Calibrating the Confidence of Large Language Models by Eliciting Fidelity

Anonymous ACL submission

Abstract

Large language models optimized with techniques like RLHF have achieved good alignment in being helpful and harmless. However, post-alignment, these language models often exhibit overconfidence, where the expressed confidence does not accurately calibrate with their correctness rate. In this paper, we decompose the language model confidence into the *Uncertainty* about the question and the *Fidelity* to the answer generated by language models. Then, we propose a plug-and-play method, *UF Calibration*, to estimate the confidence of language models. Our method has shown good calibration performance by conducting experiments with 6 RLHF-LMs on four MCQA datasets. Moreover, we propose two novel metrics, IPR and CE, to evaluate the calibration of the model, and we have conducted a detailed discussion on *Truly Well-Calibrated Confidence* for large language models. Our method could serve as a strong baseline, and we hope that this work will provide some insights into the model confidence calibration.

1 Introduction

Large language models (LLMs) acquire vast world knowledge and demonstrate powerful capabilities through pre-training (Brown et al., 2020; OpenAI, 2023; Bubeck et al., 2023). With technologies like RLHF (Ouyang et al., 2022) and RLAIF (Bai et al., 2022; Lee et al., 2023), large language models can become more helpful and harmless to align with human preferences (Askell et al., 2021). However, how to build a more honest system has not yet been fully discussed. An honest model should have a certain understanding of the boundary of its knowledge, that is, *knowing what it does not know* (Yin et al., 2023; Yang et al., 2023b; Zhou et al., 2024). A plausible method is utilizing the calibrated confidence to estimate the knowledge boundary of language models. For pre-trained language models, the per-token logit can already be considered a well-calibrated confidence score, which implies that *pre-trained language models (mostly) know what they know* (Kadavath et al., 2022).

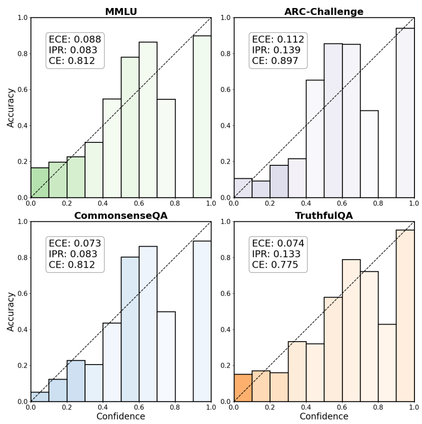

Figure 1: In four different MCQA datasets, our method has demonstrated good calibration effects, meaning it is sufficiently close to the $y = x$ curve. The experimental data is derived from `GPT-3.5-Turbo`.

However, recent studies have indicated that language models optimized with techniques like RLHF will exhibit issues of overconfidence (Lin et al., 2022a; Kadavath et al., 2022; OpenAI, 2023; He et al., 2023; Zhao et al., 2023; Tian et al., 2023; Xiong et al., 2023). This issue could be reflected in Multiple-Choice Question Answering (MCQA) tasks, where the probability of RLHF-LMs generating a token and the likelihood of that token being the correct answer are not well-calibrated. For example, an answer provided by RLHF-LMs with 95% confidence does not mean that there is a 95% probability that the answer is correct. This phenomenon may be due to the optimization objective of RLHF, which is to make the model generate responses aligned with human preferences rather than fitting answers that appear more frequently in the corpus during the pre-training stage.

To alleviate the issue of miscalibration, previous work focuses on two perspectives: the logit-based method and the verbalization-based method. Logit-based methods are usually post-hoc. We need to find a higher temperature (usually above 2.0), known as Temperature-Tuning (Guo et al., 2017), to make the distribution of the model's token logit smoother for mitigating overconfidence (Kadavath et al., 2022; He et al., 2023). The verbalization-based method usually requires prompt engineering to elicit the model's confidence, and it also necessitates the model to have strong Self-Awareness (Lin et al., 2022a; Tian et al., 2023; Yin et al., 2023). Aggregating the model's logit-based and verbalization-based confidence can also calibrate the model confidence to some extent (Xiong et al., 2023).

As shown in Figure 2 and Appendix Tabel 5, by replacing the model's answer with "*All other options are wrong.*", we can assess whether the model had high fidelity to its previously given answer. Inspired by this phenomenon, we decompose the language model confidence into two dimensions: the ***Uncertainty*** about the question and the ***Fidelity*** to the answer generated by language models. First, if the answers provided by language model are consistent under multiple samplings, it indicates that language model has lower uncertainty regarding that question. Thus, we could utilize the information entropy of the frequency distribution of sampled answers to calculate the model's uncertainty about a question. Second, we design a novel method to estimate the model's fidelity to each of its sampled answers. Last, the uncertainty regarding question $\mathcal{Q}$ and the fidelity to the answer a_i together determine the model's confidence. As shown in Figure 1, our proposed UF Calibration achieved good calibration across different MCQA datasets. Meanwhile, UF Calibration does not require knowledge of the model's per-token log-probability, making it broadly applicable to various Black-box RLHF-LMs, which do not provide the per-token log-probability.

To have a closer look at the calibration of model confidence, we propose two novel metrics for evaluating and observation: **1)** ***I****nverse* ***P****air* ***R****atio* (IPR), which is the proportion of inverse pairs in the Reliability Diagram. This metric could reflect whether the model is well-calibrated from the perspective of the monotonicity of the Reliability Diagram. If the reliability diagram is monotonic, it indicates that the average accuracy of low-confidence answers is always lower than that of high-confidence answers. **2)** As shown in Table 10, we find that as the number of model parameters increases, language models still tend to consistently express uncertainty within certain fixed ranges. Thus, we design the ***C****onfidence* ***E****venness* (CE) to observe to the uniformity of the density of each bar in the reliability diagram. Our experimental results indicate that, after calibration, even within the same dataset, there is a significant difference in the confidence of the answers provided by language models for different questions. We summarize our main contributions as follows:

1) Our proposed method could be viewed as a strong baseline for eliciting model confidence, where answer set is known. And the calibrated confidence could be viewed as a soft label.
2) We propose two new metrics, IPR and CE, to evaluate the calibration of LM's confidence.
3) We conduct a detailed discussion of a research question: "*What kind of Confidence is Truly Well-Calibrated?*", and we hope our discussion can bring some insights to the community.

2 Related Work

Recent work has focused on LLM calibration (Lin et al., 2022a; Kadavath et al., 2022; OpenAI, 2023). In this section, we will briefly introduce two mainstream methods for eliciting the confidence from language models, namely the Logit-based Method and the Verbalization-based Method.

2.1 Logit-based Method

When we can obtain the per-token logits from language models, we can directly use the probability of generating candidate answers as its confidence.

$$\mathrm{Conf}(a_i) = \frac{\exp(\mathrm{logit}_{a_i}/t)}{\sum_{j=1}^{|\mathcal{A}|} \exp(\mathrm{logit}_{a_j}/t)}, \quad (1)$$

where t is the sampling temperature of language models and $|\mathcal{A}|$ is the size of candidate answer set $\mathcal{A}$. Recent studies indicate that good calibration can be achieved by adjusting the temperature of RLHF-LMs (Kadavath et al., 2022; He et al., 2023). However, temperature-scaling (Guo et al., 2017) often requires higher temperatures, such as above 2.0 (Kadavath et al., 2022), which might cause the outputs of the language models to become too random. When the probabilities for model-generated tokens are inaccessible, a straightforward solution is to deploy sampling and use the frequency of the sampled result to estimate the probability of generating this token. For instance, given a question $\mathcal{Q}$, we could sample K times to acquire a set of

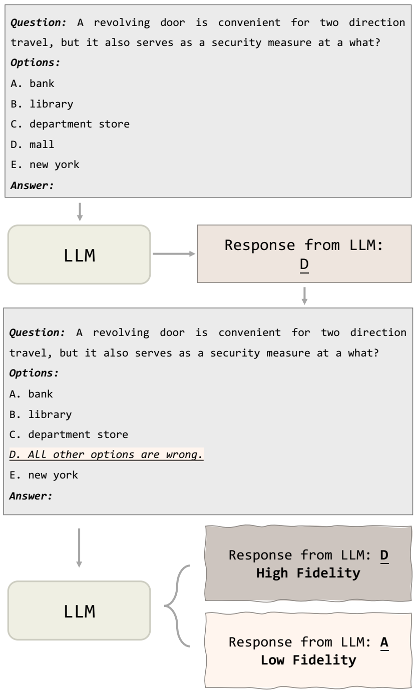

Figure 2: If the model's choice of answer changes after replacing the content of its previous selected option with "*All other options are wrong*", it could be considered that the model's fidelity to its previous answer is not high enough.

answers $\mathcal{A}$ containing N distinct answers, and each answer with an associated frequency n_i. The probability of the model generating answer a_i can be estimated by $\frac{n_i}{K}$. Therefore, we could estimate the confidence of language models by $\mathcal{P}_{\text{sampled}}(a_i)$. Recently, Kumar et al. (2023) also propose to utilize the conformal prediction to calibrate the confidence of LLMs.

$$\text{Conf}(a_i) = \mathcal{P}_{\text{sampled}}(a_i) = \frac{n_i}{K}, a_i \in \mathcal{A} \quad (2)$$

2.2 Verbalization-based Method

However, some commercial models, such as ChatGPT and Claude, usually do not provide per-token logits. Benefiting from instruction fine-tuning(Chung et al., 2022; Zhang et al., 2023), language models could generate responses corresponding to the input instructions. Another intuitive method is to prompt large language models to provide their verbalized confidence along with their responses as follows (Jiang et al., 2021; Lin et al., 2022a; Tian et al., 2023):

$$(\text{Answer}, \ \text{Conf}) = \text{LLM}(\text{Question}), \quad (3)$$

This method requires the model to have a strong ability to follow instructions and strong self-awareness (know whether it knows something or not (Yin et al., 2023)). Accordingly, verbalized confidence can be a floating-point number between 0 and 1, i.e., `'0.8'`. And it can be linguistic expressions, i.e., `'Almost Certain'`, `'About Even'`, `'Unlikely'`. Although this method is quite easy to implement, we find various different LMs always tend to output some fixed high confidence expressions, as show in Table 10.

