Self-Ensemble: Mitigating Confidence Distortion for Large Language Models

Zicheng Xu[1,3*], Guanchu Wang[2*†], Guangyao Zheng[3], Yu-Neng Chuang[1], Alexander Szalay[3], Xia Hu[1], Vladimir Braverman[3†]

[1]Rice University, [2]University of North Carolina at Charlotte, [3]Johns Hopkins University

Abstract

Although Large Language Models (LLMs) perform well in general fields, they exhibit a ***confidence distortion*** problem on multi-choice question-answering (MCQA), particularly as the number of answer choices increases. Specifically, on MCQA with many choices, LLMs suffer from under-confidence in correct predictions and over-confidence in incorrect ones, leading to a substantially degraded performance. To solve this problem, we propose `Self-Ensemble` in this work. Our method splits the choices into several groups and ensembles LLM predictions across these groups to reach a final decision. The advantage of `Self-Ensemble` is its plug-and-play nature, where it can be integrated into existing LLM architecture based on a designed attention mask and positional encoding, without requiring labeled datasets for parameter tuning. Experimental results on three LLMs and datasets demonstrate that `Self-Ensemble` comprehensively addresses the confidence distortion problem of LLMs, outperforming standard inference as well as baseline methods. The source code is available at `https://github.com/ZichengXu/Self-Ensemble`.

1 Introduction

Large Language Models (LLMs) have exhibited remarkable performance in processing natural language information, such as the LLaMA (Touvron et al., 2023), Mistral (Jiang et al., 2024), and Deepseek (Guo et al., 2025). Among various NLP tasks, multi-choice question-answering (MCQA) stands out as a standard and challenging benchmark to evaluate the reliability and reasoning ability of LLMs (Wang et al., 2024c). It can significantly reduce the hallucinations by limiting the answer to a predefined set of choice aligned with human knowledge (Anjum et al., 2024; Neeley et al., 2025).

*Equal contribution, ordered by rolling dices.

†Correspondence to: Vladimir Braverman vova@cs.jhu.edu and Guanchu Wang gwang16@charlotte.edu

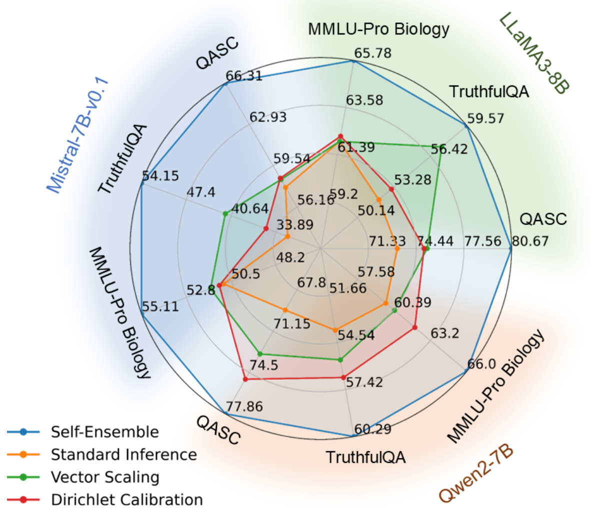
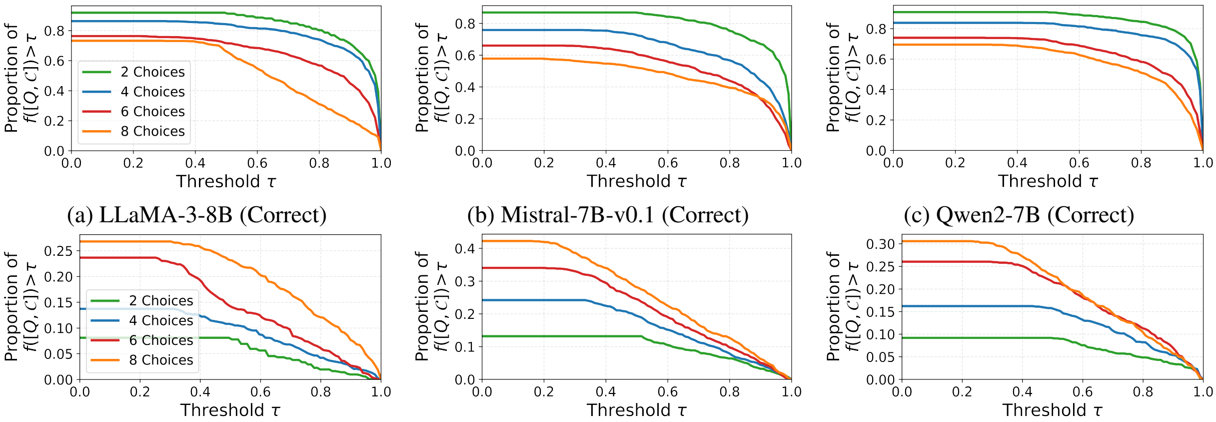


Figure 1: `Self-Ensemble`'s comprehensive performance on the QASC, TruthfulQA, and MMLU-Pro Biology datasets compared with baseline methods.

While advanced LLMs perform well on standard MCQA benchmarks, such as MMLU (Hendrycks et al., 2021), MathQA (Amini et al., 2019), and GSM-8K (Cobbe et al., 2021), they still face challenges when it comes to numerous choices that are closely related (Wang et al., 2024a). In particular, we identify a critical problem: LLMs suffer from ***confidence distortion*** on MCQA tasks based on a comprehensive benchmark analysis. Specifically, LLM's confidence in the correct choice tends to degrade, while its confidence in incorrect choices increases as the number of choices grows. This problem grows even more pronounced in the presence of additional noisy or partially relevant choices, thereby increasing the unreliable generations and erroneous predictions.

To calibrate LLM predictions on MCQA, existing work focuses on post-processing LLM predictions to better align confidence with correctness. One common method is the vector scaling (Guo et al., 2017). It adjusts the model outputs by applying a learned scaling vector and bias vector to the

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 16603–16615
November 4-9, 2025



(a) LLaMA-3-8B (Correct) (b) Mistral-7B-v0.1 (Correct) (c) Qwen2-7B (Correct)

(d) LLaMA-3-8B (Incorrect) (e) Mistral-7B-v0.1 (Incorrect) (f) Qwen2-7B (Incorrect)

Figure 2: Proportion of model prediction probability exceeding a threshold on the QASC dataset, for each model under both correct- and incorrect-answer conditions.

logits before applying the softmax function, which can smooth or sharpen the probabilities to better align with correctness. Another approach is Dirichlet Calibration (Zong et al., 2024). It generalizes LLM's prediction probability on each choice using a Dirichlet distribution to maximize the likelihood function on a validation set. However, a major limitation of these methods is their reliance on a labeled validation dataset to optimize calibration parameters. In practical scenarios, collecting high-quality validation data can be both labor-intensive and costly, which limits these techniques.

