Scoring, Reasoning, and Selecting the Best!
Ensembling Large Language Models via a Peer-Review Process

Zhijun Chen[1†], Zeyu Ji[1†], Qianren Mao[2], Junhang Cheng[1], Bangjie Qin[3], Hao Wu[4], Zhuorran Li[1], Jingzheng Li[2], Kai Sun[4], Zizhe Wang[5], Zhu Sun[6], Xiangyang Ji[5], Hailong Sun[1*]

[1]Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Hong Kong University of Science and Technology, Hong Kong, China
[4]Xi'an Jiaotong University, Xi'an, China
[5]Tsinghua University, Beijing, China
[6]Singapore University of Technology and Design, Singapore, Singapore
{zhijunchen, zeyuji, sunhl}@buaa.edu.cn

Abstract

We propose LLM-PeerReview, an unsupervised LLM Ensemble method that selects the most ideal response from multiple LLM-generated candidates for each query, harnessing the collective wisdom of multiple models with diverse strengths. LLM-PeerReview is built on a novel, peer-review-inspired framework that offers a clear and interpretable mechanism, while remaining fully unsupervised for flexible adaptability and generalization. Specifically, LLM-PeerReview operates in three stages: For *scoring*, we use the emerging LLM-as-a-Judge technique to evaluate each response by reusing multiple LLMs at hand; For *reasoning*, we can apply a principled graphical model-based truth inference algorithm or a straightforward averaging strategy to aggregate multiple scores to produce a final score for each response; Finally, the highest-scoring response is selected as the ensemble output. LLM-PeerReview is conceptually simple and empirically powerful. The two variants of the proposed approach obtain new state-of-the-art results across four datasets, including outperforming the recent advanced model Smoothie-Global by 6.9% and 7.3% points, respectively.

1. Introduction

In recent years, the landscape of artificial intelligence been dramatically reshaped by the rise of Large Language Models (LLMs), including Gemini (Team et al. 2023), GPT-4 (Achiam et al. 2023), Llama (Touvron et al. 2023), and the recently introduced DeepSeek (Liu et al. 2024a). The success of these models has led to a surge in research activity, with over 182,000 models now available on Hugging Face.

Behind this research enthusiasm, we can observe two main point (Chen et al. 2025; Jiang, Ren, and Lin 2023): 1) *Persistent performance concerns*: Although large language models can be easily deployed for zero-shot or in-context few-shot inference, they still face common performance issues, such as limited accuracy, hallucinations, and misalignment with human goals; 2) *The varying strengths and weaknesses of LLMs*: These models exhibit considerable variation in their responses due to factors like architectural differences, parameter size, tokenization, training data, and methodology. As a result, their outputs can differ substantially. With the above two aspects in mind and drawing on the spirit of Ensemble Learning (Dong et al. 2020), it is logical to consider that, for each task, rather than persistently relying on a single LLM based on public rankings or other criteria, it might be more advantageous to simultaneously consider multiple LLM candidates (usable out-of-the-box) and harness their distinct strengths. This is exactly what the recently emerging field of *LLM Ensemble* (Chen et al. 2025) explores.

As LLM Ensemble gains increasing attention, one well-established class of solutions—*ensemble-after-inference* (also known as *post-hoc ensemble*) methods(Jiang, Ren, and Lin 2023; Lv et al. 2024; Tekin et al. 2024; Li et al. 2024c; Guha et al. 2024; Si et al. 2023)—has emerged. These methods include the following two representative approaches (Chen et al. 2025):

- **Selection-then-regeneration approach** (Jiang, Ren, and Lin 2023; Lv et al. 2024; Tekin et al. 2024), during inference on a downstream task, first employs a pre-trained "PairRanker" module to select the top-K candidate responses—those deemed most likely to be of high quality—from a pool of LLM-generated responses. This selected subset is then fed into another fine-tuned LLM (e.g., Flan-T5-XL (Chung et al. 2024)) to synthesize a *final* response. While this line of work has attracted significant attention (Jiang, Ren, and Lin 2023), they rely heavily on carefully curated task-specific training data and the need to fine-tune an additional LLM, limiting their generalization and adaptability.
- **Similarity-based selection approach** (Li et al. 2024c; Guha et al. 2024; Si et al. 2023), instead, are mostly fully unsupervised (Guha et al. 2024; Si et al. 2023). These methods follow a simple and intuitive principle: *for a given query, select the response with the highest total similarity to all other responses*. While such methods pioneered unsupervised post-hoc LLM ensemble, their design remains coarse-grained—they rely on the naive

† Equal contribution

* Corresponding author

similarity-based selection strategy (Li et al. 2024c; Guha et al. 2024), along with shallow similarity measure of BLEU (Li et al. 2024c) or limited informational utilization (Guha et al. 2024). Thus, the true potential of selection-based post-hoc ensemble remains largely untapped.

When we revisit this research problem, we ask the most fundamental question: *In the real world, how would humans select the most ideal text from a set of candidate texts?* Perhaps the most immediate and relatable real-world example is: the academic peer-review process. Motivated by this, we propose a new, fully unsupervised LLM Ensemble method called LLM-PeerReview. Specifically, LLM-PeerReview is structured sequentially around three components: 1) *Scoring (analogous to paper reviewing)*: Given multiple candidate responses to the same query, we adopt the LLM-as-a-Judge approach—leveraging available LLMs as evaluators that assess each response and assign a score (e.g., 5.0 indicating Strong Accept, 4.0 indicating Weak Accept, etc.); 2) *Reasoning (analogous to final score estimation made by the senior reviewer)*: Besides being able to perform direct averaging calculations, we can invoke graphical model-based truth inference techniques from crowdsourcing and weak supervision literature, to perform refined, reliability-aware weighted score aggregation, deriving a final score for each response; 3) *Selection (analogous to final decision made by the senior reviewer)*: This step is analogous to how a senior reviewer or area chair selects the most suitable paper from a small set of submissions. For each query, once final scores have been inferred for all responses, we select the highest-scoring response as the ensemble result.

LLM-PeerReview is built upon the unsupervised, selection-based paradigm, and introduces a novel peer-review-inspired framework for LLM Ensmeble, offering a clear and interpretable mechanism. Within the methodology, LLM-PeerReview leverages the emerging LLM-as-a-Judge technique (Li et al. 2024a,b) to evaluate each candidate response, thereby effectively reusing the collective intelligence of multiple existing LLMs at hand. Moreover, the use of a graphical-model-based truth inference algorithm allows us to benefit from the principled graphical model for refined and reliability-aware aggregation of multiple scoring signals. Empirically, extensive experiments conducted with 7B-scale LLMs show that the proposed LLM-PeerReview outperforms recent advanced similarity-based methods (Li et al. 2024c; Guha et al. 2024).

