Two heads are better than one: Enhancing LLMs Reasoning with Model Ensemble

Anonymous ACL submission

Abstract

Each Large Language Model (LLM) possesses unique strengths and limitations, urging the model ensemble to take full advantage of complementary strengths of different LLMs. To achieve this, we propose novel model ensemble methods which combine the confidence and popularity scores to generate the final outputs. The confidence is measured by the belief degree of one LLM to produce its output and the popularity is evaluated through the consistency degree of its output to other LLMs. Experimental results demonstrate that our methods markedly improve the performance on seven commonly used reasoning benchmarks, surpassing both the top-performing model and other strong baselines. Additionally, we explore the effects of varying ensemble sizes, offering valuable insights for optimizing model ensemble strategies for LLMs reasoning.

1 Introduction

Large Language Models (LLMs) have achieved remarkable progress in recent years. Many LLMs, including notable ones like GPT-4 (OpenAI, 2023), Claude (Anthropic, 2023), Bard (Google, 2023) and Llama-2 (Touvron et al., 2023), have shown impressive general capabilities, attributed to pre-training on large-scale corpora, instruction fine-tuning and alignment with human feedback. According to research conducted by Zhang et al. (2023), a single model faces significant challenges during reasoning tasks when the required knowledge was not encountered during the pre-training phase. Consequently, the combination of multiple LLMs, utilizing the unique inherent knowledge in each model, has the great potential to enhance the results across various reasoning tasks.

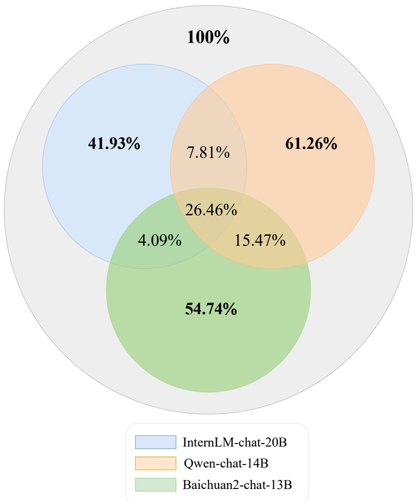


Figure 1: The Venn diagram shows the intersection and union relationships among the sets of samples correctly answered by three models on the GSM8K dataset. The overlapping sections of the circles in the diagram represent the proportion of samples correctly answered by different models in common.

Figure 1 shows a Venn diagram of the sample sets correctly answered by the three models on the GSM8K (Cobbe et al., 2021) dataset. The samples correctly answered by all three models account for only 26.46%. If we can effectively integrate the responses generated by the three models, the theoretical upper bound of accuracy could reach 77.64%, which is much higher than the accuracy of the three individual models at 41.93%, 54.74%, and 61.26% respectively. Therefore, by leveraging the complementary strengths of various LLMs through an ensemble approach, it is feasible to construct an all-encompassing model that outperforms individual models. Just as the saying goes *'two heads are better than one'*, ensemble methods can bring together the unique capabilities of each model, addressing their individual limitations and amplifying their collective strengths.

Previous studies on LLMs ensemble (Farinhas et al., 2023; Jiang et al., 2023) often relied on task-specific judge models (Farinhas et al., 2023) or specially trained generative models (Jiang et al., 2023). The use of these methods is limited in certain scope, and they require additional supervised training. Other studies (Wang et al., 2023) primarily concentrates on the internal integration within a single model, which we refer to as "self-ensemble", rather than ensemble among multiple models. To address the shortcomings of existing ensemble methods, we present three novel unsupervised LLMs ensemble methods: Confidence-Based, Popularity-Based and MBR-Based methods. The methods integrates the confidence scores of individual models in their responses and the consistency scores of one response to others (called 'popularity') to get the final answer. The methods eliminate the need for additional parameter training and can be used for a variety of tasks. Experimental results demonstrate that our method outperforms all single models and strong ensemble baselines in various benchmarks.

In this paper, we make the following contributions:

- We introduce three novel unsupervised ensemble methods for multiple LLMs. These methods combine the generative confidence of single models and the majority consensus of multiple models. The methods are not limited to any specific domain and do not require additional supervised training.
- Experiments demonstrate that our approaches exhibit substantial improvements in performance, consistently surpassing the top-performing single models and strong baselines across seven benchmarks.

2 Related Work

2.1 Model ensemble

Currently, common ensemble methods can be broadly categorized into two types: supervised methods (Jiang et al., 2023) and unsupervised methods (Wang et al., 2023; Farinhas et al., 2023).

