00 | 86.30 | 8.00 |
| | ASC | 79.10 | **4.35** | 85.90 | <u>2.34</u> |
| | ESC | 79.60 | 6.06 | 86.30 | 4.69 |
| | GrACE-SC | **81.80** | 8.00 | **87.20** | 8.00 |
| | GrACE-ES | <u>80.50</u> | <u>4.48</u> | **87.20** | **2.14** |
| Qwen2.5 | SC | 85.00 | 8.00 | 89.50 | 8.00 |
| | ASC | 84.10 | **3.79** | 88.70 | **2.23** |
| | ESC | 84.70 | 5.08 | 89.30 | 4.32 |
| | GrACE-SC | **88.30** | 8.00 | **90.30** | 8.00 |
| | GrACE-ES | <u>87.70</u> | <u>3.87</u> | <u>90.10</u> | <u>2.47</u> |

Table 2: **Performance comparison for test-time scaling with sampling budget $T = 8$.** All the results are reported in percentage (%). The best results are **bolded**. The second-best is <u>underlined</u>.

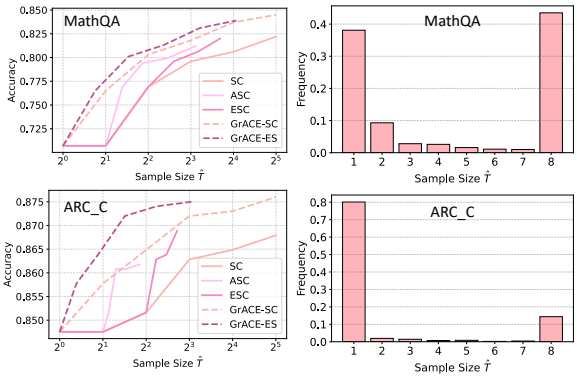


Figure 4: **(Left)** Accuracy over actual sample size from $2^0$ to $2^5$. **(Right)** Distribution of actual sample size $\hat{T}$ for GrACE-ES with sampling budget $T = 8$, based on Llama-3.1-8B-Instruct.

AUROC to show how indicative the confidence is of the factuality of the answer. The implementation details and detailed introduction of comparison methods are in Appendix A.2 and Appendix A.3.1.

**GrACE produces reliable confidence in open-ended generation.** We report our results in Table 1 and the corresponding reliability diagrams in Appendix B.3. GrACE outperforms one state-of-the-art method (Apricot (Ulmer et al., 2024)) by 3.9%, 2.5%, and 5.0% in terms of ECE, Brier and AUROC. Fine-tuning with GrACE does not affect the model utility as the accuracies on both datsets maintains at the same level with the original model. Moreover, GrACE achieves the best AUROC among all baselines, indicating that its elicited confidence scores are highly predictive of response factuality. On TriviaQA with Llama2-7B, ActCab yields the lowest ECE; however, as Figure 7 shows, it pushes most samples toward a confidence around 0.5, offering limited insight into answer correctness. We find verbalized confidence calibrates the worst. Most verbalized confidences are in the range of (0.9,1], indicating the model becomes overconfident when expressing verbalized confidence, which aligns with the conclusion of Xiong et al.. Our findings further demonstrate that LLMs learn the true distribution of prediction correctness better without any auxiliary model.

**Generalization to unseen domain.** We study the generalization ability of confidence elicitation methods that need training (i.e., Apricot, ActCab, GrACE). We conduct cross-validation using TriviaQA and SciQ, that is, train the model on one dataset and test on another to find out if the method performs consistently well in both in-domain (IND) and out-of-domain (OOD) setting.

The results are reported in Figure 3. GrACE exhibits the strongest generalization, with only marginal increases in ECE (+3.3%) and Brier score (+2.4%), and a slight AUROC drop (-3.9%) under domain shift. Moreover, it achieves the highest absolute performance in the OOD setting, outperforming all baselines by a large margin. Apricot is prone to domain shift as the AUROC decreases by 16.0% on average, indicating that much reliance on the question semantics provides a shortcut towards calibration with limitations in generalization. ActCab only trains a linear probe as a calibrator, which is not capable of fully capturing the correspondence between questions and correct answers. In contrast, GrACE teaches LLM itself the true distribution of the prediction correctness, explicitly encouraging the LLM to internalize the mapping between quesion-answer pairs and the accuracy. This enables GrACE to generalize more effectively across domains with different semantics. Moreover, we conduct an ablation study in Appendix B.4.

### 4.2 GrACE boosts Test-Time Scaling

**Experiment Settings.** We evaluate how GrACE helps TTS with two widely-used reasoning LLMs: Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen2.5-7B-Instruct (Yang et al., 2024a) as they are better at instruction-following, which is essential to ensure steady performance in the test-time scaling. The experiment is conducted on two challenging benchmark datasets: MathQA (Amini et al., 2019) and ARC_Challenge (ARC_C) (Clark et al., 2018). We compare GrACE-SC and GrACE-ES with original Self-Consistency (**SC**) (Wang et al., 2022) and two adaptive TTS methods: Adaptive

| Method | MathQA | | | | ARC_C | | | |
|---|---|---|---|---|---|---|---|---|
| | *Cali.* | *Disc.* | *Test-Time Scaling* | | *Cali.* | *Disc.* | *Test-Time Scaling* | |
| | ECE ↓ | AUROC ↑ | Acc ↑ | $\hat{T}$ ↓ | ECE ↓ | AUROC ↑ | Acc ↑ | $\hat{T}$ ↓ |
| P(True)-SC | 33.09 | 57.14 | 79.30 | 8.00 | 42.97 | 56.19 | 86.20 | 8.00 |
| P(True)-ES | - | - | 77.70 | 6.78 | - | - | 86.00 | 6.69 |
| Verbal-SC | 22.99 | 67.20 | 81.10 | 8.00 | 11.20 | 63.65 | 87.00 | 8.00 |
| Verbal-ES | - | - | 73.30 | **1.30** | - | - | 86.40 | **1.73** |
| GrACE-SC | **17.89** | **85.87** | **81.80** | 8.00 | **3.12** | **87.02** | **87.20** | 8.00 |
| GrACE-ES | - | - | 80.50 | 4.48 | - | - | **87.20** | 2.14 |

Table 3: **Performance comparison for test-time scaling with different confidence estimations for sampling budget $T = 8$.** All the results are reported in percentage (%). The best results are **bolded**. The second-best is underlined.

