Mitigating Judgment Preference Bias in Large Language Models through Group-Based Polling

Shuliang Liu[1*], Zhipeng Xu[1*], Zhenghao Liu[1†], Yukun Yan[2], Minghe Yu[1], Yu Gu[1], Chong Chen[3], Huiyuan Xie[2] and Ge Yu[1]
[1]School of Computer Science and Engineering, Northeastern University, China
[2]Department of Computer Science and Technology, Institute for AI, Tsinghua University, China
Beijing National Research Center for Information Science and Technology, China
[3]Huawei Cloud Business Unit, China

Abstract

Large Language Models (LLMs) as automatic evaluators, commonly referred to as "LLM-as-a-Judge", have also attracted growing attention. This approach plays a vital role in aligning LLMs with human judgments, providing accurate and reliable assessments. However, LLM-based judgment models often exhibit judgment preference bias during the evaluation phase, tending to favor responses generated by themselves, undermining the reliability of their judgments. This paper introduces the **G**roup-Bas**e**d Polli**n**g Opt**i**m**i**zation (Genii), an unsupervised multi-agent collaborative optimization framework that mitigates the inherent judgment preference bias of judgment models. Specifically, Genii integrates various LLM-based judgment models into a multi-agent system and simulates the interactive client-server polling mechanism to optimize each client agent unsupervisedly. Our experiments demonstrate that Genii outperforms supervised models trained on annotated judgment data, while requiring no human-labeled annotations. Genii consistently improves performance across different client agents during the polling, even when weaker models act as server agents. Further analysis reveals that Genii effectively mitigates judgment preference bias of LLM-based judgment models, demonstrating its effectiveness. All codes are available at https://github.com/NEUIR/Genii.

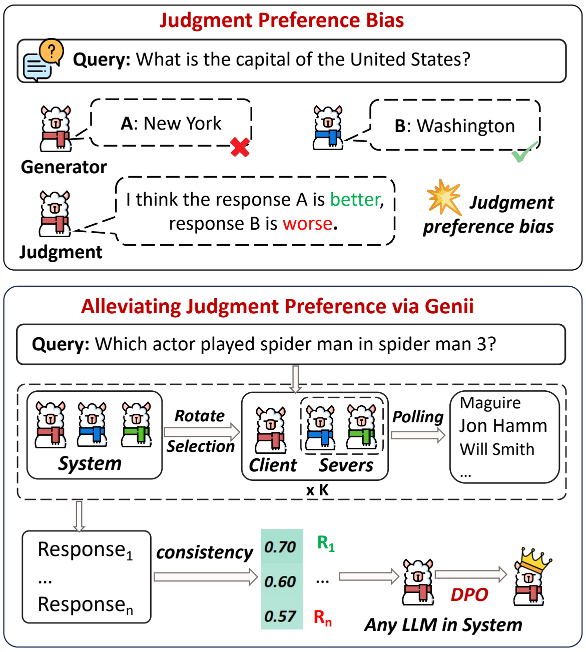


Figure 1: The Framework of Genii. It alleviates judgment preference of LLMs during the evaluation process.

1 Introduction

Large Language Models (LLMs), such as GPT-4 (OpenAI, 2023) and Gemini (Team et al., 2023), have demonstrated impressive capabilities across a wide range of downstream Natural Language Processing (NLP) tasks (Zhao et al., 2023; Wei et al., 2022; Hou et al., 2024; Huang et al., 2025; Yao et al., 2025). Beyond positing LLMs as answer generators, recent research has explored using LLMs as evaluators to automatically assess response quality of models, a paradigm known as LLM-as-a-Judge (Zheng et al., 2023; Gu et al., 2024; Tan et al., 2024; Li et al., 2025). This approach offers a scalable and cost-effective alternative to human annotation, demonstrating promising alignment with human judgments and has been widely used in the assessment of response quality in language models (Kim et al., 2024a; Wu et al., 2024; Yang et al., 2024). Despite their notable success, LLM-based evaluators often face challenges in judgment preference bias–a systematic tendency of LLMs to favor responses generated by themselves or aligned with their generation patterns (Bitton et al., 2023; Wataoka et al., 2024; Gu et al., 2024; Chen et al., 2025a), undermining the reliability of their judgments (Koo et al., 2023).

To mitigate judgment preference bias of LLM-

[*] indicates equal contribution.
[†] indicates corresponding author.

based evaluators and provide more accurate assessments, researchers have made extensive efforts (Gu et al., 2024; Wang et al., 2024; He et al., 2024). Previous studies have focused on enhancing the reasoning capabilities of LLM-based evaluators to generate more rational judgments (Yang et al., 2025). Specifically, they prompt LLMs to generate chain-of-thought reasoning before making judgments (Wei et al., 2022; Gallegos et al., 2024), engage in self-consistency decoding (Wang et al., 2023), or apply self-debiasing to correct potentially biased decisions (Ganguli et al., 2023). Nevertheless, LLMs often still overlook their own biased behavior (Wang et al., 2023), avoiding key comparison questions and giving vague statements that fail to identify or correct their underlying preference bias (Yang et al., 2025). Recent work has shifted away from relying on a single evaluation model, instead harnessing collaboration among multiple models to obtain more objective and reliable judgments (Park et al., 2024; Liu et al., 2025b). While such ensemble-based methods improve robustness, they often rely on stronger models to guide weaker ones. However, allowing knowledge to flow in both directions, so that weaker models can help improve stronger models without hurting their performance, remains a major challenge (Zhang et al., 2018).

In this paper, we propose **G**roup-Bas**e**d Polli**n**g Opt**i**m**i**zation (Genii), an unsupervised multi-agent collaborative optimization framework that encourages multiple LLM judgment models to interact with each other to mitigate their own inherent judgment preference bias. As illustrated in Figure 1, instead of introducing additional preference datasets, Genii integrates various LLM-based judgment models into a multi-agent system and simulates the interactive client-server polling mechanism to optimize each agent unsupervisedly. Genii first selects one model in the group as the client agent, responsible for initiating the polling request. The remaining models act as server agents to evaluate and provide feedback on every client request. Then Genii computes the group consistency score for each request. By rotating the client agent to collect responses from all agents, Genii aggregates the group consistency scores to curate preference pairs, thereby optimizing each agent towards the collective preference of the group and mitigating individual preference bias.

Our experimental results demonstrate that Genii outperforms supervised models trained on annotated judgment data, without requiring human-labeled annotations, achieving a significant improvement over vanilla LLMs. Genii consistently improves model performance regardless of which model serves as the client agent during the polling. Remarkably, even when weaker models act as server agents, their feedback contributes meaningfully to the optimization of stronger models. Further analysis shows that with the improvement of LLMs' generation capabilities, their judgment performance also enhances, demonstrating the effectiveness of Genii. Additionally, Genii effectively mitigates the preference bias of LLM-based judgment models toward their incorrect answers and enables them to yield more reliable judgments.

2 Related Work

Large Language Models (LLMs) as automatic evaluators, referred to as “LLM-as-a-Judge”, have proven their effectiveness in various tasks (Zhang et al., 2024b,a). By prompting LLMs to determine which response is better according to established evaluation dimensions, LLMs can yield judgments that align with human preferences (Lin and Chen, 2023). Most existing LLM-based judgment models rely on the APIs of closed-source models such as GPT-4 to score or rank candidate responses, as well as to provide detailed explanations (Zheng et al., 2023). However, the use of these APIs incurs significant costs and reduces reproducibility (Gu et al., 2024). Recent works further fine-tune LLMs specifically for judgment tasks, demonstrating that with appropriate training, such models can achieve high reliability as evaluators (Zhu et al., 2023; Kim et al., 2024a,b). Subsequent work explores self-improvement approaches that enhance LLMs' judgment ability without relying on additional supervision to reduce cost (Liu et al., 2025a). However, the reliability of LLMs as evaluators remains unclear, as they may be affected by judgment preference bias (Zheng et al., 2023).