2. LLM-PeerReview

This section presents our proposed method, LLM-PeerReview, with an overview shown in Figure 1. We begin by formalizing the research problem, followed by a detailed introduciton of the three components of our method in Sections 2.1, 2.2, and 2.3.

Problem Formulation: (Unsupervised) LLM Ensemble. Without access to any reference responses (i.e., ideal/truth responses), we are given a set of queries $\{\mathbf{x}^{(i)}\}_{i=1}^{I}$ for a generative task. We have access to J large language models $\{\mathcal{M}_j\}_{j=1}^{J}$, where each model $\mathcal{M}_j$ generates a response $\mathbf{r}^{(i,j)} = \mathcal{M}_j(\mathbf{x}^{(i)})$—which is often not ideal—for a given query $\mathbf{x}^{(i)}$. Thus, for each query, we have a set of zero-shot inference responses $\mathbf{R}^{(i)} = [\mathbf{r}^{(i,1)}, \ldots, \mathbf{r}^{(i,J)}]$ from heterogeneous LLMs $\{\mathcal{M}_j\}_{j=1}^{J}$, while the underlying reference response $\mathbf{y}^{(i)}$ is *unobserved* to us. Our goal is to ensemble the LLM responses to produce a single, high-quality final response for each query $\mathbf{x}^{(i)}$, using the available data $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{R}^{(i)}\}_{i=1}^{I}$.

2.1. Scoring

Naive point-wise scoring. As shown in Figure 1 and within our proposed LLM-PeerReview, the scoring phase occurs after the LLMs have first generated responses to the input queries. Using the LLM-as-a-Judge technique (Gu et al. 2024), each LLM judge can assign a *point-wise score* to each response, representing its overall quality. For example, the score can range from [1, 2, 3, 4, 5], representing the levels of [“Very Poor”, “Poor”, “Acceptable”, “Good”, “Excellent”]. [1]

Flipped-triple scoring trick. The above naive point-wise scoring technique can provide scores; however, we propose a technique called *flipped triple scoring*, which we recommend when applying our approach. Specifically: 1) For multiple responses from different models to the same query $\mathbf{x}^{(i)}$, we first shuffle them; 2) Then, for each LLM judge $\mathcal{M}_{j'}$, we score the response triplet $[\mathbf{r}^{(i,j-1)}, \mathbf{r}^{(i,j)}, \mathbf{r}^{(i,j+1)}]$ sequentially (with J times), and for each iteration, we also score the flipped version (i.e., $[\mathbf{r}^{(i,j+1)}, \mathbf{r}^{(i,j)}, \mathbf{r}^{(i,j-1)}]$). As a result, each response receives six scores from the same LLM judge. We can simply average these scores to obtain a final score, which serves as the score $y^{(i,j;j')}$ for response $\mathbf{r}^{(i,j)}$ by LLM judge $\mathcal{M}_{j'}$. In short, this technique mitigates two common scoring biases in LLM-as-a-Judge (Wang et al. 2023; Zheng et al. 2023; Gu et al. 2024). First, in point-wise scoring, models tend to show a consistent bias toward certain scores (e.g., consistently assigning a score of “1”), as they evaluate a single response without the reference effect of multiple responses. Also, when multiple responses (such as two or three) are presented for evaluation at one time, models frequently exhibit a *position bias*, tending to favor responses that appear either at the beginning or at the end.

2.2. Reasoning: a Truth Inference Process

First variant: LLM-PeerReview-Average. After the scoring phase, as shown in Figure 1, each response $\mathbf{r}^{(i,j)}$ corresponding to a query $\mathbf{x}^{(i)}$ receives multiple scores $\{y^{(i,j;j')}\}_{j'=1}^{J}$ provided by multiple LLM judges. Then, how can we aggregate these scores meaningfully to compute a final, reliable score $\hat{t}^{(i,j)}$ for each response $\mathbf{r}^{(i,j)}$? This problem is analogous to designing an algorithm that simulates a senior reviewer who consolidates evaluations from multiple

[1] 1) It is worth noting that each scoring prompt contains both the corresponding query and the response to be evaluated, rather than presenting the response alone; 2) The scoring prompts that we designed and utilized in the experiments are provided in the supplementary materials.

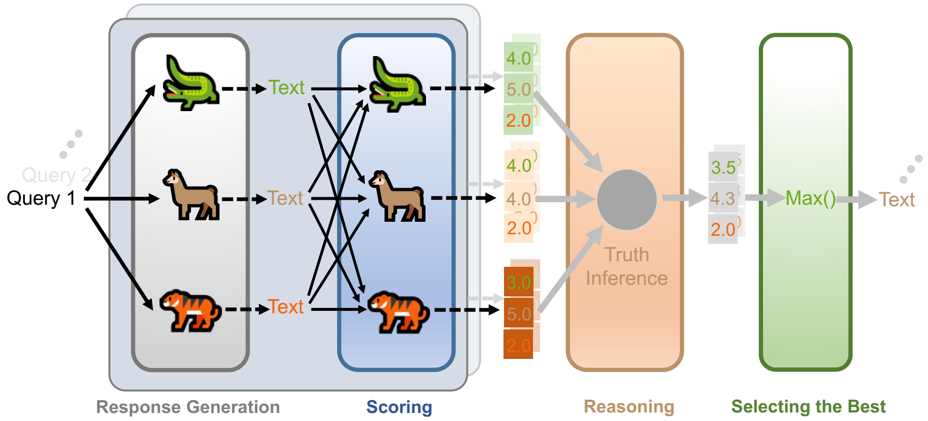


Figure 1: The proposed LLM-PeerReview contains three steps: **(1) Scoring**: For a given query, after each LLM independently generates a response (analogous to a submitted academic paper), LLM-PeerReview applies the LLM-as-a-Judge technique, treating each model as a reviewer to assign scores to all candidate responses; **(2) Reasoning**: LLM-PeerReview then uses a truth inference algorithm—analogous to a senior reviewer—to estimate a final score for each response. Notably, the graphical model-based inference algorithm is performed using score information across all queries, allowing the model to learn each LLM's scoring behavior using global information from the dataset, thereby enabling fine-grained, reliability-aware score aggregation; **(3) Selecting the best:** Finally, for each query, LLM-PeerReview selects the response with the highest final score as the ensemble output—analogous to how a senior reviewer chooses the best paper from a specific submission pool.