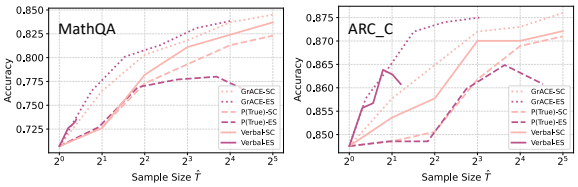


Figure 5: **Accuracy over actual sample size $\hat{T}$ with sampling budget $T$ from $2^0$ to $2^5$.** The results compare adaptive test-time scaling strategies incorprating different confidence estimations.

Self-Consistency (**ASC**) (Aggarwal et al.), Early-Stopping Self-Consistency (**ESC**) (Li et al.). The details of implementation and comparison methods are in Appendix A.2 and Appendix A.3.2.

**Confidence improves consistency-based strategy.** We show the experiment results for TTS when the sampling budget is 8 in Table 2. Moreover, we evaluate the performance of all the methods with sampling budgets ranging from $2^0$ to $2^5$ and plot the results in Figure 4. We observe two advantages of GrACE when applying to TTS. ***i)* GrACE improves the accuracy of vanilla self-consistency.** As shown in Figure 4, GrACE-SC and GrACE-ES consistently outperform the methods based on frequency over a wide range of sample budgets. As we examine a fixed sample budget (e.g., 8), we find that GrACE-SC improves the accuracy over SC on MathQA and ARC_C by 2.6%/0.9% with Llama3.1-8B-Instruct and 3.3%/0.8% with Qwen2.5-7B-Instruct, respectively. It demonstrates that there is a gap between consistency and calibrated confidence. We provide a case study for this phenomenon in Appendix B.5. Therefore, incorporating a good confidence score as a voting weight is beneficial for making decision from samples in TTS. ***ii)* GrACE-ES improves the efficiency of TTS while preserving, even enhancing performance of the TTS.** As shown in Figure 4, GrACE-ES improves the accuracy in all settings with a great reduction of sample size. There are 38.6% of queries stopping sampling at the first sample whose confidence reaches the threshold for MathQA and 80.1% of the queries only need to sample once in the less challenging benchmark ARC_C. These results demonstrate that GrACE enhances both the efficacy and efficiency of TTS, helping to avoid self-consistency errors (Tan et al., 2025).

**GrACE offers the best confidence for Test-Time Scaling.** We examine how different confidence elicitation methods affect TTS by incorprating two representative confidence elicitation methods *P*(True) and Verbal into the weighted self-consistency and early-stopping strategy proposed in §3.2.4 and test the methods with Llama-3.1-8B-Instruct. The results are reported in Table 3 and Figure 5. We observe that GrACE provides the best confidence estimation for long generation, such as Chain-of-Thought, and improves self-consistency more effectively, demonstrated by the highest accuracy when applied to self-consistency while the least calibrated confidence *P*(True) performs worst in TTS. Moreover, the efficiency of early-stopping strategy is influenced by the confidence calibration. With the same threshold, overconfident estimates (Verbal) often lead to premature termination on incorrect responses, whereas underconfident scores (*P*(True)) tend to exhaust more samples and approach the sampling budget. A well-calibrated confidence estimate serves as a reliable indicator for determining early-stopping thresholds that meet the desired accuracy.

## 5 Conclusion

In this paper, we propose GrACE, an approach to eliciting calibrated generative confidence from LLMs. Unlike post-generation and verbalized confidence elicitation methods, GrACE helps the model to output calibrated high-fidelity confidence scores on-the-fly. The LLM is trained to express its confidence by internally measuring the similarity between its last hidden state and the embedding of a special token <CNF>. GrACE first constructs confidence targets by connecting the uncalibrated confidence to accuracy using k-fold binning. The model is then fine-tuned to assign a probability mass that correlates with calibration targets to the <CNF> at the end of its response. Extensive experiments show that GrACE outperforms six confidence elicitation methods in terms of reliability in open-ended generation, while demonstrating strong domain generalization ability. Moreover, GrACE improves both the efficacy and efficiency of test-time scaling.

## Limitations

Despite the calibration performance and extended applications, GrACE has several limitations. First, GrACE is only able to provide a single confidence score that reflects the overall confidence for the generation. It cannot give different confidence scores for every claim in the generation or each step in Chain-of-Thoughts, which will be addressed in our future work. Second, GrACE focuses only on reflecting the factuality of response through confidence, while other aspects regarding model response, such as completeness, should also be considered in future work. Finally, GrACE only assesses but does not improve the model confidence.

## Ethics Statement

Our work focus on improving the reliability of large language models. We discourage any malicious use of our work, especially attempts to compromise LLM systems. The artifacts and datasets in our work are all under the restriction of the license and follow the intended use. We used GPT-5 as an AI writing assistant to refine and improve the clarity of our text. All AI-generated suggestions were carefully reviewed and edited by the authors to ensure the integrity of the work. The final manuscript reflects the authors' original contributions, with AI assistance limited solely to enhancing the presentation of our findings.