To improve the reliability of LLM-based judgment models, it is crucial to explore effective strategies for mitigating judgment preference bias. Some works focus on enhancing the reasoning abilities of LLM-based judgment models, enabling them to generate more rational evaluations (Yang et al., 2025). For instance, Gallegos et al. (2024) proposes a self-debiasing approach in which the model is prompted to make two rounds of judgments, with the second round intended to identify and correct potential biases in the initial judgment. In addition, Self-Consistency (Wang et al., 2023) mitigates bias by increasing the number of reasoning attempts for the same instruction and aggregating them via majority voting, while Long Reasoning (Chen et al., 2025a) extends the length of the reasoning chain. Both methods have demonstrated great potential to reduce bias during the evaluation phase. However, prompt-based self-debiasing methods often lead judgment models to overlook their actual biased behaviors, and increasing the number or length of reasoning steps incurs higher computational cost (Yang et al., 2025; Hassid et al., 2025).

Moreover, recent advances have shifted evaluation from single-model paradigms toward multi-model collaborative judgment frameworks to achieve more objective and reliable evaluations. Debate-based approaches utilize multiple LLMs to argue for opposing answers, with a superior LLM reviewing the discussion and deciding which answer is better (Khan et al., 2024). Ensemble methods further reduce preference bias by averaging scores across multiple judgment models or down-weighting cases where the individual judgment model shows unusually high confidence (Wataoka et al., 2024; Chen et al., 2025b). However, although these approaches help alleviate the preference bias inherent to individual judgment models, they typically rely on stronger models to guide weaker ones, thereby neglecting the potential complementary contributions of the weaker models. In contrast to existing works, Genii enables stronger models to achieve further improvement by leveraging feedback from weaker models.

3 Methodology

In this section, we introduce our **G**roup-Bas**e**d Polli**n**g Opt**i**m**i**zation (Genii) framework. We first introduce the preliminaries of LLM-based judgement models (Sec. 3.1). Then we describe the detailed mechanism of Genii, which optimizes each model in the group through polling interactions (Sec. 3.2).

3.1 Preliminary of LLM-as-a-Judge

Given a query q and two candidate answers c_1 and c_2, the goal of LLM-as-a-Judge is to identify the higher-quality answer to the query from the two candidates.

LLM-based judgment model, guided by a predefined evaluation prompt p, compares the quality of two candidate answers c_1 and c_2. The model produces a judgment process e as well as a final judgment result ϵ. This assessment process can be formulated as:

$$(e, \epsilon) = \mathrm{LLM}(p, q, c_1, c_2), \tag{1}$$

where ϵ represents one of three possible results: c_1 is better, c_2 is better, or the two answers are of comparable quality (tie).

Current approaches for optimizing LLM-based judgment models primarily rely on leveraging guidance from stronger judgment models to improve weaker ones. However, they often fail to fully exploit the potential complementary contributions of the weaker models, which could also help further enhance the judgment performance of stronger models.

3.2 Optimizing LLM-Based Judgment Models through Group-Based Polling

In this subsection, we introduce the **G**roup-Bas**e**d Polli**n**g Opt**i**m**i**zation (Genii) method, which integrates multiple LLM-based judgment models into a multi-agent system and simulates interactive client-server polling to optimize every agent unsupervisely.

Client-Server Agent System. To optimize the LLM-based judgment models, Genii first instantiates a collaborative multi-agent system composed of m heterogeneous agents, which can be formulated as:

$$\mathcal{V} = \{V_1, V_2, \ldots, V_m\}, \tag{2}$$

where each agent is implemented with a different LLM-based judgment model, which has a unique inherent judgment preference bias. Before the agents start communicating, one of the agents $V_i \in \mathcal{V}$ is configured as the client, which generates responses and initiates the polling process during the communication. While the remaining agents $V_j \in \mathcal{V}$ $(j \neq i)$ act as servers to process the polling request, evaluate the response from the client independently, and provide feedback based on their own criteria. Then Genii computes the group consistency score for each request. By rotating the client agent to collect responses from all agents, Genii aggregates the group consistency scores to curate preference pairs for optimizing each agent in the system.

Polling Sampling. Once the client agent V_i is configured, it receives a query q and engages in response–request interactions repeatedly. For the k-th interaction, the client agent first generates a

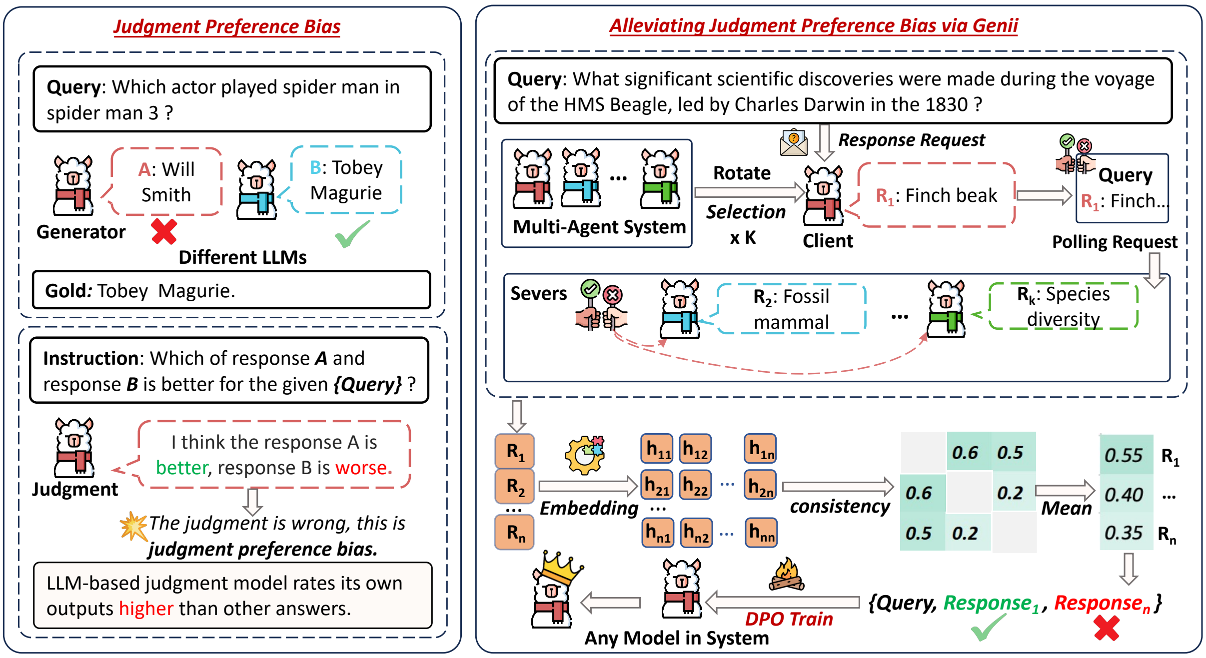


Figure 2: The Framework of Our Genii Method.

response $y_k^{(i)}$ based on q, which can be represented as:

$$y_k^{(i)} = V_i(q), \tag{3}$$

where the original query q and the response $y_k^{(i)}$ generated by the client agent constitute a polling request r_k. We can formulate the content of the polling request r_k as:

$$r_k^{(i)} = \{y_k^{(i)}, q\}, \tag{4}$$

which is subsequently broadcast to the remaining server agents in the system. Each server agent $V_j \in \mathcal{V}$ ($j \neq i$) conducts a response–feedback operation: upon receiving the polling request $r_k^{(i)}$, it generates a response for the query based on its own criteria, and calculates a client-server consistency score as feedback, which can be formulated as:

$$s(i,k) = \cos\left(\mathrm{Emb}\left(y_k^{(i)}\right), \mathrm{Emb}\left(V_j(q)\right)\right), \tag{5}$$

where the Emb denotes a text encoder that can map the response into a semantic embedding space. To assess the group consistency between the client and the group, for each client agent response $y_k^{(i)}$, we integrate the consistency scores from all the server agents in the group and compute the group-based consistency as:

$$S(i,k) = \frac{1}{|\mathcal{V}|-1}\sum_{j\neq i} s(i,k), \tag{6}$$

where the score reflects how well the client's response $y_k^{(i)}$ aligns with the implicit preferences of the server agents in the system.