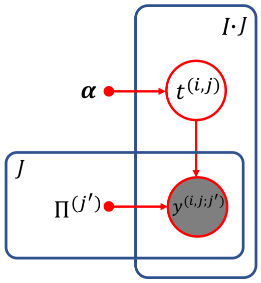


Figure 2: Probabilistic graphical representation.

reviewers with different scoring preferences and evaluation capabilities. A straightforward and intuitive approach is "averaging" (Zheng et al. 2017; Zhang et al. 2021)—simply taking the mean of all the scores for a given response. As a simple variant of our approach, we refer to it as *LLM-PeerReview-Average*.

Second variant: LLM-PeerReview-Weighted. We observe that the above averaging strategy assumes all models to be equally reliable, ignoring the inherent differences in evaluation quality across models. To address this, we propose a weighted variant, referred to as *LLM-PeerReview-Weighted*. Here, we invoke the well-established Dawid-Skene (DS) model (Dawid and Skene 1979), a canonical truth-inference graphical model widely used in weak supervision learning and crowdsourcing(Zheng et al. 2017; Zhang et al. 2021), and adapt it to our context. In the following, we introduce the construction of the graphical model in Section 2.2.1, and present the optimization objective and optimization (to obtain the final score value for each response) in Section 2.2.2.

2.2.1. Graphical Model Overall, to infer the underlying "truth" score (*unobserved*) behind the multiple weak/non-ideal score annotations (*observed*) for each response, we construct a *latent variable graphical model* (Bishop and Nasrabadi 2006; Everett 2013; Chen et al. 2023b) that includes a *latent variable* representing the truth score. As depicted the graphical representation in Figure 2, we next introduce the probabilistic generative process we construct from truth scores to weak scores labeled by LLM judges.

First, for each response $\mathbf{r}^{(i,j)}$, we assume that its true score $t^{(i,j)}$ is drawn from a categorical distribution:

$$t^{(i,j)} \sim \mathrm{Cat}(t^{(i,j)}; \boldsymbol{\alpha}), \tag{1}$$

where the distribution is parametrized by $\boldsymbol{\alpha}$. Next, similar to the concept of *confusion matrix* commonly used in machine learning (Bishop and Nasrabadi 2006; Goodfellow 2016), we introduce an annotator-specific *transition matrix* $\boldsymbol{\Pi}^{(j')}$ to model the probability that an LLM confuses one score category for another, capturing its scoring tendencies and potential biases:

$$p(y^{(i,j;j')} = n | t^{(i,j)} = m; \boldsymbol{\Pi}^{(j')}) = \pi_{mn}^{(j')}, \tag{2}$$

where $m, n \in \{1, \ldots, K\}$ and K denotes the number of categories (i.e., the number of score levels).

2.2.2. Objective and Optimization

Objective. Based on the model construction above, the optimization objective is to maximize the log conditional likelihood of the observed scoring labels $\mathbf{Y} = \{y^{(i,j;j')} \mid 1 \leq i \leq I,\ 1 \leq j \leq J,\ 1 \leq j' \leq J\}$ contributed by J LLM judges, i.e., $\log p(\mathbf{Y}; \Theta)$ w.r.t. the parameters $\Theta = \{\boldsymbol{\alpha}, \boldsymbol{\Pi}^{(1)}, \ldots, \boldsymbol{\Pi}^{(J)}\}$.

Optimization. In brief, as with most latent variable models (Bishop and Nasrabadi 2006; Dawid and Skene 1979), we apply the Expectation-Maximization (EM) algorithm (Dempter 1977) to solve the optimization problem.[2] First, the log-likelihood can be written as:

$$\begin{aligned}\log p(\mathbf{Y};\Theta) &= \sum_{i=1}^{I}\sum_{j=1}^{J}\log p(\mathbf{y}^{(i,j)};\Theta)\\ &\geq \sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t^{(i,j)}} q(t^{(i,j)})\log\frac{p(\mathbf{y}^{(i,j)},t^{(i,j)};\Theta)}{q(t^{(i,j)})},\end{aligned} \tag{3}$$

where $\mathbf{y}^{(i,j)} = \{y^{(i,j;j')} \mid 1 \leq j' \leq J\}$ denotes the set of scores assigned to response $\mathbf{r}^{(i,j)}$ by the J models. The derivation which obtains the *Evidence Lower Bound* (ELBO) $\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t^{(i,j)}} q(t^{(i,j)})\log\frac{p(\mathbf{y}^{(i,j)},t^{(i,j)};\Theta)}{q(t^{(i,j)})}$ used Jensen's inequality (Bishop and Nasrabadi 2006); and $q(t^{(i,j)})$ is a discrete distribution over the variable $t^{(i,j)}$. In the following, we then proceed to apply the general EM recipe to perform iterative calculations (concerning E-step and M-step) to solve the optimization problem $\Theta := \underset{\Theta}{\operatorname{argmax}} \log p(\mathbf{Y};\Theta)$.

E-step (inference): $q(t^{(i,j)}=k) := p(t^{(i,j)}=k \mid \mathbf{y}^{(i,j)};\Theta)$

$$\propto p(t^{(i,j)}=k;\boldsymbol{\alpha})\cdot\prod_{j'=1}^{J} p(y^{(i,j;j')} \mid t^{(i,j)}=k;\mathbf{\Pi}^{(j')}). \tag{4}$$

The posterior $q(t^{(i,j)})$ is obtained by using of Bayes's theorem given the parameters $\Theta = \{\boldsymbol{\alpha},\mathbf{\Pi}^{(1)},\ldots,\mathbf{\Pi}^{(J)}\}$ learned on the last M-step. Given that we are likely to obtain decimal values rather than integers on $y^{(i,j;j')}$ after the scoring phase, we make the adaptation:

$$\begin{aligned}q(t^{(i,j)}=k) \propto\,& p(t^{(i,j)}=k;\boldsymbol{\alpha})\cdot\prod_{j'=1}^{J}[\phi_l\cdot p(y_l^{(i,j;j')} \mid t^{(i,j)}=k;\mathbf{\Pi}^{(j')})\\ &+\phi_u\cdot p(y_u^{(i,j;j')} \mid t^{(i,j)}=k;\mathbf{\Pi}^{(j')})],\end{aligned} \tag{5}$$

where ϕ_l and ϕ_u represent the confidences for the decimal $y^{(i,j;j')}$ corresponding to its lower and upper nearest integer neighbors (i.e., $y_l^{(i,j;j')}$, $y_u^{(i,j;j')}$).