## References


Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Pranjal Aggarwal, Aman Madaan, Yiming Yang, and 1 others. Let's sample step by step: Adaptive-consistency for efficient reasoning and coding with llms. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. 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 2357–2367.

Anthropic. 2024. Claude 3: A conversational ai model.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Neil Band, Xuechen Li, Tengyu Ma, and Tatsunori Hashimoto. 2024. Linguistic calibration of long-form generations. In *Proceedings of the 41st International Conference on Machine Learning*, pages 2732–2778.

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

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

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 preprint arXiv:1803.05457*.

Roi Cohen, Konstantin Dobler, Eden Biran, and Gerard de Melo. 2024. I don't know: Explicit modeling of uncertainty with an [idk] token. *Advances in Neural Information Processing Systems*, 37:10935–10958.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *The Twelfth International Conference on Learning Representations*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Sophia Hager, David Mueller, Kevin Duh, and Nicholas Andrews. 2025. Uncertainty distillation: Teaching language models to express semantic confidence. *arXiv preprint arXiv:2503.14749*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Chengsong Huang, Langlin Huang, Jixuan Leng, Jiacheng Liu, and Jiaxin Huang. 2025. Efficient test-time scaling via self-calibration. *arXiv preprint arXiv:2503.00031*.

Ziwei Ji, Lei Yu, Yeskendir Koishekenov, Yejin Bang, Anthony Hartshorn, Alan Schelten, Cheng Zhang, Pascale Fung, and Nicola Cancedda. 2025. Calibrating verbal uncertainty as a linear feature to reduce hallucinations. *arXiv preprint arXiv:2503.14477*.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, and 1 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Katherine M Collins, Arka Pal, Umang Bhatt, Adrian Weller, Samuel Dooley, Micah Goldblum, and Andrew Gordon Wilson. Large language models must be taught to know what they don't know. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

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

Volodymyr Kuleshov and Percy S Liang. 2015. Calibrated structured prediction. *Advances in Neural Information Processing Systems*, 28.

Chengzu Li, Han Zhou, Goran Glavaš, Anna Korhonen, and Ivan Vulić. 2025. Large language models are miscalibrated in-context learners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 11575–11596.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth International Conference on Learning Representations*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Xin Liu, Farima Fatahi Bayat, and Lu Wang. 2024. Enhancing language model factuality via activation-based confidence calibration and guided decoding. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 10436–10448.

Varun Magesh, Faiz Surani, Matthew Dahl, Mirac Suzgun, Christopher D Manning, and Daniel E Ho. 2024. Hallucination-free? assessing the reliability of leading ai legal research tools. *arXiv preprint arXiv:2405.20362*.

Andrey Malinin and Mark Gales. Uncertainty estimation in autoregressive structured prediction. In *International Conference on Learning Representations*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

Alexandru Niculescu-Mizil and Rich Caruana. 2005. Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 625–632.

John Platt and 1 others. 1999. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, and 1 others. 2023. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and 1 others. 2025. Too consistent to detect: A study of self-consistent errors in llms. *arXiv preprint arXiv:2505.17656*.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dennis Thomas Ulmer, Martin Gubri, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. 2024. Calibrating large language models using their generations only. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 15440–15459. Association for Computational Linguistics.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. 2025. Reasoning aware self-consistency: Leveraging reasoning paths for efficient llm sampling. In *The 2025 Annual Conference of the Nations of the Americas Chapter of the ACL*.

Xinyi Wang, Lucas Caccia, Oleksiy Ostapenko, Xingdi Yuan, William Yang Wang, and Alessandro Sordoni. Guiding language model reasoning with planning tokens. In *First Conference on Language Modeling*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Qianqian Xie, Weiguang Han, Zhengyu Chen, Ruoyu Xiang, Xiao Zhang, Yueru He, Mengxi Xiao, Dong Li, Yongfu Dai, Duanyu Feng, and 1 others. 2024. Finben: A holistic financial benchmark for large language models. *Advances in Neural Information Processing Systems*, 37:95716–95743.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *The Twelfth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024a. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Ruihan Yang, Caiqi Zhang, Zhisong Zhang, Xinting Huang, Sen Yang, Nigel Collier, Dong Yu, and Deqing Yang. 2024b. Logu: Long-form generation with uncertainty expressions. *arXiv preprint arXiv:2410.14309*.

Ruihan Yang, Caiqi Zhang, Zhisong Zhang, Xinting Huang, Dong Yu, Nigel Collier, and Deqing Yang. 2025b. Uncle: Uncertainty expressions in long-form generation. *arXiv preprint arXiv:2505.16922*.

Caiqi Zhang, Ruihan Yang, Zhisong Zhang, Xinting Huang, Sen Yang, Dong Yu, and Nigel Collier. 2024. Atomic calibration of llms in long-form generations. *arXiv preprint arXiv:2410.13246*.

Zhaohan Zhang, Ziquan Liu, and Ioannis Patras. 2025. Get confused cautiously: Textual sequence memorization erasure with selective entropy maximization. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10924–10939.

# A Implementation Details

## A.1 Prompt Template

We elaborate on the prompt templates used for open-ended generation and test-time scaling in this section.