To overcome this limitation, we introduce `Self-Ensemble` for LLM calibration in this work. Unlike existing work, `Self-Ensemble` calibrates LLM predictions without relying on a labeled validation dataset for parameter tuning. Specifically, `Self-Ensemble` mitigates the confidence distortion problem through a divide-and-conquer process: it splits the choices into several groups of choice subsets; then ensembles LLM predictions on each group to achieve the final decision. To plug into existing LLM architecture, we design attention masking and positional re-encoding mechanisms that enable the divide-and-conquer process to be executed internally during inference. Figure 1 shows the comprehensive performance of `Self-Ensemble` on three datasets and LLMs, where our method shows significant improvement over standard inference and baseline methods by addressing the confidence distortion problem. To summarize, the contributions of this work are as follows:

- **Confidence distortion.** We identify the confidence distortion problem of LLMs in MCQA task, where LLMs have under-confident problems on correct answers and over-confident problems on incorrect answers.
- **Self-Ensemble.** We introduce `Self-Ensemble` to calibrate LLM confidence in MCQA task. We design attention masking and positional re-encoding to plug `Self-Ensemble` into existing LLM architectures.
- **Evaluation.** Experimental results demonstrate that `Self-Ensemble` can comprehensively mitigate the under-confident problems on correct answers and over-confident problems on incorrect answers. It can effectively improve accuracy on benchmark datasets by 8% on average.

2 Preliminary

2.1 Notations

We consider an LLM f on MCQA tasks in this work. We define an MCQA task with K answer choices as *K-choice QA*, where the set of options is denoted as $\mathcal{C} = \{op_1, op_2, \cdots, op_K\}$. In this work, we refer to the setting with $K \leq 4$ as ***few-choice QA***, and with $K \geq 8$ as ***many-choice QA***. Our goal is to solve the degradation of LLM's performance and confidence on the task of many-choice QA.

2.2 LLMs for MCQA

The two most effective ways for LLMs doing MCQA are *verbal generation* and *token probability*. In the verbal generation approach, the LLM generates the final answer as text outputs, corresponding to one of the choices. However, a key drawback is that it may fail to follow the expected output format, leading to answer extraction errors. In contrast, the token probability approach directly computes the probability of each choice token at the model's final layer, selecting the answer with the highest probability. It offers a more reliable evaluation result and often reflects a model's best

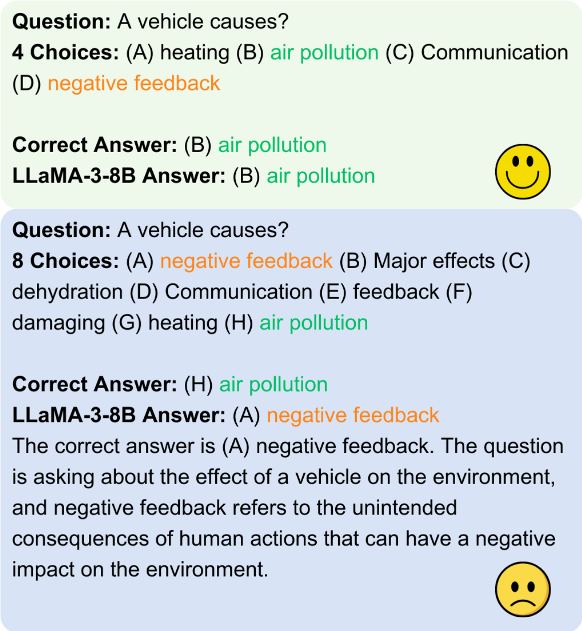


Figure 3: LLMs ignore the correct choice and pick the incorrect one in the many-choice setting.

Model	2-choice	4-choice	6-choice	8-choice
LLaMA-3-8B	91.90	86.29	76.35	73.22
Mistral-7B-v0.1	86.83	75.81	65.98	57.78
Qwen2-7B	90.82	83.80	73.97	69.44
Average	89.85	81.97	72.10	66.81

Table 1: Accuracy of LLMs on the QASC dataset with different choice numbers.

achievable performance. In this work, we consider the token probability approach for MCQA.

3 Confidence Distortion and `Self-Ensemble`

We demonstrate the ***confidence distortion*** problem for LLMs and introduce our `Self-Ensemble` method to overcome this problem.

3.1 Confidence Distortion on Many-choice QA

We give Figure 2 to demonstrate the confidence distortion problems of LLMs on many-choice QA, where the experiment is on the QASC datasets (Khot et al., 2020). Specifically, Figure 2 shows the proportion of $f([Q,\mathcal{C}])>\tau$ versus τ for $0<\tau<1$, where $f([Q,\mathcal{C}])>\tau$ represents LLM's predicted probability on the correct or incorrect choice exceeds a threshold τ. A high value of $f([Q,\mathcal{C}])>\tau$ proportion across $0\leq\tau\leq 1$ indicates strong model confidence. To show the confidence distortion problem, we compare the $f([Q,\mathcal{C}])>\tau$ proportion across different choice settings with $K=2,4,6,8$ in Figure 2. On the correctly answered questions, the LLM achieves the highest $f([Q,\mathcal{C}])>\tau$ proportion when $K=2$, showing that the model becomes under-confident in many-choice scenarios. In contrast, for the incorrectly answered questions, the LLM shows the highest $f([Q,\mathcal{C}])>\tau$ proportion when $K=8$, indicating an over-confidence issue in many-choice scenarios.

The confidence distortion problem causes a degradation of LLM performance on many-choice QA. As shown in Figure 3, for the LLaMA-3-8B model, it can originally solve the 4-choice QA. However, when the same question is extended to 8 choices, while keeping the correct answer unchanged, the model selects an incorrect option. This failure is due to a decrease in confidence in the correct answer and an increase in confidence in incorrect options. We show a comprehensive result in Table 1, where each column shares the same question context but varies in the number of answer choices. A many-choice question contains all the options from the corresponding few-choice versions. For example, each question in the 8-choice task includes all the choices from the 6-choice task. As shown in Table 1, LLMs generally perform worse on many-choice QA than few-choice QA, This indicates the confidence distortion problem for LLM on many-choice QA.

3.2 LLM Inference with `Self-Ensemble`

We propose `Self-Ensemble` to solve the confidence distortion problem on many-choice QA. The intuition of `Self-Ensemble` is to divide a K-choice (many-choice) QA into multiple m-choice (few-choice) QA, where $m \ll K$. Then, it collects LLM's answer probability in each few-choice case, and estimates the expected probability of each choice for the final decision. Specifically, `Self-Ensemble` has three steps as follows:

Step 1. Given the K choices $\mathcal{C}=\{op_1,\cdots,op_K\}$, `Self-Ensemble` randomly splits them into $\lceil\frac{K}{m}\rceil$ groups with seed s, which is given by $\mathcal{G}_s(\mathcal{C}) = \{G_1,\cdots,G_{\lceil\frac{K}{m}\rceil}\}$. Each group G_i has m choices, while the last group may have less than m choices if $K \bmod m \neq 0$. Different groups take random subsets of $\mathcal{C}$ without overlaps, i.e. $G_i \cap G_j = \varnothing$ for $i \neq j$, and $G_1 \cup \cdots \cup G_{\lceil\frac{K}{m}\rceil} = \mathcal{C}$.