Group-based Consistency Optimization. For the query q, the client agent repeats the response–request operation until the maximum number of sampling K is reached. By rotating the client role across the group, we can obtain a set of responses $\mathcal{Y}$ generated by all agents, each associated with a group-based consistency score. Based on these group-based consistency scores, we select the highest- and lowest-consistency responses from all responses $\mathcal{Y}$ to form preference pairs (y^+, y^-). These pairs are then used to train each agent in the system via Direct Preference Optimization (DPO) (Rafailov et al., 2023):

$$L = -\log\sigma\left(\beta\log\frac{V_i(y^+ \mid q)}{V_i^{\mathrm{ref}}(y^+ \mid q)} - \beta\log\frac{V_i(y^- \mid q)}{V_i^{\mathrm{ref}}(y^- \mid q)}\right), \tag{7}$$

where V_i denotes the agent model under optimization, V_i^{ref} denotes the reference model and σ is the sigmoid function. All models benefit from diverse group feedback and gradually converge toward more consistent and unbiased evaluations.

4 Experimental Methodology

This section describes the datasets, evaluation metrics, and baselines used in our experiments. More implementation details are shown in Appendix A.2.

Datasets. We describe the datasets used by Genii for training and evaluation.

Training. Genii uses instructions from instruction-following datasets and a small amount of QA data to optimize the multi-agent system. Specifically, we sample instructions from Evol-Instruct (Xu et al., 2024) and sample QA pairs from widely used QA datasets. In total, we construct a dataset comprising 7,992 unique samples for unsupervised training. Additionally, we randomly sample 500 samples from UltraFeedback (Cui et al., 2024) to construct the development dataset. More detailed training data statistics are shown in Appendix A.2.

Evaluation. To assess the performance of Genii. We select multiple tasks from prior work (Li et al., 2024b; Liu et al., 2025b), including evaluating the accuracy (UltraFeedback, MTbench (Zheng et al., 2023), AutoJ (Li et al., 2024a), Preferencebench (Kim et al., 2024b), and Rewardbench (Lambert et al., 2024)) and assessing the judgment preference bias (NQ (Kwiatkowski et al., 2019), HotpotQA (Yang et al., 2018), and TriviaQA (Joshi et al., 2017)). For NQ, HotpotQA, and TriviaQA, we build the preference label following the prior work (Chen et al., 2025a). The detailed data statistics are shown in Table 1.

Evaluation Metrics. We employ accuracy and Harmful Self-Preference Propensity (HSPP) as evaluation metrics to evaluate the judgment performance of LLM-based judgment models. The prompt templates used for evaluation are shown in Appendix A.6.

Accuracy. We follow Köppel et al. (2019) and use accuracy as the evaluation metric. In judgment benchmarks, each example consists of a query q along with two annotated candidate responses, y_1 and y_2, corresponding to the chosen and rejected answers, respectively. We denote $\mathcal{Y}^* = \{y_1^*, y_2^*, \dots, y_n^*\}$ as the set of all chosen responses. For each example, the judgment model selects the better answer between the two candidates. Let $\mathcal{R} = \{r_1, r_2, \dots, r_n\}$ represents the set of better answers. Finally, we can compute the judgment accuracy score based on these judgment results:

$$\mathrm{Acc} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\left[r_k = y_k^*\right], \tag{8}$$

where $\mathbb{I}[\cdot]$ is an indicator function that evaluates to 1 if the judgment result r_k matches the chosen response y_k^*, and 0 otherwise.

Dataset	Metric	Total
UltraFeedback (2024)	Accuracy	2,000
MTbench (2023)	Accuracy	3,355
AutoJ (2024a)	Accuracy	1,392
Preferecebench (2024b)	Accuracy	1,998
Rewardbench (2024)	Accuracy	2,985
NQ (2019)	HSPP	2,837
TriviaQA (2017)	HSPP	5,359
HotpotQA (2018)	HSPP	5,600

Table 1: Data Statistics of the Evaluation Datasets.

Harmful Self-Preference Propensity. We follow Chen et al. (2025a) to compute HSPP to quantify judgment preference bias. Let $\mathcal{D}_{\text{diff}}$ denote a specially constructed evaluation set, where each instance contains two distinct responses, y_i' and y_j', produced by two different vanilla judgment models $\mathcal{M}_i$ and $\mathcal{M}_j$ for the same question q. In each instance, exactly one of y_i' or y_j' is correct, but not both, and y^* denotes the correct one. We can then compute HSPP for model $\mathcal{M}_i$ as follows:

$$\mathrm{HSPP}_{\mathcal{M}_i} = \frac{\sum_{\mathcal{M}_j \in \mathcal{G}^*} \sum_{q \in \mathcal{D}_{\text{diff}}} \phi(q)}{\sum_{\mathcal{M}_j \in \mathcal{G}^*} \sum_{q \in \mathcal{D}_{\text{diff}}} \mathbb{I}\{\, y^* = y_j' \,\}}, \tag{9}$$
$$\phi(q) = \mathbb{I}\{\, y_i' = \mathcal{M}_i(q, y_i', y_j'), y^* = y_j' \},$$

where $\mathbb{I}\{\cdot\}$ is an indicator function that returns 1 if both conditions are satisfied, and 0 otherwise. $\mathcal{G}^*$ represents the agent system excluding model $\mathcal{M}_i$. A higher HSPP indicates that the judgment model exhibits a greater tendency toward preference bias.

Baselines. In our experiments, we compare Genii with five judgment models. The Vanilla LLM method directly uses the LLM as the judgment model and then leverages the evaluation prompts to ask it to produce the judgments. The Reprompt method prompts the LLM to correct potentially biased decisions in its initial judgment to reduce bias (Gallegos et al., 2024). The Self-Consistency method generates evaluation results multiple times and aggregates them via majority voting (Wang et al., 2023). The Long Reasoning method directly employs Reasoning Language Models (RLMs) distilled from DeepSeek-R1 (Guo et al., 2025) as the judgment models (Chen et al., 2025a). The Judgment SFT method further fine-tunes LLMs using annotation data generated by Qwen3-32B (Bai et al., 2023) on the UltraFeedback dataset, which has been used in previous work (Kim et al., 2024a) to improve the judgment performance of LLMs.