**Prompt Templete for Open-ended Generation**

```
Provide your best guess for the following question. Give ONLY the guess, no other words or explanation.

For example:

Question:<User's question>
Guess: <most likely guess based on the support, as short as possible; not a complete sentence, just the guess!>

Question: {Question}
```

**Prompt Templete for Test-Time Scaling**

```
For the following question, provide a step-by-step explanation of your thought process.
Use the format demonstrated below for your response.

"'Example Format:
Explanation: <Your detailed explanation here, outlining how you arrived at your answer.>

Answer: <Insert your concise answer here, which should include a {answer_type}>
Ensure that your response strictly adheres to this format. Explicitly include the words 'Explanation:', 'Answer:'.
```

## A.2 Experiment Settings and Hyperparameter Selection

In the open-ended generation calibration task, we randomly sample 2000 queries from the training data in the benchmarks for training GrACE and sample 1000 queries from the test data for evaluation. We train GrACE with a learning rate of $1 \times 10^{-5}$ for 3 epochs, using the AdamW optimizer. The batch size for training is 8. We apply LoRA with rank $r = 16$, scaling factor $\alpha = 16$, and dropout factor of 0.05. The LoRA module is applied to the MLP block for both backbone models. The trainable parameters are negligible compared with the model parameters, accounting for only 0.41%, 0.34%, and 0.32% of Phi3-3.8B's, Llama2-7B's, and Llama3.1-8B-Instruct's total parameters, respectively. The threshold for ROUGE score is set to 0.3 following Ulmer et.al (Ulmer et al., 2024). The experiment is conducted on two NVIDIA GeForce RTX 3090 GPUs.

In the test-time scaling task, we randomly sample 2000 queries from the training data in the benchmarks for training GrACE and sample 1000 queries from the test data for evaluation. To align the training process of GrACE with the implementation of TTS, we generate 16 different responses for each query in the training data with temperature $t = 0.8$. The trainable parameters account for 0.17% and 0.13% of Llama-3.1-8B-Instruct's and Qwen2.5-7B-Instruct's total parameters, respectively. The model is trained with a learning rate of $lr = 5 \times 10^{-5}$ for 3 epochs with AdamW optimizer. The batch size is set to 4. We apply LoRA with rank $r = 32$, scaling factor $\alpha = 16$, and dropout factor of 0.05. The LoRA adapter is applied to the query and value weight of attention blocks. The threshold for the early-stopping strategy is set to 0.8. The experiment is conducted on two NVIDIA A100 80GB GPUs.

We ensure the training data is equally distributed in all the confidence intervals to prevent overfitting on any specific confidence level. This strategy helps stabilize the training process and reduce the bias for confidence elicitaion. The hyperparameter $\gamma$ in the loss function is set to 0.1 for all the experiments. We report the average performance over 5 random seeds. The artifacts used in our work are all under the restriction of the license and follow the intended use.

## A.3 Comparison Methods

### A.3.1 Comparison Methods for Open-end Generation Calibration

**Sequence Likelihood** (Malinin and Gales) uses the length-normalized likelihood of generated sequences as predicted confidence.

**Platt Scaling** (Platt et al., 1999) fits the Seq. Likelihood to minimize the mean squared error on the

validation set with scalers $a, b \in \mathbb{R}$.

***P*(True)** (Kadavath et al., 2022) prompts the LLMs to decide whether or not the generation is true and uses the probability of true as the predictive confidence.

**Verbal** (Tian et al., 2023) directly asks the model to assess its confidence in percentage.

**Apricot** (Ulmer et al., 2024) sets the accuracy of semantic clusters as the calibration target and trains an auxiliary model on the generation to predict confidence.

**ActCab** (Liu et al., 2024) trains a linear probe with the activation from LLMs as the calibrator. Note that Platt Scaling, Apricot, and ActCab need to train auxiliary models while other comparison methods are training-free.

#### A.3.2 Comparison Mehthods for TTS

**SC** (Wang et al., 2022) takes the majority vote from samples generated in TTS to decide the final answer.

**ASC** (Aggarwal et al.) calculates the cumulative frequency of each answer in the TTS process and terminates sampling when the response's relative frequency reaches the threshold.

**ESC** (Li et al.) divides the sampling steps into windows and keep sampling until the samples within the window reach high agreement.

### A.4 Pseudocode for GrACE-ES

We provide the pseudocode for GrACE-ES in Algorithm 1.

**Algorithm 1** GrACE-ES

**Require:** Input $\boldsymbol{x}$, Confidence Threshold $\tau$, Sample Budget $T$

Initialize answer list $\mathcal{A} \leftarrow [\,]$
Initialize confidence list $\mathcal{C} \leftarrow [\,]$
**for** $i = 1$ to $T$ **do**
    Sample answer and confidence $\boldsymbol{y}_i, a_i, c_i \leftarrow \text{Model}(\boldsymbol{x})$
    Append $a_i$ to $\mathcal{A}$, $c_i$ to $\mathcal{C}$
    **if** $c_i \geq \tau$ **then**
        **return** $a_i$ ▷ Early stop
    **end if**
**end for** ▷ Use weighted voting if threshold was never met
GrACE-SC:
Choose $a^*_{es} = a^*_{sc}$ according to equation 8
**return** $a^*_{es}$

## B Discussions and Additional Results

### B.1 Target Distribution

We plot the distribution of target confidence in Figure 6. We observe that the distribution of target confidence is not concentrated around a single value. Instead, it exhibits a sharp distribution across the entire range of [0, 1], with relatively higher densities at both the lower and upper ends. Such a distribution suggests an enhanced capacity for discriminativeness.