Step 2. For each group $G_j = \{\tilde{op}_1,\cdots,\tilde{op}_{\tilde{m}}\}$, `Self-Ensemble` collects the probability of each choice within this group from LLMs as follows:

$$Pr(\tilde{op}_1),\cdots,Pr(\tilde{op}_{\tilde{m}}) = f([Q,G_j]) \quad (1)$$

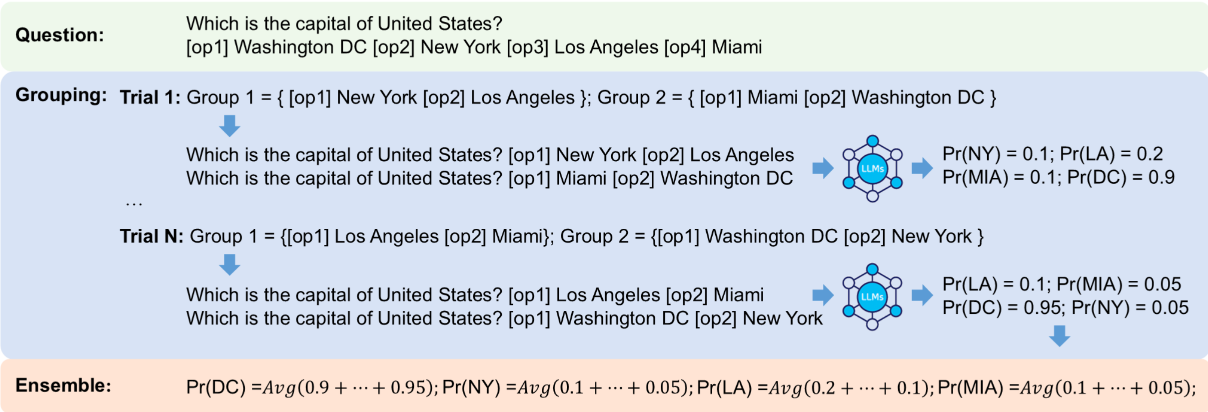


Figure 4: Example of `Self-Ensemble` process on 4-choice QA.

Step 3. To estimate the final probability of each choice op_i, `Self-Ensemble` averages all the probabilities of choice op_i as follows:

$$Pr(op_i) = \mathbb{E}_{\substack{s\sim U(\mathbb{N}), G_j \sim \mathcal{G}_s(\mathcal{C}),\\ \widetilde{op}\sim G_j}}[\mathbb{I}(op_i = \widetilde{op}) Pr(\widetilde{op})] \quad (2)$$

where $s \sim U(\mathbb{N}), G_j \sim \mathcal{G}_s(\mathcal{C})$ represents randomly splitting the choice set using different integer random seed s; and $\mathbb{I}(op_i = \widetilde{op}) = 1$ if $op_i = \widetilde{op}$; otherwise it takes 0. The intuition of Equation (2) is to estimate the expected probability of each choice by averaging over different group partitions.

An example of `Self-Ensemble` is given in Figure 4. Specifically, in each trial, `Self-Ensemble` simplifies the given 4-choice QA into 2-choice QAs by randomly grouping the choices, and achieves the choice probability within each group from LLMs. After N trials, it estimates the expected probability of each choice by averaging different trials.

4 Plugging `Self-Ensemble` into LLMs

We propose integrating `Self-Ensemble` into existing LLM architectures to enable intrinsic self-ensemble processing during inference. As illustrated in Equations (1) and (2), `Self-Ensemble` requires multiple forward passes through the LLM, which are $f([Q, G_1])$, $f([Q, G_2])$, $f([Q, G_3]), \cdots$. For efficient self-ensemble processing, we design specialized attention masks and positional encoding, such that these multiple calls of LLM can be executed in a single forward pass over the concatenated sequence $f([Q, G_1, G_2, G_3, \cdots])$. Here, each group $G_j \sim \mathcal{G}_s(\mathcal{C}), s \sim \mathbb{N}$ is sampled from the choice set following Section 3.2 Step 1; and $[Q, G_1, G_2, G_3, \cdots]$ denotes combining the question and different groups of choice into a sequence of tokens as prompts for LLMs.

4.1 Attention Masking

The standard auto-regressive attention mask follows $\mathbb{I}(j \leq i)$, ensuring that each token attends to all preceding tokens in the sequence. However, following `Self-Ensemble`, the LLM should restrict its attention to a single group of choices during each trial, ignoring the influence of other groups. Therefore, for the concatenated input sequence with the question and all choice groups $[Q, G_1, G_2, G_3, \cdots]$, ***when processing a certain choice group G_i, the LLM should not have attention to other groups G_j, $j \neq i$.***

To implement this constraint, `Self-Ensemble` designs a custom attention mask to block the cross-group attention. As shown in the left-hand side of Figure 5, a token within group G_1 (highlighted in red) attends only to the question and other tokens within G_1. This ensures that the model focuses solely on the current group when generating the answer. In general, let $\mathtt{Tk}_i$ denote a certain token at position i. Its attention mask to a previous token $\mathtt{Tk}_j$, $j \leq i$ is given by

$$M_{i,j} = \begin{cases} 1, & \text{if } \mathtt{Tk}_j \in \mathtt{Tk}(Q), \\ \prod_{n=1}^{N} \mathbb{I}(\mathtt{Tk}_i, \mathtt{Tk}_j \in \mathtt{Tk}(G_n)) & \text{otherwise}, \end{cases} \quad (3)$$

where $\mathtt{Tk}(Q)$ denotes the tokens of question Q; $\mathbb{I}(\mathtt{Tk}_i, \mathtt{Tk}_j \in \mathtt{Tk}(G_n)) = 1$ if $\mathtt{Tk}_i$ and $\mathtt{Tk}_j$ belong to the same group, and 0 otherwise; and $M_{i,j} = 0$ for $j > i$ as standard attention masks. This attention mask enforces pairwise independence between choice groups during inference.

4.2 Positional Re-encoding

The standard auto-regressive positional encoding follows $\mathtt{Pos}_i = i$, where $\mathtt{Pos}_i$ denotes the positional index of a token at position i. Following