Supervised ensemble methods require additional training of specialized models and are usually limited by the tasks and domains. Jiang et al. (2023) presented a question to eleven different models to generate responses. Then a ranking model (He et al., 2023), PairRanker, is trained to select the top-ranked responses and a T5-like model (Chung et al., 2022) is trained to generate the final answers. The introduction of trainable parameters can improve performance, but it also leads to higher time and computational demands, and reduces the model's flexibility for direct application to other tasks.

Unsupervised ensemble methods currently mainly focus on the integration of multiple responses from a single model. Majority voting (Lam and Suen, 1997) is the most commonly used unsupervised ensemble method. Wang et al. (2023) introduced Self-Consistency decoding, selecting answers through a majority vote among multiple responses. Zhou et al. (2020) and Farinhas et al. (2023) utilized the Minimum Bayes-risk decoding (Kumar and Byrne, 2002; Eikema and Aziz, 2020) to integrate multiple candidate results from a single model in machine translation tasks, achieving impressive results. But their approach still involves integration only on a single model. Unsupervised ensemble methods for multiple LLMs remains a highly promising research field.

2.2 Reasoning of Large Language Models

Reasoning is the process of integrating various types of knowledge from both explicit and implicit sources, to derive new conclusions about real or hypothetical situations in the world (Yu et al., 2023). According to Qiao et al. (2023), reasoning tasks can be categorized into several types, such as Arithmetic Reasoning , Commonsense Reasoning and more. In addition, academic examination style multiple-choice questions (Hendrycks et al., 2021) also represent an important form of reasoning tasks.

Existing studies have explored numerous approaches aimed at unlocking reasoning capabilities of large language models. A notable development is the Chain-of-Thought (CoT) reasoning (Wei et al., 2022), which navigates models through step-by-step thinking to tackle complex reasoning tasks. Wang et al. (2023) introduced Self-Consistency, a technique that involves sampling multiple reasoning paths and selecting the final answer through majority voting. However, no model is capable of handling all reasoning tasks, especially when applying the model to domains that were not seen during the pre-training phase (Zhang et al., 2023).

3 Methods

Task Definition. Consider a reasoning task with an input denoted by x. Let there be K LLMs represented as $M_1, M_2, ..., M_K$. For each model M_k, we define a specific prompt p_k. We require each model to generate responses to the input x with its prompt. The response produced by model

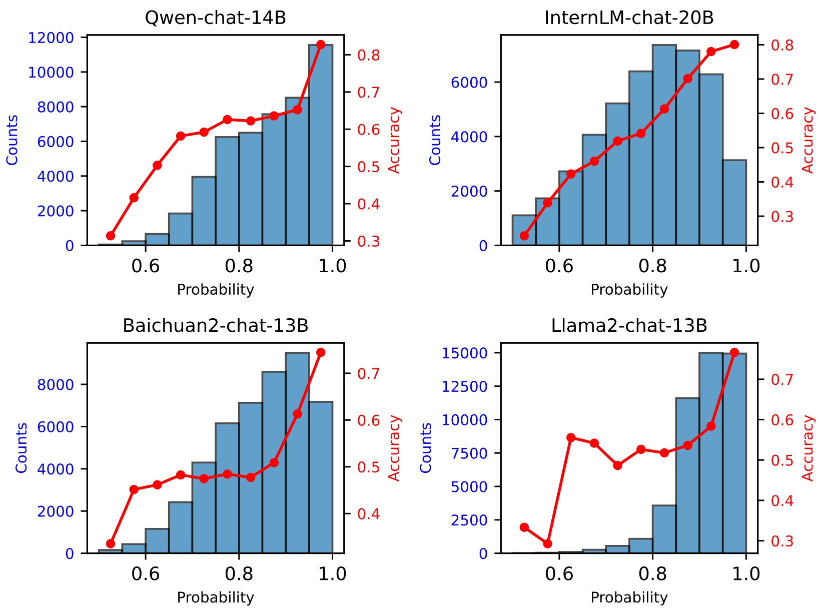


Figure 2: We evaluate the accuracy of responses generated by four fundamental models across different probability intervals, utilizing the aggregation of all datasets described in Section 4.1.

M_k when provided with input x is denoted by $r_k = M_k(p_k, x)$.

In this paper, the task of model ensemble is conceptualized as the process of choosing the most appropriate response among those generated by different models. The chosen response of ensemble is represented as R^*.

We employ three unsupervised methods for model ensemble, as Figure 3, namely **Confidence-Based**, **Popularity-Based**, and **MBR-Based** methods. Each method employs its respective set of criteria to choose responses.

3.1 Confidence-Based Method

Generally speaking, the greater the model's confidence in a response, the more likely it is that this response accurately represents the correct answer to the input. Accurately estimating the confidence of a model in generating responses remains an unresolved issue. Existing methods (Lu et al., 2022) require the introduction of additional training or parameters. We avoid introducing new parameters and instead use the average probability of each token generated by the model as a measure of the model's confidence.

