PICO: PEER REVIEW IN LLMS BASED ON CONSISTENCY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Existing large language models (LLMs) evaluation methods typically focus on testing the performance on some closed-environment and domain-specific benchmarks with human annotations. In this paper, we explore a novel **unsupervised evaluation direction**, utilizing *peer-review* mechanisms to measure LLMs automatically without any human feedback. In this setting, both open-source and closed-source LLMs lie in the same environment, capable of answering unlabeled questions and evaluating each other, where each LLMs response score is jointly determined by other anonymous ones. During this process, we found that those answers that are more recognized by other "reviewers" (models) usually come from LLMs with stronger abilities, while these models can also evaluate others' answers more accurately. We formalize it as a *consistency assumption*, *i.e.*, the ability and score of the model usually have consistency. We exploit this to optimize each model's confidence, thereby re-ranking the LLMs to be closer to human rankings. We perform experiments on multiple datasets with standard rank-based metrics, validating the effectiveness of the proposed approach.

1 INTRODUCTION

Goodhart's Law: *"When a measure becomes a target, it ceases to be a good measure."*

Large language models (LLMs) [11; 2; 12; 45] have achieved remarkable success across a variety of real-world applications [56; 34; 38; 54]. With the increasingly widespread application of these models, there is an urgent need for an effective evaluation method to ensure that their performance and usability meet the growing demands. To assess the ability level of LLMs, a large number of evaluation benchmarks have been proposed by using some small and domain-specific datasets with human-curated labels, such as MMLU [26], HELM [32], Big-Bench [41], GLUE [46]. However, these benchmarks can only measure LLMs' core capability on a confined set of tasks (e.g. multi-choice knowledge or retrieval questions), which fails to assess their alignment with human preference in open-ended tasks adequately [16; 30; 36]. On the other hand, these evaluations may suffer from *benchmark leakage* issue, referring that the evaluation data is unknowingly used for model training, which can also lead to misleading evaluations [51; 58]. Therefore, blindly improving scores on these public benchmarks cannot always yield a large language model that truly satisfies human requirements.

For assessing human preferences, recent studies have focused on building crowdsourced battle platforms with human ratings as the primary evaluation metric. Typical platforms include Chatbot Arena [57], MT-Bench [57], and AlpacaEval [31]. It constructs anonymous battles between chatbots in real-world scenarios, where users engage in conversations with two chatbots at the same time and rate their responses based on personal preferences. While human evaluation is the gold standard for measuring human preferences, it is exceptionally slow and costly [57]. In addition, adding a new LLM to the crowdsourced battle platforms also poses a cold-start issue [15]. Thus, a fundamental question arises: *can we construct an unsupervised LLMs evaluation system without relying on any human feedback*?

Actually, in real human evaluation systems, people build the human-ability hierarchy based on different empirical assumptions. For example, majority voting [22; 10; 42] and rating voting [5] methods

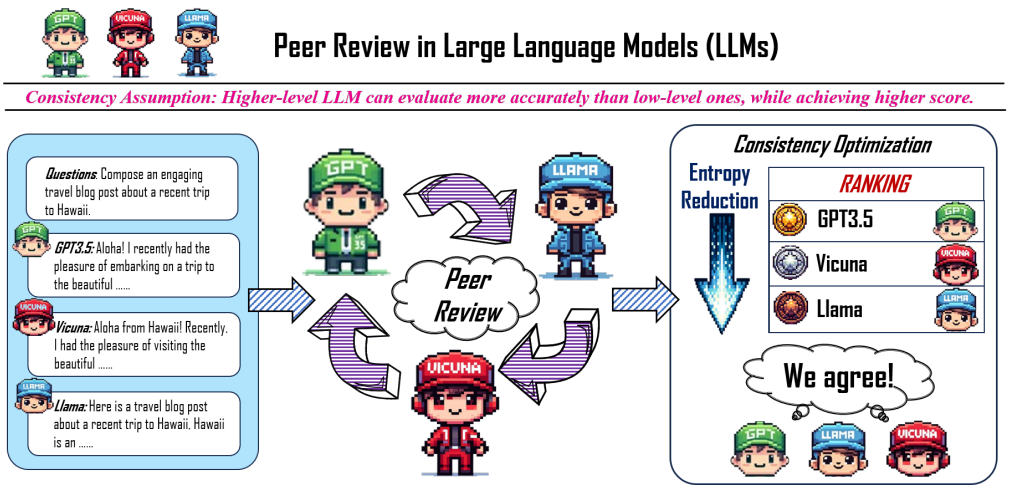


Figure 1: The framework of PiCO. In this framework, both open-source and closed-source LLMs lie in the same environment, capable of answering unlabeled questions and evaluating each other, where each LLM's response score is jointly determined by other anonymous ones. We assign each LLM a learnable capability weight to optimize the score ranking based on the *consistency assumption*, while reducing the entropy of the *peer-review* evaluation system. The goal is to find a final score ranking that all LLMs "agree" it.

are widely used during the decision-making process, which are based on the wisdom of the crowds [42; 13; 52] and have been proven to lead to better results than that of an individual. Moreover, in the established practice of *peer-review* in academic research, scholars evaluate their academic level rankings based on the *consistency assumption*, *i.e.*, scholars with stronger abilities usually have stronger persuasiveness for evaluating others, and these scholars can also obtain higher achievements. This paper attempts to explore whether a similar phenomenon exists in the LLMs evaluation systems.

In this paper, we propose **PiCO**, a **P**eer review approach **i**n LLMs based on **C**onsistency **O**ptimization. In this setting, LLMs themselves act as "reviewers", engaging in mutual assessments to achieve comprehensive, efficient, and performance evaluations without relying on manually annotated data. This method aims to address the limitations of existing evaluation approaches and provide insights into LLMs' real-world capabilities. As shown in Figure 1, both open-source and closed-source LLMs lie in the same environment and answer the open-ended questions from an unlabeled dataset. Then, we construct anonymous answer pairs, while randomly selecting other LLMs as "reviewers" to evaluate both responses with a learnable confidence weight w. Finally, we employ this weight and calculate the response scores G for each LLM based on the weighted joint evaluation. It is worth noting that the whole *peer-review* process works in an unsupervised way, and our goal is to optimize the confidence weights w that re-rank the LLMs to be closer to human rankings.

To achieve this, we formalize it as a constrained optimization based on the consistency assumption. We maximize the consistency of each LLM's capability w and score G while adjusting the final ranking to align with human preference more closely. The key assumption behind this is that high-level LLM can evaluate others' answers more accurately (confidence) than low-level ones, while higher-level LLM can also achieve higher answer-ranking scores. As a result, the entropy (controversy) of the whole *peer-review* evaluation system can be minimized. In other words, the consistency optimization aims to find a final score ranking that all LLMs have no "disputes" regarding.

We perform experiments on multiple crowdsourcing datasets with standard rank-based metrics, the results demonstrate that the proposed PiCO framework can effectively obtain a large language models' leaderboard closer to human preferences. The contributions of this paper can be summarized as follows:

- We explore a novel unsupervised LLM evaluation direction without human feedback, *i.e.*, utilizing *peer-review* mechanisms to measure LLMs automatically. All LLMs can answer unlabeled questions and evaluate each other.
- A constrained optimization based on the consistency assumption is proposed to re-rank the LLMs to be closer to human rankings.

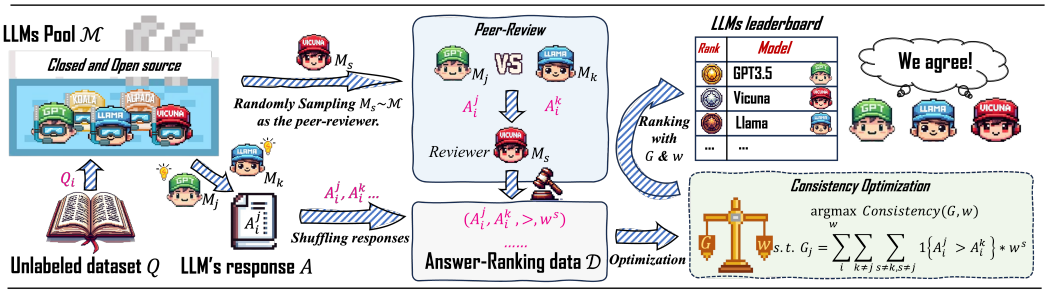


Figure 2: The pipeline of the PiCO. It is mainly composed of two components: the peer-review and consistency optimization stages. Specifically, in the peer-review stage, the unlabeled dataset $\mathcal{Q}$ and the LLMs pool $\mathcal{M}$ are given. Then, we let all LLMs answer each unlabeled question to obtain the response set $\mathcal{A}$. We shuffle the set and construct anonymous answer pairs, while randomly selecting other LLMs to evaluate both responses with a learnable confidence w. As a result, we can obtain the answer-ranking data $\mathcal{D}$ which is a quadruple that records the partial order between two answers and the evaluator's confidence weight. In the consistency optimization stage, we update the parameter w by maximizing the consistency of each LLM's capability and score, while re-ranking the LLMs to be closer to human rankings.

- We conduct extensive experiments on three crowdsourcing datasets with three standard rank-based metrics validating the effectiveness of the proposed PiCO approach.

2 The Proposed Approach

2.1 Problem Definition

This paper aims to re-rank the ability of LLMs to be closer to human (ground-truth) rankings $\mathcal{R}^*$ in an unsupervised way (without relying on any human annotations). Specifically, we have a large language models (LLMs) pool $\mathcal{M} = \{M_j\}_{j=1}^m$, which includes both open-source and closed-source models. Write $M_1 \succ M_2$ to indicate that the LLM M_1 has stronger capabilities than the LLM M_2. Thus, we can assume that the ground-truth ranking $\mathcal{R}^*$ is as follows,

$$\mathcal{R}^* := [M_1 \succ M_2 \succ M_3 \succ ... \succ M_m]. \tag{1}$$

Assuming that the learned ranking $\hat{\mathcal{R}}$ by different evaluation methods is as follows,

$$\hat{\mathcal{R}} := [M_3 \succ M_1 \succ M_2 \succ ... \succ M_m]. \tag{2}$$

The goal is to learn an LLM ranking $\hat{\mathcal{R}}$ that aligns with human ranking $\mathcal{R}^*$ as much as possible.

2.2 Algorithm Details

The pipeline of the proposed PiCO, depicted in Figure 2, involves peer-review and consistency optimization stages. Next, we will introduce the two stages in detail.

Peer Review Stage. In our *peer-review* system, we consider an unsupervised LLM evaluation scenario with an unlabeled dataset $\mathcal{Q}$ consisting of n open-ended questions, where $\mathcal{Q} = \{Q_i\}_{i=1}^n$. All LLMs will answer each unlabeled question to obtain the set $\mathcal{A} = \{\{A_i^j\}_{i=1}^n\}_{j=1}^m$, where A_i^j is as follows,

$$A_i^j = M_j(Q_i) \tag{3}$$

which infers the model M_j response an answer A_i^j with question Q_i. In addition, LLMs themselves also act as "reviewers" to evaluate other answers. Specifically, for the same question $Q_i \in \mathcal{Q}$, we randomly construct a battle pair $< A_i^j, A_i^k >$ for review. Each battle pair will randomly assign "reviewers" to determine the winners or declare ties,

$$(A_i^k, A_i^s, >, w^j) = M_j(A_i^k; A_i^s | Q_i). \tag{4}$$

Under the same question Q_i, the quadruples $(A_i^k, A_i^s, >, w^j)$ indicate that the "reviewer" M_j believes that the answer A_i^k is better than answer A_i^k with a confidence w^j. Thus, we can collect the answer-ranking data $\mathcal{D}$ as follows,

$$\mathcal{D} = \left\{(A_i^k, A_i^s, >, w^j)\right\}_{i \sim \mathcal{Q}, j, k, M_j \sim \mathcal{M}}, \tag{5}$$

Table 1: Validation of consistency assumption. Performance comparison of Backward, Uniform, Forward weight voting, and Consistency Optimization methods with two metrics across three datasets.