3 Methodology

In this section, we will introduce the method we propose. Our method does not require any knowledge of the per-token logit of language models or trivial prompt engineering to make the language model output its confidence in a specified format.

3.1 Sampling

Firstly, as shown in the first step from Figure 3, for question $\mathcal{Q}$, by sampling K times, we can obtain a set of candidate answers $\mathcal{A}$. We take the most frequently occurring answer as the final answer. Meanwhile, we can obtain the frequency distribution $\mathcal{P}_{\text{sampled}}$ of candidate answers.

3.2 Eliciting the Fidelity of Answers

As shown in Figure 2, for question $\mathcal{Q}$ and a candidate answer (a_i, o_i), where the option index is a_i and the content is o_i, we simply replace o_i with "***All other options are wrong.***", and then query the model again. If the model has high fidelity to the previously selected answer (a_i, o_i), it should select (a_i, "*All other options are wrong.*") in the subsequent round of inquiry rather than any other option. If language models select other options, we remove the newly selected option to ensure that there is only one "*All other options are wrong*" in candidate options. By repeating this process until the model selects "*All other options are wrong*", we can establish a hierarchical fidelity chain $\mathcal{C}$, such as "A→C→D". This implies that when all options are available, the model will prefer to select option A. However, if option A is excluded, the model will tend to choose option C, which indicates that the model's fidelity to option A is not high enough. Accordingly, if the chain $\mathcal{C}$ has only one element, such as "A", this suggests that the model's fidelity to option A is high enough, which can, to a certain extent, reflect the model's confidence. Correspondingly, for a hierarchical fidelity chain $\mathcal{C}$, we assign a fidelity weight to each element from right to left. For example, for the ith element d_i from the right, we simply set its weight as τ^i. Therefore,

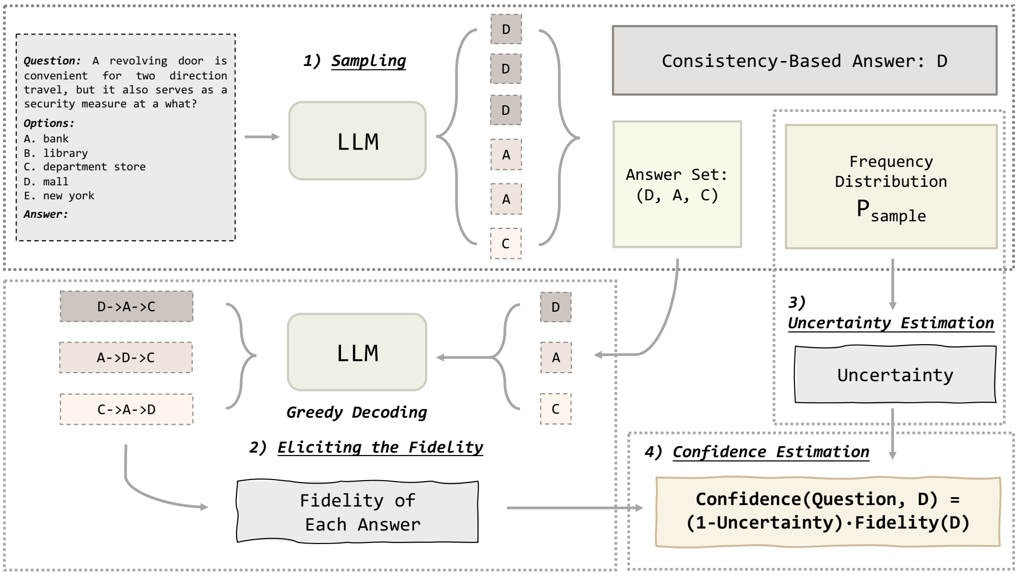

Figure 3: Our proposed UF Calibration, which requires at most two phases to invoke the model. In the Sampling phase, for black-box models, similar to the Sampled method, we need to sample 10 times. For white-box models, a single invocation is sufficient. In the eliciting the fidelity phase, the model needs to be invoked approximately 2 to 3 times to generate a fidelity chain, as show in Table 8.

the normalized fidelity of the ith element a_i can be calculated as follows:

$$\mathbf{Fidelity}_{\mathcal{C}}(a_i) = \frac{\tau^i}{\sum_{i=1}^{|\mathcal{C}|} \tau^i}, \quad (4)$$

where we usually set τ as 2. As shown in Figure 3, the answer set $\mathcal{A}$ might include multiple different answers. Consequently, we sequentially replace the candidate answer in $\mathcal{A}$ with "*All other options are wrong*." to elicit different hierarchical fidelity chains, as depicted in the second step of Figure 3. The fidelity score of each element a_i in every hierarchical fidelity chain $\mathcal{C}_j$ can be calculated using (4). Thus, the model's fidelity of answer a_i can be calculated by the weighted average fidelity score across different hierarchical chains. Since the hierarchical fidelity chain is elicited by greedy decoding, the frequency of occurrence of different chains is consistent with the frequency of occurrence of the first element $a_{|\mathcal{C}|}$ from left to right. Therefore, the frequency $\mathcal{P}_{\text{sampled}}(a_{|\mathcal{C}|})$ can be viewed as a proxy for the probability $\mathcal{P}_{\text{sampled}}(\mathcal{C}_j)$ of different hierarchical fidelity chains to calculate the overall fidelity score $\mathbf{F}(\cdot)$ of each answer.

$$\mathbf{F}(a_i) = \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{\text{sampled}}(\mathcal{C}_j) \cdot \mathbf{Fidelity}_{\mathcal{C}_j}(a_i), \quad (5)$$

3.3 Uncertainty Estimation

As shown in Section 3.1, through sampling, we can obtain the frequency of each answer generated by the model and use it to estimate the generation probability of each answer token. Previous works (Kadavath et al., 2022; OpenAI, 2023) have revealed that RLHF-LMs often exhibit overconfidence in token generation probability, especially in the temperature range we commonly use, such as between 0 and 1.0. However, these probabilities could still reveal, to some extent, the model's confidence regarding the current question $\mathcal{Q}$. For instance, if the distribution of $\mathcal{P}_{\text{sampled}}$ is flatter, it indicates that the language model has more significant uncertainty regarding the question $\mathcal{Q}$. An intuitive method is calculating the information entropy of the distribution $\mathcal{P}_{\text{sampled}}$ to estimate the model's uncertainty about question $\mathcal{Q}$ as follows:

$$\mathbf{Uncertainty}(\mathcal{Q}) = -\frac{\sum_{i=1}^{\mathrm{M}} p_i \cdot \log p_i}{\log \mathrm{M}}, \quad (6)$$

where M is the option number of question $\mathcal{Q}$. Since the range of the information entropy for $\mathcal{P}_{\text{sampled}}$ is from 0 to $\log \mathrm{M}$, we normalize the information entropy using $\log \mathrm{M}$.

3.4 Confidence Estimation

Given the model's Uncertainty for a given question $\mathcal{Q}$ and the fidelity $\mathbf{F}(\cdot)$ among different candidate

Method	ARC-Challenge				MMLU				CommonSenseQA				TruthfulQA			
	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$
GPT-3.5-TURBO																
Verb	**0.069**	0.200	0.681	75.597	0.138	0.200	0.795	59.028	0.087	0.178	0.660	71.253	0.215	0.178	0.792	57.405
Ling	0.083	0.464	0.451	75.683	0.197	0.472	0.441	56.019	0.109	0.250	0.451	71.499	0.271	0.667	0.669	59.241
Sampled	0.095	**0.067**	0.793	**79.266**	0.120	**0.022**	**0.922**	**63.151**	0.135	**0.067**	0.782	**74.590**	0.147	**0.044**	**0.901**	**59.333**
Ours	0.112	0.139	**0.897**	**79.266**	**0.088**	0.083	0.812	**63.151**	**0.073**	0.083	**0.812**	**74.590**	**0.074**	0.133	0.775	**59.333**
GPT-4-TURBO																
Verb	0.080	0.400	0.642	**92.833**	**0.045**	0.095	0.706	**81.25**	0.083	0.111	0.713	83.210	0.056	**0.044**	0.598	83.109
Ling	**0.040**	**0.036**	0.520	89.505	0.066	**0.083**	0.627	78.762	**0.056**	**0.071**	0.637	**83.702**	0.059	0.139	0.635	79.437
Sampled	0.067	0.200	0.221	**92.833**	0.153	0.311	0.536	80.324	0.121	0.133	0.541	83.866	0.091	0.178	0.478	**87.515**
Ours	0.127	0.083	**0.757**	**92.833**	0.089	**0.083**	**0.906**	80.324	0.109	0.083	**0.925**	83.866	**0.042**	**0.044**	**0.764**	**87.515**

Table 1: Experimental results derived from `GPT-3.5-Turbo` and `GPT-4-Turbo`. For each column in the table, the closer the color is to blue, the better the calibration. And the closer it is to orange, the worse the performance. We also have bolded the best results, and for the second-best results, we have added an underline beneath them.