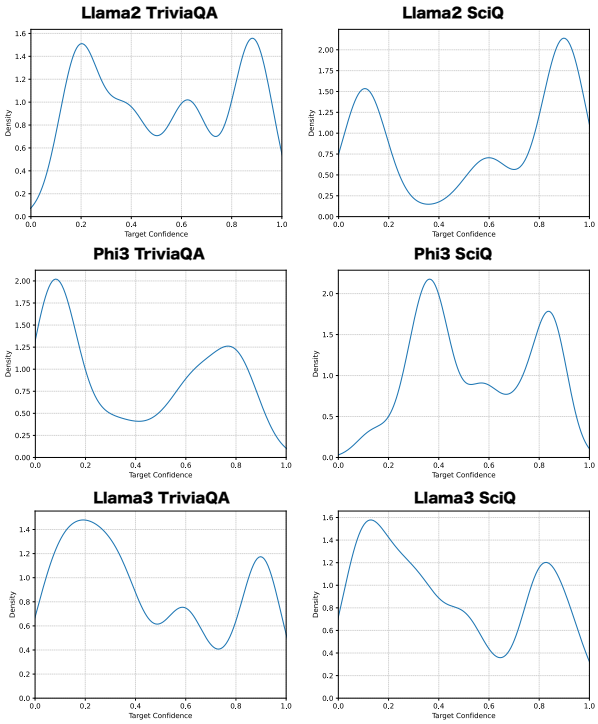


Figure 6: **The distribution of confidence targets.**

### B.2 Relationship between the target confidence and calibration

Given the question-answer pair $(\boldsymbol{x}_i, \boldsymbol{y}_i)$, the calculation of expected calibration error (ECE) is formulated as:

$$\sum_{m=1}^{M} \frac{|\mathcal{B}_m|}{N} |\text{Acc}(m) - \frac{1}{|\mathcal{B}_m|} \sum_{i \in \mathcal{B}_m} c_i|, \quad (9)$$

where $\mathcal{B}_m$ denotes the set of indices belonging to bin $m$, $c_i$ is the confidence estimation associated with the model response $\boldsymbol{y}_i$. Recalling that the target confidence $t_i$ in Section §3.2.2 is the empirical accuracy within bin $m$, reducing the gap between the confidence $c_i$ and target confidence $t_i$ is thus equvalent to minimizing the ECE. Therefore, optimizing loss $\mathcal{L}_C$ in Eq. 7 directly contributes to improving model calibration.

### B.3 Reliability Diagrams

The reliability diagrams for the calibration of different confidence elicitation methods are shown in Figure 7 (Llama2-7B on TriviaQA), Figure 8 (Llama2-7B on SciQ), Figure 9 (Phi3-3.8B on TriviaQA), Figure 10 (Phi3-3.8B on SciQ), Figure 11 (Llama3.1-8B-Instruct on TriviaQA), and Figure 12 (Llama3.1-8B-Instruct on SciQ).

### B.4 Ablation Study

We conduct an ablation study using three variants of GrACE to investigate the contributions of different components, evaluated on the TriviaQA dataset with Llama2-7B. **GrACE-Disc** expresses confidence with discrete tokens. In the training stage, each data point is assigned to one of $N = 10$ bins as described in §3.2.2. The bins are represented by a set of special tokens {<B1>, ..., <BN>}, which are appended to the end of the response. The model is trained to predict the correct token using the $\mathcal{L}_{\text{SFT}}$. During evaluation, each bin corresponds to a representative confidence score from the set $\{0.05, 0.15, \ldots, 0.95\}$ , which approximates the midpoint of each confidence interval. **GrACE-Pre** places the <CNF> immediately after the input prompt rather than at the end of the response, thereby eliciting confidence based solely on the question context. **GrACE-SFT** includes the <CNF> in the calculation of $\mathcal{L}_{\text{SFT}}$ when calculating Eq. 7, explicitly training the model to allocate probability mass to it.

The results are reported in Table 4. GrACE-Disc leads to a 1.93% increase in ECE and a 7.01% drop in AUROC, indicating that a discretized representation of confidence is insufficiently precise for reliably reflecting response quality. Although GrACE-Disc shares similarity with verbalized confidence methods which give a discrete expression of confidence, it performs better in terms of calibration because the new tokens are specifically trained to express confidence associated with factuality, while verbalized confidence methods are interfered with natural language generation. GrACE-Pre drops by 28.55% in AUROC, suggesting that the confidence scores derived solely from question information are not reliable indicators of the correctness of the model response. GrACE-SFT suffers a great increase in ECE (+4.39%) but a rather trivial decrease in AUROC (-2.98%), because optimizing <CNF> with $\mathcal{L}_{\text{SFT}}$ puts extra probability mass on <CNF>, leading to overconfidence. However, the relatively small drop in AUROC indicates that, despite the increased miscalibration, the elicited confidence scores still retain a strong correlation with response actuality.

| Method | ECE↓ | Brier↓ | AUROC↑ |
|---|---|---|---|
| **GrACE-Disc** | 10.29 | 20.23 | 77.21 |
| **GrACE-Pre** | 10.45 | 26.32 | 55.67 |
| **GrACE-SFT** | 12.75 | 19.28 | 81.24 |
| **GrACE** | **8.36** | **16.78** | **84.22** |

Table 4: **Ablation study results on Triviaqa using Llama2-7B.** All results are reported in percentage (%). The best results are **bolded**.

### B.5 Case Study

We provide a case in which the query is from MathQA and the samples are generated by Llama-3.1-8B-Instruct in Table 5. In this case, the four samples give the consistent answer (D) to the query, which is also the correct answer. However, the model generates the fourth sample with confidence 0.1 while the confidences are higher than 0.9 for other samples. Taking a closer look at the samples, we find that the confident samples explicitly express the correct answer (8 days) in the thinking process. The sample with confidence 0.1 skips the step where the specific answer is derived, leading to low confidence to the final answer. This case demonstrates that within the samples that are consistent in final answer, there are also samples with low confidence. GrACE does not capture the superfacial self-consistancy information but the internal confidence in the model.