Methods	MT-Bench		Chatbot Arena		AlpacaEval	
	$S(\uparrow)$	$\tau(\uparrow)$	$S(\uparrow)$	$\tau(\uparrow)$	$S(\uparrow)$	$\tau(\uparrow)$
Backward Weight	0.70	0.50	0.72	0.52	0.69	0.50
Uniform Weight	0.74	0.54	0.80	0.58	0.77	0.58
Forward Weight	0.75	0.56	0.82	0.59	0.79	0.60
Random Weight + Consistency Optimization	**0.90**	**0.77**	**0.89**	**0.72**	**0.84**	**0.68**

where i denotes the question index, and j, k, s indicate the model indices. $w^s \in (0, 1]$ is a learnable confidence weight of model M_s, and $>$ is a partial order relationship from $\{>, <, =\}$. After that, we can calculate the response score G_j of each LLM,

$$G_j = \sum_{(A_i^k, A_i^s, >, w^j) \sim \mathcal{D}} \mathbf{1}\{A_i^j > A_i^k\} \cdot w^s, \tag{6}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function that the value is 1 when the condition is met, otherwise, it is 0. We can define the LLM M_1 is better than M_2 as its score is larger, *i.e.*, $M_1 \succ M_2 := G_1 > G_2$. Thus, we can re-write the learned LLM ranking $\hat{\mathcal{R}}$ as follows,

$$\hat{\mathcal{R}} := [G_3 > G_1 > G_2 > ... > G_m]. \tag{7}$$

Thus, the goal is to learn the confidence weights w to adjust the final ranking $\hat{\mathcal{R}}$ to be closer to ground-truth ranking $\mathcal{R}^*$.

Validation of Consistency Assumption. First of all, we start with a toy experiment to study the role of confidence w in Table 1. Specifically, we manually construct three methods: Backward Weight, Uniform Weight, and Forward Weight. That is, the ability weights of the model are respectively weighted forward ($w = [1, 0.9, ..., 0]$), uniformly ($w = [1, 1, ..., 1]$), and backward ($w = [0, 0.1, ..., 1]$) according to the ground-truth human ranking. In other words, the Forward Weight means manually assigning higher weights to those models with stronger abilities, and so on for others. Then, we can calculate the response score G_j for each model using Eq.6, and obtain the LLM ranking $\hat{\mathcal{R}}$. We measure the alignment between $\hat{\mathcal{R}}$ and $\mathcal{R}^*$ with Spearman's $S(\uparrow)$ and Kendall's $\tau(\uparrow)$ rank correlation coefficient in Table 1. Note that this is an ideal experiment, as we only use the ground-truth human ranking to validate the feasibility of our idea.

As shown in Table 1, it can be observed that the Forward Weight achieves better results than the Uniform and Backward ones in all cases, while the Backward one always achieves worse results. **It validates that assigning larger weights to those models with stronger capabilities can obtain better results.** In other words, those answers that are more recognized by other "reviewers" (models) usually come from LLs with stronger abilities. We formalize it as a *consistency assumption*, *i.e.*, high-level LLM can evaluate others' answers more accurately (confidence) than low-level ones, while higher-level LLM can also achieve higher answer-ranking scores, the ability and score of the model usually have consistency.

Consistency Optimization Stage. Based on this observation, we propose to maximize the consistency of each LLM's capability w and score G with constrained optimization as follows,

$$\begin{aligned} &\underset{w}{\text{argmax}}\ \text{Consistency}(G, w) \\ &\text{s.t. } G_j = \sum_{(A_i^j, A_i^k, >, w^s) \sim \mathcal{D}} \mathbf{1}\{A_i^j > A_i^k\} \cdot w^s, \end{aligned} \tag{8}$$

where the Pearson correlation [40] is used to measure the consistency between w and G. Note that we only introduce this straightforward implementation to validate our idea of PiCO. Other more advanced strategies may be employed to further improve the performance.

Discussion: It is worth noting that the whole process (Eq. 5 and 8) works in an unsupervised way. The only thing we do is to adaptively adjust the score of each LLM that match its abilities. Most importantly, we also validate the effectiveness of the proposed *consistency optimization* in Table 1.

Specifically, we randomly initialize the ability weights and employ our *consistency optimization* to adjust the weight. It can be observed that the learned w by our consistency optimization algorithm (Eq.8) can further improve the performance of the evaluation system, making the LLM ranking $\hat{\mathcal{R}}$ closer to human ranking $\mathcal{R}^*$. Another intuitive example is as follows: in a real peer-review system, if the academic level of three scholars a, b, and c satisfies the following relationship, $w^a > w^b > w^c$. So, in the ultimate ideal scenario, the ranking of the scores submitted by these three scholars should also be, $G_a > G_b > G_c$. In other words, the sorting of G and w satisfies high consistency. On the other hand, scholars with stronger abilities (*i.e.*, scholar a) evaluate $A^b > A^c$ have stronger persuasiveness, so scholar b should also receive higher weighted scores $1 * w^a$.

Reviewer Elimination Mechanism. Realizing that not all LLMs have sufficient ability to evaluate the responses of other models. We thus introduce an unsupervised elimination mechanism to remove those LLMs that have low scores. It iteratively removes the lowest-scoring LLM from the "reviewer queue" for the next consistency optimization stage, until 60% of models are eliminated. The discussion of the elimination mechanism can also be found in the Experiment 3.3.

3 EXPERIMENTS

Datasets. To validate the effectiveness of the proposed approach, we perform experiments on Chatbot Arena[57], MT-Bench[57], and AlpacaEval[31]. The MT-Bench dataset assesses six LLMs' responses to 80 multi-category questions. The Chatbot Arena Conversations Dataset, with 33K conversations from 13K IPs during April-June 2023, evaluates real dialogue performance. AlpacaEval dataset integrates 805 evaluations from diverse tests (e.g., Self-Instruct[49], OASST, Anthropics helpful[7], Vicuna[16] and Koala[24] test sets) to align evaluations real-world interactions[21]. These datasets are collected by crowdsourcing platforms from human feedback, so they have a ground-truth ranking LLMs $\mathcal{R}^*$ to measure the alignment performance of different evaluation methods.

LLMs Pool. In our experiments, we employ 15 LLMs with diverse architectures to construct the LLMs pool, including GPT-3.5-Turbo[37], WizardLM-13B[53], Guanaco-33B[1], Vicuna-7B[16], Vicuna-13B[16], Koala-13B[25], Mpt-7B[44], gpt4all-13B[6], ChatGLM-6B[55], Oasst-sft-4-pythia-12B[19], FastChat-T5-3B[57], StableLM-7B[3], Dolly-12B[18], LLaMA-13B[45], Alpaca-13B[43]. All models use the same prompt template, which can be found in Appendix C.

Baselines. To validate the effectiveness of the proposed PiCO approach, we compare the following methods in the experiments.

- *The wisdom of the crowds*: The two methods that perform LLMs evaluation based on the wisdom of the crowds [42; 13; 52] are compared in this experiment. *1)* **Majority Voting** [42]: Multiple review models vote for the better answer for the same response pair, and the model with the most votes gets 1 score; *2)* **Rating Voting** [5]: Multiple review models also vote on the same response pair, and the number of votes obtained is the score.
- *State-of-the-art methods*: The four recent SOTA methods of using either single or multiple models for self-evaluation are compared in this experiment. **PandaLM[48]:** It is a fine-tuned language model based on Llama-7b designed for the preference judgment tasks to evaluate and optimize LLMs. **GPTScore[23]:** It employs generative pre-trained models to assess the quality of generated text. It calculates the likelihood that the text was generated in response to specific instructions and context, indicative of high quality. In our implementation, GPT-3 (davinci-002) and flan-t5-xxl serve as the base models. **PRD[30]:** It transforms the LLMs win rates into weights for competitive ranking, while evaluating each LLM based on its preference for all possible pairs of answers, enabling a tournament-style ranking system. **PRE[17]:** It employs a supervised process to evaluate LLMs using a qualification exam, aggregates their scores based on accuracy, and assigns weights accordingly. **Claude-3 (API):** Another SOTA closed-source LLM developed by Anthropic. **PiCO (Ours)**: the proposed approach in this paper.

Metrics. For all experiments, we employ three popular rank-based metrics to evaluate the aforementioned experimental setups and our PiCO method: **Spearman's Rank Correlation Coefficient** $S(\uparrow)$ [28], **Kendall's Rank Correlation Coefficient** $\tau(\uparrow)$ [27] and **Permutation Entropy** $H(\downarrow)$ [8]. The details of these metrics can be found in the Appendix A. Moreover, we perform the experiments for 4 runs and record the average results over 4 seeds ($seed = 1, 2, 3, 4$).

Table 2: Comparison of all methods on three datasets under data volumes of 1, 0.7 and 0.4, where the top value is highlighted by blod font. Higher S and τ scores indicate better performance, while a lower H score signifies improved performance.