Method	ARC-Challenge				MMLU				CommonSenseQA				TruthfulQA			
	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$	$ECE_{10}\downarrow$	$IPR_{10}\downarrow$	$CE_{10}\uparrow$	**Acc** $\uparrow$
Verb.	0.135	0.178	0.752	58.191	0.199	0.178	0.802	45.891	0.107	0.083	0.806	59.214	0.373	0.133	0.874	26.928
Ling	0.298	0.286	0.613	50.853	0.399	0.333	0.709	30.921	0.097	0.222	0.771	60.770	0.594	0.571	0.681	23.990
Sampled	0.121	0.044	**0.890**	**67.702**	0.162	0.067	**0.919**	52.315	0.110	0.044	0.857	70.762	0.236	0.133	**0.891**	34.517
Token	0.064	0.067	0.521	67.235	0.135	0.067	0.647	**54.803**	0.064	**0.022**	0.477	**71.007**	0.176	0.133	0.577	**34.761**
Ours	**0.063**	**0.028**	0.887	**67.702**	**0.076**	**0.028**	0.829	52.315	**0.051**	0.056	**0.886**	70.762	**0.080**	**0.028**	0.704	34.517

Table 2: Experimental results derived from `Baichuan2-13B-Chat`.

answers, the confidence of the model in its answer a_i for question $\mathcal{Q}$ is defined as follows:

$$\mathbf{Conf}(\mathcal{Q}, a_i) = \big(1 - \mathbf{Uncertainty}(\mathcal{Q})\big) \cdot \mathbf{F}(a_i), \tag{7}$$

4 Experiments

To validate the effectiveness of our proposed method, we conducted experiments on different RLHF-LMs such as `GPT-3.5-Turbo`[1], `GPT-4-Turbo` (OpenAI, 2023), `LLaMA2-Chat` (Touvron et al., 2023) and `Baichuan2-13B-Chat` (Yang et al., 2023a). To mitigate the influence of the sampling algorithm, unless specifically stated otherwise, we use hyper-parameters with a temperature of 1.0 and set top_p as 1.0.

4.1 Experimental Setting

Dataset. We have conducted experiments on four MCQA datasets to verify the effectiveness of our proposed confidence estimation method. ARC (Clark et al., 2018) is a dataset of 7,787 grade-school-level questions. We use the test split of the ARC-Challenge with 1,172 questions for our experiments. MMLU (Hendrycks et al., 2021) is a dataset designed to measure knowledge acquired during pretraining and covers 57 subjects. To reduce the cost of API calls, we sampled $\frac{1}{8}$ of the data for testing for each subject. CommonSenseQA (Talmor et al., 2019) is a dataset for commonsense question answering, and we use the validation split with 1,221 questions for experiments. TruthfulQA (Lin et al., 2022b) is a dataset that contains 817 questions designed to evaluate language models' preference to mimic some human falsehoods. All the experiments are conducted under a 0-shot setting.

Metrics. We utilize multiple metrics to evaluate. We bin the predictions from the model by their confidence and report the ECE (expected calibration error). We also report the Brier Score of different methods in Table 7. In this paper, we also defines two novel metrics to evaluate the calibration. The first one is IPR (**I**nverse **P**air **R**atio), which is used to measure the monotonicity of the reliability diagram. If the reliability diagram is monotonic, it indicates that the average accuracy of answers with low confidence is lower than the average accuracy of answers with high confidence.

$$\mathrm{IPR}_M = \frac{\mathrm{IP}}{C_K^2}, \tag{8}$$

where IP is the inverse pair number in the reliable diagram, and K is the bin number with a density larger than 0. We found that as the number of model parameters increases, the accuracy of the model improves across various datasets. However, language models still tend to consistently express uncertainty within certain fixed ranges, and ECE cannot clearly reflect this phenomenon. Therefore, we suggest using the CE (**C**onfidence **E**venness) to evaluate the uniformity of the density of each bar in the reliability diagram.

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

Method	ARC-Challenge				MMLU				CommonSenseQA				TruthfulQA			
	$\text{ECE}_{10}\downarrow$	$\text{IPR}_{10}\downarrow$	$\text{CE}_{10}\uparrow$	**Acc** $\uparrow$	$\text{ECE}_{10}\downarrow$	$\text{IPR}_{10}\downarrow$	$\text{CE}_{10}\uparrow$	**Acc** $\uparrow$	$\text{ECE}_{10}\downarrow$	$\text{IPR}_{10}\downarrow$	$\text{CE}_{10}\uparrow$	**Acc** $\uparrow$	$\text{ECE}_{10}\downarrow$	$\text{IPR}_{10}\downarrow$	$\text{CE}_{10}\uparrow$	**Acc** $\uparrow$
LLAMA2-7B-CHAT																
Verb	0.294	**0.083**	0.482	45.904	0.325	0.267	0.531	41.551	0.208	0.267	0.516	52.662	0.499	0.200	0.626	21.787
Ling	0.452	0.333	0.283	44.625	0.478	0.357	0.315	38.542	0.385	0.250	0.275	51.597	0.647	0.607	0.406	24.113
Sampled	0.329	0.156	0.781	**50.683**	0.316	0.222	**0.900**	**43.056**	0.294	0.178	0.765	54.627	0.389	0.133	**0.875**	**27.540**
Token	0.161	0.156	0.430	50.256	0.224	0.333	0.593	42.419	0.148	**0.133**	0.417	**54.791**	0.234	0.289	0.484	27.417
Ours	**0.073**	0.111	**0.921**	**50.683**	**0.102**	**0.167**	0.890	**43.056**	**0.053**	0.167	**0.903**	54.627	**0.121**	**0.083**	0.762	**27.540**
LLAMA2-13B-CHAT																
Verb	0.198	0.143	0.495	57.594	0.286	0.214	0.572	45.614	0.204	0.278	0.497	56.260	0.443	0.167	0.732	27.138
Ling	0.327	0.333	0.393	57.301	0.448	0.333	0.378	45.040	0.316	0.133	0.449	56.692	0.627	0.733	0.508	26.864
Sampled	0.297	0.200	0.653	**60.239**	0.351	0.267	0.788	47.251	0.287	0.156	0.717	**58.722**	0.461	0.422	**0.798**	29.131
Token	0.135	0.178	0.408	59.898	0.225	0.244	0.502	**47.512**	0.142	0.222	0.403	57.007	0.238	0.200	0.429	**30.845**
Ours	**0.069**	**0.111**	**0.886**	**60.239**	**0.070**	**0.083**	**0.852**	47.251	**0.043**	**0.083**	**0.883**	**58.722**	**0.121**	**0.083**	0.762	29.131
LLAMA2-70B-CHAT																
Verb	**0.071**	0.286	0.369	70.819	0.236	0.194	0.351	53.183	**0.069**	0.222	0.286	70.680	0.311	**0.028**	0.522	43.452
Ling	0.223	0.333	0.119	67.833	0.375	0.333	0.096	51.794	0.189	**0.067**	0.117	70.106	0.507	0.400	0.289	36.597
Sampled	0.220	0.311	0.475	72.867	0.325	0.289	0.289	56.308	0.212	0.089	0.551	**72.809**	0.351	0.156	0.622	51.897
Token	0.091	0.200	0.315	**73.208**	0.190	0.378	0.378	**56.597**	0.093	0.178	0.339	72.645	0.173	0.267	0.352	**52.020**
Ours	0.085	**0.111**	**0.908**	72.867	**0.066**	**0.083**	**0.898**	56.308	0.094	0.111	**0.918**	**72.809**	**0.093**	0.089	**0.804**	51.897

Table 3: Experimental results derived from `LLaMA-2-Chat`.

$$\text{CE}_M = -\frac{\sum_{i=1}^{\text{M}} p_i \cdot \log p_i}{\log \text{M}}, \quad (9)$$

In this paper, we adopt 10 equal-size bins to calculate ECE_{10}, IPR_{10} and CE_{10}. We also report the accuracy on these benchmarks to measure whether calibration reduces the accuracy.

Baselines. We compared our approach with different baselines for eliciting the confidence of language model. First, we reproduced the **Verb** and **Ling** method proposed by Tian et al. (2023). The **Verb** method involves prompting the model to output a floating-point number between 0 and 1 to represent its confidence immediately after providing an answer (Tian et al., 2023; Lin et al., 2022a). The **Ling** method entails having the language model express its confidence level in natural language (Tian et al., 2023). Since commercial models like `ChatGPT` do not provide per-token logits, we employed a sampling technique to estimate the probability of token generation, referred to as the **Sampled** method. Unless otherwise specified, the Sampled method involves sampling 10 times. For open-source models like `LLaMA2-Chat`, we directly use the probability of token generation as the measure of the language model's confidence, which we refer to as the **Token** method. We also compare the **Conformal** Prediction Baseline proposed by Kumar et al. (2023) with our UF calibration in Appendix B.1. All the prompt templates we use are shown in Appendix E.

4.2 Main Results

Tables 1–3 show our experimental results on `GPT-3.5-Turbo`, `GPT-4-Turbo`, `Baichuan2-13B-Chat`, and `LLaMA2-Chat`. Based on the experimental results, the following conclusions can be drawn:

1) Our proposed method demonstrates a clear improvement over the various baselines in terms of three metrics: ECE_{10}, IPR_{10}, and CE_{10}, which demonstrates the effectiveness of our method.
2) The Verb and Ling methods might, to some extent, impair the language model's accuracy on multiple-choice question answering tasks, which might be caused by more complicated instructions. Additionally, since the Ling method is more complex, it has a greater impact on the overall accuracy than the Verb method.
3) Similar to the conclusion from Tian et al. (2023), the calibration of the Verb method tends to be better than that of the Ling method. This is because the linguistic expressions used in the Ling method are based on human psychology. However, the confidence represented by the same expression may have a gap between humans and models and among different models and different sentences might mean the same thing (Kuhn et al., 2023).
4) The CE_{10} of the Verbalization-based Method is relatively low, which suggests that language models tends to prefer outputting expressions of certain confidence, such as `'Highly Likely'`, 0.8 and 0.9. This phenomenon can also explain why the ECE_{10} of the Verbalization-based Method improves when the overall average accuracy of the model is between 70-90%.

4.3 Ablation Study

As shown in Table 4, removing Uncertainty and only relying on Fidelity to estimate the model's

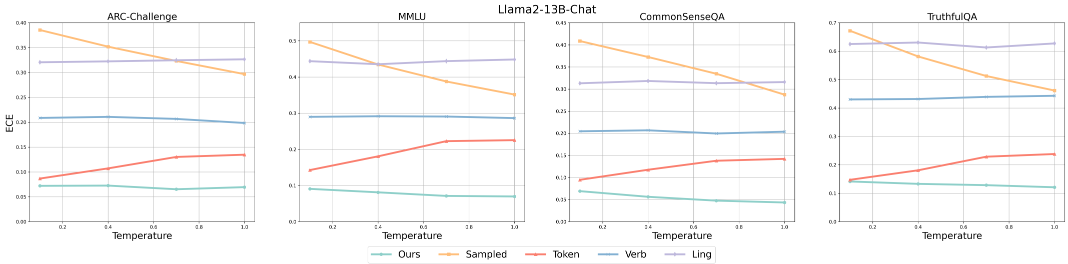

Figure 4: Our proposed method achieved well-calibrated results across all temperatures. The experimental results are derived from `LLaMA2-13B-Chat`. The results from `Baichuan2-13B-Chat` are presented in Appendix Figure 7.