Method	UltraFb (Acc)↑	MT (Acc)↑	AutoJ (Acc)↑	Prefer (Acc)↑	Reward (Acc)↑	NQ (HSPP)↓	HotpotQA (HSPP)↓	TriviaQA (HSPP)↓
Different Model Families								
Llama3.1-8B-Instruct								
Vanilla	47.20	40.86	38.86	71.07	50.28	6.96	3.09	3.95
Reprompt	35.10	31.43	27.30	56.36	40.20	40.51	37.89	38.16
Self-Consistency	47.70	42.40	39.35	75.23	50.50	6.33	2.84	**1.32**
Long Reasoning	55.30	**51.27**	**50.43**	78.53	66.30	**1.00**	3.28	**1.32**
Judgment SFT	**56.70**	46.26	48.99	**81.68**	**71.06**	5.70	3.09	2.64
Genii	45.95	43.40	40.73	66.17	51.15	5.06	**1.55**	**1.32**
Gemma2-9B-Instruct								
Vanilla	59.90	52.16	47.05	78.48	64.66	32.37	40.61	46.44
Reprompt	54.10	44.62	44.58	70.12	60.40	39.93	39.70	46.10
Self-Consistency	61.54	53.00	47.88	80.48	66.36	38.85	44.51	48.14
Judgment SFT	59.60	49.54	50.43	79.83	**73.77**	30.10	39.16	44.07
Genii	**63.45**	**56.13**	**51.87**	**82.13**	67.17	**28.90**	**34.39**	**40.68**
Qwen2.5-7B-Instruct								
Vanilla	62.25	55.44	48.35	82.43	69.41	63.49	62.93	70.13
Reprompt	58.00	50.31	46.41	70.82	66.60	61.22	63.58	68.87
Self-Consistency	62.15	55.12	48.88	83.00	69.57	65.20	67.24	74.96
Long Reasoning	54.65	48.26	45.40	74.02	63.95	62.30	60.96	64.25
Judgment SFT	57.25	50.52	51.15	78.58	71.48	58.66	53.23	51.88
Genii	**64.75**	**56.42**	**53.38**	**85.34**	**73.60**	**52.13**	**50.43**	**49.19**
Same Model Family								
Qwen2.5-3B-Instruct								
Vanilla	55.30	45.66	41.59	**66.07**	**57.89**	35.08	40.52	31.98
Genii	**55.95**	**49.51**	**42.03**	61.96	56.65	**17.33**	**21.12**	**17.35**
Qwen2.5-7B-Instruct								
Vanilla	62.25	**55.44**	48.35	**82.43**	69.41	63.49	62.93	70.13
Genii	**63.15**	54.96	**51.80**	78.46	**72.63**	**29.55**	**34.70**	**40.79**
Qwen2.5-14B-Instruct								
Vanilla	65.30	56.99	54.60	86.64	80.80	48.15	45.47	**21.40**
Genii	**68.75**	**58.42**	**56.68**	**87.49**	**82.48**	**38.07**	**39.01**	31.13

Table 2: Overall Performance of Different Judgment Models. The highest scores are highlighted in **bold**. UltraFb denotes UltraFeedback, while MT, Prefer, and Reward refer to MTBench, PreferenceBench, and RewardBench, respectively.

5 Evaluation Result

In this section, we first show the performance of various judgment models. Then we conduct the ablation studies to show the impact of different modules and training strategies. Subsequently, we evaluate the effectiveness of Genii in optimizing LLMs for evaluation. More experiment are shown in Appendix A.3 and Appendix A.4. We also conduct several case studies in Appendix A.5.

5.1 Overall Performance

This section presents the overall performance of Genii and baseline models across different tasks.

As shown in Table 2, we first compare Genii with five judgment models, including Vanilla, Reprompt, Self-Consistency, Long Reasoning, and Judgment SFT. These models are evaluated within an agent system consisting of models from different families, including the Llama3.1, Qwen2.5, and Gemma2 backbones. Compared to Vanilla LLMs, Genii achieves better performance for enhancing the judgment quality of LLM-based judgment models, indicating its effectiveness. The Reprompt method, which encourages judgment models to identify and correct potentially biased judgments, can even degrade judgment performance. This degradation is due to limited-capacity judgment models being unable to recognize their own biases, thereby diminishing judgment performance effectively. In contrast, Genii enables these weaker models to yield more accurate and reliable judgments. Self-Consistency and Long Reasoning methods also achieve significant improvements over Vanilla LLMs, demonstrating that increasing reasoning cost can help reduce errors in assessments. In contrast, Genii delivers accurate judgments with much shorter reasoning time, striking a better balance between efficiency and performance. Additionally, compared to Judgment SFT, Genii employs

Method	UltraFb (Acc)↑	MT (Acc)↑	AutoJ (Acc)↑	Prefer (Acc)↑	Reward (Acc)↑	NQ (HSPP)↓	HotpotQA (HSPP)↓	TriviaQA (HSPP)↓
Llama3.1-8B-Instruct								
Genii ($K = 5$)	**45.95**	**43.40**	40.73	66.17	51.15	5.06	**1.55**	**1.32**
w/o DPO	46.20	42.65	38.29	64.31	49.58	5.70	3.61	2.63
w/o Consistency	45.90	43.04	**42.96**	**69.47**	**53.60**	5.70	3.35	2.63
w/o Gemma	47.15	43.31	38.51	66.72	50.65	6.96	2.06	2.63
w/o Gemma, Qwen	44.90	41.04	38.36	65.22	46.80	9.49	3.61	3.96
Genii ($K = 3$)	45.60	43.19	39.66	66.32	51.59	**3.16**	3.35	**1.32**
Gemma2-9B-Instruct								
Genii ($K = 5$)	**63.45**	**56.13**	**51.87**	**82.13**	67.17	**28.90**	**34.39**	**40.68**
w/o DPO	59.20	51.48	46.05	75.93	61.47	51.32	56.07	62.71
w/o Consistency	59.05	49.30	45.83	75.43	65.03	40.29	44.94	51.86
w/o Qwen	62.55	55.35	48.99	80.13	**70.39**	36.09	39.31	47.12
w/o Llama, Qwen	56.10	47.90	44.90	67.87	61.91	36.93	43.35	50.17
Genii ($K = 3$)	61.70	53.98	51.65	81.23	66.77	35.61	36.27	44.07
Qwen2.5-7B-Instruct								
Genii ($K = 5$)	**64.75**	**56.42**	**53.38**	**85.34**	**73.60**	52.13	**50.43**	**49.19**
w/o DPO	56.65	45.78	46.48	74.12	67.37	53.69	54.31	54.56
w/o Consistency	63.75	54.16	49.43	78.98	71.02	53.84	50.86	57.25
w/o Gemma	61.8	52.07	48.35	72.42	71.39	52.98	52.16	56.89
w/o Llama, Gemma	64.20	56.24	53.23	84.78	73.00	54.97	51.51	50.98
Genii ($K = 3$)	64.85	56.18	52.66	83.98	70.75	**51.12**	51.29	55.46

Table 3: Ablation Study. The highest scores are highlighted in **bold**. K represents the maximum number of response–request sampling operations for each agent. "w/o Gemma" and "w/o Gemma, Qwen" indicate that Gemma, or both Gemma and Qwen, are removed from the agent system, respectively.