Datasets	Chatbot Arena			MT-Bench			AlpacaEval		
Methods	1	0.7	0.4	1	0.7	0.4	1	0.7	0.4
	Spearman's Rank Correlation Coefficient $S(\uparrow)$								
Majority Voting [42]	$0.76^{\pm 0.00}$	$0.75^{\pm 0.01}$	$0.73^{\pm 0.03}$	$0.73^{\pm 0.00}$	$0.77^{\pm 0.01}$	$0.75^{\pm 0.01}$	$0.80^{\pm 0.00}$	$0.79^{\pm 0.01}$	$0.78^{\pm 0.01}$
Rating Voting [5]	$0.74^{\pm 0.00}$	$0.72^{\pm 0.02}$	$0.71^{\pm 0.02}$	$0.80^{\pm 0.00}$	$0.78^{\pm 0.02}$	$0.74^{\pm 0.03}$	$0.77^{\pm 0.00}$	$0.77^{\pm 0.01}$	$0.78^{\pm 0.01}$
GPTScore(flan-t5-xxl)[23]	$-0.09^{\pm 0.00}$	$-0.09^{\pm 0.01}$	$-0.12^{\pm 0.02}$	$0.05^{\pm 0.00}$	$0.01^{\pm 0.07}$	$0.04^{\pm 0.09}$	$0.34^{\pm 0.00}$	$0.34^{\pm 0.00}$	$0.34^{\pm 0.01}$
GPTScore(davinci-002)[23]	$0.15^{\pm 0.00}$	$0.13^{\pm 0.02}$	$-0.02^{\pm 0.14}$	$0.52^{\pm 0.00}$	$0.42^{\pm 0.05}$	$0.45^{\pm 0.05}$	$0.76^{\pm 0.00}$	$0.77^{\pm 0.07}$	$0.75^{\pm 0.06}$
PandaLM[48]	$0.43^{\pm 0.00}$	$0.44^{\pm 0.03}$	$0.44^{\pm 0.10}$	$0.50^{\pm 0.00}$	$0.50^{\pm 0.08}$	$0.52^{\pm 0.17}$	$0.57^{\pm 0.00}$	$0.55^{\pm 0.01}$	$0.48^{\pm 0.08}$
PRD[30]	$0.84^{\pm 0.00}$	$0.84^{\pm 0.00}$	$0.82^{\pm 0.03}$	$0.86^{\pm 0.00}$	$0.84^{\pm 0.03}$	$0.81^{\pm 0.03}$	$0.81^{\pm 0.00}$	$0.81^{\pm 0.01}$	$0.81^{\pm 0.02}$
PRE[17]	$0.86^{\pm 0.00}$	$0.86^{\pm 0.01}$	$0.86^{\pm 0.01}$	$0.86^{\pm 0.00}$	$0.84^{\pm 0.03}$	$0.82^{\pm 0.04}$	$0.83^{\pm 0.00}$	$0.81^{\pm 0.01}$	$0.83^{\pm 0.02}$
Claude-3 (API)	$0.90^{\pm 0.01}$	$0.88^{\pm 0.03}$	$0.87^{\pm 0.04}$	$0.85^{\pm 0.06}$	$0.82^{\pm 0.08}$	$0.80^{\pm 0.07}$	$0.79^{\pm 0.03}$	$0.78^{\pm 0.02}$	$0.75^{\pm 0.04}$
PiCO (Ours)	$\mathbf{0.90^{\pm 0.00}}$	$\mathbf{0.89^{\pm 0.01}}$	$\mathbf{0.89^{\pm 0.01}}$	$\mathbf{0.89^{\pm 0.01}}$	$\mathbf{0.89^{\pm 0.01}}$	$\mathbf{0.84^{\pm 0.11}}$	$\mathbf{0.84^{\pm 0.00}}$	$\mathbf{0.83^{\pm 0.03}}$	$\mathbf{0.85^{\pm 0.01}}$
	Kendall's Rank Correlation Coefficient $\tau(\uparrow)$								
Majority Voting [42]	$0.58^{\pm 0.00}$	$0.56^{\pm 0.02}$	$0.52^{\pm 0.05}$	$0.56^{\pm 0.00}$	$0.61^{\pm 0.02}$	$0.60^{\pm 0.02}$	$0.62^{\pm 0.00}$	$0.60^{\pm 0.02}$	$0.58^{\pm 0.02}$
Rating Voting [5]	$0.54^{\pm 0.00}$	$0.53^{\pm 0.02}$	$0.52^{\pm 0.02}$	$0.58^{\pm 0.00}$	$0.57^{\pm 0.02}$	$0.54^{\pm 0.01}$	$0.58^{\pm 0.00}$	$0.57^{\pm 0.01}$	$0.57^{\pm 0.02}$
GPTScore(flan-t5-xxl) [23]	$-0.06^{\pm 0.00}$	$-0.06^{\pm 0.02}$	$-0.09^{\pm 0.02}$	$-0.05^{\pm 0.00}$	$-0.07^{\pm 0.05}$	$-0.02^{\pm 0.06}$	$0.25^{\pm 0.00}$	$0.26^{\pm 0.01}$	$0.26^{\pm 0.01}$
GPTScore(davinci-002) [23]	$0.20^{\pm 0.00}$	$0.23^{\pm 0.02}$	$0.03^{\pm 0.11}$	$0.36^{\pm 0.00}$	$0.30^{\pm 0.05}$	$0.31^{\pm 0.05}$	$0.60^{\pm 0.08}$	$0.61^{\pm 0.05}$	$0.59^{\pm 0.08}$
PandaLM [48]	$0.30^{\pm 0.00}$	$0.31^{\pm 0.03}$	$0.31^{\pm 0.07}$	$0.39^{\pm 0.00}$	$0.37^{\pm 0.06}$	$0.40^{\pm 0.12}$	$0.41^{\pm 0.00}$	$0.39^{\pm 0.02}$	$0.32^{\pm 0.05}$
PRD [30]	$0.68^{\pm 0.00}$	$0.69^{\pm 0.01}$	$0.67^{\pm 0.03}$	$0.68^{\pm 0.06}$	$0.66^{\pm 0.02}$	$0.63^{\pm 0.03}$	$0.64^{\pm 0.00}$	$0.63^{\pm 0.03}$	$0.63^{\pm 0.02}$
PRE [17]	$0.71^{\pm 0.00}$	$0.73^{\pm 0.02}$	$0.72^{\pm 0.02}$	$0.68^{\pm 0.00}$	$0.68^{\pm 0.02}$	$0.65^{\pm 0.03}$	$0.64^{\pm 0.00}$	$0.66^{\pm 0.01}$	$0.66^{\pm 0.03}$
Claude-3 (API)	$0.76^{\pm 0.04}$	$0.72^{\pm 0.05}$	$0.70^{\pm 0.07}$	$0.67^{\pm 0.07}$	$0.66^{\pm 0.11}$	$0.61^{\pm 0.10}$	$0.64^{\pm 0.06}$	$0.61^{\pm 0.04}$	$0.66^{\pm 0.06}$
PiCO (Ours)	$\mathbf{0.77^{\pm 0.00}}$	$\mathbf{0.76^{\pm 0.01}}$	$\mathbf{0.77^{\pm 0.02}}$	$\mathbf{0.72^{\pm 0.01}}$	$\mathbf{0.72^{\pm 0.03}}$	$\mathbf{0.70^{\pm 0.12}}$	$\mathbf{0.68^{\pm 0.00}}$	$\mathbf{0.66^{\pm 0.04}}$	$\mathbf{0.67^{\pm 0.02}}$
	Permutation Entropy $H(\downarrow)$								
Majority Voting [42]	$1.27^{\pm 0.05}$	$1.30^{\pm 0.03}$	$1.36^{\pm 0.06}$	$1.37^{\pm 0.03}$	$1.30^{\pm 0.06}$	$1.27^{\pm 0.04}$	$1.26^{\pm 0.02}$	$1.28^{\pm 0.03}$	$1.29^{\pm 0.03}$
Rating Voting [5]	$1.39^{\pm 0.02}$	$1.43^{\pm 0.03}$	$1.42^{\pm 0.07}$	$1.32^{\pm 0.03}$	$1.35^{\pm 0.04}$	$1.38^{\pm 0.04}$	$1.34^{\pm 0.03}$	$1.37^{\pm 0.03}$	$1.34^{\pm 0.08}$
GPTScore(flan-t5-xxl)[23]	$1.68^{\pm 0.01}$	$1.68^{\pm 0.02}$	$1.65^{\pm 0.02}$	$1.72^{\pm 0.02}$	$1.70^{\pm 0.02}$	$1.68^{\pm 0.03}$	$1.55^{\pm 0.02}$	$1.57^{\pm 0.03}$	$1.60^{\pm 0.01}$
GPTScore(davinci-002)[23]	$1.54^{\pm 0.02}$	$1.64^{\pm 0.02}$	$1.68^{\pm 0.05}$	$1.51^{\pm 0.02}$	$1.61^{\pm 0.01}$	$1.61^{\pm 0.04}$	$1.25^{\pm 0.02}$	$1.23^{\pm 0.08}$	$1.26^{\pm 0.14}$
PandaLM[48]	$1.65^{\pm 0.01}$	$1.64^{\pm 0.02}$	$1.63^{\pm 0.05}$	$1.55^{\pm 0.03}$	$1.59^{\pm 0.05}$	$1.52^{\pm 0.08}$	$1.56^{\pm 0.01}$	$1.58^{\pm 0.01}$	$1.64^{\pm 0.05}$
PRD[30]	$1.15^{\pm 0.04}$	$1.12^{\pm 0.05}$	$1.13^{\pm 0.06}$	$1.15^{\pm 0.05}$	$1.17^{\pm 0.06}$	$1.23^{\pm 0.04}$	$1.21^{\pm 0.04}$	$1.22^{\pm 0.06}$	$1.23^{\pm 0.07}$
PRE[17]	$1.07^{\pm 0.01}$	$1.03^{\pm 0.03}$	$1.06^{\pm 0.04}$	$1.17^{\pm 0.04}$	$1.13^{\pm 0.05}$	$1.19^{\pm 0.05}$	$1.18^{\pm 0.03}$	$1.21^{\pm 0.04}$	$1.15^{\pm 0.05}$
PiCO (Ours)	$\mathbf{0.94^{\pm 0.02}}$	$\mathbf{0.96^{\pm 0.04}}$	$\mathbf{0.95^{\pm 0.08}}$	$\mathbf{1.01^{\pm 0.07}}$	$\mathbf{1.02^{\pm 0.11}}$	$\mathbf{1.06^{\pm 0.24}}$	$\mathbf{1.17^{\pm 0.02}}$	$\mathbf{1.17^{\pm 0.08}}$	$\mathbf{1.13^{\pm 0.05}}$

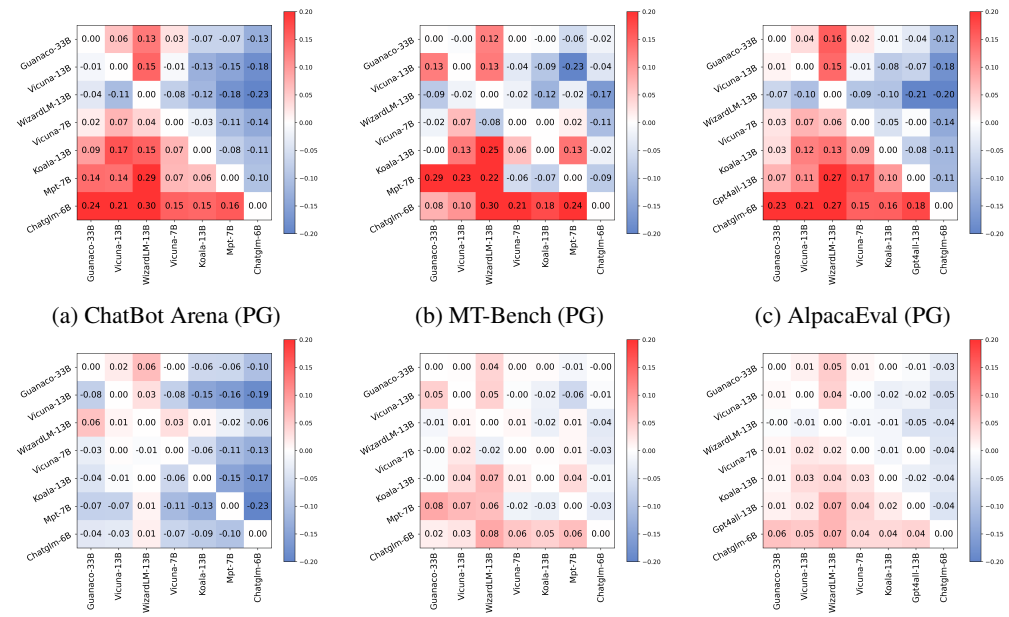
3.1 PERFORMANCE COMPARISON

We validate the effectiveness of the proposed PiCO method on three datasets by comparing the following two types of methods, *i.e.*, the wisdom of the crowds and recent SOTA LLMs evaluation methods. The average results with different rank-based metrics and datasets are demonstrated in Table 2. The ratios of response sets $\mathcal{D}$ are 1, 0.7, and 0.4, respectively.

The results presented in Table 2 demonstrate that the proposed PiCO method consistently outperforms competing approaches across most evaluated metrics, including surpassing all baselines, such as **Claude-3 (API)**. Specifically, PiCO achieves improvements of 0.027, 0.047, and 0.14 on Spearman's Rank Correlation Coefficient, Kendall's Rank Correlation Coefficient, and Permutation Entropy metrics, respectively, compared to the runner-up. These results underscore the superiority of aggregating evaluations from multiple models, such as Majority Voting, Rating Voting, PRD, and PRE, as opposed to relying solely on single-model methods like GPTScore and PandaLM. This collective model approach, leveraging 'the wisdom of the crowds', aligns with human rankings more accurately in our open-question evaluation framework.

In comparison with existing SOTA evaluation methods(*i.e.,* PRD and PRE), it is evident that PiCO exhibits improvements across various evaluation metrics. Despite PRD's adjustment of model weights based on their win rates and PRE's reliance on supervised human feedback data to assign weights through a qualification exam, neither method achieves performance superior to the fully unsupervised PiCO approach. These methods rely on predefined criteria and human feedback, potentially leading to biases or suboptimal performance. In contrast, PiCO leverages unsupervised learning techniques, allowing it to autonomously adapt and discover patterns in the data without explicit human intervention.