M-step (learning):

$$\begin{aligned}\Theta :=& \underset{\Theta}{\operatorname{argmax}}\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{t^{(i,j)}} q(t^{(i,j)})\log\frac{p(\mathbf{y}^{(i,j)},t^{(i,j)};\Theta)}{q(t^{(i,j)})}\\ :=& \underset{\Theta}{\operatorname{argmax}}\sum_{i=1}^{I}\sum_{j=1}^{J}[\mathbb{E}_{q(t^{(i,j)})}\log p(t^{(i,j)};\boldsymbol{\alpha})+\\ &\mathbb{E}_{q(t^{(i,j)})}\log\prod_{j'=1}^{J}p(y^{(i,j;j')} \mid t^{(i,j)};\mathbf{\Pi}^{(j')})]\end{aligned} \tag{6}$$

Algorithm 1: LLM-PeerReview

```
Input: Data $\mathcal{D} = \{\mathbf{x}^{(i)}, \mathbf{R}^{(i)}\}_{i=1}^{I}$, where for each query $\mathbf{x}^{(i)}$, we have responses $\mathbf{R}^{(i)} = \{\mathbf{r}^{(i,j)} \mid 1 \leq j \leq J\}$ from heterogeneous LLMs $\{\mathcal{M}_j\}_{j=1}^{J}$
Output: results $\{\mathbf{r}^{(i)}_{\text{ensemble}}\}_{i=1}^{I}$
# Scoring:
Each LLM $\mathcal{M}_{j'}$ acts as a judge and assigns a score $y^{(i,j;j')}$ to each response $\mathbf{r}^{(i,j)}$
# Reasoning:
# Reasoning the posterior probabilities $q(t^{(i,j)})$ over score categories for each response:
Initialize posterior $\{q(t^{(i,j)}) \mid 1 \leq i \leq I, 1 \leq j \leq J\}$ by averaging scores on $\mathbf{r}^{(i,j)}$
while not converge do
    Update $\boldsymbol{\alpha} = \{\alpha^{(k)} \mid 1 \leq k \leq K\}$ by using Eq. 7
    Update $\{\mathbf{\Pi}^{(j')}\}_{j'=1}^{J}$ by using Eq. 8
    Update the posterior $\{q(t^{(i,j)}) \mid 1 \leq i \leq I, 1 \leq j \leq J\}$ by using Eq. 5
end while
# Obtain the final score value for each response:
Obtain the final score value $S(t^{(i,j)})$ for each individual response $\mathbf{r}^{(i,j)}$ by using Eq. 9
# Selecting the best:
Obtain the final results $\{\mathbf{r}^{(i)}_{\text{ensemble}}\}_{i=1}^{I}$ by using Eq. 10
```

Furthermore, by maximizing optimization objective in Equation 5 and using the standard Lagrange multiplier method (Bishop and Nasrabadi 2006), we can obtain the closed-form solution for $\boldsymbol{\alpha} = \{\alpha^{(k)} \mid 1 \leq k \leq K\}$ in Equation 7 shown below; similar to Equation 5 and by equating the gradient of Equation 5 to zero, we can obtain the closed-form solution for $\{\mathbf{\Pi}^{(j')}\}_{j'=1}^{J}$ in Equation 8 shown below.

$$\alpha_k = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} q(t^{(i,j)}=k)}{I\cdot J}, \tag{7}$$

$$\pi_{mn}^{(j')} = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J} q(t^{(i,j)}=m)\cdot\Psi(y^{(i,j;j')},n)}{\sum_{i=1}^{I}\sum_{j=1}^{J} q(t^{(i,j)}=m)}, \tag{8}$$

where $\Psi(y^{(i,j;j')}) = [\phi_l\cdot\mathbb{I}(y_l^{(i,j;j')}=n)+\phi_u\cdot\mathbb{I}(y_u^{(i,j;j')}=n]$, and $\mathbb{I}(\cdot)$ is an indicator function that takes the value 1 when the internal declaration is true, and 0 otherwise.

Obtain the final score value for each response. After the EM-based optimization, we obtain the posterior probabilities $q(t^{(i,j)})$ over score categories for each response.

Then, we compute the final score for each response through the following simple summation:

$$S(t^{(i,j)}) = \sum_{k=1}^{K} q(t^{(i,j)}=k)\cdot s_k, \tag{9}$$

where s_k denotes the score value corresponding to the k-th scoring category.[3]

[2] We provide the detailed derivations in Appendix.

[3] For example, suppose that for a given response, we have

Types	Methods	TriviaQA↑	GSM8k↑	MATH↑	AlpacaEval↑	Average↑
Single LLM	Llama-3.1-8B-Instruct	75.3	79.3	52.3	7.3	53.5
	Mistral-7B-Instruct	72.7	64.3	26.5	10.4	43.5
	Qwen2-7B-Instruct	63.0	88.5	59.8	15.2	56.6
	Qwen2.5-7B-Instruct	62.5	91.5	69.3	27.6	62.7
	Theoretical average	68.4	80.9	51.9	15.1	54.1
LLM Ensemble	Random	68.4 ± 0.3	81.2 ± 1.2	52.2 ± 1.1	15.2 ± 0.6	54.2
	Smoothie-Global	63.0	91.5	59.8	27.6	60.5
	Smoothie-Local	73.6	85.5	61.8	18.3	59.8
	Agent-Forest	70.5	86.8	61.0	22.1	60.1
	LLM-PeerReview-Average	76.9 ± 0.1	92.7 ± 0.3	69.5 ± 0.2	**30.4** ± 0.1	67.4
	LLM-PeerReview-Weighted	**77.0** ± 0.1	**93.0** ± 0.2	**71.0** ± 0.2	30.2 ± 0.1	**67.8**
Our variants	Llama-3-8B-Selection	76.5 ± 0.2	90.8 ± 0.6	68.8 ± 0.5	29.6 ± 0.3	66.4
	Mistral-7B-Selection	75.6 ± 0.3	90.8 ± 0.1	66.4 ± 0.3	25.9 ± 0.4	64.7
	Qwen2-7B-Selection	74.2 ± 0.2	88.8 ± 0.6	61.7 ± 0.7	23.7 ± 0.3	62.1
	Qwen2-7B-Selection	75.5 ± 0.2	92.1 ± 0.4	66.2 ± 0.6	28.1 ± 0.1	65.5

Table 1: Main results (%).