We find that the accuracy of reasoning is closely related to the probability of generating responses. As illustrated in Figure 2, our experiments on four commonly used LLMs reveal that the higher the generation probability, the greater the chance that this response is accurate. Thus, the probability of response serves as an indicator of the model's confidence in the correctness of its own answer.

Therefore, we use the probability of responses as a measure of each model's confidence, and we perform model ensemble based on these confidence values. This method is illustrated in Figure 3(a). In particular, given the input x and the prompt p_k, the confidence of responses r_k generated by model M_k can be represented as Equation 1 :

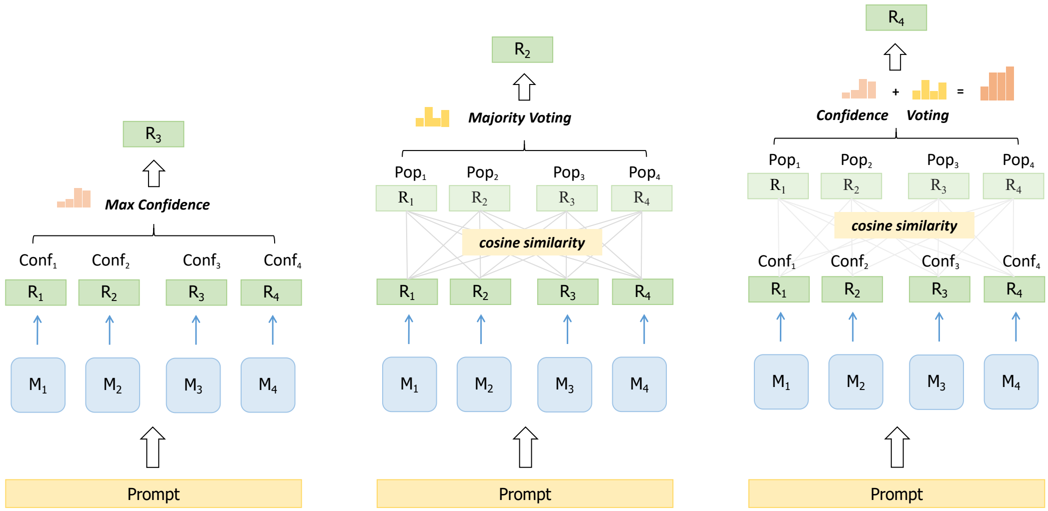
$$\mathrm{Conf}(r_k|M_k) = \exp^{\frac{1}{T}\sum_{t=0}^{T}\log P(r_k^t|p_k,x,r_k^1,\ldots,r_k^{t-1})} \tag{1}$$

We select the response that maximizes the value, as detailed in the specified Equation 2.

$$R^* = \arg\max_{r_k}\{\mathrm{Conf}(r_k|M_k)\} \tag{2}$$

3.2 Popularity-Based Method

Majority voting plays a crucial role in ensemble learning. When experts, each offering unique insights, come together, majority voting often outperforms relying on a single opinion (Lam and Suen, 1997). Inspired by Self-Consistency decoding (Wang et al., 2023), we integrate the majority voting into our model ensemble methods. Our aim is to choose a response that reflects **popularity**, representing the agreement of most models and demonstrating consensus across multiple models.



(a) Confidence-Based **(b) Popularity-Based** **(c) MBR-Based**

Figure 3: We utilize three distinct unsupervised approaches for model ensemble: **(a) Confidence-Based** method, **(b) Popularity-Based** method, **(c) MBR-Based** method. We prompt every model participating in the ensemble to answer the same reasoning question, thereby generating their responses. Subsequently, each method assess responses from various models, ultimately selecting the final appropriate response.

Considering that matching-Based voting of Self-Consistency decoding is not suitable for all reasoning tasks, We propose a new voting mechanism. We calculate semantic similarity between each response generated by one model and all responses from other models, aggregating these similarity scores as votes for popularity:

$$\text{Popularity}(r_k) = \sum_{j=0, j\neq k}^{K} \text{SIM}(r_k, r_j) \quad (3)$$

where $\text{SIM}(.)$ denotes the semantic similarity between two responses.