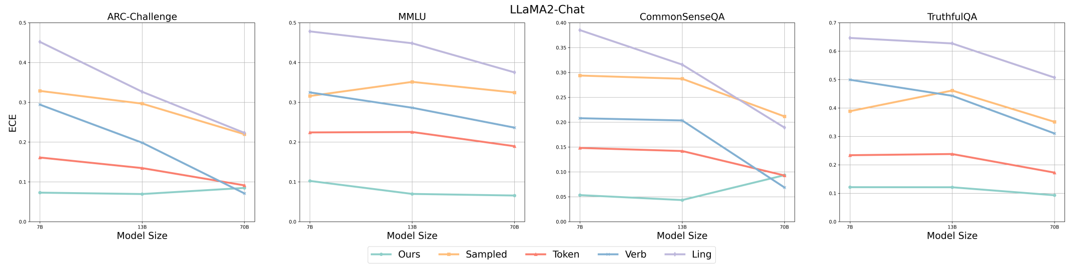

Figure 5: The experimental results are derived from `LLaMA2-Chat`.

confidence, we can also achieve comparatively better calibration than other methods. This phenomenon indicates that our proposed method reflects the language model's Fidelity to its answers very well. Meanwhile, it is difficult to estimate the model's confidence only depending on Uncertainty. As mentioned in 3.3, Uncertainty is designed for measuring the model's uncertainty regarding the question $\mathcal{Q}$, rather than its confidence for a particular answer. In the section 3.2, we utilize (4) to calculate the language model's normalized fidelity in a hierarchical fidelity chain, where τ is a hyper-parameter. The larger the value of τ, the lower the estimated fidelity for answers closer to the end of the fidelity chain. Our experiments in Table 4 indicate that setting τ to around 2 is a relatively appropriate choice for the fidelity estimation process. If τ is too large, the ECE_{10} will also increase, which will cause the issue of overconfidence of our estimated confidence.

Method	ARC	MMLU	CSQA	TruthfulQA	Avg.
Ours	**0.069**	**0.070**	**0.043**	**0.121**	**0.076**
w/o. Uncertainty	0.122	0.184	0.115	0.202	0.156
w/o. Fidelity	0.675	0.614	0.704	0.677	0.668
$\tau = 1.5$	0.103	**0.064**	0.066	**0.082**	0.079
$\tau = 2.0$ (Default)	0.069	0.070	0.043	0.121	**0.076**
$\tau = 2.5$	**0.067**	0.089	**0.040**	0.142	0.085
$\tau = 3.0$	0.074	0.107	0.050	0.155	0.097
$\tau = 4.0$	0.085	0.138	0.075	0.165	0.116
$\tau = 5.0$	0.102	0.158	0.094	0.183	0.134
Best Result (Others)	0.135	0.225	0.142	0.238	0.185

Table 4: Ablation study of our method. The results (ECE_{10}) are derived from `LLaMA2-13B-Chat`.

5 Analysis and Discussion

To take a closer look at the difference between different calibration methods tailored for language models, in this section, we verify the robustness of our method from two aspects: *Temperature-Scaling* and *Parameter-Scaling*. Meanwhile, we also conducted a detailed discussion of a research question: *What kind of Confidence is Truly Well-Calibrated?*

Temperature-Scaling In the main experiments, we evaluate various methods using a constant temperature of 1.0. In this section, we will explore the influence of sampling temperature on the performance of different methods. As illustrated in Figures 4 and 7, our proposed calibration method consistently achieves the lowest expected calibration error across all temperatures, showing remarkable robustness to temperature variations. This is because, in eliciting model fidelity, our method always employs Greedy Decoding rather than Sampling. Thus, the hierarchical chains we obtain are usually consistent across different sampling temperatures. In contrast, the expected calibration error of Logit-based Methods is usually affected by temperature. For the Sampling method with limited sampling budgets, the lower the temperature, the more significantly the diversity of the sampled results will decrease, exacerbating the overconfidence of language models. For the Token Method, the impact of temperature on its calibration shows a trend of "*first increasing and then remaining relatively*

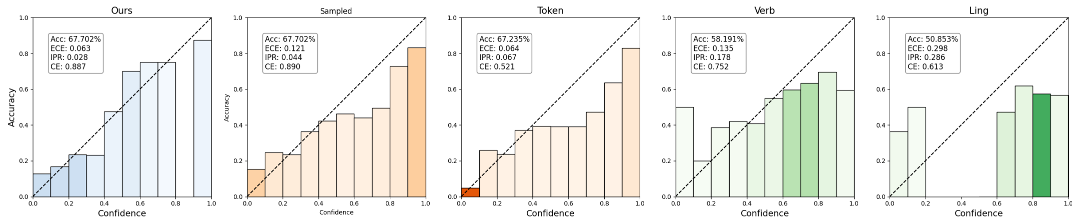

Figure 6: Reliability diagrams of `Baichuan2-13B-Chat` on ARC-Challenge. In these diagrams, the darker the color, the higher the density. The reliability diagrams of other models we evaluated are shown in Appendix Figures 8–13.

stable" or "*first increasing and then decreasing*". This is because we could directly utilize (1) to estimate the confidence of each option, and if the temperature is too low (i.e., 0.1), it will lead to the confidence of a large number of options approaching zero. This phenomenon might contribute to reducing expected calibration error, but it does not necessarily indicate that the model's confidence is well-calibrated. The Verbalization-based method is less affected by temperature, which indicates that the expressions which language models prefer to output are relatively consistent across different temperatures.

Parameter-Scaling As shown in Figure 5, we evaluate the calibration of various methods at different parameter scales on the `LLaMA2-Chat` series models. Our proposed method exhibits good calibration across different amounts of model parameters. With the size of model parameters increasing, the calibration of the Verbalization-based method and the Logit-based method is improving. This phenomenon indicates that as the scale of model parameters increases, the model's Self-Awareness is improving. However, the relatively high expected calibration error suggests that language models still have issues with overconfidence.

Truly Well-Calibrated Confidence Previous work mainly evaluates the calibration of language models through ECE. This section will discuss the research question: "*What Kind of Confidence is Truly Well-Calibrated?*". Figure 6 demonstrates the calibration of various methods. From the calibration perspective, we hope that the confidence and accuracy relationship is close to the curve $y = x$. Thus, we need to reduce the ECE by calibrating confidence. Meanwhile, we hope that the reliability diagram should be as monotonic as possible to ensure that the accuracy of the results generated with low confidence is lower than that of the results with high confidence. Therefore, we propose the *Inverse Pair Ratio* (IPR) to evaluate monotonicity. From the perspective of building a more honest system, we hope the model's confidence should be distributed across different confidence intervals. For example, if a language model has an overall accuracy of 75% on the TruthfulQA dataset and the confidence of each question from the language model is always 75%, its ECE and IPR would be 0. And we find that different models tend to express confidence within a fixed interval. In this case, we think that the confidence may not necessarily be a truly well-calibrated confidence because we could not exclude some low-confidence results based on the confidence from the language model. Although the prior distribution of the model's confidence is unknown, our confidence estimation method finds that language models have different confidence for different questions. Thus, we propose a metric called *Confidence Evenness* (CE) to measure whether the model confidence always is located in a fixed interval. We believe ECE, IPR, and CE evaluate calibration from different perspectives and there is a trade-off between these three metrics. We suggest that truly well-calibrated confidence should achieve a balance among ECE, IPR, and CE, rather than over-optimizing any of them.

6 Conclusion

In this paper, we decompose the language model confidence into the *Uncertainty* about the question and the *Fidelity* to the answer generated by language models. Through the decomposition, we propose a plug-and-play method, UF CALIBRATION, to calibrate the confidence of language models. Through experiments with 6 RLHF-LMs on 4 multiple-choice question answering benchmarks, our method exhibits good calibration. Besides, we propose two novel metrics, IPR and CE, to evaluate the calibration of language models. Finally, we conduct a detailed discussion on *Truly Well-Calibrated Confidence*. We believe our method can serve as a strong baseline, and we hope that this work could provide some insights into the language model confidence calibration.

Limitations

Although our method has shown good calibration, it is mainly applicable to scenarios where the set of answers is known, i.e., multiple-choice question answering, text classification, sentiment classification, and preference labeling in RLHF. Eliciting the model's fidelity in open-ended generation scenarios is a direction worth exploring. Meanwhile, our method involves multiple invocations of language models, and how to estimate the probability distribution of tokens generated by the language model with as few callings as possible remains to be studied.

References

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. 2021. A general language assistant as a laboratory for alignment.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022. Constitutional ai: Harmlessness from ai feedback.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. ArXiv preprint arXiv:2303.12712.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

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, abs/1803.05457.

Wade Fagen-Ulmschneider. 2023. Perception of probability words. Ms., UIUC, 05-24-2023.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1321–1330. PMLR.

Guande He, Peng Cui, Jianfei Chen, Wenbo Hu, and Jun Zhu. 2023. Investigating uncertainty calibration of aligned language models under the multiple-choice setting.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In International Conference on Learning Representations.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. Transactions of the Association for Computational Linguistics, 9:962–977.

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

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In The Eleventh International Conference on Learning Representations.

Bhawesh Kumar, Charlie Lu, Gauri Gupta, Anil Palepu, David Bellamy, Ramesh Raskar, and Andrew Beam. 2023. Conformal prediction with large language models for multi-choice question answering. arXiv preprint arXiv:2305.18404.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022a. Teaching models to express their uncertainty in words. Transactions on Machine Learning Research.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022b. TruthfulQA: Measuring how models mimic human falsehoods. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In Advances in Neural Information Processing Systems, volume 35, pages 27730–27744. Curran Associates, Inc.