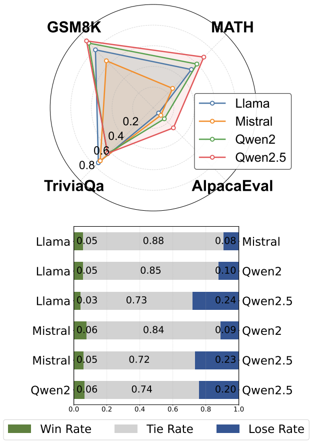


Figure 3: LLM performances (bottom: dataset MATH).

2.3. Selecting the Best

Finally, for each query $\mathbf{x}^{(i,j)}$, we can easily determine its optimal response:

$$\mathbf{r}^{(i)}_{\text{ensemble}} = \underset{\mathbf{r}^{(i,j)}}{\operatorname{argmax}} \{S(t^{(i,j)}) \mid 1 \leq j \leq J\}, \qquad (10)$$

which is selected as the final result after ensemble.

The overall procedure for our proposed LLM-PeerReview is summarized in Algorithm 1.

3. Experiments

3.1. Setup

We provide more details on the setup and the code in the supplementary materials. [4]

Datasets and evaluation. We evaluate four widely-used datasets, grouped into three categories: **(1) Factual Recall:** TriviaQA (Joshi et al. 2017; Guha et al. 2024) evaluates the *accuracy* of model responses to factual questions across various domains, including history, science, and geography. **(2) Arithmetic Reasoning:** GSM8k (Chen et al. 2021a; Hu et al. 2024) and MATH (Hendrycks et al. 2021; Hu et al. 2024) assess basic arithmetic and more advanced mathematical reasoning, respectively, with *accuracy* as the evaluation metric, focusing on correct numerical answers. **(3) Instruction Following:** AlpacaEval (Dubois et al. 2023; Hu et al. 2024) tests models' ability to follow various instructions. We use GPT-4o-mini to evaluate the *accuracy* of model responses, assessing whether the model's response exceeds the reference answer in the dataset.

Seed LLMs for ensemble. Considering 7B-scale models are widely used by researchers and generally regarded as having acceptable judging capabilities (Wang, Zhang, and Choi 2025; Kim et al. 2024), we use these well-established 7B models for ensemble: Llama-3.1-8B-Instruct, Mistral-7B-Instruct, Qwen2-7B-Instruct, and Qwen2.5-7B-Instruct.

Baselines. We compare the proposed LLM-PeerReview with the two categories of baselines. **(1) Single LLMs:** The four 7B-scale models mentioned before. **(2) LLM Ensemble baselines:** (*i*) Random (Lu et al. 2024) is a random-selection baseline that simply returns the response from a randomly chosen LLM in the ensemble. As one of the simplest ensemble strategies for large language models, this method has previously been applied to dialogue tasks (Lu et al. 2024); (*ii*) Smoothie-Global (Guha et al. 2024), Smoothie-Local (Guha et al. 2024), and Agent-Forest (Li et al. 2024c) are recently proposed, strong similarity-based ensemble methods, as introduced in detail in Section 1.

Configurations. (1) For each individual large language model, we follow the setup of Smoothie (Guha et al. 2024), where the model responds once to each query. The responses from all models are stored for integration by the LLM Ensemble methods. [5] (2) For the two variants of the baseline Smoothie (Guha et al. 2024), we set the number of neighbors as specified in the original paper. Agent-Forest (Li et al. 2024c) does not require any hyperparameter configuration. For our method, we set the model temperature to 0 during the scoring process to eliminate suboptimal results caused by randomness. Additionally, the scoring prompts used across the four datasets are provided in the Appendix. (3) All experiments were performed using 6 or 4 parallel Nvidia V100 32GB GPUs. All experiments with stochastic outputs were conducted three times.

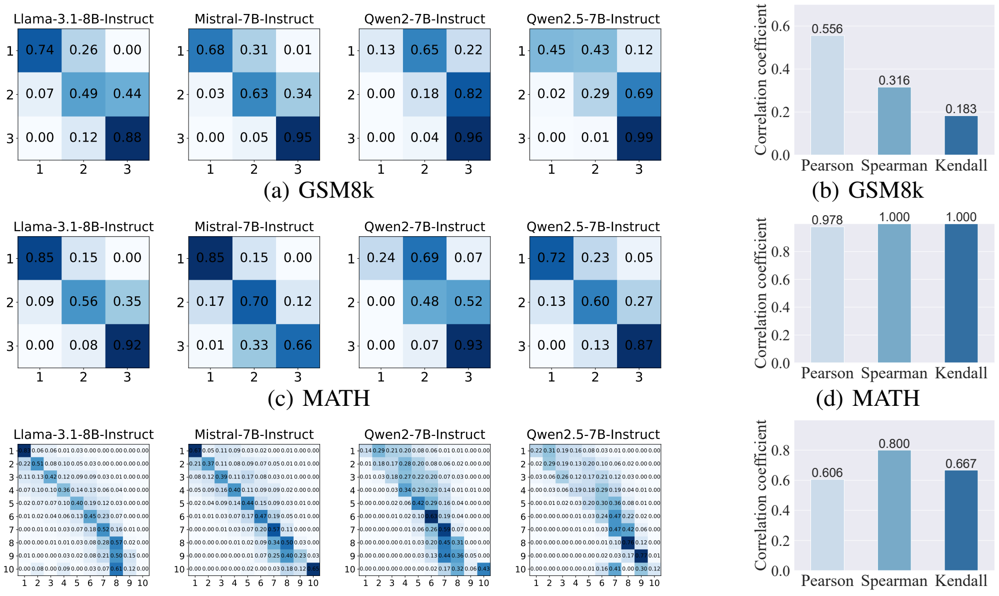
3.2. Main Results

The ensemble of the proposed LLM-PeerReview is effective. The main results are shown in Table 1. First, by examining the results in the "Single LLM" and "LLM Ensemble"

$q(t^{(i,j)} = 4) = 0.5$, $q(t^{(i,j)} = 5) = 0.5$, along with the score values $s_4 = 4.0$, $s_5 = 5.0$. Then, the final aggregated score is $S(t^{(i,j)}) = 0.5 \cdot 4.0 + 0.5 \cdot 5.0 = 4.5$.

[4] Also, the code for all datasets will be made open-source.

[5] All LLM responses will also be open-sourced to promote reproducibility and further research.