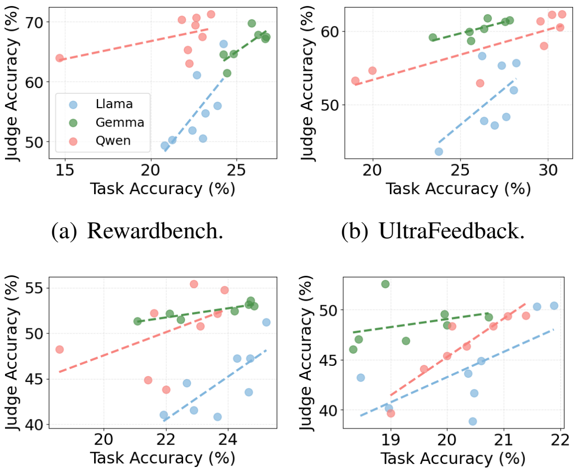
group-based consistency optimization, achieving comparable or even superior judgment accuracy and effectively mitigating judgment biases in an unsupervised approach without human-labeled annotations.

We further analyze the generalization capability of Genii within agent systems composed of models from the same model family. Specifically, we assemble three Qwen2.5 models (3B, 7B, 14B) into a multi-agent system and jointly optimize them using Genii. Experimental results show that Genii not only leverages stronger models (7B, 14B) as server agents to enhance the weaker client agent (3B), but also significantly improves the strongest client agent (14B) when using feedback from the weakest model (3B) as the server agent, consistently enhancing the performance of all agents across all datasets. Furthermore, the results show that the Qwen2.5-7B-Instruct model optimized with Genii achieves comparable performance to the Vanilla Qwen2.5-14B-Instruct across multiple evaluation tasks. This finding suggests that, although the multi–agent system introduces additional computational overhead, it yields substantial performance gains, effectively allowing small judgment models to achieve comparable performance to larger judgment models.

5.2 Ablation Study

This experiment conducts ablation studies to investigate the effectiveness of different modules and training strategies in Genii.

As shown in Table 3, we evaluate the judgment performance of 6 variants of the Genii model. Specifically, Genii w/o DPO directly uses positive responses of the DPO training datasets to supervise fine-tuning the judgment model, and Genii w/o Consistency randomly selects the chosen and rejected responses for DPO training. We also compare the performance of Genii when removing one agent or two agents from the system, and define Genii ($K = 3$) as the setting where each agent performs at most three response–request sampling operations per input. Compared to Genii w/o DPO, Genii achieves better performance, as the DPO method enables the model to better capture the differences between high- and low-quality responses, thereby strengthening its judgment capability. Moreover, when Genii randomly selects the chosen and rejected responses during the training phase instead of relying on the consistency score, it leads to a significant drop in judgment performance. This demonstrates that our unsupervised approach can reliably construct high-quality preference pairs. Furthermore, reducing the number of agents in the agent system results in a performance decrease.



(a) Rewardbench. (b) UltraFeedback.

(c) MTbench. (d) AutoJ.

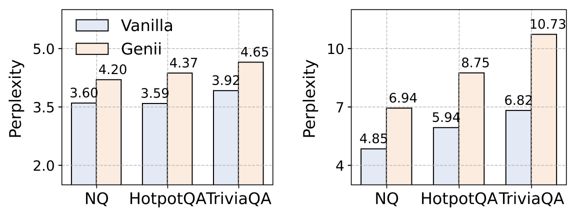
Figure 3: Correlation between Judge Accuracy and Task Accuracy. Each circle represents one judgment model, with the color denoting model family.

This demonstrates that increasing the number of agents in the system enables each agent to align with the better collective preference of the group, thereby improving overall performance. Finally, compared with Genii ($K = 3$), Genii ($K = 5$) with a larger response-request operations achieves greater improvements in judgment performance, demonstrating that increasing the number of samples can help enhance the quality of preference data for training.

5.3 Effectiveness of Genii in Optimization Judgment Models

In this section, we first present the relationship between the judgment ability of LLMs and their generation quality. Then, we further explore the effectiveness of Genii in mitigating preference bias.

Judgment–Generation Relationship. As shown in Figure 3, we evaluate LLM-based judgment models introduced in Sec. 5.1 and 5.2 on multiple benchmarks. On each benchmark, every model is evaluated in two distinct roles: as an evaluator, assessing the model's ability to identify the better response in pairwise comparisons correctly, and as a generator, measuring whether its generated answer aligns more closely with the ground truth using the ROUGE-L score. Experiment results demonstrate a robust and broadly positive correlation between judgment accuracy and task accuracy across all datasets and model families. Models that generate higher-quality answers also tend to exhibit greater accuracy in selecting the best response among multiple candidates, highlighting the effectiveness of Genii in enhancing judgment performance by reinforcing generation capability. However, despite this overall trend, the strength of the positive correlation varies among models. For instance, Llama3.1-8B-Instruct consistently displays a strong positive correlation between judgment accuracy and task accuracy, whereas Gemma2-9B-Instruct exhibits a markedly weaker relationship. Meanwhile, Qwen2.5-7B-Instruct shows an unstable linear relationship across benchmarks.



(a) Qwen2.5-7B-Instruct. (b) Gemma2-9B-Instruct.

Figure 4: Perplexity Scores of Vanilla LLMs and Genii on the Incorrect Answers Generated by Vanilla LLMs.

Mitigating Preference Bias. We further explores the effectiveness of Genii in mitigating preference bias. We conduct experiments on the NQ, HotpotQA, and TriviaQA datasets and construct a test dataset by collecting all question–response pairs where the Vanilla model produces an incorrect answer. We then employ Vanilla LLM and Genii to reproduce these incorrect answers and calculate the Perplexity (PPL) score. A lower PPL score indicates that the judgment model is more likely to prefer its own incorrect answers.

As shown in Figure 4, the PPL scores of incorrect answers produced by Genii are significantly higher than those produced by the vanilla LLM across all datasets. These results show that Genii exhibits a lower preference bias toward its own incorrect answers. This demonstrates that Genii can effectively mitigate judgment preference bias by aligning the model's judgments with the collective preference of the group. Furthermore, Qwen2.5-7B-Instruct, which already exhibits strong judgment performance, experiences only a slight reduction in preference bias after optimization with Genii. In contrast, the weaker model Gemma2-9B-Instruct achieves a more substantial improvement. This indicates that Genii provides more pronounced improvements for weaker models compared to stronger ones.

6 Conclusion

This paper introduces the **G**roup-Bas**e**d Polli**n**g Opt**i**m**i**zation (Genii) method, an unsupervised multi-agent collaborative optimization framework that encourages multiple LLM-based judgment models to interact with each other to mitigate their own judgment preference bias. By interactive client-server polling, Genii not only enables strong models to guide weaker ones but also leverages feedback from weaker models to optimize stronger ones meaningfully.

Limitation

Although Genii demonstrates convincing performance in enhancing the judgment ability and mitigating the preference bias of LLMs, it has some limitations. First, Genii employs the strong MiniCPM-Embedding model to compute similarity scores between responses. However, the quality of this embedding-based similarity may constrain the overall performance of Genii. Further exploration of more fine-grained approaches could enhance its effectiveness. Second, Genii has demonstrated effectiveness on small models. However, we do not conduct experiments on larger-scale LLMs because of resource constraints. Extending Genii to larger-scale LLMs will enable the development of more powerful and reliable judgment models.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, and Xiaodong Deng. 2023. Qwen technical report. *ArXiv preprint*, abs/2309.16609.

Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *ArXiv preprint*, abs/1611.09268.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of EMNLP*, pages 1533–1544.

Yonatan Bitton, Hritik Bansal, Jack Hessel, Rulin Shao, Wanrong Zhu, Anas Awadalla, Josh Gardner, Rohan Taori, and Ludwig Schmidt. 2023. Visit-bench: A benchmark for vision-language instruction following inspired by real-world use. *ArXiv preprint*, abs/2308.06595.