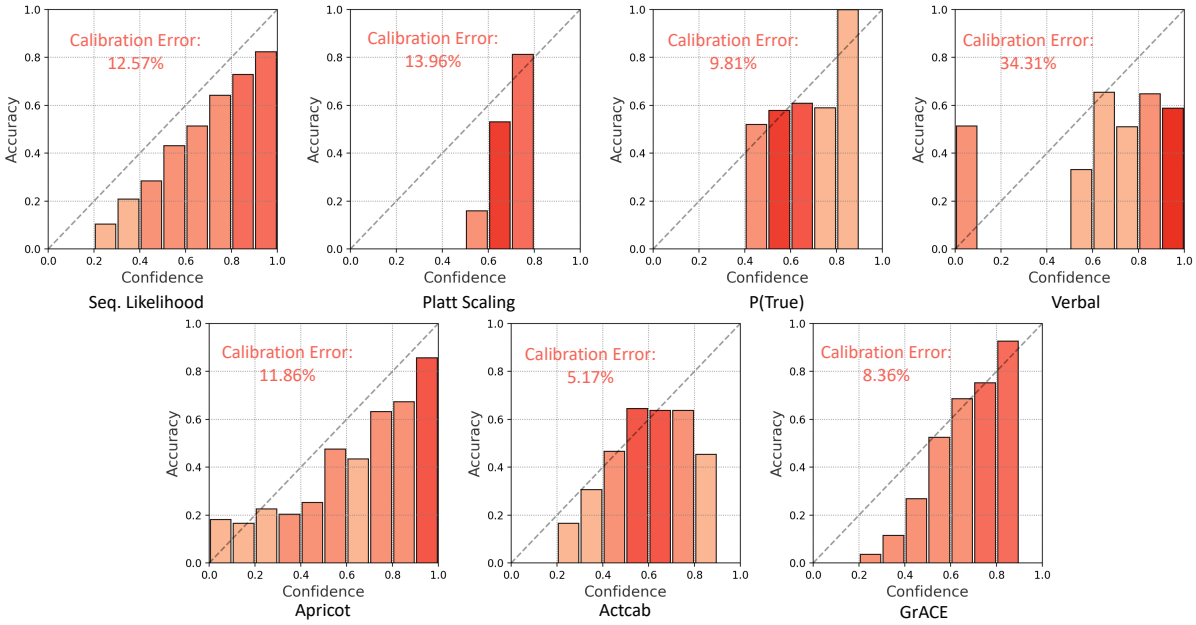


Figure 7: **The reliability diagram of various confidence elicitation methods applied to Llama2-7B on the TriviaQA test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests a larger value.

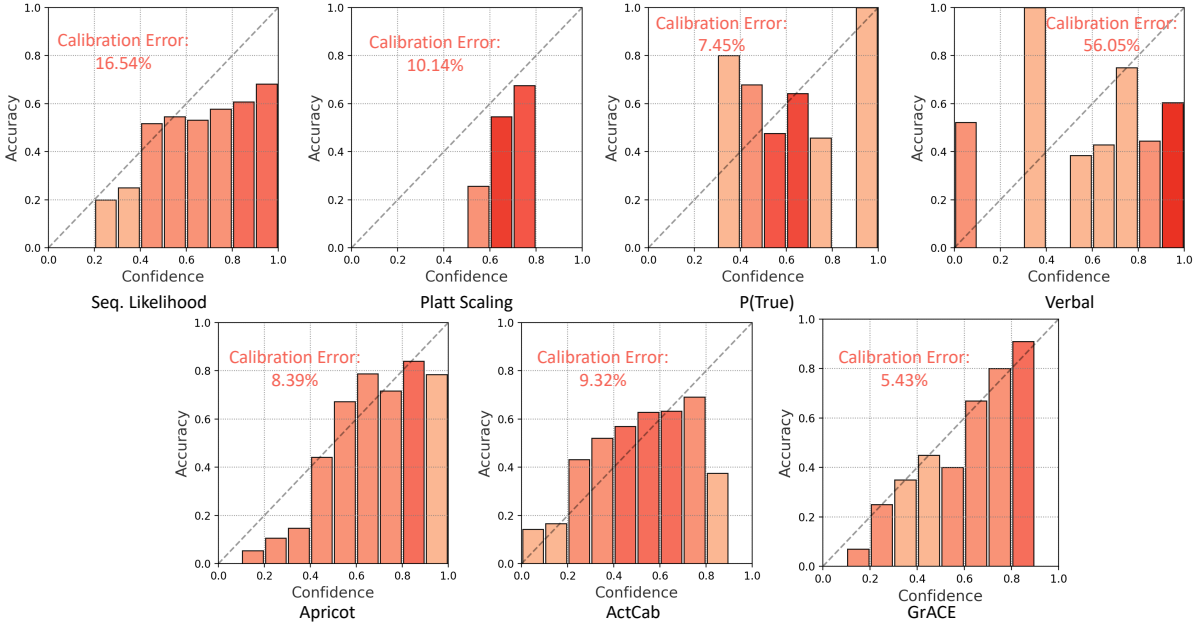


Figure 8: **The reliability diagram of various confidence elicitation methods applied to Llama2-7B on the SciQ test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests larger value.