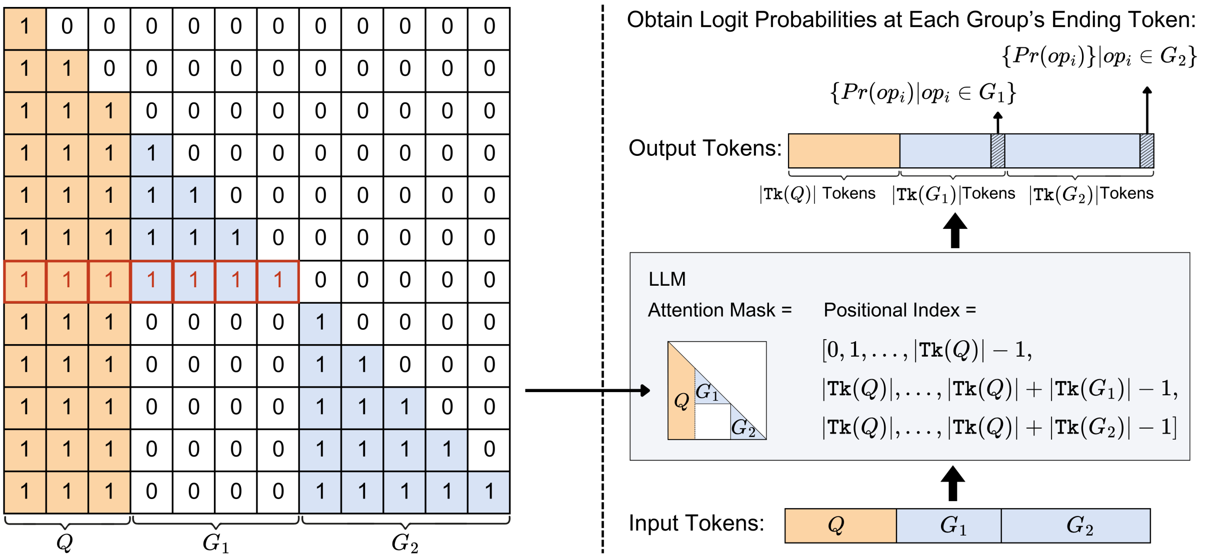


Figure 5: Plug-in `Self-Ensemble`: by incorporating the attention mask and positional re-encoding, LLMs can achieve the ensembled results in a single forward pass.

`Self-Ensemble`, the LLM should process each question-group pair $[Q, G_j]$ as a continuous input sequence for $1 \leq j \leq N$. However, this continuity is disrupted in the concatenated input sequence $[Q, G_1, G_2, G_3, \cdots]$, because starting from G_2, the question Q and a choice group are no longer physically adjacent. For example, G_1 lies between Q and G_2, breaking the positional continuity between Q and G_2. To preserve the relative positional relationships between the question and each choice group, ***each choice group should be encoded into the relative contextual position to the question***. Specifically, the positional indexes of question tokens and choice tokens are given as follows:

Question tokens. A question token $\mathtt{Tk}_i \in \mathtt{Tk}(Q)$ should have absolute position $\mathtt{Pos}_i = i$.

Choice tokens. A choice token $\mathtt{Tk}_i$, if it is with choice group G_n, its absolute position should be replaced into a relative position as follows:

$$\mathtt{Pos}_i = i - \sum_{j=1}^{n} |\mathtt{Tk}(G_j)|, \tag{4}$$

where $|\mathtt{Tk}(G_j)|$ indicates the token number of a choice group G_j; and n takes maximal value that satisfies $i \geq \sum_{j=1}^{n} |G_j|$.

Following Equation (4) to replace the position index of choice tokens, the position index of input sequence $[Q, G_1, \cdots, G_n]$ is given by

$$\Big[\overbrace{0, 1, \cdots, |\mathtt{Tk}(Q)|-1}^{\text{Position index of Q}}, \tag{5}$$
$$\overbrace{|\mathtt{Tk}(Q)|, \cdots, |\mathtt{Tk}(Q)|+|\mathtt{Tk}(G_1)|-1}^{\text{Position index of } G_1}, \cdots,$$
$$\overbrace{|\mathtt{Tk}(Q)|, \cdots, |\mathtt{Tk}(Q)|+|\mathtt{Tk}(G_n)|-1}^{\text{Position index of } G_n}\Big]$$

By combining the attention mask with positional re-encoding, we ensure that processing the concatenated sequence $[Q, G_1, G_2, G_3, \cdots]$ is functionally equivalent to independently processing each question-group pair $[Q, G_1], [Q, G_2], [Q, G_3], \cdots$. This enables `Self-Ensemble` to seamlessly integrate into existing LLM architectures.

4.3 Choice Probability

By using the attention mask and positional re-encoding in Equations (3) and (5), respectively, the LLM can simultaneously process all groups of choices in a single forward pass. As shown in Figure 5, within the output sequence, the ending token of each group captures the choice probabilities within each choice group. Formally, let $\boldsymbol{a} = f([Q, G_1, G_2, G_3, \cdots])$ denote the sequence of output logits. For a certain choice $\widetilde{op}_i \in G_j$, the probability of $\widetilde{op}_i$ is estimated by

$$Pr(\widetilde{op}_i) = \boldsymbol{a}[\mathtt{index}_j, \mathtt{index}_i], \tag{6}$$

where $\mathtt{index}_j = |Q| + \sum_{k=1}^{j-1} |\mathtt{Tk}(k)| - 1$ takes the ending token position of group G_j; $\mathtt{index}_i$ takes the token ID of $\widetilde{op}_i$ defined by the LLM tokenizer. The final probability of each choice takes the expected probability $Pr(op_i) = \mathbb{E}[\mathbb{I}(op_i = \tilde{op}_i)Pr(\tilde{op}_i)]$.

The advantage of `Self-Ensemble` lies in its seamless integration with existing LLM architectures. It enables the LLM to achieve the expected probability of each choice in a single forward pass, avoiding separate inference over individual choice groups. This enables `Self-Ensemble` to behave both accurately and efficiently on MCQA.

5 Experiment

In this experiment, we conduct experiments to evaluate `Self-Ensemble`, aiming to answer the following research questions: **RQ1:** Can `Self-Ensemble` improve on LLM's accuracy on many-choice QA? **RQ2:** Can `Self-Ensemble` mitigate the confidence distortion problems for LLMs? **RQ3:** How does `Self-Ensemble` reshape the scaling of model accuracy with respect to parameter size?

5.1 Experimental Setup

We specify the datasets, LLMs, evaluation metrics, and implementation details.

Models. We evaluate `Self-Ensemble` using three popular model families: LLaMA-3-8B (Touvron et al., 2023), Mistral-7B-v0.1 (Jiang et al., 2024), and Qwen-2-7B (Yang et al., 2024). We download these models from the Huggingface Transformers library (Wolf et al., 2019).

Dataset. The evaluation of `Self-Ensemble` is based on the QASC (Khot et al., 2020), TruthfulQA (Lin et al., 2021), and MMLU-Pro (Wang et al., 2024d) datasets. **QASC:** It is a multi-hop reasoning dataset comprised of 8-choice QA. We evaluate LLM performance on its validation set with 927 questions under a closed-book setting. **TruthfulQA:** It has 817 multiple-choice and short-answer questions across 38 categories, such as health, law, and finance. Following many-choice setting, we filter out questions with fewer than five incorrect choices. As a result, 277 questions have at least one correct choice and five incorrect choices. **MMLU-Pro:** Enhancement of the MMLU dataset by selecting questions where state-of-the-art LLMs consistently fail. We use the Biology subset for our experiments, which has 450 10-choice questions.

Baseline Methods. **Standard inference:** We perform standard inference by comparing the probability of each choice token and selecting the option with the highest likelihood. **Vector scaling:** Vector scaling (Guo et al., 2017) applies a learnable scale vector w and bias b to the logits before softmax, enabling finer-grained calibration. The parameters are optimized to maximize accuracy on a validation set. **Dirichlet calibration:** Dirichlet calibration (Zong et al., 2024) is a lightweight multiclass calibration method that adjusts the model's predicted probability to better match observed frequencies on a validation set.