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 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, Singapore. Association for Computational Linguistics.

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

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

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023a. Baichuan 2: Open large-scale language models.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. 2023b. Alignment for honesty.

Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuanjing Huang. 2023. Do large language models know what they don't know? In Findings of the Association for Computational Linguistics: ACL 2023, pages 8653–8665, Toronto, Canada. Association for Computational Linguistics.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2023. Instruction tuning for large language models: A survey.

Theodore Zhao, Mu Wei, J. Samuel Preston, and Hoifung Poon. 2023. Automatic calibration and error correction for generative large language models via pareto optimal self-supervision.

Yunhua Zhou, Pengyu Wang, Peiju Liu, Yuxin Wang, and Xipeng Qiu. 2024. The open-world lottery ticket hypothesis for OOD intent classification. In Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), pages 15988–15999, Torino, Italia. ELRA and ICCL.

A Algorithm

The pseudo code of our proposed method is shown in Algorithm 1. It should be clarified that, as long as a candidate answer a_i appears in the answer set $\mathcal{A}$ or the Fidelity chain set $\mathcal{S}$, we could estimate its confidence through (7).

Algorithm 1 Algorithm

Require: Input question $\mathcal{Q}$, Option list $\mathcal{O}$, Answer set $\mathcal{A} = \varnothing$, Sampling budget K, RLHF-LM LM, o^* is "All other options are wrong.", Fidelity chain set $\mathcal{S}$, $\mathbf{U}(\cdot)$ refers to (6).

$t \leftarrow 0$
while $t < K$ **do**
  $a_i \leftarrow \mathrm{LM}(Q, \mathcal{O})$ ▷ Sampling answer
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{a_i\}$
  $\mathcal{P}_{sampled}(a_i) \leftarrow \mathcal{P}_{sampled}(a_i) + 1$
  $t \leftarrow t + 1$ ▷ Continue sampling
end while
$\mathcal{P}_{sampled}(a_i) \leftarrow \mathcal{P}_{sampled}(a_i)/K$

$\mathbf{Uncertainty}(\mathcal{Q}) = \mathbf{U}(\mathcal{P}_{sampled})$ ▷ Get uncertainty
$i \leftarrow 0$
while $|\mathcal{A}| > 0$ **do**
  $\mathcal{A} \leftarrow A \setminus \{a_i\}$ ▷ Select a answer
  $\mathcal{O}^* \leftarrow (\mathcal{O} \setminus \{o_i\}) \cup o_*$ ▷ Replace option
  $\mathcal{C}_i = a_i$ ▷ Init a fidelity chain
  while $|\mathcal{O}^*| > 0$ **do**
    $a^* \leftarrow \mathrm{LM}(\mathcal{Q}, \mathcal{O}^*)$ ▷ Greedy decoding
    if $a^* \neq a_i$ **then** ▷ Low fidelity
      $\mathcal{O}^* \leftarrow \mathcal{O}^* \setminus \{o_i\}$ ▷ Delete option
      $a_i = a^*$
      $\mathcal{C}_i = (\mathcal{C}_i \rightarrow a_*)$ ▷ Add element
    else
      break ▷ High fidelity
    end if
  end while
  $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{C}_i$
  $i \leftarrow i + 1$
end while

$\mathbf{F}(a_i) = \sum_{j=1}^{|\mathcal{A}|} \mathcal{P}_{\mathrm{sampled}}(\mathcal{C}_j) \cdot \mathbf{Fidelity}_{\mathcal{C}_j}(a_i)$ ▷ Get fidelity
$\mathrm{Conf}(\mathcal{Q}, a_i) = (1 - \mathbf{Uncertainty}(\mathcal{Q})) \cdot \mathbf{F}(a_i)$ ▷ Get confidence
return $\mathrm{Conf}(\mathcal{Q}, a_i)$ ▷ Return the confidence of answer a_i

Model	Is the answer chosen in the first round correct?	Choose "All other options are wrong." after replacing	Do not choose "All other options are wrong." after replacing
GPT-3.5-TURBO	**True**	25.99%	33.27%
	False	5.85%	34.88%
	Acc.	81.61%	48.82%
GPT-4-TURBO	**True**	70.75%	16.83%
	False	3.00%	9.42%
	Acc.	95.93%	64.10%
BAICUAN2-13B-CHAT	**True**	5.14%	29.40%
	False	4.22%	61.24%
	Acc.	54.90%	32.43%
LLAMA2-7B-CHAT	**True**	3.92%	23.50%
	False	4.83%	67.75%
	Acc.	44.76%	25.75%
LLAMA2-13B-CHAT	**True**	3.55%	25.64%
	False	2.82%	67.99%
	Acc.	55.77%	27.39%
LLAMA2-70B-CHAT	**True**	13.59%	38.43%
	False	3.98%	44.00%
	Acc.	77.35%	46.62%

Table 5: We found that if the option chosen by the model in the first round is replaced with "All other options are wrong," the model then chooses "All other options are wrong" in the second round. In this case, the accuracy of the model's first-round choice is significantly higher compared to when it chooses other options in the second round. The results are derived from TruthfulQA.

B Additional Results

B.1 Compared with Conformal Prediction

We reproduce Conformal Prediction for RLHF-LMs (Kumar et al., 2023) in our dataset and setting. Specifically, for each dataset, we select 50% samples as the calibration set and the other samples as the test set. We also set the error rate to $\alpha = 0.1$ meaning the prediction answer set has a 90% probability of containing the correct answer. We then calculate the conformal scores in the calibration set, where the specific calculation formula is $Score = 1 - \max SoftmaxScore$. For the test set, we take the $1 - \alpha$ quantile of the conformal scores from the calibration set as the threshold q. During the testing stage, for a given sample, it is only added to the prediction set if its generated probability is greater than or equal to $1 - q$. For each sample in the prediction set, we consider its confidence to be $(1 - \alpha) \cdot (SoftmaxScore)$. as shown in the following table 6, our proposed UF Calibration still demonstrates good calibration compared to conformal prediction for RLHF-LMs. It is also important to note that conformal prediction requires a calibration set to determine a threshold to build a prediction set. However, our method is a plug-and-play approach that can accurately estimate the model's confidence without requiring any prior knowledge.

B.2 Brier Score

Besides the ECE metric, the Brier Score is also commonly used as an evaluation criterion for model calibration.

$$\text{BrierScore} = \frac{1}{N}\sum_{t=1}^{N}(f_t - o_t)^2, \qquad (10)$$

where f_t is the probability and o_t is the label. Accordingly, f_t can be referred to as the model's confidence, while o_t represents whether it is the correct answer (0 indicating an incorrect answer, 1 indicating a correct answer). In Table 7, we present the Brier Scores of various baselines and our proposed method. It can be seen that our method still exhibits good calibration, especially for closed-source models such as `GPT-3.5-Turbo`, `GPT-4 Turbo`.

C Why could CE be used as a metric?

As mentioned in section 4.2, we found that Language models tend to prefer outputting expressions of certain confidence, such as 'Highly Likely', 0.8, and 0.9. In the table 10, we have counted the occurrence of different confidence levels for various models on different datasets to demonstrate the model's preference for certain confidence levels when using the Verb and Ling method.

We also notice that as the model parameters increased, the accuracy of the model improved, but the language model's preference for certain confidence levels do not change and even became stronger. Therefore, we introduced the Confidence Evenness to assess whether the model's confidence is overly concentrated in certain intervals.

Can existing metrics (such as ECE) capture this phenomenon? There is an example: on CommonSenseQA, as the parameters of Llama2-Chat increasing, the accuracy rises from 51% to 70%,

Model	Dataset	Method	$\text{ECE}_{10} \downarrow$	BS $\downarrow$	$\text{CE}_{10} \uparrow$	$\text{IPR}_{10} \downarrow$
GPT-3.5-TURBO	MMLU	Conformal Prediction	**0.086**	0.189	**0.897**	0.111
		Ours	0.088	**0.170**	0.812	**0.083**
	TruthfulQA	Conformal Prediction	0.115	0.197	**0.884**	**0.028**
		Ours	**0.074**	**0.153**	0.775	0.133
	CommonSenseQA	Conformal Prediction	0.079	0.173	0.699	0.139
		Ours	**0.073**	**0.139**	**0.812**	**0.083**
	ARC	Conformal Prediction	**0.039**	0.142	**0.670**	0.143
		Ours	0.112	**0.141**	0.897	**0.139**
GPT-4-TURBO	MMLU	Conformal Prediction	**0.084**	0.164	0.482	0.472
		Ours	0.089	**0.142**	**0.906**	**0.083**
	TruthfulQA	Conformal Prediction	0.046	0.112	0.425	0.222
		Ours	**0.042**	**0.102**	**0.764**	**0.044**
	CommonSenseQA	Conformal Prediction	**0.040**	**0.130**	0.509	0.194
		Ours	0.109	0.134	**0.925**	**0.083**
	ARC	Conformal Prediction	**0.084**	**0.026**	0.000	**0.000**
		Ours	0.127	0.095	**0.757**	0.083
BAICHUAN2-13B-CHAT	MMLU	Conformal Prediction	0.130	0.218	**0.888**	0.056
		Ours	**0.076**	**0.193**	0.829	**0.028**
	TruthfulQA	Conformal Prediction	0.209	0.239	**0.865**	0.250
		Ours	**0.080**	**0.149**	0.704	**0.028**
	CommonSenseQA	Conformal Prediction	0.056	0.162	0.801	**0.056**
		Ours	**0.051**	**0.153**	**0.886**	**0.056**
	ARC	Conformal Prediction	**0.061**	0.173	**0.848**	**0.028**
		Ours	0.063	**0.166**	0.887	**0.028**
LLAMA2-7B-CHAT	MMLU	Conformal Prediction	0.253	0.290	0.864	0.361
		Ours	**0.102**	**0.214**	**0.890**	**0.167**
	TruthfulQA	Conformal Prediction	0.353	0.361	**0.825**	0.361
		Ours	**0.121**	**0.186**	0.762	**0.083**
	CommonSenseQA	Conformal Prediction	0.234	0.283	0.655	0.333
		Ours	**0.053**	**0.181**	**0.907**	**0.167**
	ARC	Conformal Prediction	0.260	0.308	**0.701**	**0.083**
		Ours	**0.073**	**0.204**	0.921	0.111
LLAMA2-13B-CHAT	MMLU	Conformal Prediction	0.279	0.317	0.740	0.250
		Ours	**0.070**	**0.196**	**0.852**	**0.083**
	TruthfulQA	Conformal Prediction	0.429	0.416	0.728	0.611
		Ours	**0.121**	**0.180**	**0.762**	**0.083**
	CommonSenseQA	Conformal Prediction	0.220	0.274	0.647	0.250
		Ours	**0.043**	**0.166**	**0.883**	**0.111**
	ARC	Conformal Prediction	0.212	0.260	0.611	0.361
		Ours	**0.069**	**0.178**	**0.886**	**0.111**
LLAMA2-70B-CHAT	MMLU	Conformal Prediction	0.260	0.305	0.592	0.250
		Ours	**0.066**	**0.189**	**0.898**	**0.083**
	TruthfulQA	Conformal Prediction	0.281	0.301	0.558	0.306
		Ours	**0.093**	**0.162**	**0.804**	**0.089**
	CommonSenseQA	Conformal Prediction	0.156	0.221	0.479	0.333
		Ours	**0.094**	**0.156**	**0.908**	**0.111**
	ARC	Conformal Prediction	0.118	0.189	0.427	0.361
		Ours	**0.085**	**0.154**	**0.908**	**0.111**