(a) GSM8k (b) GSM8k

(c) MATH (d) MATH

(e) AlpacaEval (f) AlpacaEval

Figure 4: **Left:** The transition matrix of each LLM estimated by LLM-PeerReview-Weighted. **Right:** The correlation coefficients between the sum of the diagonal elements of the transition matrix and its performance as a single judge (corresponding to ”our variants” in Table 1).

sections of Table 1, one key finding is that both of our variants consistently outperform any single LLM and all LLM Ensemble baselines across all datasets. In the last column, which presents the average performance, our two variant methods (with results of 67.4% and 67.8%) surpass the strongest single model, Qwen2.5, by 4.7% and 5.1%, respectively, and outperform the strongest ensemble method, Smoothie-Global, by 6.9% and 7.3%. These results directly demonstrate the effectiveness of our method, as it achieves superior performance by integrating the collective knowledge of multiple models across factual-recall QA tasks, mathematical reasoning tasks, and instruction-following tasks. Additionally, the ensemble task across these four datasets is challenging, as the performance of the four LLMs varies significantly for each dataset. In contrast, ensembling four LLMs with similar performance would make it easier to achieve superior results compared to any single LLM.

Each LLM has its strengths and weaknesses. In Figure 3, the upper subplot presents a radar chart of individual LLM performance, while the lower subplot displays the win-tie-loss chart for models on the challenging instruction-following dataset, AlpacaEval. This chart highlights that models with the best overall performance may underperform on specific tasks compared to those with weaker overall results. In summary, the results in Table 1 and Figure 3 demonstrate that a strong LLM does not excel across all datasets. Each model has its strengths and weaknesses, highlighting the substantial practical significance of LLM Ensemble.

3.3. Significance of Aggregating Multiple Judges

Simply averaging the scores from multiple judges is also quite effective. In the "Our variants" of Table 1, we present the performance of using a single LLM as a judge to select the optimal response. From the average performances in the last column of Table 1, we observe that these variants perform quite well (surpassing the overall best model, Qwen2.5, in 3/4 cases). However, when comparing the performance of these variants with that of our prototype LLM-PeerReview-Average, it becomes clear that aggregating and averaging the scores from multiple judges is highly beneficial, compared to relying solely on the score of a single large model.

The weighted truth inference has the potential for further performance improvement. By observing the average results in Table 1, we find that LLM-PeerReview-Weighted leads to further performance gains compared to simple averaging. In the left subplot of Figure 4, we observe subtle variations in the transition matrices learned for each model. On the other hand, the right subplot in Figure 4, displaying positive correlation coefficients, demonstrates that our method can effectively identifies stronger and weaker judges.

3.4. Alternative Designs and Related Analyses

The flipped-triple scoring trick represents a performance-efficiency trade-off. As introduced in Section 2.1, it is intuitive that, in addition to the our recommended *flipped-triple scoring* method, several variant scoring methods could be employed (their definitions are provided in the caption of Table 2). Overall, the performance of these four variants follows the order: *quadruple-half* > *flipped-triple* > *double* > *single*. Variants quadruple-half, flipped-triple, and double all offer noticeable *de-biasing* performance advantages over the single-scoring strategy. On the other hand, in terms of theoretical computational complexity, the complexities for de-biased strategies double/flipped-triple/quadruple-half are

Method	TriviaQA	GSM8k	MATH	AlpacaEval	Average
Variant Performance (↑):					
Random	65.7± 1.3	81.3± 0.6	51.2± 1.8	14.7± 0.5	53.2
Single	69.2± 0.6	85.5± 2.1	60.3± 1.6	23.8± 1.0	59.7
Double	73.3± 0.5	90.0± 0.7	71.3± 0.2	29.3± 0.2	66.0
Flipped-triple	74.5± 0.0	90.8± 0.2	71.5± 0.4	30.5± 0.0	66.8
Quadruple-half	74.7± 0.2	91.5 ± 0.4	73.3± 0.2	29.2± 0.2	67.2
Computation Efficiency (seconds/scoring the 4 responses for each sample; ↓):					
Single ($\mathcal{O}(J)$)	7.89	10.2	10.6	16.9	11.4
Double ($\mathcal{O}(J^2)$)	37.1	49.4	51.6	77.4	53.9
Flipped-triple ($\mathcal{O}(J)$)	29.7	43.4	47.1	74.3	48.6
Quadruple-half ($\mathcal{O}(J!)$)	51.3	83.8	90.0	137.65	90.7

Table 2: **Top:** Performance of the proposed method with different scoring strategies. Note that, for computational efficiency, we use 200 samples from each dataset (all shuffled as necessary). (*i*) Random refers to the same baseline in Table 1; (*ii*) *Single* refers to scoring a single response at a time; (*iii*) *Double* refers to scoring all *response pairs* within the response set at a time; (*iv*) *Quadruple-half* refers to scoring all possible *response quadruple sequences* in the response set; given the high computational cost, we used a relaxed version and only calculated half of the cases. **Bottom:** Computation efficiency.

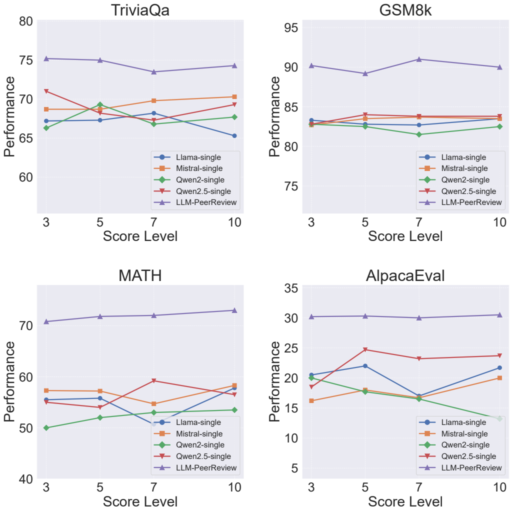


Figure 5: Performance of various variants across different scoring levels.

$\mathcal{O}(J)$/$\mathcal{O}(J^2)$/$\mathcal{O}(J)$/$\mathcal{O}(J!)$, with strategy flipped-triple having the lowest computational complexity. Furthermore, in Table 3, we present the scoring efficiency of these four strategies. Compared to strategies double and quadruple-half, strategy flipped-triple is the most time-efficient.

Common scoring levels can generally be attempted. In Figure 5, we conduct a further analysis of how different scoring levels influence the performance of both our method and the four individual scoring models.[6] For each scoring level, we have carefully crafted meaningful descriptions and corresponding prompts. Under these conditions, our method exhibits slightly varying performance, showing no consistent tendencies across the levels of 3, 5, 7, and 10.