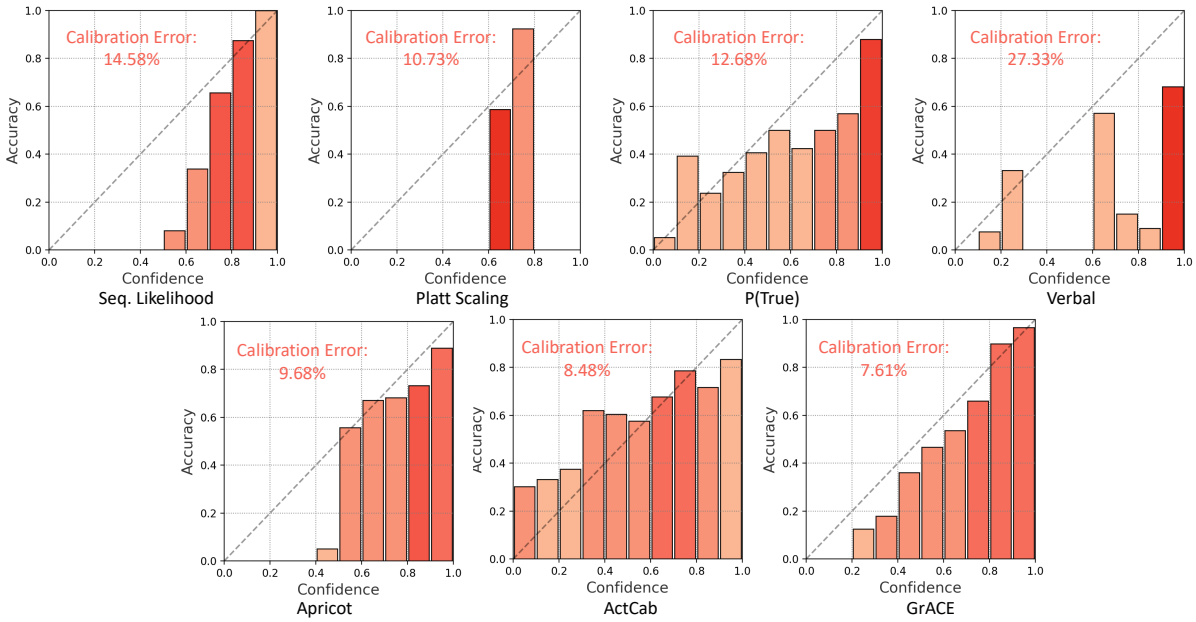


Figure 9: **The reliability diagram of various confidence elicitation methods applied to Phi3-3.8B on the TriviaQA test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests larger value.

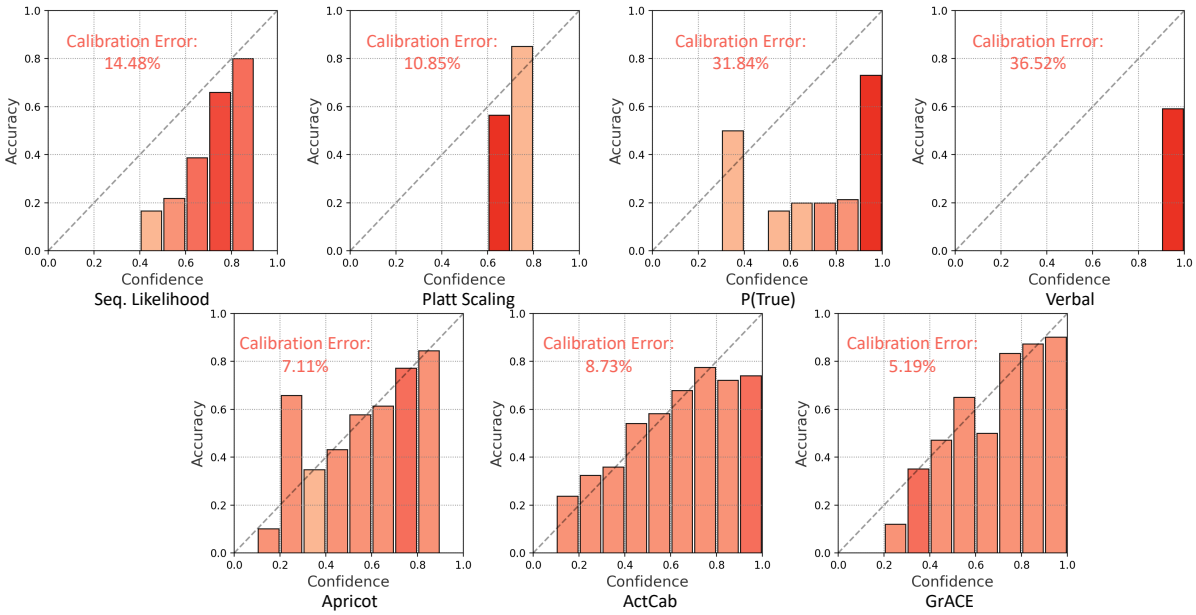


Figure 10: **The reliability diagram of various confidence elicitation methods applied to Phi3-3.8B on the SciQ test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests larger value.