It is important to highlight that PandaLM, a language model equipped with 7 billion parameters, was fine-tuned using labels generated by GPT-3.5-turbo as the ground truth, achieving stable performance across various datasets. However, in our unsupervised, open-ended experimental setup, which focuses on ranking-based metrics, GPTScore exhibits less robustness regardless of whether the base model is GPT-3 (davinci-002) or flan-t5-xx.



(a) ChatBot Arena (PG) (b) MT-Bench (PG) (c) AlpacaEval (PG)

(d) ChatBot Arena (weighted PG) (e) MT-Bench (weighted PG) (f) AlpacaEval (weighted PG)

Figure 3: Heatmap distribution of preference gap (PG) metric among seven LLMs across three datasets. Higher values (above 0) indicate greater evaluation bias[17]. The first row shows original PG values in three datasets, while the second row displays PG values re-weighted using our learned confidence weights.

3.2 Exploring the Role of Confidence Weight

In this subsection, we show that the confidence weight w learned by our *consistency optimization* can reduce the system evaluation bias. Specifically, we first study whether the "review" model would prefer a particular model's response. Following [17], we employ the preference gap (PG) to evaluate the bias as follows,

$$PG(i,j) = P_i(i > j) - P_j(i > j), \tag{9}$$

where $P_i(i > j)$ represents the winning rate of model i as the "reviewer" believes that i defeated j. The heatmap distribution of the PG value $PG(i,j)$ among seven LLMs across three datasets is demonstrated in the first row of Figure 3. It can be observed that the evaluation system exhibits severe bias. Especially on ChatGLM-6B and Mpt-7B models, they often believe that their results are better than other ones, as their PG values are greater than 0 across three datasets.

After the *consistency optimization*, we assign the learned confidence weight w to the corresponding model and ultimately obtain the re-weighting PG value $\hat{PG}(i,j)$ as follows,

$$\hat{PG}(i,j) = w_i \times P_i(i > j) - w_j \times P_j(i > j). \tag{10}$$

The results of the re-weighting PG value $\hat{PG}(i,j)$ are displayed on the second row of Figure 3. It can be observed that the learned confidence weight w can significantly mitigate the preference gaps of the whole evaluation system. In our consistency optimization, LLMs such as ChatGLM-6B and Mpt-7B have lower weights, and reducing their confidence can effectively alleviate the system evaluation bias.

3.3 Study of Elimination Mechanism

Performance Comparison of Elimination Mechanisms. The PiCO and PRE[17] methods both employ elimination mechanisms to remove those weakest LLMs from the "reviewer queue" during the evaluation process. As shown in Figure 4, the x-axis quantifies the number of reviewers eliminated, and the y-axis measures the PEN, where lower scores denote higher performance. It can be observed that both PiCO and PRE exhibit better performance with an increasing number of eliminated "reviewers". The proposed PiCO approach can achieve better performance than PRE in most cases. It is worth noting that the PRE method employs the accuracy of "qualification exams" to elim-

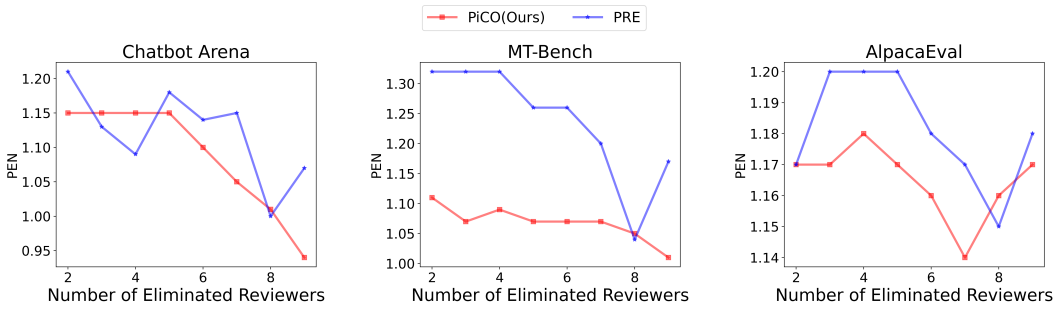


Figure 4: Performance comparison of the PiCO (Ours) and PRE[17] methods on the Chatbot Arena, MT-Bench, and AlpacaEval datasets, with the number of eliminated reviewers on the x-axis. The y-axis is PEN, where lower values indicate better performance.

inate weak LLMs, and this process requires human annotation [17]. On the contrary, the elimination process of our PiCO method is unsupervised and can still achieve better evaluation results than PRE.

Automatic Learning of Elimination Thresholds. We observed that weaker LLMs tend to have poorer evaluation abilities, introducing significant noise into the peer-review system. Therefore, eliminating weaker models instead of retaining them enhances the robustness of the system. We employed an unsupervised approach to automatically learn the elimination threshold, as shown in Figure 5, by using the average training loss curve as the number of eliminated reviewers increases. It can be seen that removing weaker reviewers reduces the average loss of the entire system, indicating that eliminating noisy evaluations benefits the overall process. Notably, when 60% (or 9) of the weaker reviewers are removed, the system's loss reaches its minimum. This trend is consistent across all three datasets, suggesting that the elimination threshold is learned automatically. However, removing more than 9 stronger reviewers harms the evaluation process.

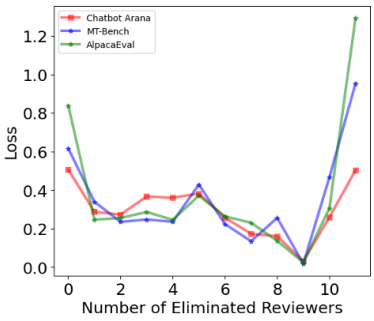


Figure 5: The average loss for different numbers of eliminated reviewers(↓). It shows how the iterative elimination of weaker reviewers affects the overall loss in the peer-review system.

3.4 Other Results

Validation on more metrics (Precision@K and RBP@K). We demonstrated the results of precision and RBP (K=8,9,10) with other baselines in Table 3 (left). The results show that the proposed PiCO approach can achieve better precision and RBP performance in all cases. These results once again validate that PiCO can predict the LLM ranking more accurately than other baselines.

Comparison of tokens consumed. We compute the token consumption of each method in Table 3 (right). It can be observed that the proposed PiCO approach has a similar token consumed with other baselines (*e.g.*, PRD and PRE) while achieving better evaluation performance. Although Chatbot Arena has a smaller token consumption, it requires 33k human annotations, while PiCO does not require any human annotations.

Stability validation of consistency optimization. We repeated the experiment with different seeds for 1000 times, and plotted the training loss curve and weight distribution in Figure 6. The results show that the proposed consistency optimization process is stable and the learned w is convergence.

4 Related Work

Evaluation Benchmarks for Diversity. LLMs are designed to handle a variety of tasks, necessitating comprehensive benchmarks [15]. Notable benchmarks include GLUE[46] and SuperGLUE [47], which simulate real-world scenarios across tasks such as text classification, translation, reading comprehension, and dialogue generation. HELM [32] provides a holistic evaluation of LLMs, assessing language understanding, generation, coherence, and reasoning. BIG-bench [41] pushes LLM capabilities with 204 diverse tasks. MMLU [26] measures multitask accuracy across domains

Table 3: Comparison of more metrics (Precision@K and RBP@K) and token consumption on Chatbot Arena.

Methods	RBP@K (↑)			Precision@K (↑)			Input Token	Output Token	Annotation Cost
	8	9	10	8	9	10			
Chatbot Arena Platforms [57]	-	-	-	-	-	-	∼ 7500k	∼ 10944k	∼ 32k
GPTScore(flan-t5-xxl) [23]	26.2%	29.6%	45.1%	50.0%	55.6%	70.0%	∼ 22882k	∼ 12260k	0
GPTScore(davinci-002) [23]	42.0%	50.6%	53.3%	62.5%	77.8%	80.0%	∼ 22882k	∼ 12260k	0
PandaLM [48]	63.5%	63.5%	66.2%	62.5%	55.6%	60.0%	∼ 22882k	∼ 10355k	0
PRD [30]	67.2%	73.8%	81.3%	87.5%	88.9%	80.0%	∼ 25087k	∼ 10935k	0
PRE [17]	78.0%	81.3%	81.3%	87.5%	88.9%	80.0%	∼ 24120k	∼ 11115k	∼ 7k
PiCO (Ours)	**83.2%**	**83.2%**	**85.9%**	**100.0%**	**100.0%**	**90.0%**	∼ 23823k	∼ 11685k	0

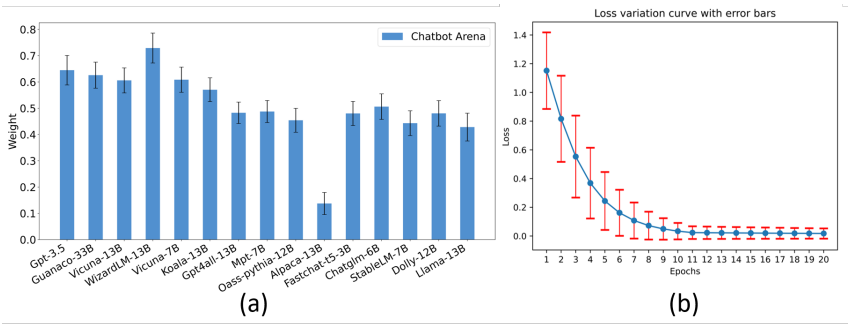


Figure 6: Stability validation of consistency optimization. We repeated the experiment with different seeds for 1000 times, and plotted the training loss curve and weight distribution. The results show that the learning process is stable and the learned w is convergence.

like mathematics and law. However, these evaluations can be compromised by benchmark leakage, where evaluation data inadvertently used for training leads to inflated performance metrics [4; 58].

Human Evaluation. Human evaluation provides reliable feedback that closely aligns with real-world applications [15]. Liang et al. [32] evaluated summary and misinformation scenarios across multiple models. Ziems et al. [59] involved experts to assess model outputs in various domain-specific tasks. Bang et al. [9] examined ChatGPT's performance in summarization, translation, and reasoning using human-annotated datasets. The LMSYS initiative introduced platforms like Chatbot Arena [57], relying on human ratings as the primary evaluation metric. Despite its effectiveness, human evaluation is costly and subject to bias and cultural differences[39].

Large Language Models for Evaluation. The development of open-source LLMs has led to the use of LLMs as evaluators. GPTScore[23] uses models like GPT-3 to assign probabilities to high-quality content through multidimensional evaluation. Bubeck et al.[12] tested GPT-4, finding it rivaling human capabilities. Lin and Chen introduced LLM-EVAL[33] for evaluating dialogue quality with single prompts. PandaLM[48] employs LLMs as "judges" for evaluating instruction tuning. However, reliance on a single model can introduce biases such as positional[20], verbosity[50], and self-favoring biases[35; 57]. ChatEval[14] proposes a multi-agent framework to simulate human evaluation processes. Similarly, PRE[17] and PRD[30] use LLMs as evaluators, combining multiple evaluation outcomes for automated assessment. However, the PRE method, which relies on human feedback for supervised evaluation throughout the process, still incurs relatively high costs.

5 Conclusion

In this paper, we propose PiCO, a novel unsupervised evaluation method to automatically evaluate Large Language Models (LLMs) without relying on human feedback. PiCO utilizes *peer-review* mechanisms to autonomously assess LLMs in a shared environment, where both open-source and closed-source models can respond to unlabeled questions and evaluate each other. In this setup, each LLM's response score is determined collectively by other anonymous models, aiming to maximize consistency across capabilities and scores. The extensive experiment results across multiple datasets and standard rank-based metrics demonstrate that PiCO effectively generates an LLM ranking that aligns closely with human preferences. In the future, we plan to extend the peer-review mechanism to evaluate the capabilities of multi-modality large models.