Table 6: Comparing calibration results of Conformal Prediction of RLHF-LMs (Kumar et al., 2023) and our proposed method.

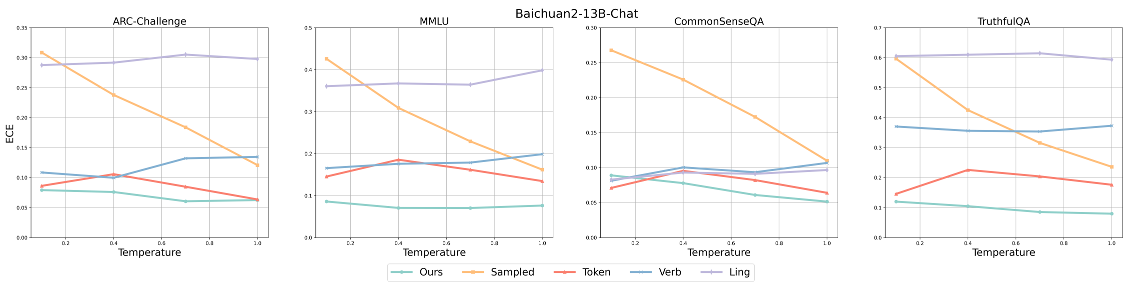

Figure 7: The Impact of Temperature on Different Methods. Our proposed method achieved well-calibrated results across all temperatures. The experimental results are derived from `Baichuan2-13B-Chat`.

Model	Method	ARC-Challenge	MMLU	CommonSenseQA	TruthfulQA	Avg.
GPT-3.5-TURBO	Verb	0.181	0.247	0.189	0.274	0.223
	Ling	0.197	0.278	0.204	0.318	0.249
	Sampled	0.157	0.202	0.216	0.206	0.195
	Conformal	0.142	0.189	0.173	0.197	0.175
	Ours	**0.141**	**0.170**	**0.139**	**0.153**	**0.151**
GPT-4-TURBO	Verb	0.181	0.247	0.204	0.274	0.227
	Ling	0.198	0.278	0.216	0.318	0.253
	Sampled	0.074	0.174	0.147	0.112	0.127
	Conformal	**0.026**	0.164	**0.130**	0.112	**0.108**
	Ours	0.095	**0.142**	0.134	**0.102**	0.118
BAICHUAN2-13B-CHAT	Verb	0.257	0.294	0.239	0.363	0.288
	Ling	0.336	0.407	0.235	0.553	0.383
	Sampled	0.196	0.236	0.186	0.262	0.220
	Token	**0.095**	**0.168**	**0.092**	0.198	**0.138**
	Conformal	0.173	0.218	0.162	0.239	0.198
	Ours	0.166	0.193	0.153	**0.149**	0.165
LLAMA2-7B-CHAT	Verb	0.332	0.348	0.283	0.449	0.353
	Ling	0.451	0.471	0.396	0.609	0.4821
	Sampled	0.358	0.350	0.323	0.411	0.360
	Token	**0.171**	0.238	**0.158**	0.246	0.203
	Conformal	0.308	0.290	0.283	0.361	0.311
	Ours	0.204	**0.214**	0.181	**0.186**	**0.196**
LLAMA2-13B-CHAT	Verb	0.277	0.320	0.272	0.394	0.316
	Ling	0.352	0.448	0.343	0.599	0.435
	Sampled	0.318	0.374	0.317	0.470	0.370
	Token	**0.141**	0.233	**0.150**	0.242	0.192
	Conformal	0.260	0.317	0.274	0.416	0.317
	Ours	0.178	**0.196**	0.166	**0.180**	**0.180**
LLAMA2-70B-CHAT	Verb	0.206	0.297	0.208	0.332	0.261
	Ling	0.267	0.390	0.240	0.496	0.348
	Sampled	0.236	0.347	0.237	0.360	0.295
	Token	**0.094**	0.196	**0.098**	0.174	**0.141**
	Conformal	0.189	0.305	0.221	0.301	0.254
	Ours	0.154	**0.189**	0.156	**0.162**	0.165

Table 7: The Brier Score of different methods from six RLHF-Models on four MCQA datasets.

and the ECE using the Ling method decrease from 0.385 to 0.189. But the 70B model shows a stronger preference for outputting a confidence of 0.9. Focusing solely on the ECE metric cannot fully observe the changes in model preferences. Fortunately, this phenomenal could be reflected by the CE metrics.

Another extreme case is if models of varying parameter sizes always output a 0.9 confidence level, and as the model size increases, the average accuracy just shifts from 70% to 90%, then the ECE would drop to 0. If we only use existing metrics for observation, we might conclude that the model with the largest parameters has the strongest self-awareness. However, by evaluating the CE metric across different models, we can identify a potential preference in how models express confidence. Its ECE becoming 0 might just coincidentally be because the average accuracy on a certain dataset equals the confidence level it prefers to output. Therefore, we believe the CE metric provides a new perspective for observing model confidence calibration.

Finally, it should be noted that we believe an over-concentration of model confidence in a particular value or interval is not conducive to using model confidence as a simple metric to filter out low-confidence answers.

D The Computation Cost of Eliciting Fidelity

In this section, we will display the average length of the fidelity chains for different models across various datasets in the Table 8. Since we deploy greedy decoding during the process of eliciting fidelity,the average length of the fidelity chain is equal to the average number of requests. At the same time, it should be noted that, when eliciting the Fidelity Chain, only 1 token needs to be generated. Therefore, the average length of the fidelity chain can also be regarded as the average number of tokens generated.

Model	ARC-Challenge	MMLU	CommonSenseQA	TruthfulQA	Avg.
GPT-3.5-Turbo	2.774	2.984	3.052	3.275	3.021
GPT-4-Turbo	1.492	1.915	2.157	1.616	1.795
Baichuan2-13B-Chat	2.830	2.820	2.889	4.345	3.221
Llama2-7B-Chat	2.467	2.631	2.771	3.944	2.953
Llama2-13B-Chat	2.725	2.875	2.956	4.100	3.164
Llama2-70B-Chat	2.384	2.563	2.455	3.284	2.671

Table 8: The average length of the fidelity chains for different models across various datasets

E Prompt Templates

We use the prompt template from Tian et al. (2023) for a fair comparison. The prompt template for each baseline is provided in Table 11. The question is substituted for the variable `${THE_QUESTION}` in each prompt. Table 9 shows the linguistic expression list of confidence we used for the Ling Method, which originates from Fagen-Ulmschneider (2023).

F Reliability Diagram

We provide the reliability diagrams of all the RLHF-LMs we evaluated in Figures 8-13. In a reliability diagram, the darker the color of the bar, the greater its density is, which indicates a preference for the confidence the language models express. Although the average accuracy of various RLHF-LMs is quite different, these model always prefer to express their confidence about 70-90% in verbalized methods.

Linguistic Expression	Confidence Score
'Certain'	1.0
'Almost Certain'	0.95
'Highly Likely'	0.9
'Very Good Chance'	0.8
'We Believe'	0.75
'Probably'	0.7
'Probable'	0.7
'Likely'	0.7
'Better than Even'	0.6
'About Even'	0.5
'Probably Not'	0.25
'We Doubt'	0.2
'Unlikely'	0.2
'Little Chance'	0.1
'Chances are Slight'	0.1
'Improbable'	0.1
'Highly Unlikely'	0.05
'Almost No Chance'	0.02
'Impossible'	0.0

Table 9: The `EXPRESSION_LIST` we used for the Ling Method.