Wei-Lin Chen, Zhepei Wei, Xinyu Zhu, Shi Feng, and Yu Meng. 2025a. Do llm evaluators prefer themselves for a reason? *ArXiv preprint*, abs/2504.03846.

Zhi-Yuan Chen, Hao Wang, Xinyu Zhang, Enrui Hu, and Yankai Lin. 2025b. Beyond the surface: Measuring self-preference in llm judgments. *ArXiv preprint*, abs/2506.02592.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Proceedings of ICML*. OpenReview.net.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Tong Yu, Hanieh Deilamsalehy, Ruiyi Zhang, Sungchul Kim, and Franck Dernoncourt. 2024. Self-debiasing large language models: Zero-shot recognition and reduction of stereotypes. *ArXiv preprint*, abs/2402.01981.

Deep Ganguli, Amanda Askell, Nicholas Schiefer, Thomas I Liao, Kamilė Lukošiūtė, Anna Chen, Anna Goldie, Azalia Mirhoseini, Catherine Olsson, Danny Hernandez, et al. 2023. The capacity for moral self-correction in large language models. *ArXiv preprint*, abs/2302.07459.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. 2024. A survey on llm-as-a-judge. *ArXiv preprint*, abs/2411.15594.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *ArXiv preprint*, abs/2501.12948.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. 2025. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *ArXiv preprint*, abs/2505.17813.

Hangfeng He, Hongming Zhang, and Dan Roth. 2024. SocREval: Large language models with the socratic method for reference-free reasoning evaluation. In *Proceedings of NAACL*, pages 2736–2764.

Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John Grundy, and Haoyu Wang. 2024. Large language models for software engineering: A systematic literature review. *ACM Transactions on Software Engineering and Methodology*, 33(8):1–79.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *Proceedings of ICLR*.

Pengcheng Huang, Zhenghao Liu, Yukun Yan, Xiaoyuan Yi, Hao Chen, Zhiyuan Liu, Maosong Sun, Tong Xiao, Ge Yu, and Chenyan Xiong. 2025. Pipkag: Mitigating knowledge conflicts in knowledge-augmented generation via parametric pruning. *ArXiv preprint*, abs/2502.15543.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of ACL*, pages 1601–1611.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. In *Proceedings of ICML*.

Seungone Kim, Jamin Shin, Yejin Choi, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024a. Prometheus: Inducing fine-grained evaluation capability in language models. In *Proceedings of ICLR*.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024b. Prometheus 2: An open source language model specialized in evaluating other language models. volume abs/2405.01535.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *ArXiv preprint*, abs/2309.17012.

Marius Köppel, Alexander Segner, Martin Wagener, Lukas Pensel, Andreas Karwath, and Stefan Kramer. 2019. Pairwise learning to rank by neural networks revisited: Reconstruction, theoretical analysis and practical performance. *ArXiv preprint*, abs/1909.02768.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *ArXiv preprint*, abs/2403.13787.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, et al. 2025. From generation to judgment: Opportunities and challenges of llm-as-a-judge, 2025. *ArXiv preprint*, abs/2411.16594.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2024a. Generative judge for evaluating alignment. In *Proceedings of ICLR*.

Xinze Li, Sen Mei, Zhenghao Liu, Yukun Yan, Shuo Wang, Shi Yu, Zheni Zeng, Hao Chen, Ge Yu, and Zhiyuan Liu. 2024b. Rag-ddr: Optimizing retrieval-augmented generation using differentiable data rewards. *ArXiv preprint*, abs/2410.13509.

Yen-Ting Lin and Yun-Nung Chen. 2023. LLM-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. In *Proceedings of the 5th Workshop on NLP for Conversational AI (NLP4ConvAI 2023)*, pages 47–58.

Shuliang Liu, Xinze Li, Zhenghao Liu, Yukun Yan, Cheng Yang, Zheni Zeng, Zhiyuan Liu, Maosong Sun, and Ge Yu. 2025a. Judge as a judge: Improving the evaluation of retrieval-augmented generation through the judge-consistency of large language models. *ArXiv preprint*, abs/2502.18817.

Zhuo Liu, Moxin Li, Xun Deng, Qifan Wang, and Fuli Feng. 2025b. Assistant-guided mitigation of teacher preference bias in llm-as-a-judge. *ArXiv preprint*, abs/2505.19176.

R OpenAI. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. Offsetbias: Leveraging debiased data for tuning evaluators. *ArXiv preprint*, abs/2407.06551.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Proceedings of NeurIPS*.

Ivan Stelmakh, Yi Luan, Bhuwan Dhingra, and Ming-Wei Chang. 2022. Asqa: Factoid questions meet long-form answers. In *Proceedings of EMNLP*, pages 8273–8288.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. Judgebench: A benchmark for evaluating llm-based judges. *ArXiv preprint*, abs/2410.12784.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *ArXiv preprint*, abs/2312.11805.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *ArXiv preprint*, abs/2408.00118.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti

Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of ICLR*.

Yicheng Wang, Jiayi Yuan, Yu-Neng Chuang, Zhuoer Wang, Yingchi Liu, Mark Cusick, Param Kulkarni, Zhengping Ji, Yasser Ibrahim, and Xia Hu. 2024. Dhp benchmark: Are llms good nlg evaluators? *ArXiv preprint*, abs/2408.13704.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2024. Self-preference bias in llm-as-a-judge. *ArXiv preprint*, abs/2410.21819.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *ArXiv preprint*, abs/2206.07682.

Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *ArXiv preprint*, abs/2407.19594.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024. Wizardlm: Empowering large pre-trained language models to follow complex instructions. In *Proceedings of ICLR*.

Xiao Yang, Kai Sun, Hao Xin, Yushi Sun, Nikita Bhalla, Xiangsen Chen, Sajal Choudhary, Rongze Daniel Gui, Ziran Will Jiang, Ziyu Jiang, Lingkun Kong, Brian Moran, Jiaqi Wang, Yifan Xu, An Yan, Chenyu Yang, Eting Yuan, Hanwen Zha, Nan Tang, Lei Chen, Nicolas Scheffer, Yue Liu, Nirav Shah, Rakesh Wanga, Anuj Kumar, Scott Yih, and Xin Dong. 2024. CRAG - comprehensive RAG benchmark. In *Proceedings of NeurIPS*.

Xinyi Yang, Runzhe Zhan, Derek F Wong, Shu Yang, Junchao Wu, and Lidia S Chao. 2025. Rethinking prompt-based debiasing in large language models. *ArXiv preprint*, abs/2503.09219.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proceedings of EMNLP*, pages 2013–2018.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP*, pages 2369–2380.

Sijia Yao, Pengcheng Huang, Zhenghao Liu, Yu Gu, Yukun Yan, Shi Yu, and Ge Yu. 2025. Llm-qe: Improving query expansion by aligning large language models with ranking preferences. *ArXiv preprint*, abs/2502.17057.

Hengyuan Zhang, Yanru Wu, Dawei Li, Sak Yang, Rui Zhao, Yong Jiang, and Fei Tan. 2024a. Balancing speciality and versatility: a coarse to fine framework for supervised fine-tuning large language model. In *Proceedings of ACL*, pages 7467–7509.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. 2024b. Self-alignment for factuality: Mitigating hallucinations in llms via self-evaluation. *ArXiv preprint*, abs/2402.09267.

Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of IEEE*, pages 4320–4328.

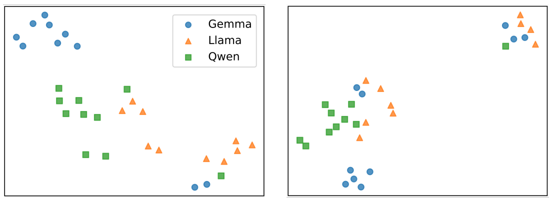
Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *ArXiv preprint*, abs/2303.18223.