The response receiving the highest number of votes is considered the most popular response and thus becomes the final choice, as illustrated in Figure 3(b). This is formally expressed as:

$$R^* = \arg\max_{r_k} \{\text{Popularity}(r_k)\} \quad (4)$$

3.3 MBR-Based Method

In machine translation tasks, Minimum Bayesian Risk (MBR) is frequently employed in the decoding process, commonly referred to as MBR decoding (Eikema and Aziz, 2020; Farinhas et al., 2023), typically taking the following form:

$$h* := \arg\max_{h\in H} \mathbb{E}_{p_{(y|x,\theta)}}[u(y,h)], \quad (5)$$

where $u(y,h)$ is utility function which assesses the hypothesis h against a reference y. Equation 5 aims to identify the candidate that optimally maximizes the expected utility function $u(y,h)$ (or minimises expected loss) over the entire set of translation hypotheses H. Furthermore, the expected value is generally estimated through a Monte Carlo sampling (Farinhas et al., 2023):

$$\mathbb{E}_{p_{(y|x,\theta)}}[u(Y,h)] \approx \frac{1}{M}\sum_{i=1}^{M} u(y_i,h) \quad (6)$$

This process involves using M samples drawn from $p_{(y|x,\theta)}$, which provides an unbiased estimate of the expected utility.

Inspired by the MBR decoding method, we migrate this approach to the field of model ensemble. We utilize a popularity voting mechanism as the utility function and define M in Equation 6 as the number of models participating in the ensemble. Additionally, we consider the unique attributes of each model by incorporating the confidence of the model at the time of generating each response, as formulated in Equation 7

$$R^* = \arg\max_{r_k} \frac{1}{K-1} \sum_{j=0,j\neq k}^{K} \text{Conf}(r_k|M_k)\text{SIM}(r_k,r_j) \quad (7)$$

We name it MBR-Based ensemble method and Figure 3(c) provides a visual illustration of this method. MBR-Based method integrates generation confidence from one model and majority voting from multiple models, effectively capturing both self-assurance of single model and the collective agreement among all models.

4 Experiments

4.1 Setup

We conduct a comprehensive evaluation of our ensemble methodology, which integrates eight different large language models (LLMs), across seven reasoning benchmarks:

Models. We select four types of large language models with unique foundational architectures: Llama-2 (Touvron et al., 2023), Baichuan-2 (Baichuan, 2023), Qwen (Bai et al., 2023), and InternLM (InternLM, 2023) from OpenCompass Leaderboard[1]. For each foundational model, we investigate two configurations varied by scale, specifically 7B and 13B parameters. These models will be denoted as M1-M8 in the following sections.

Benchmarks. We employ seven different reasoning tasks as evaluation benchmarks. In terms of task types, we select datasets from multiple domains following the categorizations established in prior research (Qiao et al., 2023; Hendrycks et al., 2021), including academic examination, arithmetic reasoning, and commonsense reasoning datasets. These tasks encompass two main formats: multiple-choice reasoning tasks and generative answer-Based reasoning tasks. This approach aims to comprehensively assess reasoning capabilities of our ensemble methods across various fields.

- **Academic Examination.** We use the Measuring Massive Multitask Language Understanding (MMLU; Hendrycks et al., 2021) dataset, including fifty seven tasks covering areas such as mathematics, American history, computer science and more.
- **Arithmetic Reasoning.** We use GSM8K (Cobbe et al., 2021) and AQUA-RAT (Ling et al., 2017) datasets. Both of them require multi-step arithmetic reasoning to solve math world problems.
- **Commonsense Reasoning.** We use CommonsenseQA (CSQA; Talmor et al., 2019) and OpenBookQA (OBQA; Mihaylov et al., 2018 2018) that require multi-step reasoning using commonsense knowledge. We also include TriviaQA (Joshi et al., 2017) without its reference documents, which requires various knowledge. Additionally, we use SQuAD 2.0 (Rajpurkar et al., 2018) for reading comprehension task.

Settings. We apply our three ensemble techniques: Confidence-Based, Popularity-Based and MBR-Based methods on four selected models. Typically, we select the top four models based on their performance for each specific dataset, leading to a variation in the ensemble's model selection. However, prior research has indicated that when there is a substantial performance gap among different models on a particular dataset, models with inferior performance can significantly degrade the overall efficacy of the ensemble (Wang et al., 2023). Therefore, if the performance gap between the top four models is too large, we select four models with more similar performance for ensemble instead. Regarding the prompt, we adopt a zero-shot (Kojima et al., 2022) testing framework for all models. For each dataset, we create a specific prompt that includes a brief task description and relevant questions or options. For more information on the prompt, please refer to Appendix B.

Baselines. As baselines, we select three types of methods:

- **Single Model.** We evaluate all eight LLMs separately on each dataset using a greedy decoding strategy.
- **Self-Ensemble.** The best-performing single model is employed to generate answers four times using sampling decoding strategy and our three ensemble methods as self-ensemble baseline.
- **Supervised Multi-Model Ensemble.** PairRanker (Jiang et al., 2023) is used to select the most appropriate answer from candidates for the same question. It has been trained with responses from various models in instruction-following tasks.