Implementation Details. For QASC's 8-choice QA, `Self-Ensemble` splits each question into a 4-choice QA for 20 trials. In addition, for the TruthfulQA dataset, it splits the 6-choice QAs into 3-choice QAs for 6 trials. Moreover, for the MMLU-Pro Biology dataset, it splits the 10-choice QAs into 5-choice QAs for 40 trials. `Self-Ensemble` does not reply on a validation set for optimizing any parameter or special setting.

5.2 Accuracy of MCQA (RQ1)

Table 2 shows the accuracy (%) of `Self-Ensemble`. These results are compared with baseline methods and standard inference of LLMs.

Accuracy Improvement. As shown in Table 2, `Self-Ensemble` consistently outperforms LLM standard inference and baseline methods, demonstrating its potential in enhancing LLM's performance on MCQA. Moreover, compared with baseline methods, `Self-Ensemble` does not require a validation set for maximizing performance.

Model Agnosticism. By integrating with different families of LLMs, `Self-Ensemble` shows consistent performance and improvement, as shown in Table 2. This generality indicates that `Self-Ensemble` can potentially serve as a versatile enhancement for a range of LLMs in practice.

Stable Improvement. For datasets of different difficulty levels, standard inference shows less accuracy on TruthfulQA and MMLU-Pro than QASC. This implies that the confidence distortion is general problem on both easy and hard MCQA tasks. In contrast, `Self-Ensemble` consistently improves performance across all difficulty levels, demonstrating its general effectiveness in mitigating the confidence distortion problem.

5.3 Confidence Calibration by `Self-Ensemble` (RQ2)

In this section, we show that `Self-Ensemble` mitigates the confidence distortion problem of LLMs in

Model	Method	QASC Accuracy	QASC Improve	TruthfulQA Accuracy	TruthfulQA Improve	MMLU-Pro Biology Accuracy	MMLU-Pro Biology Improve	Average Accuracy	Average Improve
LLaMA-3-8B	Standard inference	73.22	-	51.99	-	62.00	-	62.40	-
	Vector Scaling	75.16	+1.94	57.40	+5.41	62.00	+0.00	64.85	+2.45
	Dirichlet Calibration	74.96	+1.74	53.07	+1.06	62.24	+0.24	63.42	+1.02
	`Self-Ensemble`	**80.67**	**+7.45**	**59.57**	**+7.58**	**65.78**	**+3.78**	**68.67**	**+6.27**
Mistral-7B-v0.1	Standard inference	57.78	-	32.13	-	50.89	-	46.93	-
	Vector Scaling	58.42	+0.64	41.52	+9.39	51.56	+0.67	50.50	+3.57
	Dirichlet Calibration	58.53	+0.75	35.38	+3.25	51.11	+0.22	48.34	+1.41
	`Self-Ensemble`	**66.31**	**+8.53**	**54.15**	**+22.02**	**55.11**	**+4.22**	**58.52**	**+11.59**
Qwen-2-7B	Standard inference	69.44	-	53.79	-	59.78	-	61.00	-
	Vector Scaling	73.00	+3.56	55.60	+1.81	60.44	+0.66	63.01	+2.01
	Dirichlet Calibration	75.05	+5.61	56.68	+2.89	62.00	+2.22	64.58	+3.58
	`Self-Ensemble`	**77.86**	**+8.42**	**60.29**	**+6.50**	**66.00**	**+6.22**	**68.05**	**+7.05**

Table 2: Accuracy of `Self-Ensemble` on the QASC, TruthfulQA, MMLU-Pro Biology datasets.

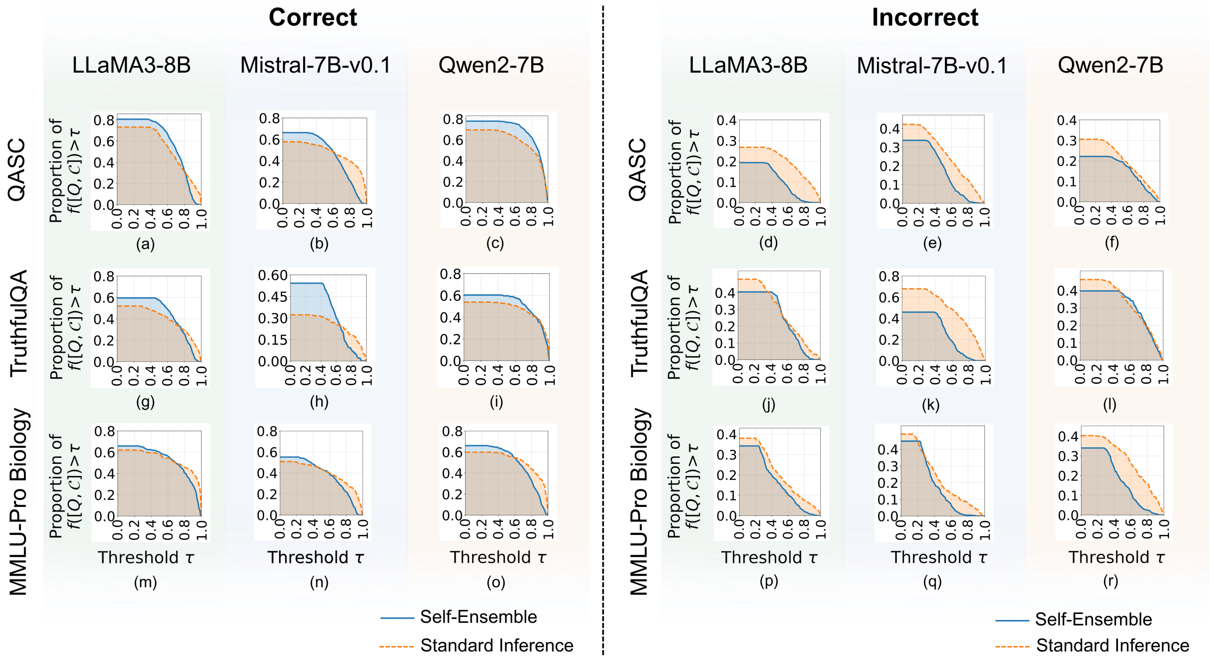


Figure 6: Probability of model confidence exceeding a threshold on the QASC, TruthfulQA, and MMLU-Pro-Biology dataset, for each model under both correct- and incorrect-answer conditions.

Figure 6. Specifically, Figure 6 shows the proportion of $f([Q,\mathcal{C}])>\tau$ versus τ for $0<\tau<1$ on three different LLMs and datasets, where f takes different LLMs; and $f([Q,\mathcal{C}])>\tau$ represents LLM's predicted probability exceeds a threshold τ. A higher value of $f([Q,\mathcal{C}])>\tau$ proportion across $0<\tau<1$ indicates strong model confidence.

In Figure 6's sub-figures (a)-(c), (g)-(i), (m)-(o), `Self-Ensemble` has a higher confidence level on its correct answers; while in sub-figures (d)-(f), (j)-(l), (p)-(r), `Self-Ensemble` has a lower confidence level on its incorrect answers. This indicates `Self-Ensemble` can effective mitigate the under-confident problems on correct answers and over-confident problems on incorrect answers, improving the reliability of LLMs on MCQA in practice.