REFERENCES

[1] Guanaco - generative universal assistant for natural-language adaptive context-aware omnilingual outputs. https://guanaco-model.github.io/, 2023. Accessed: 15 April 2024.

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

[3] Stability AI. Stablelm-tuned-alpha-7b: A fine-tuned language model for diverse applications. https://huggingface.co/stabilityai/stablelm-tuned-alpha-7b, 2023. Accessed: 15 April 2024.

[4] Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. Can we trust the evaluation on chatgpt?, 2023.

[5] Mohammad Allahbakhsh and Aleksandar Ignjatovic. Rating through voting: An iterative method for robust rating. *arXiv preprint arXiv:1211.0390*, 2012.

[6] Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. https://github.com/nomic-ai/gpt4all, 2023.

[7] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[8] Christoph Bandt and Bernd Pompe. Permutation entropy: a natural complexity measure for time series. *Physical review letters*, 88(17):174102, 2002.

[9] Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*, 2023.

[10] Robert S Boyer and J Strother Moore. Mjrtya fast majority vote algorithm. In *Automated reasoning: essays in honor of Woody Bledsoe*, pp. 105–117. Springer, 1991.

[11] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[12] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[13] David V Budescu and Eva Chen. Identifying expertise to extract the wisdom of crowds. *Management science*, 61(2):267–280, 2015.

[14] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

[15] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 2023.

[16] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90% chatgpt quality. https://vicuna.lmsys.org, 2023. Accessed: 15 April 2024.

[17] Zhumin Chu, Qingyao Ai, Yiteng Tu, Haitao Li, and Yiqun Liu. Pre: A peer review based large language model evaluator. *arXiv preprint arXiv:2401.15641*, 2024.

[18] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

[19] Open-Assistant Contributors. Oasst-sft-4-pythia-12b: A supervised fine-tuning model for language understanding. https://huggingface.co/OpenAssistant/oasst-sft-4-pythia-12b-epoch-3.5, 2023. Accessed: 15 April 2024.

[20] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36, 2024.

[21] Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

[22] Allan M. Feldman. Majority voting. *SpringerLink*, 2006.

[23] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166*, 2023.

[24] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1, 2023.

[25] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala-13b: Dialogue model for effective human-ai interaction. https://bair.berkeley.edu/blog/2023/04/03/koala/, 2023. Accessed: 15 April 2024.

[26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[27] Maurice G Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

[28] Ann Lehman, Norm O'Rourke, Larry Hatcher, and Edward Stepanski. *JMP for basic univariate and multivariate statistics: methods for researchers and social scientists*. Sas Institute, 2013.

[29] Charles Eric Leiserson, Ronald L Rivest, Thomas H Cormen, and Clifford Stein. *Introduction to algorithms*, volume 3. MIT press Cambridge, MA, USA, 1994.

[30] Ruosen Li, Teerth Patel, and Xinya Du. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762*, 2023.

[31] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

[32] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[33] Yen-Ting Lin and Yun-Nung Chen. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*, 2023.

[34] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[35] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

[36] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[37] OpenAI. Introducing chatgpt. `https://openai.com/blog/chatgpt`, 2022. Accessed: [insert date here].

[38] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[39] Kaiping Peng, Richard E Nisbett, and Nancy YC Wong. Validity problems comparing values across cultures and possible solutions. *Psychological methods*, 2(4):329, 1997.

[40] Philip Sedgwick. Pearsons correlation coefficient. *Bmj*, 345, 2012.

[41] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*, 2022.

[42] James Surowiecki. *The wisdom of crowds*. Anchor, 2005.

[43] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[44] MosaicML NLP Team. Introducing mpt-7b: A new standard for open-source, commercially usable llms, 2023. URL `www.mosaicml.com/blog/mpt-7b`. Accessed: 2023-05-05.

[45] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[46] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[47] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[48] Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*, 2023.

[49] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[50] Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36, 2024.

[51] Tianwen Wei, Liang Zhao, Lichang Zhang, Bo Zhu, Lijie Wang, Haihua Yang, Biye Li, Cheng Cheng, Weiwei Lü, Rui Hu, et al. Skywork: A more open bilingual foundation model. *arXiv preprint arXiv:2310.19341*, 2023.

[52] Susan C Weller. Cultural consensus theory: Applications and frequently asked questions. *Field methods*, 19(4):339–368, 2007.

[53] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions, 2023.

[54] Jia-Yu Yao, Kun-Peng Ning, Zhen-Hui Liu, Mu-Nan Ning, and Li Yuan. Llm lies: Hallucinations are not bugs, but features as adversarial examples. *arXiv preprint arXiv:2310.01469*, 2023.

[55] Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*, 2022.

[56] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[57] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023.

[58] Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

[59] Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *arXiv preprint arXiv:2305.03514*, 2023.

A Detailed Explanation of Metrics

In this section, we provide a comprehensive explanation of the metrics used to evaluate the alignment between learned LLM rankings and human rankings. These metrics assess the strength of correlations, complexity, and the level of agreement between rankings. Specifically, we discuss five key metrics: Spearman's Rank Correlation Coefficient, Kendall's Rank Correlation Coefficient, Permutation Entropy, Count Inversions, and Longest Increasing Subsequence, detailing their formulations and intuitive interpretations.

i) **Spearman's Rank Correlation Coefficient** $S(\uparrow)$ [28] measures the strength and direction of the monotonic relationship between two ranked variables. It is computed as:

$$S(\hat{\mathcal{R}}, \mathcal{R}^*) = 1 - \frac{6 \sum_{i=1}^{m} d_i^2}{m(m^2-1)}, \tag{11}$$

where $d_i = \text{rank}_{\hat{\mathcal{R}}}(M_i) - \text{rank}_{\mathcal{R}^*}(M_i)$ is the difference between the ranks of LLM M_i in the learned ranking $\hat{\mathcal{R}}$ and the human ranking $\mathcal{R}^*$, and m is the total number of LLMs. A higher Spearman coefficient indicates a stronger correlation between the rankings.

ii) **Kendall's Rank Correlation Coefficient** $\tau(\uparrow)$ [27] evaluates the similarity between two rankings by counting the number of concordant and discordant pairs. It is given by:

$$\tau(\hat{\mathcal{R}}, \mathcal{R}^*) = \frac{C - D}{\frac{1}{2}m(m-1)}, \tag{12}$$

where C represents the number of concordant pairs, and D represents the number of discordant pairs. A pair (M_i, M_j) is concordant if M_i and M_j have the same order in both $\hat{\mathcal{R}}$ and $\mathcal{R}^*$, meaning if $M_i \succ M_j$ in $\hat{\mathcal{R}}$, then $M_i \succ M_j$ in $\mathcal{R}^*$. Conversely, a pair is discordant if their relative order differs between the two rankings. A higher τ value indicates a closer alignment between the rankings.

iii) **Permutation Entropy** $H(\downarrow)$ [8] measures the complexity or randomness of sequences, which is formulated as follows:

$$H(\hat{\mathcal{R}}, \mathcal{R}^*) := -\sum p(\pi) \log p(\pi), \tag{13}$$

where

$$p(\pi) = \frac{\#\{t | 0 \leq t \leq m-k, (M_{t+1}, ..., M_{t+k}) \in \pi\}}{m-k+1}.$$

π denotes different permutations, k is a hyper-parameter recommended to be set to 3 to 7, and we set $k = 3$ in this paper. Intuitively, it samples some subsequences and calculates the entropy for all permutation types. And the lower the permutation entropy in the learned LLM rankings, the closer it is to the ground-truth human rankings.

iv) **Count Inversions** $C(\downarrow)$. Counting inversions [29] aims to measure the degree of disorder or "invertedness" in an array or sequence of elements. We thus define it as follows,

$$C(\hat{\mathcal{R}}, \mathcal{R}^*) := \sum_{M_i, M_j \sim \mathcal{M}} \mathbf{1}\{M_i \succ M_j \wedge i < j\}. \tag{14}$$

Where $\mathbf{1}\{\cdot\}$ is the indicator function that the value is 1 when the condition is met, otherwise it is 0. Intuitively, the fewer inverse pairs in the learned LLM rankings, the closer it is to the ground-truth human rankings.

v) **Longest Increasing Subsequence** $L(\uparrow)$. The longest increasing subsequence aims to find the length of the longest subsequence in a given sequence of elements, where the subsequence is in increasing order. We utilize it to measure the degree of match with human rankings as follows,

$$L(\hat{\mathcal{R}}, \mathcal{R}^*) := \max\left\{dp[i] \mid 1 \leq i \leq m\right\}, \tag{15}$$

where

$$dp[i] = 1 + \max\left\{dp[j] \mid 1 \leq j < i \wedge M_j \prec M_i\right\}.$$

$dp[i]$ represents the length of the longest increasing subsequence that ends with M_i. LIS allows for a nuanced understanding of the degree to which the learned ranking aligns with the ideal human ranking, with a higher LIS length indicating greater alignment.

B Dataset Format

Focusing on the MT-Bench dataset, we demonstrate the ensuing data format utilizing dataset $\mathcal{Q}$. As Figure 7 illustrates, the Question dataset $\mathcal{Q}$ contains "Question id," "Category," "Question," and "Reference." In categories with definitive answers like "reasoning" or "math," the "Reference" field is populated with standard answers; otherwise, it remains blank. Each model M in our pool processes the Question dataset $\mathcal{Q}$ to generate the LLMs answer data $\mathcal{A}$, consisting of "Question id," "Answer id," "Model id," and "Answer." Finally, we combine pairs in $\mathcal{A}$ and appoint judges to evaluate, creating the Answer-Ranking data $\mathcal{D}$, featuring "Question id," "Model 1," "Model 2," "G1 winner," "G2 winner," and "Judge." Here, "G1 winner" and "G2 winner" indicate the outcomes of inputting reversed order responses of Model 1 and Model 2 into the judge model, a method employed to mitigate biases stemming from models' preferences for input order.

C Detailed Prompt for Reviewers

The evaluation prompts, as detailed in Section 2.2.1, are employed during the Peer Review Stage. These prompts are provided to the Reviewer Language Model Systems (LLMs), enabling them to generate evaluative preferences. In our experimental framework, we devised four distinct prompt settings. For each setting, a tailored prompt template was meticulously crafted as illustrated below:

Template for Single-Turn Interaction: This template is designed for single-turn interactions between users and LLMs, where there is no predetermined correct answer. It facilitates open-ended dialogue, allowing for a wide range of user inquiries without the expectation of specific responses.

Referenced Template for Single-Turn Interaction: Tailored for single-turn dialogues between users and LLMs, this template incorporates predefined correct answers. It is particularly suited for

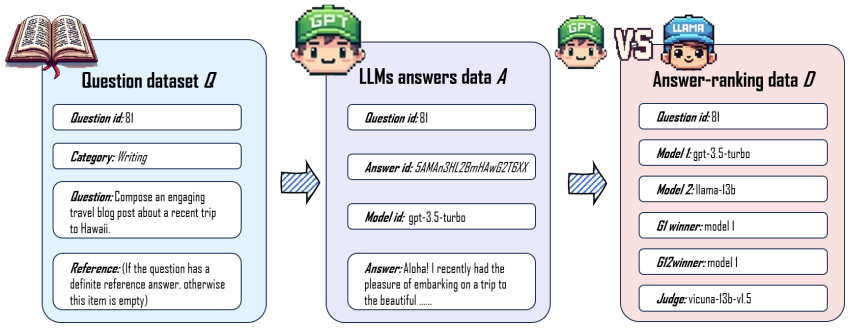


Figure 7: Format of the Question dataset $\mathcal{Q}$, LLMs responses data $\mathcal{A}$, and the Answer-Ranking data $\mathcal{D}$ for Peer Review

interactions involving factual inquiries, such as mathematics or logic problems, where accuracy and reference to correct information are paramount.

