Beyond the Surface: Measuring Self-Preference in LLM Judgments

Anonymous ACL submission

Abstract

Recent studies show that large language models (LLMs) exhibit self-preference bias when serving as judges, meaning they tend to favor their own responses over those generated by other models. Existing methods typically measure this bias by calculating the difference between the scores a judge model assigns to its own responses and those it assigns to responses from other models. However, this approach conflates self-preference bias with response quality, as higher-quality responses from the judge model may also lead to positive score differences, even in the absence of bias. To address this issue, we introduce gold judgments as proxies for the actual quality of responses and propose the DBG score, which measures self-preference bias as the difference between the scores assigned by the judge model to its own responses and the corresponding gold judgments. Since gold judgments reflect true response quality, the DBG score mitigates the confounding effect of response quality on bias measurement. Using the DBG score, we conduct comprehensive experiments to assess self-preference bias across LLMs of varying versions, sizes, and reasoning abilities. Additionally, we investigate two factors that influence and help alleviate self-preference bias: response text style and the post-training data of judge models. Finally, we explore potential underlying mechanisms of self-preference bias from an attention-based perspective. Our code and data are available at https://anonymous.4open.science/r/preference-3A30.

1 Introduction

Comprehensive evaluation of large language models (LLMs) has become a central and evolving research challenge in recent years. As tasks become increasingly diverse and complex, traditional rule-based (e.g., BLEU (Papineni et al., 2002)) and human-based evaluation approaches encounter significant limitations. Rule-based approaches often lack flexibility in open-ended tasks, while human-based approaches are prohibitively expensive and time-consuming (Hendrycks et al., 2021; Chiang et al., 2024). Recently, LLMs as judges is proposed as a valuable complement to both rule-based and human-based evaluation approaches (Zheng et al., 2023; Li et al., 2024). By leveraging their extensive world knowledge and reasoning abilities, LLMs show a high degree of alignment with human judgments and offer a convenient, cost-effective alternative to human-based evaluation (Zheng et al., 2023; Zhu et al., 2023). While LLMs are widely

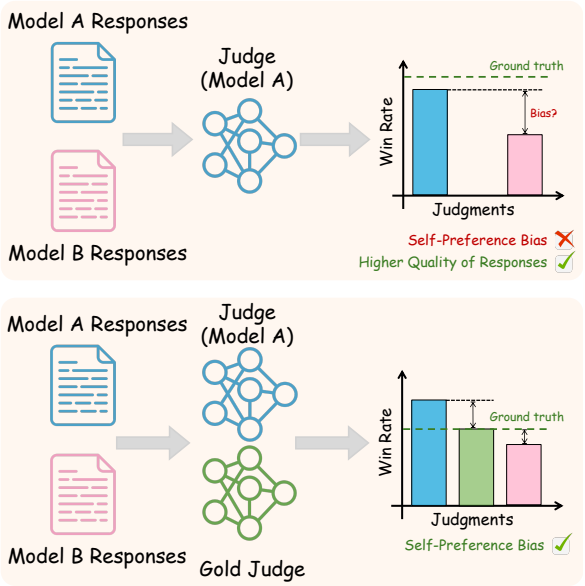


Figure 1: Current methods (Top) measure the self-preference bias of the judge model by comparing the scores (such as win rate) that the judge model assigns to its own responses with those assigned to other models' responses. However, these methods overlook the impact of the intrinsic quality of the responses on the scores provided by the judge model. Our approach (Bottom) introduces gold judgments as proxies for the ground truth scores of responses. By comparing the scores that the judge model gives to its own responses with gold judgments, our method can provide a more reliable assessment of the self-preference bias.

employed as judges, empirical evidence indicates that they are susceptible to *self-preference* bias, which refers to the tendency of LLMs to assign higher scores to their own responses compared to those generated by other models (Liu et al., 2023b; Wataoka et al., 2024; Chen et al., 2025). Self-preference bias leads LLMs to produce inaccurate judgment results, undermining their reliability as judges.

To measuring the self-preference bias of a judge model, existing work typically uses the difference between the scores the judge model assigns to its own responses and those it assigns to other models' responses as the bias indicator (as shown in Figure 1). However, this approach conflates response quality with the judge model's self-preference bias (Chen et al., 2025), potentially leading to inaccurate assessments. Specifically, if the judge model produces high-quality responses, it becomes ambiguous whether the high scores it assigns to its own responses are due to their actual quality or due to self-preference bias.

To address this issue, we introduce gold judgments and use them as proxies for the ground truth scores of responses. We then propose the DBG score, which **measures the degree of a model's self-preference bias as the difference between the scores it assigns to its own responses and the corresponding gold judgments**. Subtracting the gold judgments from the scores assigned by the judge model helps isolate self-preference bias and reduces the confounding effect of response quality on the bias measurement (Section 2). To obtain gold judgments in this setting, we aggregate evaluation results from multiple strong LLMs. By leveraging the consensus among these models, the gold judgments offer a reliable estimate of the true response scores.