4. Related Work

LLM Ensemble, as outlined in Section 1, can be broadly categorized into three approaches (Chen et al. 2025). The first category, *ensemble-before-inference* approach (Shnitzer et al. 2023; Srivatsa, Maurya, and Kochmar 2024; Ong et al. 2024), typically necessitates custom-labeled data to pretrain a classifier that routes each query. The mandatory pretraining phase and the dependency on labeled data are the primary inconveniences of these methods. The second category, *ensemble-during-inference* approach, can be subdivided into *token-level*(Yu et al. 2024; Huang et al. 2024; Xu, Lu, and Zhang 2024), *span-level*(Liu et al. 2024b; Xu et al. 2025), and *process-level*(Park et al. 2024) methods, depending on the level of granularity of the information considered in the ensemble. These methods, however, entail substantial computational costs and require the local deployment of every LLM in the ensemble. Lastly, as mentioned in Section 1, *ensemble-after-inference* approach mainly includes selection-then-regeneration-based(Lu et al. 2024) and aggregation-based methods (Li et al. 2024c; Guha et al. 2024).

Weak Supervision (Zhang et al. 2021; Chen et al. 2023b), also commonly referred to as **Learning from Crowds** (Chen et al. 2021b, 2022), is a research problem that closely resembles post-inference ensemble methods. The key distinction is that Weak Supervision methods either focus on learning classifiers directly from imperfectly labeled data (Zhang et al. 2021) or on aggregating weak label information (Zheng et al. 2017; Chen et al. 2022), whereas ensemble-after-inference LLM Ensemble methods are exclusively concerned with aggregation and do not involve the learning of classifier models. Additionally, a significant difference is that most weakly supervised learning methods are primarily focused on classification scenarios with closed answer sets, rather than text generation tasks that involve open-ended output spaces.

LLM-as-a-Judge approaches has received a lot of attention recently. As task complexity increases and model outputs become more diverse, traditional evaluation methods—such as matching-based or embedding-based metrics—often fail to capture subtle attributes and provide reliable results (Gu et al. 2024). The recent emergence of large language models has led to the development of the "LLM-as-a-judge" paradigm, where LLMs assess the quality of model outputs. These methods can be broadly classified into three categories: Single-LLM-based, Multi-LLM-based, and Human-AI collaboration-based evaluation approaches. Single-LLM-based methods primarily focus on prompt design (Fu et al. 2023; Kotonya et al. 2023), fine-tuning (Chen et al. 2023a; Wu et al. 2024), or post-processing (Daynauth and Mars 2024). Notably, Multi-LLM-based methods, which include collaborative (Zhang et al. 2023), competitive (Owens et al. 2024), and aggregation-based (Verga et al. 2024; Chen et al. 2024; Chu et al. 2024) strategies, are closely related to our approach (especially for aggregation methods).

[6] As indicated by the prompts in the Appendix, for the main experiment in Table 1, the scoring levels applied across the four datasets were 5, 3, 3, and 10, respectively.

5. Conclusion

This paper presents LLM-PeerReview, the first unsupervised peer-review-inspired method, to address the LLM ensemble problem. PeerReview benefits both from the analytical capabilities of the powerful large language models at hand when evaluating response quality, and from the principled, fine-grained score aggregation enabled by the canonical graphical model that considers model-wise quality variation when inferring final scores. LLM-PeerReview—embedded with the well-established techniques of LLM-as-a-Judge and latent variable graphical models—closely emulates the real-world human process of selecting the best text, offering a clear and interpretable mechanism. Our empirical evaluations on four datatests demonstrate that LLM-PeerReview significantly improves the recent advanced model Smoothie-Global and provides a new solution to LLM Ensemble.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv:2303.08774*.

Bishop, C. M.; and Nasrabadi, N. M. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Chen, J.; Su, W.; Chu, Z.; Li, H.; Ai, Q.; Liu, Y.; Zhang, M.; and Ma, S. 2024. An Automatic and Cost-Efficient Peer-Review Framework for Language Generation Evaluation. *arXiv preprint arXiv:2410.12265*.

Chen, J.; Yoon, J.; Ebrahimi, S.; Arik, S. O.; Pfister, T.; and Jha, S. 2023a. Adaptation with self-evaluation to improve selective prediction in llms. *arXiv preprint arXiv:2310.11689*.

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H. P. D. O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Chen, P.; Sun, H.; Yang, Y.; and Chen, Z. 2022. Adversarial learning from crowds. In *AAAI*.

Chen, Z.; Li, J.; Chen, P.; Li, Z.; Sun, K.; Luo, Y.; Mao, Q.; Yang, D.; Sun, H.; and Yu, P. S. 2025. Harnessing Multiple Large Language Models: A Survey on LLM Ensemble. *arXiv preprint arXiv:2502.18036*.

Chen, Z.; Sun, H.; Zhang, W.; Xu, C.; Mao, Q.; and Chen, P. 2023b. Neural-hidden-crf: A robust weakly-supervised sequence labeler. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 274–285.

Chen, Z.; Wang, H.; Sun, H.; Chen, P.; Han, T.; Liu, X.; and Yang, J. 2021b. Structured probabilistic end-to-end learning from crowds. In *IJCAI*.

Chu, Z.; Ai, Q.; Tu, Y.; Li, H.; and Liu, Y. 2024. Pre: A peer review based large language model evaluator. *arXiv preprint arXiv:2401.15641*.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Dawid, A. P.; and Skene, A. M. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1): 20–28.

Daynauth, R.; and Mars, J. 2024. Aligning Model Evaluations with Human Preferences: Mitigating Token Count Bias in Language Model Assessments. *arXiv preprint arXiv:2407.12847*.

Dempter, A. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society*, 39: 1–22.

Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; and Ma, Q. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14: 241–258.

Dubois, Y.; Li, C. X.; Taori, R.; Zhang, T.; Gulrajani, I.; Ba, J.; Guestrin, C.; Liang, P. S.; and Hashimoto, T. B. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36: 30039–30069.

Everett, B. 2013. *An introduction to latent variable models*. Springer Science & Business Media.

Fu, J.; Ng, S.-K.; Jiang, Z.; and Liu, P. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*.

Goodfellow, I. 2016. Deep learning.

Gu, J.; Jiang, X.; Shi, Z.; Tan, H.; Zhai, X.; Xu, C.; Li, W.; Shen, Y.; Ma, S.; Liu, H.; et al. 2024. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*.

Guha, N.; Chen, M.; Chow, T.; Khare, I.; and Re, C. 2024. Smoothie: Label free language model routing. *Advances in Neural Information Processing Systems*, 37: 127645–127672.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hu, Z.; Zhang, J.; Xiong, Z.; Ratner, A.; Xiong, H.; and Krishna, R. 2024. Language model preference evaluation with multiple weak evaluators. *arXiv preprint arXiv:2410.12869*.