Template for Multi-Turn Interaction: This template caters to multi-turn conversations between users and LLMs, without predefined answers. It supports extended interactions, enabling users to explore topics in depth through a series of interconnected questions and responses.

Referenced Template for Multi-Turn Interaction: Designed for multi-turn dialogues with predefined correct answers, this template is ideal for complex inquiries requiring sequential reasoning or problem-solving, such as mathematical computations or logical deductions.

Each template is carefully constructed to match its intended use-case, providing a structured framework that guides the interaction between users and LLMs towards achieving desired outcomes, whether for open-ended exploration or precise problem-solving.

Template for Single-Turn Answer

System prompt: Please act as a judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You do not need to explain, just give your judgment. Output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.
User Question: {question}
Assistant A's Answer: {answer a}
Assistant B's Answer: {answer b}

Referenced Template for Single-Turn Answer

System prompt: Please act as a judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below, with reference to the provided reference answers. You do not need to explain, just give your judgment. Output your final verdict by strictly following this format: "[[A]]"if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.
User Question: {question}
Reference Answer: {reference answer}
Assistant A's Answer: {answer a}
Assistant B's Answer: {answer b}

Template for Multi-Turn Answer

System prompt: Please act as a judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You do not need to explain, just give your judgment. Output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie

Assistant A's Conversation with User:

User: {question 1}
Assistant A: {answer a1}
User: {question 2}
Assistant A: {answer a2}

Assistant B's Conversation with User:

User: {question 1}
Assistant B: {answer b1}
User: {question 2}
Assistant B: {answer b2}

Referenced Template for Multi-Turn Answer

System prompt: Please act as a judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below, in comparison to the reference answers. You do not need to explain, just give your judgment. Output your final verdict by strictly following this format: "[[A]]"if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

Reference Answer

User: {question 1}
Reference answer: {ref answer 1}
User: {question 2}
Reference answer: {ref answer 2}

Assistant A's Conversation with User:

User: {question 1}
Assistant A: {answer a1}
User: {question 2}
Assistant A: {answer a2}

Assistant B's Conversation with User:

User: {question 1}
Assistant B: {answer b1}
User: {question 2}
Assistant B: {answer b2}

D SCORING METHODOLOGY

In Section 2.2.2, Equation 8 delineates the methodology for optimizing scores. Within this framework, the function $\mathbf{1}\{A_i^j > A_i^k\}$ is more precisely defined as $f(A_i^j, A_i^k)$. Additionally, the function $f(A_i^j, A_i^k)$ is not fixed and can be implemented using various computational strategies. We introduce two distinct methodologies in this context: the Elo mechanism and the Rank mechanism.

Within the framework of the Elo mechanism, as specified by Equation 16, the $BASE$ value is set to 10, and the $SCALE$ factor is determined to be 400. This approach facilitates a dynamic adjustment of scores based on the outcomes of pairwise comparisons, allowing for a nuanced reflection of performance variations among models.

Conversely, in the context of the Rank mechanism, as outlined by Equation 17, $rank(j)$ signifies the current ranking of model j, with the constant K assigned a value of 200. This mechanism employs a model's ranking within a predefined hierarchy as a pivotal factor in score calculation, thereby providing a straightforward, yet effective, method for evaluating comparative model performance.

$$f(A_i^j, A_i^k) = \begin{cases} 1 - \frac{1}{1+\text{BASE}^{((G(k)-G(j))/\text{SCALE})}} & \text{if } A_i^j > A_i^k \\ 0.5 - \frac{1}{1+\text{BASE}^{((G(k)-G(j))/\text{SCALE})}} & \text{if } A_i^j = A_i^k \\ 0 - \frac{1}{1+\text{BASE}^{((G(k)-G(j))/\text{SCALE})}} & \text{if } A_i^j < A_i^k \end{cases} \tag{16}$$

$$f(A_i^j, A_i^k) = \begin{cases} 1 + (rank(j) - rank(k))/K & \text{if } A_i^j > A_i^k \\ 0.5 & \text{if } A_i^j = A_i^k \\ 0 & \text{if } A_i^j < A_i^k \end{cases} \tag{17}$$

E Overall Algorithm of Peer Review

The overall algorithm, as delineated in Algorithm 1, encapsulates the comprehensive process outlined in Section 2.2. This sequence commences with "Data Collection and LLMs Pool Construction," progresses through "Answer-Ranking Data Construction Based on Peer Review," advances to "Consistency Optimization," and culminates with the "Unsupervised Elimination Mechanism."

F Complete Experimental Results

In Section 3.4, we both employ elimination mechanisms to cull the weakest LLMs from the 'reviewer queue' during the evaluation process. In Figures 8 and 9, we present the results for the PEN and LIS metrics, where lower PEN scores indicate better performance, and higher LIS scores denote superior performance. It is evident that both the 'PiCO' and PRE approaches demonstrate enhanced performance as the number of eliminated 'reviewers' increases. In most cases, the proposed 'PiCO' method outperforms PRE.

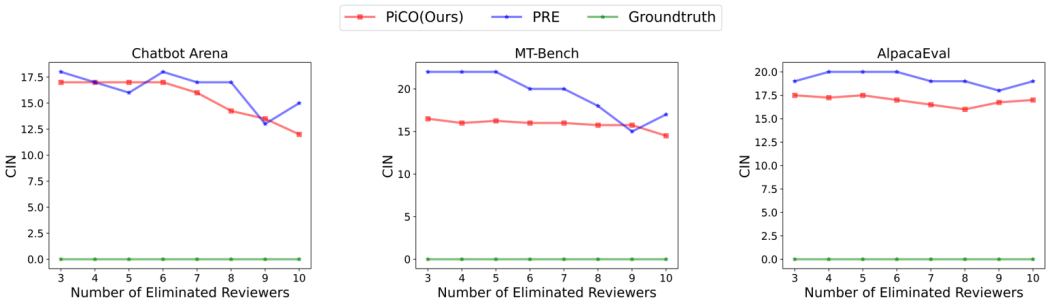


Figure 8: Performance comparison of the PiCO (Ours) and PRE[17] methods on the MT-Bench, Chatbot Arena, and AlpacaEval datasets, with the number of eliminated reviewers on the x-axis. The y-axis is CIN, where lower values indicate better performance.

In Section 3.5, we validate the effectiveness of the *consistency assumption* and compare it with the Average Performance of the Reviewer Queue, i.e., employing a single LLM as the 'reviewer' to evaluate all response pairs and then calculating the average results of all LLMs. The comprehensive results compared with the Reviewer Queue are illustrated in Table4, Figure 10, 11 and 12, revealing that in the full Reviewer Queue, the performance of the vast majority of LLMs is very poor, indicating that the evaluations from most LLMs are noise. However, our 'PiCO' approach nearly matches the evaluative prowess of the pool's most capable LLM, GPT-3.5. Remarkably, given its unsupervised nature, the 'PiCO' method demonstrates the capability to mitigate the influence of noise, reaching the evaluation upper bound (the strongest LLM) within any given unknown LLM pool M, even in the absence of prior ranking information.

G Selected Models and Optimized Ranking

For our analysis, we meticulously selected 15 LLMs spanning a variety of architectures, encompassing both open-source and closed-source models, as detailed in the subsequent table. Our curated

Algorithm 1 Overall Framework Algorithm of Peer Review

Require: Unlabeled dataset $\mathcal{Q}$, Pool of LLMs $\mathcal{M}$, Active LLM pool $\mathcal{M}^* = \mathcal{M}$
Ensure: Consistency-optimized ranking of LLMs $\mathcal{R}^*$

Initialize response matrix $A \leftarrow \emptyset$
for each question $q_i \in \mathcal{Q}$ **do**
  Initialize response vector for question q_i, $A^i \leftarrow \emptyset$
  for each model $m_j \in \mathcal{M}$ **do**
    $A^i_j \leftarrow$ response of model m_j to question q_i
    $A^i \leftarrow A^i \cup \{A^i_j\}$
  end for
  Shuffle A^i to obtain permuted response vector A^i
  $A \leftarrow A \cup \{A^i\}$
end for
Initialize answer-ranking data $D \leftarrow \emptyset$
Initialize model weights vector w with Gaussian distribution
for each permuted response vector A^i **do**
  for each pair of responses (A^j_i, A^k_i) in A^i **do**
    for $s \leftarrow 1$ to 5 **do** ▷ Randomly select 5 models for evaluation
      Evaluate the pair (A^j_i, A^k_i) with model m_s
      $D \leftarrow D \cup \{(A^j_i, A^k_i, > w^s)\}$
    end for
  end for
end for
Initialize scores G_j for each model $m_j \in \mathcal{M}$ to the Elo initial score
repeat
  while not converged **do**
    for each model $m_j \in \mathcal{M}$ **do**
      Compute G_j using updated formula:
      $G_j = \sum_i \sum_{k \neq j} \sum_{s \neq k, s \neq j} \mathbf{1}\{A^j_i, A^k_i\} \times w^s \quad (A^j_i, A^k_i, > w^s, s \in \mathcal{M}^*) \in D$
    end for
    Update weight vector w to maximize the consistency of w and G
  end while
  Sort $\mathcal{M}^*$ by G_j to identify $\mathcal{M}_{min}$, the lowest-scoring model
  if size of $\mathcal{M}^*$ > threshold **then**
    Remove $\mathcal{M}_{min}$ from $\mathcal{M}^*$
  end if
until size of $\mathcal{M}^*$ < threshold
Compute the final ranking $\mathcal{R}^*$ based on the optimized scores G_j
return $\mathcal{R}^*$

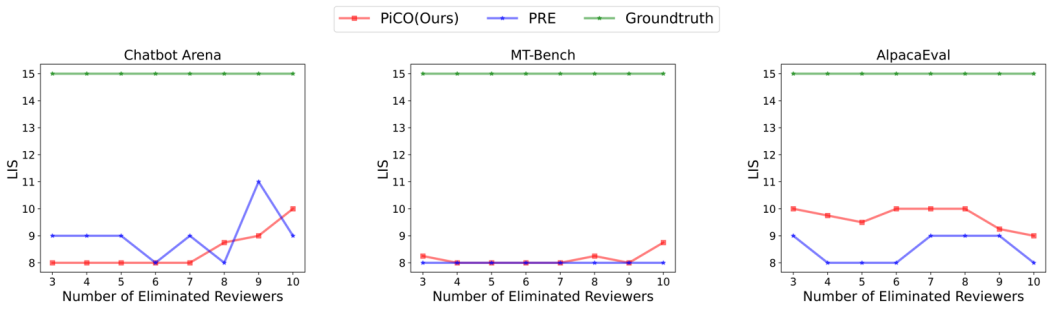


Figure 9: Performance comparison of the PiCO (Ours) and PRE[17] methods on the MT-Bench, Chatbot Arena, and AlpacaEval datasets, with the number of eliminated reviewers on the x-axis. The y-axis is LIS, where upper values indicate better performance.

Table 4: Comparison of performance across three datasets using Unsupervised methods versus using single models in reviewer queue.