Evaluation. After obtaining the model's response to a question, we extract the answer in different ways according to the dataset's features. For datasets with multiple choices, we prompt the model with final response, question and options. We then instruct the model to calculate and output the probabilities for each option following the approach of Baichuan (2023). Then we select the option with the highest probability as the final answer.

[1]https://opencompass.org.cn/leaderboard-llm

Method	MMLU	GSM8K	AQUA	CSQA	OBQA	TriviaQA	SQuAD
			M1-M4 vary across different datasets				
M1-greedy	50.94	61.26	30.71	78.87	71.60	68.03	71.09
M2-greedy	49.76	54.74	30.31	74.12	70.00	66.95	69.42
M3-greedy	49.39	52.39	29.13	73.05	65.80	61.95	69.22
M4-greedy	48.47	41.93	28.35	70.27	60.80	60.62	66.76
			Self-ensemble results derived from four samples of the best model (M1)				
Random	49.34	56.10	25.59	75.92	67.60	66.68	70.25
Confidence	<u>50.22</u>	60.80	<u>29.13</u>	76.82	70.00	<u>68.48</u>	71.73
Popularity	50.11	58.23	22.83	<u>77.48</u>	69.00	67.99	<u>72.87</u>
MBR-Based	50.14	<u>61.87</u>	27.56	76.66	<u>70.80</u>	68.32	72.66
			Ensemble results of models M1-M4				
Random	49.49	51.48	30.71	75.18	67.60	64.17	69.23
PairRanker	**52.43**	59.06	34.65	76.74	72.60	66.48	72.35
Confidence	50.87	62.09	33.07	77.31	72.60	70.05	73.24
Popularity	51.25	57.62	30.31	78.38	71.20	69.95	75.45
MBR-Based	51.58	**63.00**	**35.04**	**79.44**	**74.60**	**70.85**	**75.72**

Table 1: The overall accuracy of different methods across all datasets. The models participating in the ensemble vary across different datasets, denoted as M1-M4 (with performance ranked from high to low). Self-ensemble corresponds to the ensemble results of four sampled responses from the best-performing model M1. The methods that performs best in self-ensemble are indicated with an <u>underline</u>. The best overall results are highlighted in **bold**.

For other types of datasets, we employ rule-based methods such as regular expression matching to extract the correct answer from the response. Finally, across all datasets, we employ accuracy as the evaluation metric.

4.2 Main results

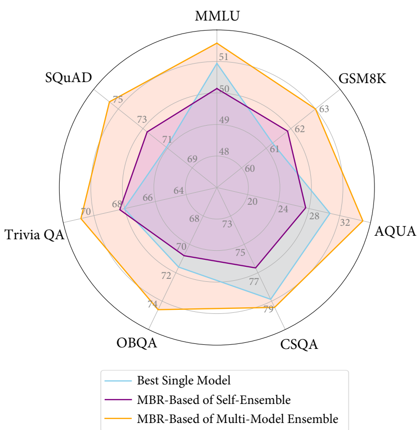


Figure 4: Comparative analysis of model performances: best single model, MBR-Based method of self-ensemble and MBR-Based method of multi-model ensemble across all datasets. To demonstrate the relative performance among methods, we normalized the original performance value.

We report all experimental results in Table 1. Models M1 to M4 (performance from high to low) represent the models participating in the ensemble. For detailed information on the selection of the four specific models (M1 to M4) and performance of all eight models for each dataset, please refer to Appendix A. In Figure 4, we present the results of the best single model, the MBR-Based method of Self-ensemble and the MBR-Based method of multi-model ensemble across all datasets.

Confidence-Based method achieves better performance than best single models on five datasets, as shown in Table 1. In most cases, when the probability assigned to a response is higher, it indicates that the model is more confident about the content. Consequently, the chance that the response contains correct answer is higher, as illustrated in Figure 2. However, sometimes the most confident model is not necessarily the one that generates the best response, and we provide further analysis in Section 4.5. These results indicate that relying on the confidence of response by a single model can be effective but insufficient.

Popularity-Based method has also achieved better performance than single models on three datasets. The core of this method is majority voting, based on comparing semantic similarities, aiming to identify the answers that most models consider to be correct. When models, each with distinct knowledge and parameters, collaborate in decision-making, majority voting often yields more effective results than relying on a single model (Lam and Suen, 1997). Furthermore, we observe that the Popularity-Based method is ineffective when there's a significant performance gap among the models, as the weaker models introduce significant