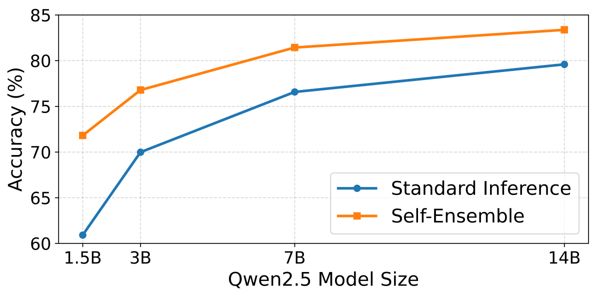


Figure 7: Qwen2.5-1.5B, 3B, 7B, and 14B with standard inference and `Self-Ensemble` on the QASC dataset.

5.4 Improving Accuracy-to-Parameter Scaling (RQ3)

`Self-Ensemble` achieves improved scaling of accuracy with parameter size. As shown in Figure 7, `Self-Ensemble` consistently yields higher accu-

Model	Accuracy
LLaMA-3-8B	73.22
DeepSeek-R1-Distill-Llama-8B	73.33
LLaMA-3-8B with CoT	74.62
LLaMA-3-8B with `Self-Ensemble`	**80.67**
Qwen2-7B	69.44
DeepSeek-R1-Distill-Qwen-7B	70.63
Qwen2-7B with CoT	74.19
Qwen2-7B with `Self-Ensemble`	**77.86**

Table 3: General LLMs with `Self-Ensemble` outperform reasoning LLMs and chain of thought prompting on the QASC dataset.

Model	4-bit	Full Precision	4-bit + `Self-Ensemble`
LLaMA-3-8B	72.57	73.22	**79.16**
Mistral-7B-v0.1	57.45	57.78	**65.23**
Qwen2-7B	67.71	69.44	**78.83**

Table 4: `Self-Ensemble` helps quantized LLMs outperform full-precision LLMs on the QASC dataset.

Model	w/o Attn Mask	w/o Pos Re-enc	`Self-Ensemble`
LLaMA-3-8B	19.22	67.39	**80.67**
Mistral-7B-v0.1	18.57	53.24	**66.31**
Qwen2-7B	15.44	71.27	**77.86**

Table 5: Comparison of `Self-Ensemble` with that w/o attention masks or positional re-encoding.

racy on the QASC dataset as the model size increases from 1.5B to 14B. It enables better scalability across model sizes and can be integrated with any LLM without requiring additional data or retraining. Notably, Qwen2.5-1.5B, 3B, 7B with `Self-Ensemble` outperform larger models using standard inference. This demonstrates the potential of `Self-Ensemble` to advance the deployment of small language models in real-world scenarios with constrained memory resources.

5.5 General-purpose LLMs with `Self-Ensemble` Outperforms Reasoning LLMs and Chain of Thought Prompting

We evaluate whether `Self-Ensemble` can match or surpass specialized reasoning methods. On the QASC dataset, general-purpose LLMs combined with `Self-Ensemble` outperform their reasoning-tuned counterparts, indicating that `Self-Ensemble` unlocks accuracy gains without dedicated reasoning fine-tuning. Furthermore, `Self-Ensemble` also surpasses chain-of-thought (CoT) prompting (Wei et al., 2022), which yields higher accuracy than CoT while avoiding the extra latency and token-generation overhead of producing reasoning traces. Both results are summarized in Table 3, highlighting that `Self-Ensemble` offers an efficient, single-pass, and fine-tuning-free alternative to reasoning LLMs and CoT prompting on MCQA.

5.6 Application to Quantized LLMs

To further explore `Self-Ensemble`'s application to memory-constrained settings, we show its effectiveness on quantized LLMs, including LLaMA-3-8B, Mistral-7B-v0.1, and Qwen2-7B under 4-bit bitsandbytes (BNB) quantization. As shown in Table 4, `Self-Ensemble` effectively offsets accuracy loss caused by the aggressive compression, enabling each quantized model to outperform its full-precision version. This indicates `Self-Ensemble`'s potential for quantized LLMs in resource-constrained environments, without requiring additional training or external information.

5.7 Ablation Study

We demonstrate the individual contribution of attention masks and positional re-encoding to `Self-Ensemble`. Specifically, we conduct experiments comparing `Self-Ensemble` with its ablated versions: without attention masks or positional re-encoding. Experimental results for the LLaMA-3-8B, Mistral-7B-v0.1, and Qwen2-7B on the QASC dataset are given in Table 5. It is observed that each model has a significant loss of accuracy without attention masks or positional re-encoding, indicating that both attention structure and position encoding contributes to the self-ensemble process. `Self-Ensemble` has a special attention mask designed in Sections 4.2 and positional re-encoding in Section 4.1 to enable intrinsic LLM inference over different choice groups and produce ensemble results efficiently. This can effectively calibrate LLM confidence and enhance the accuracy on many-choice problems.

5.8 Sensitivity to Group Size and Number of Trials

We study how `Self-Ensemble` depends on the group size m and the number of ensembles N. Using Qwen2.5-7B on the QASC dataset, we sweep $m \in \{3,4,5\}$ and $N \in \{10,20,40,80\}$. As shown in Table 6, accuracy is relatively insensitive to the group size, while increasing the number of trials improves performance and stabilizes at $N \geq 40$, indicating robustness to hyperparameters.

Trials N	$m=3$	$m=4$	$m=5$
10	78.73	78.08	78.61
20	79.59	80.67	79.91
40	79.27	81.97	81.43
80	80.45	82.40	81.97

Table 6: Accuracy comparison of `Self-Ensemble` across group sizes and trial numbers.

Consequently, `Self-Ensemble` can be used without hyperparameter tuning, reducing setup time while preserving performance.

6 Related Work

Confidence Calibration. Calibration aligns predicted probabilities with true correctness likelihoods to ensure reliable confidence estimates. Traditional methods like vector scaling (Guo et al., 2017), isotonic regression (Zadrozny and Elkan, 2002), and Dirichlet calibration (Kull et al., 2019) have been adapted to LLMs but often assume fixed prediction heads and struggle with distribution shifts or multi-choice uncertainty. Recent generative-specific techniques, such as logit perturbation (Jiang et al., 2021), prompt-based reweighting (Zhou et al., 2022b), Bayesian post-hoc methods (Wang et al., 2023), chain-of-though reasoning (Seßler et al., 2024) partially address these limitations. Moreover, pioneer work (Zhong et al., 2025) proposes a unified framework of calibrating LLMs under weight quantization, with theoretical foundations and practical design. However, consistent calibration across diverse LLM architectures and tasks remains challenging, highlighting the need for effective, model-agnostic solutions.

LLM Ensemble. Ensembling has long been a powerful strategy to improve both model accuracy and robustness. In LLMs, methods such as voting schemes (Zhou et al., 2022a), stochastic decoding (Ho and Weller, 2022), and diverse prompting (Wang et al., 2022) aggregate outputs from multiple model instances or inference runs. Notably, similar to our intuition of self-ensemble, a concurrent work (Neeley et al., 2025) also proposes a "divide-and-conquer" strategy to improve LLM performance on healthcare tasks. It has shown promising results in diagnosing rare diseases caused by genetic variants, a task that remains highly challenging. These methods are especially useful for reducing variance in generative outputs and mitigating individual model biases. Additionally, ensemble approaches support calibration via confidence averaging and self-consistency mechanisms (Wang et al., 2022). However, standard ensembles are computationally costly and less generalizable across model families and tasks, motivating for a more efficient and scalable ensemble method.