Huang, Y.; Feng, X.; Li, B.; Xiang, Y.; Wang, H.; Liu, T.; and Qin, B. 2024. Ensemble Learning for Heterogeneous Large Language Models with Deep Parallel Collaboration. In *NeurIPS*.

Jiang, D.; Ren, X.; and Lin, B. Y. 2023. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Joshi, M.; Choi, E.; Weld, D. S.; and Zettlemoyer, L. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

Kim, S.; Suk, J.; Longpre, S.; Lin, B. Y.; Shin, J.; Welleck, S.; Neubig, G.; Lee, M.; Lee, K.; and Seo, M. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535*.

Kotonya, N.; Krishnasamy, S.; Tetreault, J.; and Jaimes, A. 2023. Little giants: Exploring the potential of small llms as evaluation metrics in summarization in the eval4nlp 2023 shared task. *arXiv preprint arXiv:2311.00686*.

Li, D.; Jiang, B.; Huang, L.; Beigi, A.; Zhao, C.; Tan, Z.; Bhattacharjee, A.; Jiang, Y.; Chen, C.; Wu, T.; et al. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Li, H.; Dong, Q.; Chen, J.; Su, H.; Zhou, Y.; Ai, Q.; Ye, Z.; and Liu, Y. 2024b. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.

Li, J.; Zhang, Q.; Yu, Y.; Fu, Q.; and Ye, D. 2024c. More agents is all you need. *arXiv preprint arXiv:2402.05120*.

Liu, A.; Feng, B.; Wang, B.; Wang, B.; Liu, B.; Zhao, C.; Dengr, C.; Ruan, C.; Dai, D.; Guo, D.; et al. 2024a. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*.

Liu, C.; Quan, X.; Pan, Y.; Lin, L.; Wu, W.; and Chen, X. 2024b. Cool-fusion: Fuse large language models without training. *arXiv preprint arXiv:2407.19807*.

Lu, X.; Liu, Z.; Liusie, A.; Raina, V.; Mudupalli, V.; Zhang, Y.; and Beauchamp, W. 2024. Blending is all you need: Cheaper, better alternative to trillion-parameters llm. *arXiv preprint arXiv:2401.02994*.

Lv, B.; Tang, C.; Zhang, Y.; Liu, X.; Luo, P.; and Yu, Y. 2024. URG: A Unified Ranking and Generation Method for Ensembling Language Models. In *Findings of the ACL*.

Ong, I.; Almahairi, A.; Wu, V.; Chiang, W.-L.; Wu, T.; Gonzalez, J. E.; Kadous, M. W.; and Stoica, I. 2024. Routellm: Learning to route llms with preference data. *arXiv preprint arXiv:2406.18665*.

Owens, D. M.; Rossi, R. A.; Kim, S.; Yu, T.; Dernoncourt, F.; Chen, X.; Zhang, R.; Gu, J.; Deilamsalehy, H.; and Lipka, N. 2024. A multi-llm debiasing framework. *arXiv preprint arXiv:2409.13884*.

Park, S.; Liu, X.; Gong, Y.; and Choi, E. 2024. Ensembling Large Language Models with Process Reward-Guided Tree Search for Better Complex Reasoning. *arXiv preprint arXiv:2412.15797*.

Shnitzer, T.; Ou, A.; Silva, M.; Soule, K.; Sun, Y.; Solomon, J.; Thompson, N.; and Yurochkin, M. 2023. Large language model routing with benchmark datasets. *arXiv preprint arXiv:2309.15789*.

Si, C.; Shi, W.; Zhao, C.; Zettlemoyer, L.; and Boyd-Graber, J. 2023. Getting MoRE out of Mixture of Language Model Reasoning Experts. In *Findings of EMNLP*.

Srivatsa, K.; Maurya, K. K.; and Kochmar, E. 2024. Harnessing the Power of Multiple Minds: Lessons Learned from LLM Routing. *arXiv preprint arXiv:2405.00467*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Tekin, S.; Ilhan, F.; Huang, T.; Hu, S.; and Liu, L. 2024. LLM-TOPLA: Efficient LLM Ensemble by Maximising Diversity. In *Findings of EMNLP*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Verga, P.; Hofstatter, S.; Althammer, S.; Su, Y.; Piktus, A.; Arkhangorodsky, A.; Xu, M.; White, N.; and Lewis, P. 2024. Replacing judges with juries: Evaluating llm generations with a panel of diverse models. *arXiv preprint arXiv:2404.18796*.

Wang, P.; Li, L.; Chen, L.; Cai, Z.; Zhu, D.; Lin, B.; Cao, Y.; Liu, Q.; Liu, T.; and Sui, Z. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Wang, V.; Zhang, M. J.; and Choi, E. 2025. Improving llm-as-a-judge inference with the judgment distribution. *arXiv preprint arXiv:2503.03064*.

Wu, T.; Yuan, W.; Golovneva, O.; Xu, J.; Tian, Y.; Jiao, J.; Weston, J.; and Sukhbaatar, S. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*.

Xu, Y.; Chen, J.; Wu, J.; and Zhang, J. 2025. Hit the Sweet Spot! Span-Level Ensemble for Large Language Models. In *COLING*, 8314–8325.

Xu, Y.; Lu, J.; and Zhang, J. 2024. Bridging the Gap between Different Vocabularies for LLM Ensemble. In *NAACL*, 7133–7145.

Yu, Y.-C.; Kuo, C.-C.; Ye, Z.; Chang, Y.-C.; and Li, Y.-S. 2024. Breaking the Ceiling of the LLM Community by Treating Token Generation as a Classification for Ensembling. *arXiv preprint arXiv:2406.12585*.

Zhang, J.; Yu, Y.; Li, Y.; Wang, Y.; Yang, Y.; Yang, M.; and Ratner, A. 2021. WRENCH: A comprehensive benchmark for weak supervision. *arXiv preprint arXiv:2109.11377*.

Zhang, X.; Yu, B.; Yu, H.; Lv, Y.; Liu, T.; Huang, F.; Xu, H.; and Li, Y. 2023. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862*.

Zheng, L.; Chiang, W.-L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.; et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36: 46595–46623.

Zheng, Y.; Li, G.; Li, Y.; Shan, C.; and Cheng, R. 2017. Truth inference in crowdsourcing: Is the problem solved? *Proceedings of the VLDB Endowment*, 10(5): 541–552.