Dataset	Method	Model	0.0	0.02	0.05	0.1	0.2	0.25	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.95	1.0	$\mathrm{ECE}_{10}\downarrow$	$\mathrm{CE}_{10}\uparrow$	**Acc** $\uparrow$
CSQA	Verb	LLaMA2-7B-Chat	3	0	0	1	25	0	23	5	78	10	309	727	19	0	21	0.208	0.516	52.662
		LLaMA2-13B-Chat	11	0	0	0	9	0	1	29	7	112	108	851	61	0	32	0.204	0.497	56.260
		LLaMA2-70B-Chat	6	0	0	2	2	0	3	3	1	23	221	955	2	0	3	0.069	0.286	70.680
	Ling	LLaMA2-7B-Chat	11	0	21	0	3	0	0	0	1	5	2	13	1020	75	70	0.385	0.275	51.597
		LLaMA2-13B-Chat	18	1	11	0	6	0	0	0	0	0	3	194	892	96	0	0.316	0.449	56.692
		LLaMA2-70B-Chat	0	0	26	0	0	0	0	0	0	1	2	2	1172	2	16	0.189	0.117	70.106
MMLU	Verb	LLaMA2-7B-Chat	14	0	0	3	46	0	21	16	65	44	488	981	26	0	24	0.325	0.531	41.551
		LLaMA2-13B-Chat	23	0	0	0	41	0	0	54	7	227	278	1056	18	0	24	0.286	0.572	45.614
		LLaMA2-70B-Chat	1	0	0	0	7	0	3	1	2	9	518	1159	1	0	27	0.236	0.351	53.183
	Ling	LLaMA2-7B-Chat	47	0	101	0	21	0	0	0	6	4	7	12	1408	77	45	0.478	0.315	38.542
		LLaMA2-13B-Chat	81	1	15	0	4	2	0	0	0	0	4	84	1261	261	11	0.448	0.378	45.040
		LLaMA2-70B-Chat	3	0	31	0	0	0	0	0	0	6	2	5	1673	1	7	0.375	0.096	51.794
ARC	Verb	LLaMA2-7B-Chat	4	0	0	0	26	0	13	6	53	5	216	800	20	0	29	0.294	0.482	45.904
		LLaMA2-13B-Chat	1	0	0	0	31	0	0	13	13	68	129	851	18	0	47	0.198	0.495	57.594
		LLaMA2-70B-Chat	3	0	0	0	11	0	3	0	2	6	288	836	3	0	20	0.071	0.369	70.819
	Ling	LLaMA2-7B-Chat	3	0	24	0	10	0	0	0	0	0	5	10	1023	53	44	0.452	0.283	44.625
		LLaMA2-13B-Chat	1	0	5	0	5	0	0	0	0	0	1	76	914	162	8	0.327	0.393	57.301
		LLaMA2-70B-Chat	3	0	27	1	0	0	0	0	0	3	1	1	1121	2	13	0.223	0.119	67.833
TruthfulQA	Verb	LLaMA2-7B-Chat	10	0	0	1	23	0	8	2	125	18	167	406	17	0	40	0.499	0.626	21.787
		LLaMA2-13B-Chat	11	0	0	1	11	0	0	56	34	145	116	369	26	0	48	0.443	0.732	27.138
		LLaMA2-70B-Chat	3	0	0	0	7	0	4	4	4	22	320	404	9	0	30	0.311	0.522	43.452
	Ling	LLaMA2-7B-Chat	30	0	53	0	10	0	0	0	8	4	4	15	611	43	39	0.647	0.406	24.113
		LLaMA2-13B-Chat	39	2	19	0	4	0	0	0	0	0	4	40	526	177	6	0.627	0.508	26.864
		LLaMA2-70B-Chat	10	0	31	0	0	0	0	0	0	3	0	9	718	12	31	0.507	0.289	36.597

Table 10: Language models tend to prefer outputting expressions of certain confidence, such as 0.8, and 0.9.

Method	**Prompt Template**
Verb (Tian et al., 2023)	Provide your best guess and the probability that it is correct (0.0 to 1.0) for the following question. Give ONLY the guess and probability, no other words or explanation. For example:\nGuess: <most likely option, without any extra commentary whatsoever; just the option>\nProbability: <the probability between 0.0 and 1.0 that your guess is correct, without any extra commentary whatsoever; just the probability!>\nThe question is: {question}\nOptions:\n{choices}Answer:
Ling (Tian et al., 2023)	Provide your best guess for the following question, and describe how likely it is that your guess is correct as one of the following expressions: {EXPRESSION_LIST}. Give ONLY the guess and your confidence, no other words or explanation. For example:\n\n Guess: <most likely guess, as short as possible; not a complete sentence, just the guess!>\n Confidence: <description of confidence, without any extra commentary whatsoever; just a short phrase!>\n The question is: {question}\n Options:\n{choices}Answer:
Sampled	Provide the option you agree with most for the following question. Give ONLY the option of the answer, no other words or explanation. For example:\nAnswer: <most likely option, without any extra commentary whatsoever; just the option>\nThe question is: {question}\nOptions:\n{choices}Answer:
Token	Provide the option you agree with most for the following question. Give ONLY the option of the answer, no other words or explanation. For example:\nAnswer: <most likely option, without any extra commentary whatsoever; just the option>\nThe question is: {question}\nOptions:\n{choices}Answer:
Ours	Provide the option you agree with most for the following question. Give ONLY the option of the answer, no other words or explanation. For example:\nAnswer: <most likely option, without any extra commentary whatsoever; just the option>\nThe question is: {question}\nOptions:\n{choices}Answer:

Table 11: Prompt templates for each method evaluated.

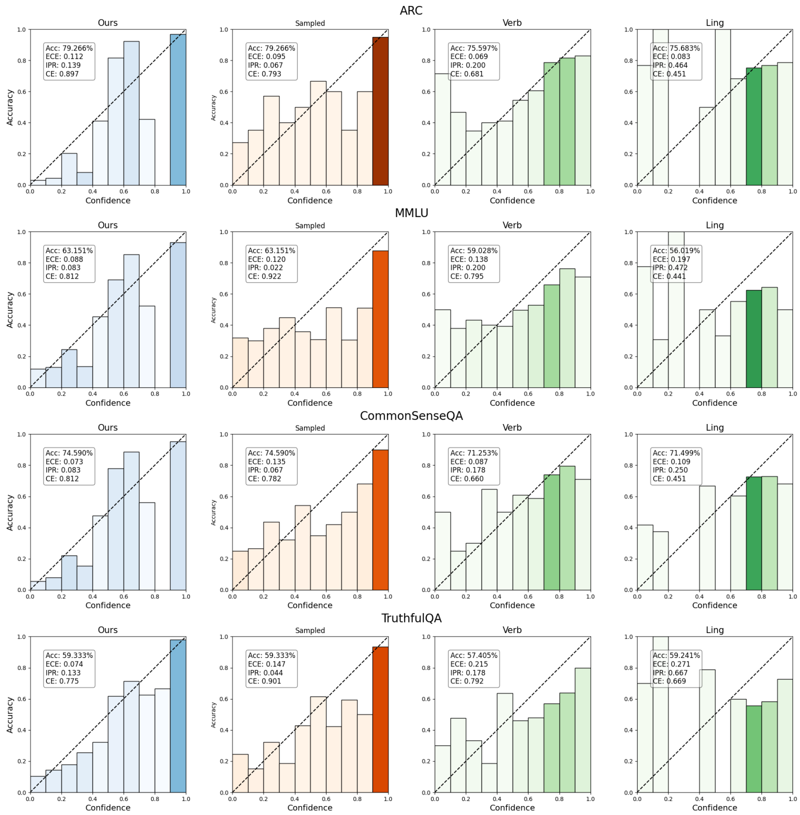

Figure 8: The experimental results are derived from GPT-3.5-Turbo on 4 MCQA datasets.

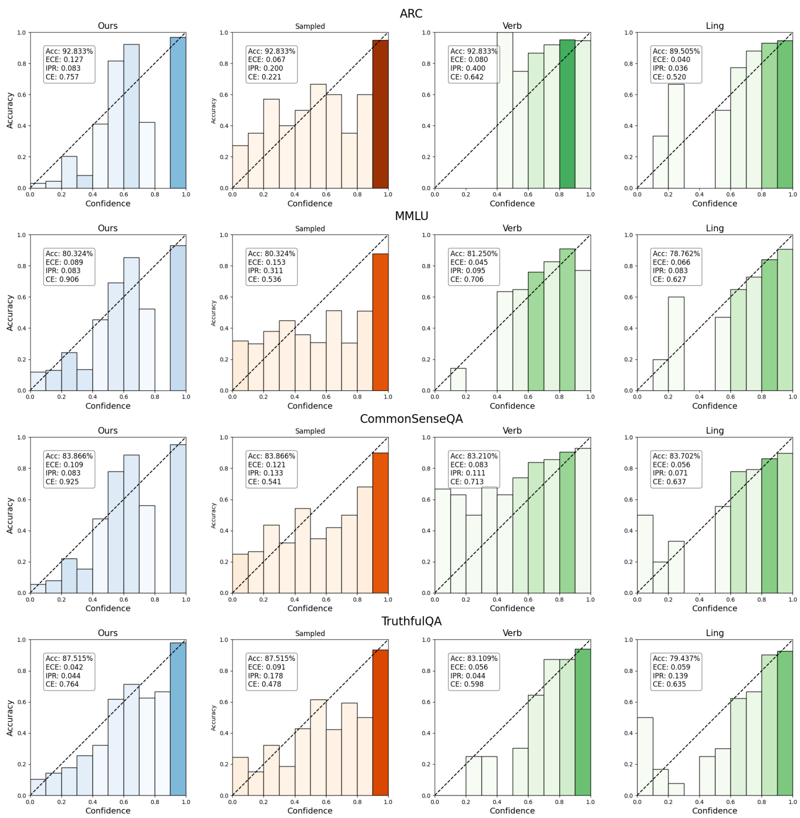

Figure 9: The experimental results are derived from GPT-4-Turbo on 4 MCQA datasets.