N	Model	Method	**MMLU**	**GSM8K**	**AQUA**	**CSQA**	**OBQA**	**TriviaQA**	**SQuAD**
1	M1	Greedy	50.94	61.26	30.71	78.87	71.60	68.03	71.09
2	M1-M2	Random	50.46	57.62	32.68	75.84	68.00	67.47	70.61
		Confidence	50.94	62.47	33.07	**79.93**	72.60	70.09	73.42
		Popularity	-	-	-	-	-	-	-
		MBR-Based	50.94	62.47	33.07	**79.93**	72.60	70.09	73.42
3	M1-M3	Random	49.91	58.38	26.38	75.02	67.00	65.79	70.21
		Confidence	50.84	62.55	31.50	77.97	74.20	70.24	73.81
		Popularity	50.82	58.76	32.28	78.05	72.20	69.47	73.54
		MBR-Based	51.56	61.49	**35.04**	78.79	**74.60**	70.66	74.19
4	M1-M4	Random	49.49	51.48	30.71	75.18	67.60	64.17	69.23
		Confidence	50.87	62.09	33.07	77.31	72.60	70.05	73.24
		Popularity	51.25	57.62	30.31	78.38	71.20	69.95	75.45
		MBR-Based	**51.58**	**63.00**	**35.04**	79.44	**74.60**	**70.85**	75.72
5	M1-M5	Random	48.21	49.89	28.35	71.25	64.80	63.49	67.92
		Confidence	46.87	44.88	28.74	74.04	69.60	69.53	71.91
		Popularity	47.69	55.27	27.95	**79.93**	71.00	69.67	76.77
		MBR-Based	46.33	50.04	29.53	78.05	72.20	70.33	**77.44**

Table 2: The accuracy of different ensemble methods across all datasets when the number of models changes. The best results in each group are underlined and the best overall results are **bolded**.

noise.

MBR-Based method has achieved the best results on almost all datasets. It surpasses the performance of the best model by a large margin on several benchmarks. At the same time, we can observe that the MBR-Based method has also shown improvements compared to Confidence-Based methods and Popularity-Based methods. This method demonstrates its ability to integrate the confidence of single models and the consensus degree of multiple models, maximizing the strengths of all models while minimizing their weaknesses.

4.3 Our methods vs. Self-ensemble

Table 1 also presents the results of self-ensemble from the best-performing model (M1) on each dataset. Self-ensemble only shows marginal improvements in just three of the datasets compared with the greedy decoding strategy. This indicates that LLMs might find the optimal solution through a single greedy decoding process for some specific tasks, making additional sampled responses for the ensemble ineffective in significantly improving performance.

The observed gap between self-ensemble and multi-model ensemble methods can be attributed to the limitations in diversity and perspective in a single model. Self-ensemble methods generate varied responses, but they are confined within the model's own learned patterns and inherent biases, which can limit the accuracy. In contrast, multi-model ensemble methods integrate the strengths of diverse models. This variety allows multi-model ensemble methods to excel in different aspects of a specific task, thereby mitigating the weaknesses inherent in individual models.

4.4 Our methods vs. Supervised methods

As shown in Table 1, PairRanker outperforms the best-performing model only on four datasets. This performance limitation stems from the fact that PairRanker's training data does not cover the comprehensive knowledge necessary for a wide range of reasoning tasks. Consequently, PairRanker struggles to accurately assess the correctness of the responses generated by different models, making it less effective than our proposed MBR-Based method across almost all datasets. Compared to supervised methods, our approach demonstrates better versatility, being able to fully leverage the knowledge of different models without being limited by training data.

4.5 Analysis of model numbers in ensemble

Table 2 shows that the number of models in the ensemble has a significant impact. We vary the number from two to five by adding a relatively weaker model sequentially. It is worth mentioning that if there are only two models participating in the ensemble, only the similarity between the responses of these two models will be calculated, so the final response can not be selected through their popularity. At the same time, the MBR-Based method degenerates into the Confidence-Based method in this case.

Overall, when the number of models in the ensemble is less than five, our MBR-Based method achieves the best performance across almost all datasets, except on GSM8K with N=3. However, when the ensemble expands to include five models, we observe notable performance degradation across several datasets. It is possible that including more models can widen the gap between the best and worst-performing models, which is not beneficial for the ensemble. Although we have made efforts to select models with similar performance, there remains a difference of more than 5% among models in some tasks, limited by the availability of publicly available LLMs.

	M1	M2	M3	M4	M5
LL	-0.094	-0.108	-0.123	-0.182	**-0.066**

Table 3: The average log-likelihood (LL) of responses generated by M1 to M5 on GSM8K dataset.