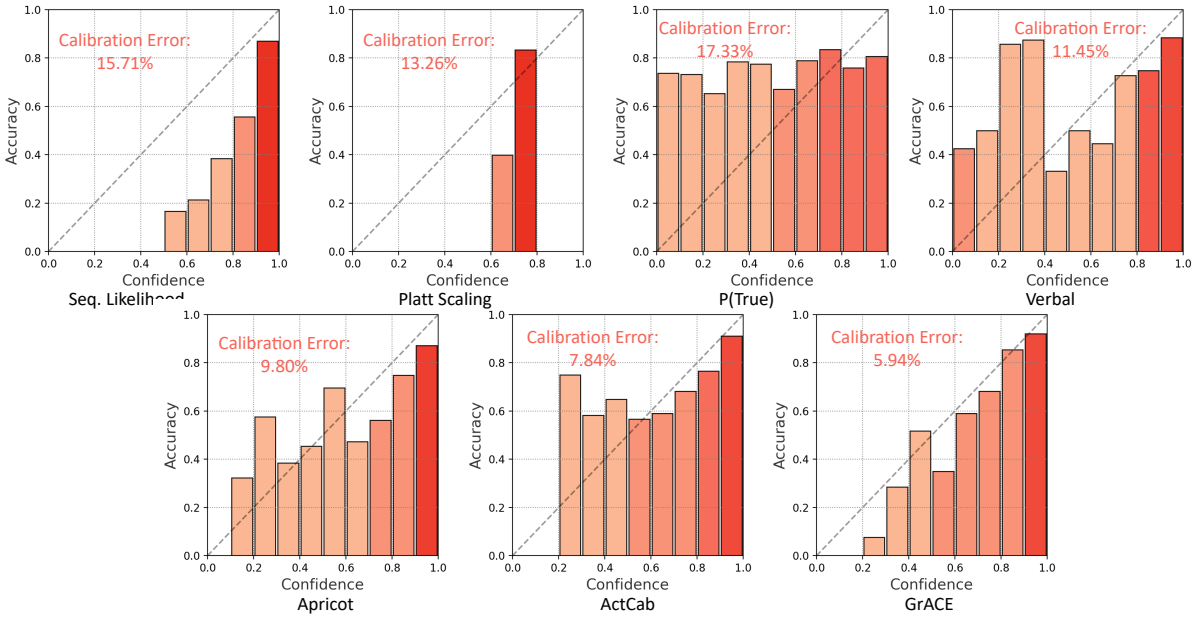


Figure 11: **The reliability diagram of various confidence elicitation methods applied to Llama3.1-8B-Instruct on the TriviaQA test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests larger value.

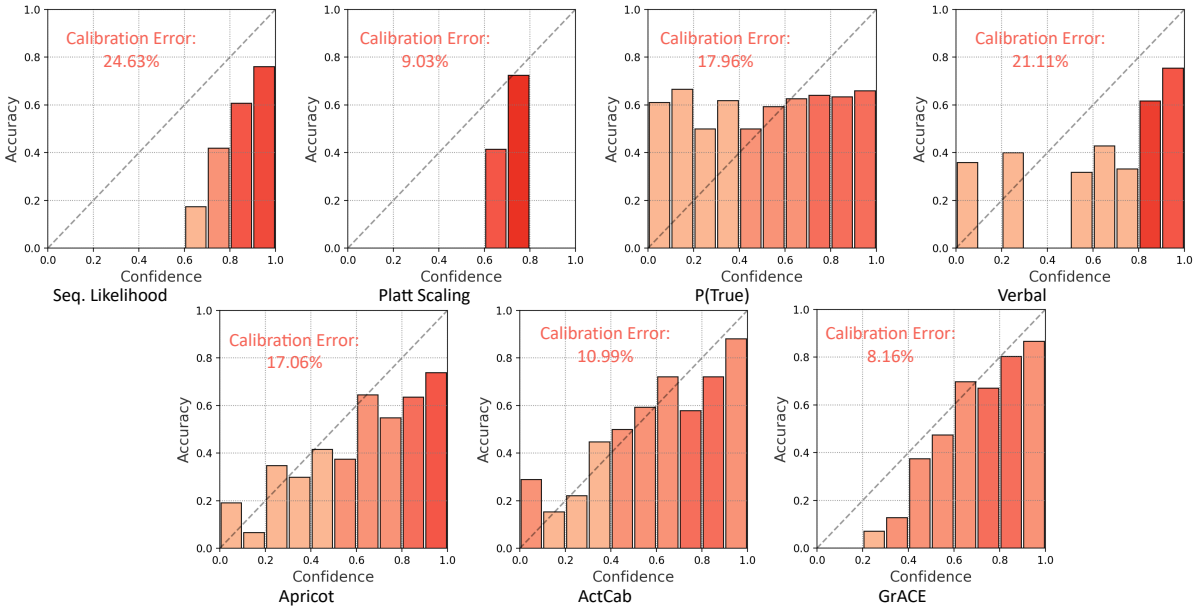


Figure 12: **The reliability diagram of various confidence elicitation methods applied to Llama3.1-8B-Instruct on the SciQ test dataset.** The color indicates the proportion of total responses contained in each bin. The color close to red suggests larger value.

| Query | Sample | Confidence |
|---|---|---|
| A contractor estimates that 10 people can paint Mr Khan's house in 4 days. If he uses 5 people instead of 10, how long will it take to complete the job?<br>Options:<br>A. 10<br>B. 4<br>C. 5<br>D. 8<br>E. 6 | Explanation: To solve this problem, we first need to understand the relationship between the number of people and the time taken to complete the job. The original information given is that 10 people can paint the house in 4 days. This means that the total amount of work done is constant.<br>...<br>New time = Total work / New number of people<br>New time = 40 man-days / 5 people<br>New time = 8 days<br><br>Therefore, if the constructor uses 5 people instead of 10, it will take them 8 days to complete the job.<br><br>**Answer: D <CNF>** | 0.992 |
| | Explanation: To solve this problem, we can use the concept of work rate. The initial situation is that 10 people can paint the house in 4 days.<br>...<br>To find out how long it will take for 5 people to complete the job, we divide the total work by the number of people: 40 person-days / 5 people = 8 days.<br><br>**Answer: D <CNF>** | 0.993 |
| | Explanation: To solve this problem, we need to understand the concept of work and time. When the number of workers decreases, the time taken to complete the job increases proportionally. We can use the concept of'man-days' to solve this problem.<br>...<br>Time = Total man-days / Number of people<br>Time = 40 / 5<br>Time = 8 days<br><br>Therefore, if the constructor uses 5 people instead of 10, they will take 8 days to complete the job.<br><br>**Answer: D <CNF>** | 0.988 |
| | Explanation: To solve this problem, we first need to understand the relationship between the number of people and the time it takes to complete the job. Since the number of people is being reduced from 10 to 5, it implies that the work will take longer to complete because there are fewer workers. The initial ratio is 10 people in 4 days. If we reduce the number of people by half (10/2 = 5), the time taken to complete the job will increase by a factor of 2, since there are fewer workers to divide the work among.<br><br>**Answer: D <CNF>** | 0.100 |

Table 5: Examples of consistent sample with different confidences.