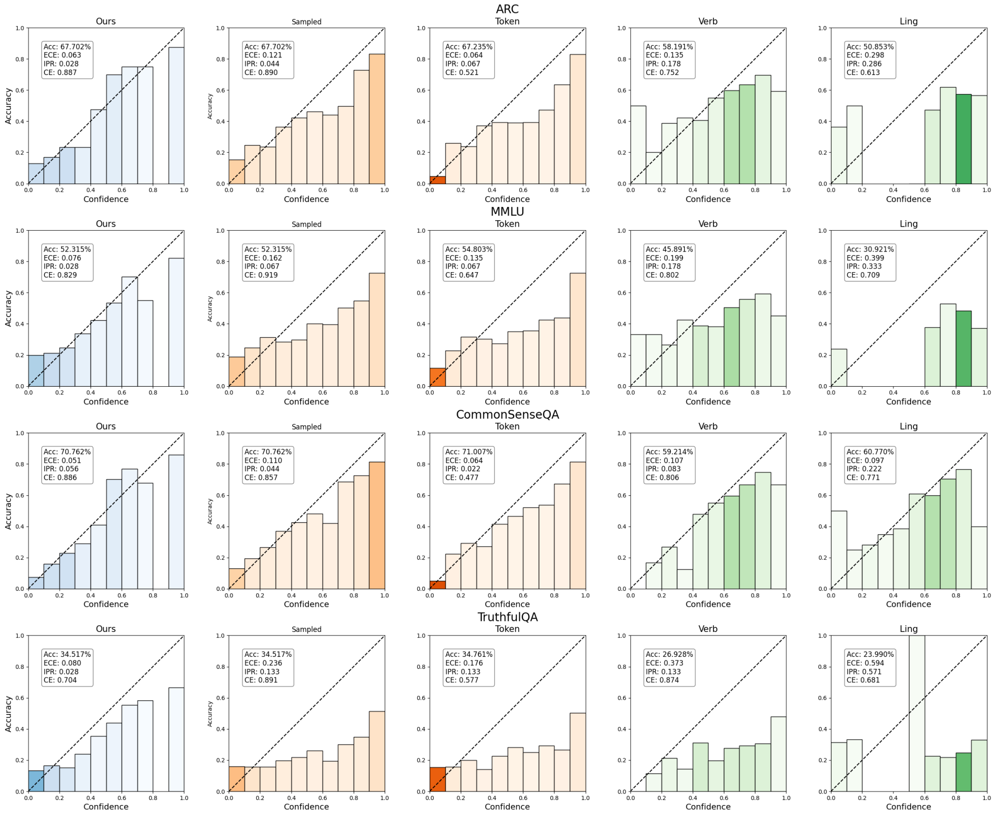

Figure 10: The experimental results are derived from `Baichuan2-13B-Chat` on 4 MCQA datasets.

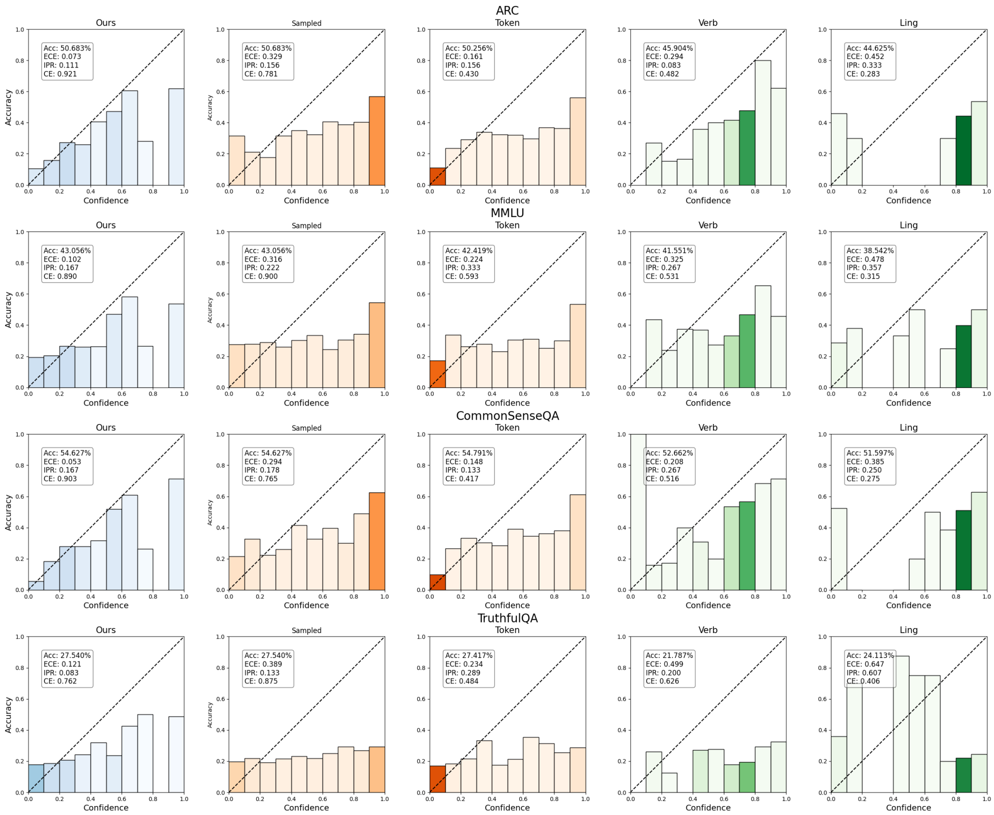

Figure 11: The experimental results are derived from `LLaMA2-7B-Chat` on 4 MCQA datasets.

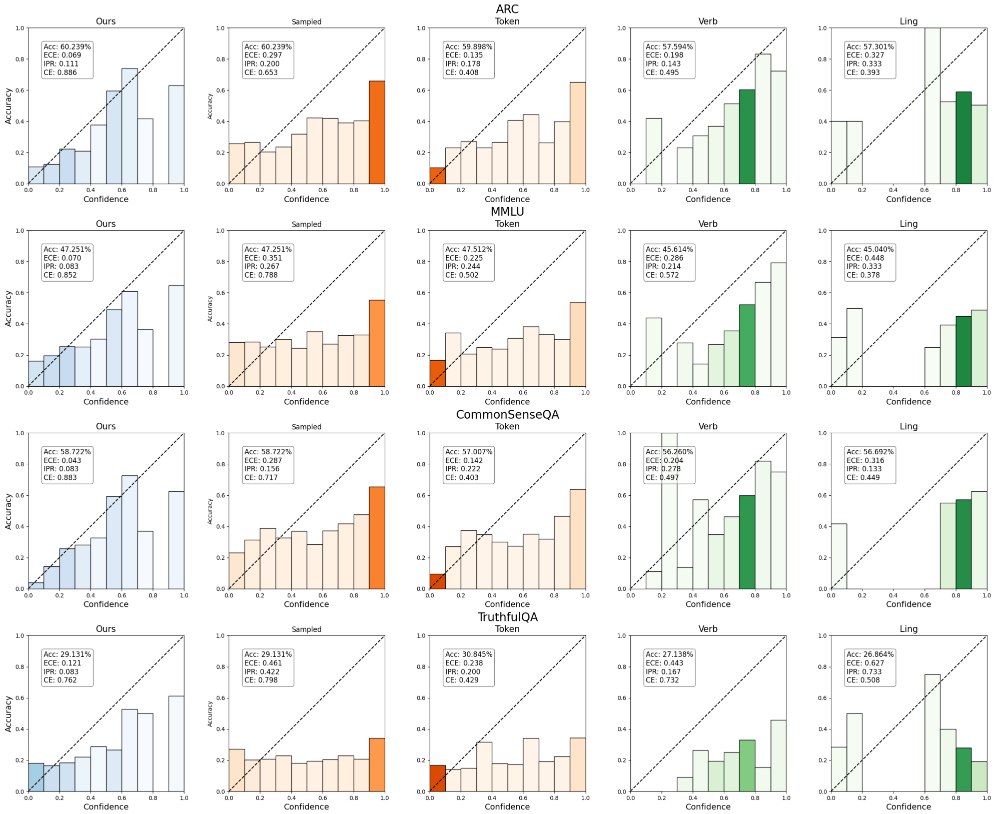

Figure 12: The experimental results are derived from `LLaMA2-13B-Chat` on 4 MCQA datasets.

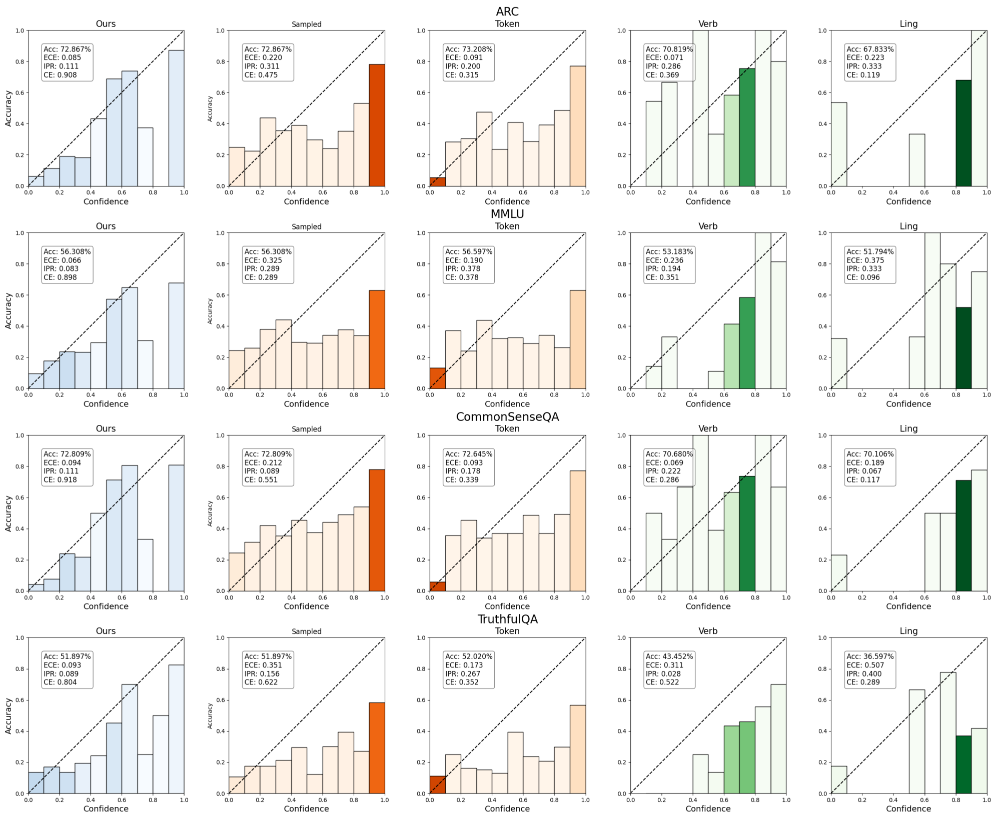

Figure 13: The experimental results are derived from `LLaMA2-70B-Chat` on 4 MCQA datasets.