Methods	MT-Bench			Chatbot Arena			AlpacaEval		
	PEN (↓)	CIN(↓)	LIS(↑)	PEN (↓)	CIN(↓)	LIS(↑)	PEN (↓)	CIN(↓)	LIS(↑)
Gpt-3.5	**0.97**	**12.00**	**10.00**	**0.85**	**11.00**	**11.00**	**1.15**	**16.00**	**9.00**
Guanaco-33B	1.25	21.00	8.00	1.50	28.00	7.00	1.26	20.00	9.00
Vicuna-13B	1.31	20.00	7.00	1.27	23.00	8.00	1.20	17.00	8.00
WizardLM-13B	1.15	17.00	9.00	1.27	19.00	8.00	1.17	17.00	9.00
Vicuna-7B	1.27	21.00	8.00	1.30	20.00	7.00	1.34	23.00	8.00
Koala-13B	1.67	43.00	6.00	1.34	23.00	8.00	1.54	31.00	7.00
gpt4all-13B	1.74	45.00	6.00	1.60	35.00	6.00	1.73	42.00	6.00
Mpt-7B	1.67	39.00	6.00	1.72	52.00	6.00	1.63	34.00	7.00
Oass-pythia-12B	1.77	50.00	5.00	1.74	42.00	5.00	1.70	47.00	6.00
Alpaca-13B	1.77	49.00	7.00	1.60	73.00	4.00	1.63	34.00	7.00
FastChat-T5-3B	1.45	29.00	7.00	1.53	30.00	7.00	1.30	22.00	7.00
ChatGLM-6B	1.59	33.00	7.00	1.71	55.00	5.00	1.63	34.00	6.00
StableLM-7B	1.68	63.00	5.00	1.75	44.00	5.00	1.72	56.00	4.00
Dolly-12B	1.76	46.00	6.00	1.57	71.00	6.00	1.75	54.00	6.00
LLaMA-13B	1.60	35.00	7.00	1.76	56.00	6.00	1.70	50.00	5.00
Average Performance of All Review LLMs	1.51	34.87	6.93	1.50	38.80	6.60	1.50	33.13	6.93
PRD[30]	1.15	17.00	8.00	1.15	17.00	8.00	1.21	19.00	9.00
PRE[17]	1.17	17.00	8.00	1.07	15.00	9.00	1.18	19.00	8.00
PiCO (Ours)	1.01	14.50	8.75	0.94	12.00	10.00	1.17	17.00	9.00

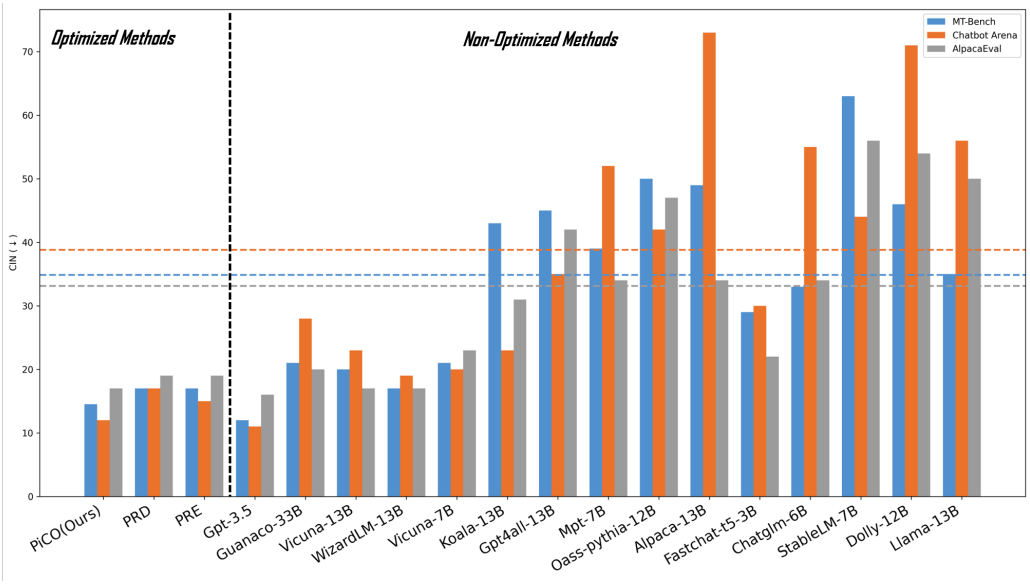


Figure 10: Comparison of performance on the CIN metric across three datasets using Unsupervised methods versus using single models, with Unsupervised methods on the left and Supervised methods on the right. The dotted line represents the average value using single models.

selection features prominent LLMs including the closed-source "gpt-3.5-turbo," "chatglm" which is predicated on the encoder-decoder framework, "fastchat-t5-3b" that leverages Google's T5 (Text-to-Text Transfer Transformer) architecture, and "llama-13b" founded on the GPT architectural principles.

We have comprehensively detailed the ranking outcomes across three distinct datasets for our comparative analysis, incorporating the optimized model rankings, names, and their respective scores. As delineated in Appendix D, the PiCO (Ours) is capable of employing various scoring mechanisms, thereby facilitating the presentation of ranking outcomes on three datasets utilizing both the Elo and Rank mechanisms. Furthermore, we have also enumerated the ranking results for PRD and PRE methodologies across the three datasets, offering a holistic view of the competitive landscape.

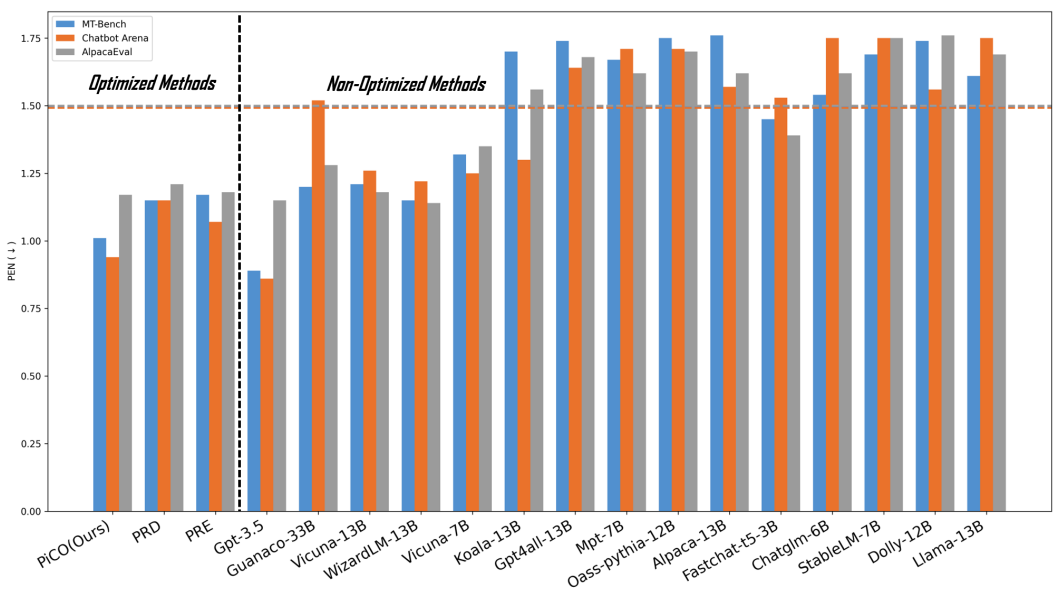


Figure 11: Comparison of performance on the PEN metric across three datasets using Unsupervised methods versus using single models, with Unsupervised methods on the left and Supervised methods on the right. The dotted line represents the average value using single models.

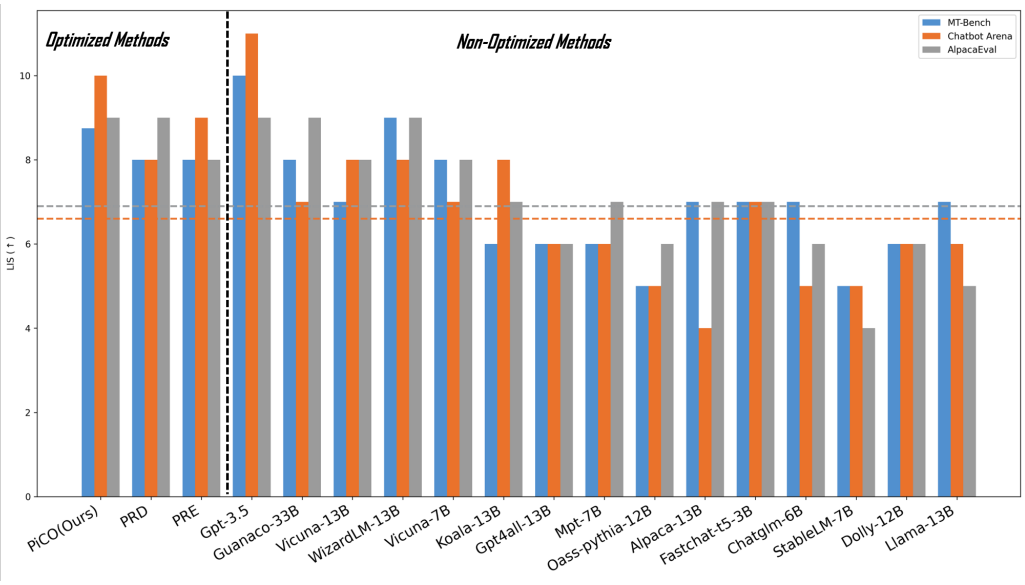


Figure 12: Comparison of performance on the LIS metric across three datasets using Unsupervised methods versus using single models, with Unsupervised methods on the left and Supervised methods on the right. The dotted line represents the average value using single models.

G.1 PiCO

Grade-Elo-Chatbot

#1 **Gpt-3.5** | Grade: 9205.162109375
#2 **WizardLM-13B** | Grade: 9143.46875
#3 **Guanaco-33B** | Grade: 5886.92626953125
#4 **Vicuna-7B** | Grade: 5368.9462890625
#5 **Vicuna-13B** | Grade: 5216.79541015625
#6 **Koala-13B** | Grade: 3545.1171875 | Eliminated
#7 **Mpt-7B** | Grade: 962.99462890625 | Eliminated
#8 **Gpt4all-13B** | Grade: 652.4602661132812 | Eliminated
#9 **Chatglm-6B** | Grade: 417.1375427246094 | Eliminated
#10 **Oasst-pythia-12B** | Grade: -898.2676391601562 | Eliminated
#11 **Fastchat-t5-3B** | Grade: -1251.7183837890625 | Eliminated
#12 **StableLM-7B** | Grade: -2232.66943359375 | Eliminated
#13 **Dolly-12B** | Grade: -3163.540283203125 | Eliminated
#14 **Llama-13B** | Grade: -3648.37841796875 | Eliminated
#15 **Alpaca-13B** | Grade: -14204.3984375 | Eliminated

Grade-Elo-AlpacaEval

#1 **WizardLM-13B** | Grade: 8662.7158203125
#2 **Vicuna-13B** | Grade: 5586.46630859375
#3 **Guanaco-33B** | Grade: 5445.341796875
#4 **Vicuna-7B** | Grade: 5374.2314453125
#5 **Gpt-3.5** | Grade: 4845.91552734375
#6 **Koala-13B** | Grade: 4338.77783203125 | Eliminated
#7 **Chatglm-6B** | Grade: 2293.4208984375 | Eliminated
#8 **Gpt4all-13B** | Grade: 2080.511962890625 | Eliminated
#9 **Mpt-7B** | Grade: 1694.4945068359375 | Eliminated
#10 **Fastchat-t5-3B** | Grade: 1371.94287109375 | Eliminated
#11 **Oasst-pythia-12B** | Grade: -665.8685302734375 | Eliminated
#12 **StableLM-7B** | Grade: -1343.5838623046875 | Eliminated
#13 **Dolly-12B** | Grade: -5377.13427734375 | Eliminated
#14 **Llama-13B** | Grade: -5847.59130859375 | Eliminated
#15 **Alpaca-13B** | Grade: -13459.6162109375 | Eliminated