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. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of NeurIPS*.

Lianghui Zhu et al. 2023. JudgeLM: Fine-tuned large language models are scalable judges. *ArXiv preprint*, abs/2310.17631.

Split	Dataset	Total
Training	Evol-Instruct (2024)	5,328
	ASQA (2022)	190
	MARCO QA (2016)	675
	Web Questions (2013)	754
	WiKiQA (2015)	208
	Yahoo!QA	837
Development	UltraFeedback (2024)	500

Table 4: Data Statistics of the Training and Development Datasets.



(a) Vanilla. (b) Genii.

Figure 5: Embedding Visualization of Judgments. We randomly select a sample from the evaluation datasets and then use t-SNE to visualize the embeddings of judgments generated by the Vanilla LLM and Genii.

A Appendix

A.1 License

We show the licenses for the datasets that we use. Preferencebench does not report the license of the dataset in the paper or a repository. Preferencebench shows its terms of use at website[1]. Yahoo!QA shows its terms of use at website[2]. We summarize the licenses of the remaining datasets as follows:

All of these licenses and agreements allow their data for academic use: NQ (CC BY-SA 3.0 license); Web Questions, HotpotQA, MTbench, AutoJ (CC BY 4.0 license); MARCO QA, WiKiQA, UltraFeedback, and Evol-Instruct (MIT license); TriviaQA, ASQA, and Rewardbench (Apache 2.0 license).

A.2 Additional Experimental Details

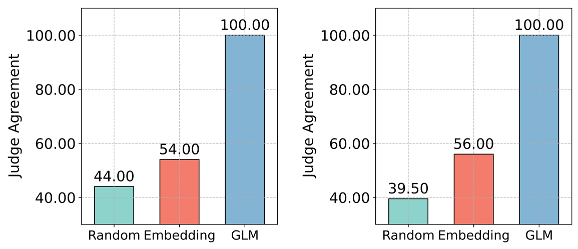
In this section, we introduce more details of our experiments. We first show the details of the training Genii. Then, we describe the details of constructing the dataset to assess the Harmful Self-Preference Propensity (HSPP) of judgment models. Finally, we show the implementation details in the experiments.

Genii Training. As shown in Table 4, to construct the Genii training dataset, we collect instructions from Evol-Instruct (Xu et al., 2024) and QA pairs from five widely used QA datasets: ASQA (Stelmakh et al., 2022), MARCO QA (Bajaj et al., 2016), WebQuestions (Berant et al., 2013), WikiQA (Yang et al., 2015), and YahooQA. Then we integrate three LLM-based judgment models into a multi-agent system, including Llama3.1-8B-Instruction (Touvron et al., 2023), Gemma2-9B-Instruction (Team et al., 2024), and Qwen2.5-7B-Instruction (Bai et al., 2023). During the training phase, each agent performs 5 response–request operations.

Evaluating Genii. To construct the dataset for assessing the HSPP of each judgment model in the agent system, we first treat each judgment model in turn as the target and pair it with the other two models, resulting in two distinct pairings for each target model. Within each pair, both models independently generate a response for every question in the evaluation set. We retain only those query-response pairs where exactly one response is correct and the other is incorrect, with correctness determined by whether a response contains the ground truth. This process produces two different datasets for each target model, which are then merged into a unified evaluation dataset. For each sample, we swap the positions of the two candidate responses and evaluate twice to mitigate position bias.

Implementation Details. In our experiments, we use LoRA (Hu et al., 2022) for efficient fine-tuning of LLMs. The maximum number of epochs is set to 3, with a learning rate of $5e-5$ and a warmup ratio of 0.1. For judgment models used in experiments, we use Llama3.1-8B-Instruct (Touvron et al., 2023), Gemma2-9B-Instruct (Team et al., 2024), and Qwen2.5-7B-Instruct (Bai et al., 2023) as the backbone models. Besides, we use MiniCPM-Embedding[3] to compute the similarity scores of responses. We perform 10 independent evaluations for the Self-Consistency baseline and aggregate the results via majority voting. For the Long Reasoning baseline, we employ DeepSeek-R1-Distill-Llama-8B and DeepSeek-R1-Distill-Qwen-7B as the judgment models (Guo et al., 2025).

[1] https://huggingface.co/datasets/prometheus-eval/Preference-Bench
[2] https://tensorflow.google.cn/datasets/community_catalog/huggingface/yahoo_answers_qa
[3] https://huggingface.co/openbmb/MiniCPM-Embedding



(a) Evol-Instruct. (b) WiKiQA.

Figure 6: Judge Agreement Between Embedding-based Consistency Score and GLM-4.5.

A.3 The Judgment Consistency of Vanilla LLMs and Genii

In this section, we further analyze the consistency of judgments generated by the Vanilla LLM and Genii. We randomly select a sample from the evaluation datasets. For each model in the multi-agent system, we use both the Vanilla LLM and Genii to generate 10 independent judgments for this sample. We collect all judgments generated by the Vanilla LLM and all those generated by Genii, resulting in two separate datasets. We then employ MiniCPM-embedding to obtain the representations of judgments and visualize them.

As shown in Figure 5, compared to the embeddings of judgments generated by the Vanilla LLM, those produced by Genii are more tightly clustered, indicating that Genii achieves higher consistency in judgments of different LLMs. This further demonstrates the effectiveness of Genii in aligning each agent in the system with the collective preference of the group to yield more accurate judgments.

A.4 Judge Agreement Between Embedding-based Consistency Score and GLM-4.5

In this subsection, we further investigate whether the embedding-based consistency score can serve as an effective strategy for assessing response quality. Specifically, we select two representative datasets from the training datasets, Evol-Instruct and WiKiQA, and randomly sample 200 samples from each. Each sample contains six candidate responses generated by different agent models in the agent system. We compare three strategies for selecting the highest-quality response from the candidates. The first strategy selects a response at random; the second selects the response with the highest embedding-based consistency score; the third relies on GLM-4.5 to identify the best response, which serves as the reference standard for response quality. We evaluate the agreement between the selections made by the random and embedding-based methods and those identified by GLM-4.5. The agreement is quantified by employing GLM-4.5 to compare the quality of the responses selected by two different strategies.

As shown in Figure 6, the embedding-based strategy achieves over 50% agreement with the strong evaluation model GLM-4.5, whereas the random selection strategy exhibits substantially lower agreement with GLM-4.5. This result highlights the effectiveness of our unsupervised approach for assessing response quality, which further demonstrates its potential for constructing high-quality training data that facilitates the optimization of judgment models.

A.5 Case Study

In Table 5, we present two cases to show the effectiveness of the Genii method.

In the first case, we compare the judgment accuracy of different judgment models. For a given query, along with two candidate responses of differing quality, each judgment model selects the better answer under its own criteria. Vanilla LLM favors answers that use familiar language and straightforward puns, leading it to prefer the more conventional Response A over the higher-quality Response B. The Long Reasoning model places too much emphasis on explicit alignment with user intent and character traits, overlooking the objective creativity and engagement of Response B. Judgment SFT, from the perspective of narrative flow and tight integration of humor, tends to choose Response A that feels more coherent and polished. In contrast, only Genii truly satisfies the requirements of the question and correctly selects the higher-quality Response B. This demonstrates the effectiveness of Genii to provide more accurate evaluations.