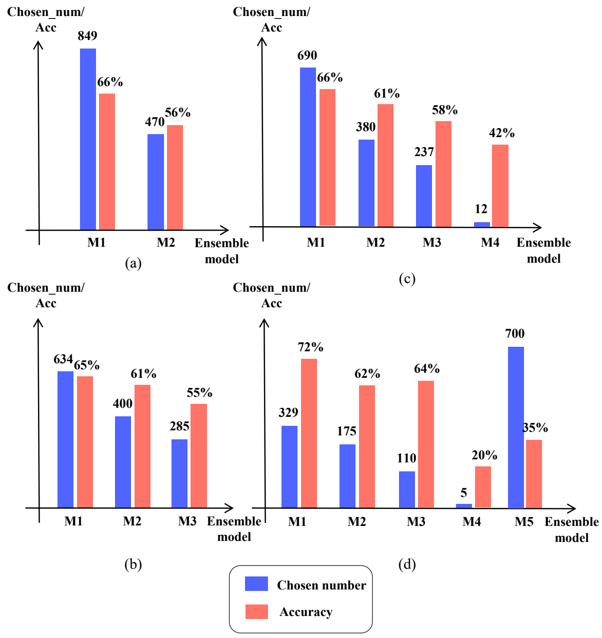


Figure 5: The chosen number of responses generated by each model and accuracy under different ensemble sizes.

In order to analyze the reasons behind the changes of overall ensemble performance after adding a relatively weaker model, we present the response allocation results of the MBR-Based ensemble on the GSM8K dataset in Figure 5 with the ensemble number varying from two to five. This includes the number of selection for each model's responses and the accuracy of these responses. Comparing the results of Figure 5 (b) and (c), we can find that when adding a relatively weaker model M4, the generated responses are not frequently chosen and thus do not significantly affect overall performance. More importantly, M4's participation appears to have a positive impact on the performance of other models (M1-M3). This is likely that M4's responses participate in the majority voting mechanism, potentially boosting the popularity of a correct answer from M1-M3. Consequently, with the inclusion of M4, there is an observed improvement in the accuracy from 61.49% to 63.00%.

Moreover, after adding M5, the performance of the MBR-Based ensemble method on GSM8K drops significantly. Comparing the change of the response allocation between Figure 5 (c) and (d), it can be found that the responses of the weakest model M5 are frequently chosen under this case. To explore the reasons for this phenomenon, we list the average log likelihood values of responses generated by M1-M5 in Table 3, reflecting the response generation probabilities of each model. We find that the probability of responses generated by M5 is significantly higher than M1-M4, thus the number of chosen responses from M5 under the MBR-Based method is more, leading to a drastic drop (from 63.00% to 50.04%) in ensemble performance.

5 Conclusions

We introduce three model ensemble approaches designed for large language models across various reasoning tasks: the confidence-Based method, popularity-Based method, and MBR-Based method. Importantly, our methods effectively take full advantages of different models across various tasks. Our MBR-Based Method, which combines the generative confidence and consensus degree from diverse models, has shown superior performance across almost all reasoning benchmarks. The empirical findings further demonstrate that model ensemble strategies, as opposed to single model, successfully utilize the combined reasoning capabilities of different models to achieve enhanced performance. Moreover, our analysis demonstrates that the number of models included in the ensemble significantly impacts its overall effectiveness.

Limitations

The limitations of our work can be summarized in two main aspects. Firstly, our study mainly focused on the ensemble methods of large language models in reasoning tasks. For generative tasks such as machine translation and summarization, their evaluation metrics only measure the similarity between the responses and reference answers, which does not necessarily indicate the correctness of the responses. Therefore, our research concentrated on reasoning tasks, which generally have clear correct answers and evaluation standards. Secondly, our experiments were conducted mainly on models with sizes of 7B and 13B. Due to resource constraints, we did not perform experiments on larger models, and thus, it remains uncertain whether our methods can be extrapolated to larger-scale models.

References

Anthropic. 2023. Introducing claude.

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

Baichuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

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.

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*.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

António Farinhas, José G. C. de Souza, and André F. T. Martins. 2023. An empirical study of translation hypothesis ensembling with large language models.

Google. 2023. Bard.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

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)*.

InternLM. 2023. Internlm: A multilingual language model with progressively enhanced capabilities. `https://github.com/InternLM/InternLM`.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-blender: Ensembling large language models with pairwise ranking and generative fusion. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14165–14178, Toronto, Canada. Association for Computational Linguistics.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *NeurIPS*.

Shankar Kumar and William Byrne. 2002. Minimum bayes-risk word alignments of bilingual texts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 140–147, USA. Association for Computational Linguistics.

L. Lam and S.Y. Suen. 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 27(5):553–568.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Yu Lu, Jiali Zeng, Jiajun Zhang, Shuangzhi Wu, and Mu Li. 2022. Learning confidence for transformer-based neural machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2353–2364, Dublin, Ireland. Association for Computational Linguistics.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. 2023. Reasoning with language model prompting: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5368–5393, Toronto, Canada. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

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.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.

Fei Yu, Hongbo Zhang, Prayag Tiwari, and Benyou Wang. 2023. Natural language reasoning, a survey.