Grade-Elo-MT_Bench

#1 **WizardLM-13B** | Grade: 2178.10302734375
#2 **Vicuna-13B** | Grade: 1720.1114501953125
#3 **Guanaco-33B** | Grade: 1704.1832275390625
#4 **Vicuna-7B** | Grade: 1659.2799072265625
#5 **Gpt-3.5** | Grade: 1535.8819580078125
#6 **Mpt-7B** | Grade: 1338.5235595703125 | Eliminated
#7 **Koala-13B** | Grade: 1267.9747314453125 | Eliminated
#8 **Chatglm-6B** | Grade: 1011.7701416015625 | Eliminated
#9 **Gpt4all-13B** | Grade: 976.5963745117188 | Eliminated
#10 **Oasst-pythia-12B** | Grade: 779.3573608398438 | Eliminated
#11 **StableLM-7B** | Grade: 512.1678466796875 | Eliminated
#12 **Alpaca-13B** | Grade: 334.9879455566406 | Eliminated
#13 **Fastchat-t5-3B** | Grade: 303.5980529785156 | Eliminated
#14 **Dolly-12B** | Grade: 72.63818359375 | Eliminated
#15 **Llama-13B** | Grade: -395.19921875 | Eliminated

Grade-Rank-Chatbot

#1 **WizardLM-13B** | Grade: 0.30809280276298523
#2 **Gpt-3.5** | Grade: 0.293962299823761
#3 **Guanaco-33B** | Grade: 0.28587597608566284
#4 **Vicuna-7B** | Grade: 0.28212910890579224
#5 **Vicuna-13B** | Grade: 0.27900218963623047
#6 **Koala-13B** | Grade: 0.2672431766986847 | Eliminated
#7 **Mpt-7B** | Grade: 0.2500302195549011 | Eliminated
#8 **Gpt4all-13B** | Grade: 0.24746862053871155 | Eliminated
#9 **Chatglm-6B** | Grade: 0.2466953843832016 | Eliminated
#10 **Oasst-pythia-12B** | Grade: 0.23637069761753082 | Eliminated
#11 **Fastchat-t5-3B** | Grade: 0.2350562959909439 | Eliminated
#12 **StableLM-7B** | Grade: 0.22843806445598602 | Eliminated
#13 **Dolly-12B** | Grade: 0.22219440340995789 | Eliminated
#14 **Llama-13B** | Grade: 0.2165679931640625 | Eliminated
#15 **Alpaca-13B** | Grade: 0.13975904881954193 | Eliminated

Grade-Rank-AlpacaEval

#1 **WizardLM-13B** | Grade: 0.4019235074520111
#2 **Vicuna-13B** | Grade: 0.36745429039001465
#3 **Guanaco-33B** | Grade: 0.3664878010749817
#4 **Vicuna-7B** | Grade: 0.36541733145713806
#5 **Gpt-3.5** | Grade: 0.36000365018844604
#6 **Koala-13B** | Grade: 0.3544933795928955 | Eliminated
#7 **Chatglm-6B** | Grade: 0.3319571018218994 | Eliminated
#8 **Gpt4all-13B** | Grade: 0.3306528627872467 | Eliminated
#9 **Mpt-7B** | Grade: 0.32641729712486267 | Eliminated
#10 **Fastchat-t5-3B** | Grade: 0.32173293828964233 | Eliminated
#11 **Oasst-pythia-12B** | Grade: 0.2999681532382965 | Eliminated
#12 **StableLM-7B** | Grade: 0.2932431995868683 | Eliminated
#13 **Dolly-12B** | Grade: 0.24777530133724213 | Eliminated
#14 **Llama-13B** | Grade: 0.24381506443023682 | Eliminated
#15 **Alpaca-13B** | Grade: 0.16114839911460876

Grade-Rank-MT_Bench

#1 **WizardLM-13B** | Grade: 0.2994651198387146
#2 **Vicuna-13B** | Grade: 0.2809261679649353
#3 **Guanaco-33B** | Grade: 0.2767307460308075
#4 **Vicuna-7B** | Grade: 0.2758147716522217
#5 **Gpt-3.5** | Grade: 0.27261608839035034
#6 **Mpt-7B** | Grade: 0.26338690519332886 | Eliminated
#7 **Koala-13B** | Grade: 0.2613368630409241 | Eliminated
#8 **Gpt4all-13B** | Grade: 0.24908888339996338 | Eliminated
#9 **Chatglm-6B** | Grade: 0.24898234009742737 | Eliminated
#10 **Oasst-pythia-12B** | Grade: 0.2415400892496109 | Eliminated
#11 **StableLM-7B** | Grade: 0.2299075722694397 | Eliminated
#12 **Alpaca-13B** | Grade: 0.22171474993228912 | Eliminated
#13 **Fastchat-t5-3B** | Grade: 0.221677765250206 | Eliminated
#14 **Dolly-12B** | Grade: 0.21185410022735596 | Eliminated
#15 **Llama-13B** | Grade: 0.192665234208107 | Eliminated

G.2 PRD

PRD-Chatbot

#1 **WizardLM-13B** | Grade: 5565.28271484375
#2 **Gpt-3.5** | Grade: 4613.22900390625
#3 **Guanaco-33B** | Grade: 3423.588134765625
#4 **Vicuna-7B** | Grade: 2985.4892578125
#5 **Vicuna-13B** | Grade: 2972.15673828125
#6 **Koala-13B** | Grade: 2237.70751953125
#7 **Chatglm-6B** | Grade: 875.373779296875
#8 **Mpt-7B** | Grade: 602.46923828125
#9 **Gpt4all-13B** | Grade: 356.06243896484375
#10 **Fastchat-t5-3B** | Grade: 184.89663696289062
#11 **Dolly-12B** | Grade: 52.10746765136719
#12 **Oasst-pythia-12B** | Grade: -307.49908447265625
#13 **StableLM-7B** | Grade: -691.4453735351562
#14 **Llama-13B** | Grade: -848.1654052734375
#15 **Alpaca-13B** | Grade: -7020.923828125

PRD-AlpacaEval

#1 **WizardLM-13B** | Grade: 5469.75634765625
#2 **Guanaco-33B** | Grade: 3707.014892578125
#3 **Vicuna-13B** | Grade: 3618.63427734375
#4 **Vicuna-7B** | Grade: 3569.389892578125
#5 **Gpt-3.5** | Grade: 3197.755615234375
#6 **Koala-13B** | Grade: 2893.642578125
#7 **Chatglm-6B** | Grade: 1847.1300048828125
#8 **Fastchat-t5-3B** | Grade: 1585.66943359375
#9 **Gpt4all-13B** | Grade: 1561.145751953125
#10 **Mpt-7B** | Grade: 1332.3753662109375
#11 **StableLM-7B** | Grade: -33.00855255126953
#12 **Oasst-pythia-12B** | Grade: -92.68387603759766
#13 **Dolly-12B** | Grade: -3013.588623046875
#14 **Llama-13B** | Grade: -3211.0302734375
#15 **Alpaca-13B** | Grade: -7432.3701171875

PRD-MT_Bench

#1 **WizardLM-13B** | Grade: 1811.64697265625
#2 **Vicuna-13B** | Grade: 1537.8084716796875
#3 **Guanaco-33B** | Grade: 1481.1739501953125
#4 **Vicuna-7B** | Grade: 1401.5194091796875
#5 **Gpt-3.5** | Grade: 1272.8072509765625
#6 **Mpt-7B** | Grade: 1186.5518798828125
#7 **Chatglm-6B** | Grade: 1166.6246337890625
#8 **Koala-13B** | Grade: 1124.2513427734375
#9 **Gpt4all-13B** | Grade: 871.2874755859375
#10 **Oasst-pythia-12B** | Grade: 855.3653564453125
#11 **StableLM-7B** | Grade: 782.702880859375
#12 **Fastchat-t5-3B** | Grade: 636.966064453125
#13 **Alpaca-13B** | Grade: 414.9374694824219
#14 **Dolly-12B** | Grade: 377.5018005371094
#15 **Llama-13B** | Grade: 78.90127563476562

G.3 PRE

PRE-Chatbot

#1 **WizardLM-13B** | Grade: 1113.7034715479742
#2 **Gpt-3.5** | Grade: 1076.1116664199608
#3 **Guanaco-33B** | Grade: 1067.441581415147
#4 **Vicuna-13B** | Grade: 1057.702184441485
#5 **Vicuna-7B** | Grade: 1043.4840340151043
#6 **Koala-13B** | Grade: 1030.4455842017508 | Eliminated
#7 **Chatglm-6B** | Grade: 1012.4487557424748 | Eliminated
#8 **Mpt-7B** | Grade: 1000.487230109001 | Eliminated
#9 **Gpt4all-13B** | Grade: 1000.4111397038492 | Eliminated
#10 **Fastchat-t5-3B** | Grade: 992.3732179832363 | Eliminated
#11 **Oasst-pythia-12B** | Grade: 977.5217305871272 | Eliminated
#12 **StableLM-7B** | Grade: 970.3665926795535 | Eliminated
#13 **Llama-13B** | Grade: 929.6268868888149 | Eliminated
#14 **Dolly-12B** | Grade: 929.1943463130976 | Eliminated
#15 **Alpaca-13B** | Grade: 798.6815779514078 | Eliminated

PRE-AlpacaEval

#1 **WizardLM-13B** | Grade: 1127.822808841937
#2 **Vicuna-7B** | Grade: 1077.1823389450524
#3 **Vicuna-13B** | Grade: 1075.4338443616266
#4 **Guanaco-33B** | Grade: 1074.8043135229418
#5 **Gpt-3.5** | Grade: 1065.305736105376
#6 **Gpt4all-13B** | Grade: 1039.4091630861865 | Eliminated
#7 **Koala-13B** | Grade: 1038.205749976473 | Eliminated
#8 **Mpt-7B** | Grade: 1032.2893401162178 | Eliminated
#9 **Chatglm-6B** | Grade: 1027.1937496918501 | Eliminated
#10 **Fastchat-t5-3B** | Grade: 992.3481168791307 | Eliminated
#11 **StableLM-7B** | Grade: 979.3894141445692 | Eliminated
#12 **Oasst-pythia-12B** | Grade: 940.6438439723215 | Eliminated
#13 **Dolly-12B** | Grade: 886.1412110662756 | Eliminated
#14 **Llama-13B** | Grade: 880.0797724297793 | Eliminated
#15 **Alpaca-13B** | Grade: 763.7505968602533 | Eliminated

PRE-MT_Bench

#1 **WizardLM-13B** | Grade: 1065.5843776639435
#2 **Vicuna-13B** | Grade: 1062.3934138040302
#3 **Guanaco-33B** | Grade: 1052.2206466556906
#4 **Vicuna-7B** | Grade: 1035.1112817247572
#5 **Gpt-3.5** | Grade: 1029.8316754711038
#6 **Koala-13B** | Grade: 1024.9307662983267 | Eliminated
#7 **Chatglm-6B** | Grade: 1020.5238960907612 | Eliminated
#8 **Mpt-7B** | Grade: 1014.0683255081057 | Eliminated
#9 **Gpt4all-13B** | Grade: 991.7142639623017 | Eliminated
#10 **StableLM-7B** | Grade: 979.8443261256327 | Eliminated
#11 **Oasst-pythia-12B** | Grade: 977.9930430111322 | Eliminated
#12 **Fastchat-t5-3B** | Grade: 953.0776159143571 | Eliminated
#13 **Alpaca-13B** | Grade: 949.129770731626 | Eliminated
#14 **Dolly-12B** | Grade: 928.511065779112 | Eliminated
#15 **Llama-13B** | Grade: 915.0655312591185 | Eliminated