7 Conclusion

In this work, we demonstrate the confidence distortion problem of LLMs generally on MCQA, particularly in the many-choice setting. To solve this problem, we propose `Self-Ensemble` by decomposing a many-choice problem into several few-choice problems, and aggregating the intermediate results into the final solution. We further integrate `Self-Ensemble` with existing LLM architecture, enabling an intrinsic self-ensemble process for LLM inference. Experimental results across various LLMs and datasets show that `Self-Ensemble` can effectively overcome this problem, improving confidence in correct answers and reducing confidence in incorrect answers. This enables `Self-Ensemble` to significantly improve the accuracy of LLMs in MCQA by 8% on average.

8 Limitations and Potential Risks

In this work, we propose a `Self-Ensemble` to calibrate the LLM decisions on many-choice QA. The application of `Self-Ensemble` is limited to the muti-choice question-answer (MCQA) problems. While MCQA stands out as a standard and challenging benchmark to evaluate the ability of LLMs, there are open-ended QA tasks in real-world scenarios. Calibrating LLMs' decisions on open-ended QA remains our future research. Furthermore, `Self-Ensemble` is methodologically orthogonal to existing calibration techniques such as Vector Scaling and Dirichlet Calibration, enabling synergistic integration to further enhance performance.

Acknowledgments

This research was supported by NSF CNS 2528780. We would like to thank The Center for Research Computing at Rice University for providing hardware and technical assistance for experiments. We extend special thanks to Xinyu Wang from the University of Kansas for help with the paper's visualizations. The views and conclusions in this paper are those of the authors and do not represent the views of any funding or supporting agencies.

References

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, Minneapolis, Minnesota. Association for Computational Linguistics.

Sumera Anjum, Hanzhi Zhang, Wenjun Zhou, Eun Jin Paek, Xiaopeng Zhao, and Yunhe Feng. 2024. Halo: Hallucination analysis and learning optimization to empower llms with retrieval-augmented context for guided clinical decision making. *arXiv preprint arXiv:2409.10011*.

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and 1 others. 2024. Main-rag: Multi-agent filtering retrieval-augmented generation. *arXiv preprint arXiv:2501.00332*.

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

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*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Daniel Ho and Adrian Weller. 2022. Large language models are few-shot learners: Generating calibration data with language models. In *NeurIPS Workshop on Trustworthy and Socially Responsible Machine Learning*.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Heinrich Jiang, Been Kim, Melody Guan, and Maya Gupta. 2021. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *International Conference on Machine Learning*, pages 4684–4694. PMLR.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Telmo Silva Filho, Hao Song, and Peter Flach. 2019. Beyond temperature scaling: Obtaining well-calibrated multi-class probabilities with dirichlet calibration. In *Advances in Neural Information Processing Systems*, volume 32.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *Preprint*, arXiv:2109.07958.

Zirui Liu, Guanchu Wang, Shaochen Henry Zhong, Zhaozhuo Xu, Daochen Zha, Ruixiang Ryan Tang, Zhimeng Stephen Jiang, Kaixiong Zhou, Vipin Chaudhary, Shuai Xu, and 1 others. 2023. Winner-take-all column row sampling for memory efficient adaptation of language model. *Advances in Neural Information Processing Systems*, 36:3402–3424.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024. Kivi: A tuning-free asymmetric 2bit quantization for kv cache. *arXiv preprint arXiv:2402.02750*.

Matthew Neeley, Guantong Qi, Guanchu Wang, Ruixiang Tang, Dongxue Mao, Chaozhong Liu, Sasidhar Pasupuleti, Bo Yuan, Fan Xia, Pengfei Liu, and 1 others. 2025. Survey and improvement strategies for gene prioritization with large language models. *arXiv preprint arXiv:2501.18794*.

Kathrin Seßler, Yao Rong, Emek Gözlüklü, and Enkelejda Kasneci. 2024. Benchmarking large language models for math reasoning tasks. *arXiv preprint arXiv:2408.10839*.

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*.

Chonghua Wang, Haodong Duan, Songyang Zhang, Dahua Lin, and Kai Chen. 2024a. Ada-leval: Evaluating long-context llms with length-adaptable benchmarks. *Preprint*, arXiv:2404.06480.

Guanchu Wang, Yu-Neng Chuang, Ruixiang Tang, Shaochen Zhong, Jiayi Yuan, Hongye Jin, Zirui Liu, Vipin Chaudhary, Shuai Xu, James Caverlee, and 1 others. 2024b. Taylor unswift: Secured weight release for large language models via taylor expansion. *arXiv preprint arXiv:2410.05331*.

Guanchu Wang, Junhao Ran, Ruixiang Tang, Chia-Yuan Chang, Yu-Neng Chuang, Zirui Liu, Vladimir Braverman, Zhandong Liu, and Xia Hu. 2024c. Assessing and enhancing large language models in rare disease question-answering. *arXiv preprint arXiv:2408.08422*.

Xuezhi Wang, Jason Wei, Nathanael Schärli, Saurav Xie, Denny Zhou, Ed H Chi, Quoc V Le, and 1 others. 2022. Self-consistency improves chain of thought reasoning in language models. In *arXiv preprint arXiv:2203.11171*.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, and 1 others. 2024d. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zixuan Wang, Xiaocheng Li, and Yang Liu. 2023. Posterior prompt tuning: Toward faithful and calibrated llms. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 40 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Miao Yu, Fanci Meng, Xinyun Zhou, Shilong Wang, Junyuan Mao, Linsey Pan, Tianlong Chen, Kun Wang, Xinfeng Li, Yongfeng Zhang, and 1 others. 2025. A survey on trustworthy llm agents: Threats and countermeasures. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 6216–6226.

Jiayi Yuan, Hongyi Liu, Shaochen Zhong, Yu-Neng Chuang, Songchen Li, Guanchu Wang, Duy Le, Hongye Jin, Vipin Chaudhary, Zhaozhuo Xu, and 1 others. 2024. Kv cache compression, but what must we give in return? a comprehensive benchmark of long context capable approaches. *arXiv preprint arXiv:2407.01527*.

Bianca Zadrozny and Charles Elkan. 2002. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 694–699.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Mingyu Zhong, Guanchu Wang, Yu-Neng Chuang, and Na Zou. 2025. Quantized can still be calibrated: A unified framework to calibration in quantized large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 30503–30517.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Daniel Bar, Xuezhi Wang, Swaroop Mishra, Andy Nguyen, Jacob Austin, Adams Wei Yu, and 1 others. 2022a. Least-to-most prompting enables complex reasoning in large language models. In *International Conference on Machine Learning*. PMLR.

Wenda Zhou, Vishwas Shah, and Alexander Rush. 2022b. Calibrating sequence-to-sequence models with post-hoc bayesian inference. In *Advances in Neural Information Processing Systems*, volume 35, pages 18136–18147.