Zhuosheng Zhang, Yao Yao, Aston Zhang, Xiangru Tang, Xinbei Ma, Zhiwei He, Yiming Wang, Mark Gerstein, Rui Wang, Gongshen Liu, and Hai Zhao. 2023. Igniting language intelligence: The hitchhiker's guide from chain-of-thought reasoning to language agents.

Long Zhou, Jiajun Zhang, Xiaomian Kang, and Chengqing Zong. 2020. Deep neural network-based machine translation system combination. *ACM Trans. Asian Low Resour. Lang. Inf. Process.*, 19(5):65:1–65:19.

A Full Results

The performance of eight models across all datasets is presented in Table 4 to Table 10. For those models participating in the ensemble, we have marked them as M1 to M5 in the "chosen" column. It is noteworthy that in three of the seven datasets, we did not select the top five performing models for ensemble. This decision was based on an observation (Wang et al., 2023): the great performance gap can adversely affect the effectiveness of the model ensemble. Therefore, we selected models with more similar performance instead.

B Prompt Template

We provide the full prompts used for each dataset in Table 11, which consist of a task description and specific questions or options.

Table 4: MMLU

Model	Chosen	MMLU
InternLM-7b	-	44.15
Llama2-7b	**M5**	45.26
Baichuan2-7b	**M4**	48.47
InternLM-20b	**M3**	49.39
Baichuan2-13b	**M2**	49.76
Llama2-13b	**M1**	50.94
Qwen-7b	-	54.86
Qwen-14b	-	63.17

Table 5: GSM8K

Model	Chosen	GSM8K
Llama2-7b	-	24.94
InternLM-7b	-	32.83
Baichuan2-7b	-	34.42
Llama2-13b	M5	35.10
InternLM-20b	M4	41.93
Qwen-7b	M3	52.93
Baichuan2-13b	M2	54.74
Qwen-14b	M1	61.26

Table 6: AQUA

Model	Chosen	AQUA
InternLM-7b	-	23.23
Llama2-7b	M5	27.17
Baichuan2-7b	M4	28.35
Llama2-13b	M3	29.13
Qwen-7b	M2	30.31
InternLM-20b	M1	30.71
Baichuan2-13b	-	37.80
Qwen-14b	-	45.67

Table 7: CSQA

Model	Chosen	CSQA
Llama2-7b	-	56.76
Llama2-13b	-	57.49
Baichuan2-7b	-	63.14
InternLM-7b	M5	65.11
Baichuan2-13b	M4	70.27
Qwen-7b	M3	73.05
InternLM-20b	M2	74.12
Qwen-14b	M1	78.87

Table 8: OBQA

Model	Chosen	OBQA
Llama2-7b	-	54.60
Llama2-13b	-	56.80
InternLM-7b	M5	57.20
Baichuan2-7b	M4	60.80
InternLM-20b	M3	65.80
Qwen-7b	M2	70.00
Baichuan2-13b	M1	71.60
Qwen-14b	-	85.80

Table 9: TriviaQA

Model	Chosen	TriviaQA
InternLM-7b	-	40.00
Baichuan2-7b	-	53.35
InternLM-20b	-	56.74
Llama2-7b	M5	60.37
Qwen-7b	M4	60.62
Baichuan2-13b	M3	61.95
Qwen-14b	M2	66.95
Llama2-13b	M1	68.03

Table 10: SQuAD

Model	Chosen	SQuAD
Baichuan2-7b	-	44.14
Baichuan2-13b	-	48.72
Qwen-7b	-	50.75
InternLM-7b	M5	63.98
Llama2-7b	M4	66.76
InternLM-20b	M3	69.22
Qwen-14b	M2	69.75
Llama2-13b	M1	71.09

Table 11: prompt

Datasets	Prompt
MMLU	"The following is a multiple choice question (with answers) about {Subject}. Please analyze carefully and provide your answer. Question: {Question} {Option} Answer: Let's think step by step."
GSM8K	"Please solve the following math problem: Question: {Question} Answer: Let's think step by step."
AQUA	"Please solve the following math problem: Question: {Question} {Option} Answer: Let's think step by step."
CSQA	"The following is a multiple choice question that requires commonsense reasoning to answer. Please analyze carefully and provide your answer.Question: {Question} {Option} Answer: Let's think step by step."
OBQA	"The following is a multiple choice question that require external facts or knowledge. Please analyze carefully and provide your answer. Question: {Question} {Option} Answer: Let's think step by step."
TriviaQA	"Please answer the following question, your answer should be as simple as possible. Question: {Question} Answer:"
SQuAD	"Given the following passage, answer the subsequent question. If the passage does not contain the information to answer the question, reply with 'unanswerable'. Passage: {Passage} Question:{Question} Answer:"