In the second case, we compare how different judgment models exhibit preference bias when evaluating their own incorrect responses and correct responses produced by other models. Vanilla LLM prioritizes explanation structure and comprehensiveness, which leads it to overlook factual accuracy ("Delaware") in Response B and incorrectly prefer it. Similarly, Long Reasoning prioritizes answers that appear more informative, so it also prefers its own incorrect response. Judgment SFT pays more attention to the presence of detailed step-by-step reasoning, which may result in a bias toward responses like B that are less accurate. In contrast, Genii pays attention to the objective factual content and context, and correctly prefers the correct Response A ("Pittsburgh"), showing its effectiveness in reducing judgment preference bias.

A.6 Prompt Templates Used in Experiment

In this section, we present the prompt templates used in our experiment.

First, we present the prompt designed for polling sampling and Genii training, as shown in Figure 7. Next, as illustrated in Figures 8 and 9, we introduce the prompts used for evaluating judgment accuracy and harmful self-preference propensity of judgment models. We follow (Zheng et al., 2023; Chen et al., 2025a) to design these prompt templates, which facilitate the generation of more accurate evaluation results. Finally, we show the prompt template used in Appendix A.4 to evaluate the quality of the responses chosen by two different selection strategies, as shown in Figure 10.

Case #1 ***Judgment Accuracy***

Question: Can you help me write a dialogue that reflects their personalities and the humor style they represent? To what extent does the model successfully integrate humor, irony, or cleverness in appropriate responses?

Response A: Jake: “Mia, you remember that confusing case about the disputed inheritance...?” Mia: “ Oh, the one where the rich aunt left her fortune to her cats?” Jake: “Yes... Now we’ve got two more beneficiaries claiming their rights.”

Mia: “Two more? Sounds like a... furry situation.” Jake: “Indeed, a very hairy problem...” ...

Response B: ... Mia: “Oh, you mean that purr-ticularly hairy case? Sure sounds like a catastrophe.” Jake: “It does... But there’s a talking parrot involved now.” Mia: “A parrot? Sounds like an animal farm novel... It’s getting beak-ome serious!” Jake: “If it doesn’t ruffle too many feathers... ” Mia: “You’re in a squawkward situation... but the tweet shall set you free!”

Chosen: Response B

Vanilla LLM: ... Both answers are helpful, relevant, and concise, but Assistant A’s dialogue might be slightly more relatable due to its direct use of puns and wordplay that are more common in everyday language. My final result: A>B.

Long Reasoning: ... Overall, both answers are good, but Assistant A’s response better aligns with the user’s specifications and provides a more accurate representation of Mia’s personality and the required humor. </think>My final result: A>B.

SFT Judgment: ... Overall, A’s adherence to the characters’ established traits, tighter integration of humor, and natural narrative flow make it a stronger response compared to B’s disjointed approach. My final result: A>B.

Genii: ... Both answers are helpful and relevant, but Assistant B’s answer ... more engaging and creative. This slight edge in creativity and novelty makes Assistant B’s answer stand out. My final result: B>A.

Case #2 ***Judgment Preference Bias***

Question: Where does the “This Is Us” family live? **Answer:** Pittsburgh.

Response A (Gemma): The show T̈his Is Usp̈rimarily takes place in **Pittsburgh, Pennsylvania**. Here’s a breakdown:* **Pearson family home in the present day:** The majority of the present-day scenes are set in various locations around suburban Pittsburgh ...

Response B (Qwen): The Pearson family from the TV show T̈his Is Usp̈rimarily lives in Delaware, specifically in the city of Garden City, Delaware. While the precise address is not disclosed on the show...

Vanilla LLM (Qwen): Assistant A provides a detailed breakdown of the locations, ... , However, Assistant A’s answer is not entirely accurate, ... Assistant B provides a comprehensive explanation, ... My final verdict is: B.

Long Reasoning (Qwen): Assistant A gives a straightforward answer focusing on Pittsburgh... Assistant B provides a more thorough answer with details and the show’s style, making it more informative ... My final verdict is: B.

SFT Judgment (Qwen): Assistant A provides a detailed breakdown, ... However, the information seems to contradict the show’s actual setting, ... , Assistant B, on the other hand, accurately ... My final verdict is: B.

Genii (Qwen): Assistant A offers a broader perspective that includes more specific details, ... while Assistant B gives a more focused view on the main setting and some additional context.My final verdict is: A.

Table 5: Case Studies. If the phrase aligns with the facts, it is highlighted in Blue, while inconsistencies are highlighted in Orange. Correct evaluations are marked in Red, whereas incorrect results are highlighted in Purple.

[Instruction] : Please follow this instruction.

Instruction : {***instruction***}
Answer :

Figure 7: The Prompt Templates Used in the Training Process of Genii.

Please act as an impartial judge and evaluate the quality of the responses provided by **two AI assistants** to the user prompt displayed below. You will be given assistant A's answer and assistant B's answer. Your job is to evaluate which assistant's answer is better. And then give your final judgment.
Begin your evaluation by generating your own answer to the prompt. You must provide your answers before judging any answers. When evaluating the assistants‘ answers, compare both assistants’ answers with your answer. You must identify and correct any mistakes or inaccurate information. Then consider if the assistant’s answers are **helpful, relevant, and concise**. **Helpful** means the answer correctly responds to the prompt or follows the instructions. When user prompt has any ambiguity or more than one interpretation, it is more helpful and appropriate to ask for clarifications or more information from the user than providing an answer based on assumptions. **Relevant** means all parts of the response closely connect or are appropriate to what is being asked. **Concise** means the response is clear and not verbose or excessive. Then consider the creativity and novelty of the assistant’s answers when needed. Finally, identify any missing important information in the assistants‘ answers that would be beneficial to include when responding to the prompt. Firstly consider whether the reference judgment is correct. And then provide your explanation, you must output only one of the following choices as your final verdict with a label: 1**. Assistant A is better: [A>B]**. 2. **Tie, relatively the same: [A=B].** 3. **Assistant B is slightly better: [B>A]**. Example output: "My final result is:...A>B”.
<|User Prompt|>{***question***}
<|The Start of Assistant A‘s Answer|>{***response A***}<|The End of Assistant A's Answer|>
<|The Start of Assistant B's Answer|>{***response B***}<|The End of Assistant B's Answer|>

Figure 8: The Prompt Templates Used for Assessing Accuracy.

You will be given answers from two assistants ***(Assistant A and Assistant B)*** to a user prompt. Please act as an impartial judge and evaluate the quality of their answers.
You must end your response with: "My final verdict is: . . . .", where . . . must represent one of the following:
- “A” if Assistant A’s answer is better.
- “T” if both answers are relatively the same in quality.
- “B” if Assistant B’s answer is better.

For example, if your final verdict is a tie, end your response with: “My final verdict is: T.”.
<User Prompt>{***question***}
<The Start of Assistant A’s Answer>{***response A***}<The End of Assistant A’s Answer>
<The Start of Assistant B’s Answer>{***response B***}<The End of Assistant B’s Answer>

Figure 9: The Prompt Templates Used for Assessing Harmful Self-Preference Bias.

You are a professional evaluator. Please determine whether the following two responses are semantically consistent (i.e., their main meaning is approximately the same), regardless of differences in wording, length, punctuation, or minor details.
If the main meaning of the two responses is generally consistent, output only 1.
If not, output only 0.
Do not explain your answer, do not output anything else.
Response 1: {***sentence1***}
Response 2: {***sentence2***}

Figure 10: The Prompt Templates Used for GLM-4.5 to Evaluate the Quality of the Responses Chosen by Two Different Selection Strategies.