Chen-Chen Zong, Ye-Wen Wang, Ming-Kun Xie, and Sheng-Jun Huang. 2024. Dirichlet-based prediction calibration for learning with noisy labels. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17254–17262.

Appendix

A Efficiency Comparison of Multi-Pass and Single-Pass Self-Ensemble

We follow existing work (Wang et al., 2024b; Liu et al., 2023; Yuan et al., 2024) to benchmark the efficiency of multi-pass Self-Ensemble in Section 3.2 and the single-pass Self-Ensemble in Section 4 using LLaMA-3-8B on the QASC dataset. As shown in Table 7, single-pass Self-Ensemble successfully accelerates inference time without sacrificing accuracy while maintaining almost the same memory cost. This demonstrates the effectiveness of our Section 4 in accelerating the inference through parallelization.

B Confidence Distortion Across Model Scales and Families

We assess generality beyond 7B–8B models by evaluating LLaMA-3.2-1B/3B, LLaMA-3-8B, and Qwen2.5-1.5B/3B/7B/14B on QASC with $K \in \{2, 4, 6, 8\}$. As shown in Table 8, accuracy consistently declines as K increases for every model, indicating that the confidence distortion phenomenon holds across architectures and scales.

C Extended Confidence Distortion Evaluation

To further illustrate Self-Ensemble mitigates confidence distortion, we provide additional evaluations based on the under-confidence and over-confidence ratios. Here, under-confidence is defined as the proportion of correct answers where the model assigns probability below a threshold τ, while over-confidence is the proportion of incorrect answers where the model assigns probability above τ. For example, with $\tau = 0.5$, the under-confidence ratio measures the percentage of correct answers where $f(Q) < 0.5$, and the over-confidence ratio measures the percentage of incorrect answers where $f(Q) > 0.5$.

We report results averaged across LLaMA-3-8B, Mistral-7B-v0.1, and Qwen2-7B on QASC, TruthfulQA, and MMLU-Pro. As shown in Table 9, Self-Ensemble consistently reduces both under-confidence and over-confidence compared to standard inference, demonstrating its ability to mitigate confidence distortion rather than exacerbate it.

	Multi-pass	Single-pass
Time (s) per question ↓	0.78	**0.16**
Memory (GB)	14.97	15.12
Accuracy (%)	78.62	80.67

Table 7: Efficiency and accuracy comparison of multi-pass vs. single-pass Self-Ensemble

Model	2-choice	4-choice	6-choice	8-choice
LLaMA-3.2-1B	81.75	66.85	51.51	50.22
LLaMA-3.2-3B	91.04	84.77	77.97	69.55
LLaMA-3-8B	91.90	86.29	76.35	73.22
Qwen2.5-1.5B	89.30	81.53	62.31	60.91
Qwen2.5-3B	91.47	81.53	75.81	69.98
Qwen2.5-7B	94.71	89.42	82.51	76.57
Qwen2.5-14B	95.68	91.68	86.29	79.59

Table 8: Accuracy of LLMs on the QASC dataset with different choice numbers.

D Additional Self-Ensemble Results Across Model Scales

To assess the generality of Self-Ensemble, we report 8-choice QASC accuracy across two model families spanning multiple scales. As shown in Table 10, Self-Ensemble consistently improves over standard inference from small (1–3B) to larger (7–14B) models.

E Robustness to Random Grouping Seeds

We quantify the effect of grouping randomness by repeating Self-Ensemble on QASC with seeds $s \in \{0, 1, 2, 3, 4\}$ that control the random partitioning of answer choices, holding all other variables fixed. Accuracy is reported as mean $\pm$ standard deviation across seeds. As shown in Table 11, the small variances (all ≤ 0.60 percentage points) indicate that Self-Ensemble is stable to grouping randomness across models.

F Group-wise Normalization with a Null Option

Self-Ensemble normalizes probabilities within each group by augmenting the group with a null choice ("None of the above") and applying a softmax over the augmented set. This guarantees per-group normalization (the probabilities over all candidates in a group sum to 1). Two immediate consequences explain the observed behavior: (i) when the correct answer is present, the null option receives negligible mass, so the sum over real options

Under-confidence ↓		
Model	Standard	`Self-Ensemble`
LLaMA-3-8B	43.45	**35.33**
Mistral-7B-v0.1	58.33	**50.60**
Qwen2-7B	41.41	**34.25**
Over-confidence ↓		
Model	Standard	`Self-Ensemble`
LLaMA-3-8B	25.09	**20.53**
Mistral-7B-v0.1	33.77	**21.66**
Qwen2-7B	30.89	**25.13**

Table 9: Under-confidence and over-confidence ratios at $\tau = 0.5$.

Model	Standard	`Self-Ensemble`
Qwen2.5 family		
Qwen2.5-1.5B	60.91	**71.81**
Qwen2.5-3B	69.98	**76.78**
Qwen2.5-7B	76.57	**81.43**
Qwen2.5-14B	79.59	**83.37**
LLaMA-3.x family		
LLaMA-3.2-1B	50.22	**56.37**
LLaMA-3.2-3B	69.55	**77.32**
LLaMA-3-8B	73.22	**80.67**

Table 10: `Self-Ensemble` consistently improves accuracy over standard inference across scales.

is close to 1; (ii) when the correct answer is absent, the null option absorbs probability mass, so the sum over real options drops below 1.

G Combination with Improvement

`Self-Ensemble` is designed to be orthogonal to existing approaches and can be seamlessly combined with them to further enhance the performance and reliability. Specifically, it can be integrated with efficient inference methods, such as KIVI (Liu et al., 2024; Yuan et al., 2024) or H2O (Zhang et al., 2023), to accelerate model execution while maintaining accuracy. Moreover, it can also be combined with secure deployment techniques, such as Taylor-Unswift (Wang et al., 2024b), which effectively protects the intellectual property of LLMs through parameter space decomposition. Finally, it is compatible with methods for multi-agent frameworks for solving complex tasks, such as (Chang et al., 2024; Yu et al., 2025).

Model	Accuracy
LLaMA-3-8B	80.24 ± 0.60
Mistral-7B-v0.1	77.91 ± 0.60
Qwen2-7B	65.87 ± 0.36

Table 11: Accuracy with error bar of `Self-Ensemble` on QASC across random seeds.

Name	Value
Data type	`torch.bfloat16`
Flash-Attention	False
Eval batch-size	1
Computing Infrastructure	GPU
GPU Model	NVIDIA-A40
GPU Memory	48GB
GPU Number	1
CUDA Version	12.3
CPU Memory	512GB

Table 12: Experiment configuration and computing infrastructure.

H Packages

In this work, we use the `transformers` along with `datasets` packages (Wolf et al., 2019) for model and dataset loading. All open-sourced packages have the Apache-2.0 license, which allows for academic research. We use public benchmarks and open models strictly for research and evaluation in accordance with their original licenses/terms of use. We release `Self-Ensemble` code, prompts, and evaluation scripts for research reproducibility; any derivatives of licensed datasets inherit their original research-only restrictions.

I Computational Infrastructure

The computational infrastructure information is given in Table 12.