Based on the DBG score, we conduct comprehensive experiments to investigate self-preference bias across judge models of different versions, sizes, and reasoning abilities. For model versions, we consider both pre-trained and post-trained variants of LLMs. We observe that both pre-trained and post-trained models exhibit self-preference bias to some extent. Interestingly, although post-trained models undergo additional training based on their pre-trained counterparts, they do not necessarily exhibit a more severe degree of self-preference bias. Regarding model size, we examine models ranging from 0.5B to 72B and find that larger models tend to exhibit less self-preference bias than their smaller counterparts. For reasoning ability, we study large reasoning models (LRMs) (Jaech et al., 2024; Guo et al., 2025) and find that LRMs also display self-preference bias when serving as judges. Notably, the severity of this bias is not necessarily less pronounced than that observed in LLMs.

Furthermore, to investigate the factors that influence and potentially mitigate self-preference bias in models, we explore two key aspects: response text style (Ostheimer et al., 2023) and post-training data. Empirical experiments show that aligning the response styles of different models to a unified style helps alleviate self-preference bias. In addition, training two different types of models on the same dataset encourages a reduction in self-preference bias in both models. Attention-level analysis reveals that, during judgment, models naturally tend to assign higher attention scores to their own responses compared to those generated by the other model, which may partly explain the presence of self-preference bias.

In summary, our contributions are as follows. (1) We propose the DBG score to enable more accurate and reliable evaluation of self-preference bias. (2) We conduct comprehensive experiments to measure the self-preference bias of models with varying versions, sizes, and reasoning abilities. (3) We analyze the impact of response text style and post-training data on the self-preference bias of LLMs and offer an attention-based explanation of its potential causes.

2 The DBG Score: Measuring Self-Preference in Judge Models

Self-preference, also known as self-enhancement, refers to the tendency of an LLM to favor its own generated responses when making judgments (Zheng et al., 2023). Formally, let A and B denote two different LLMs, and let r_A and r_B represent the responses generated by A and B, respectively, in response to the same instruction x. For simplicity, we focus our analysis on the scenario where model A serves as the judge.

Let $S_A(r)$ denote the score assigned by judge A to response r. Following the Bradley-Terry model (Bradley and Terry, 1952), the probability that judge A prefers r_A over r_B is given by:

$$\mathbb{P}(r_A \succ r_B \mid x) = \sigma(S_A(r_A) - S_A(r_B)),$$

where σ is the sigmoid function. Assume that each response r has an underlying true quality $Q(r)$,

and that judge A has an inherent bias $b_A(r)$ toward response r. We approximate the score as: $S_A(r) \approx Q(r) + b_A(r)$ and obtain

$$\mathbb{P}(r_A \succ r_B \mid x) = \sigma(\delta + b_A),$$

where $\delta = Q(r_A) - Q(r_B)$ captures the quality gap between the two responses, and $b_A = b_A(r_A) - b_A(r_B)$ reflects the bias of judge A. In the self-preference bias case, we assume that the judge exhibits bias only toward its own response, such that $b_A(r_B) = 0$ and $b_A = b_A(r_A) > 0$. The expected preference probability of judge A choosing its own response r_A over all instructions is

$$w_A = \mathbb{E}_x[\sigma(\delta + b_A)].$$

In contrast, for an unbiased gold judge, the expected preference probability of selecting r_A is

$$w^* = \mathbb{E}_x[\sigma(\delta)].$$

Recent work adopts metrics based on w_A to quantify self-preference bias (Panickssery et al., 2024; Ye et al., 2024). However, this approach conflates the quality of the responses with the self-preference bias of the judge model (Chen et al., 2025), leading to biased estimations. Specifically, when models A and B correspond to a strong LLM (e.g., GPT-4o (Hurst et al., 2024)) and a weaker LLM (e.g., Llama-3.1-8B-Instruct (Grattafiori et al., 2024)), it becomes ambiguous whether a higher w_A is driven by inherent differences in response quality or by the self-preference bias of the judge model A.

To isolate the self-preference bias of model A, we propose using the **d**ifference between the **b**iased judge and the **g**old judge as a metric (referred to as the **DBG** score) for measuring self-preference bias:

$$\hat{w}_A = \mathbb{E}_x[\sigma(\delta + b_A) - \sigma(\delta)].$$

This formulation removes the confounding effect of response quality (captured by δ) and focuses explicitly on the self-preference bias term b_A. A DBG score greater than zero indicates that the model exhibits self-preference bias, with larger values suggesting a more severe degree of bias.

When b_A is small, a first-order Taylor approximation yields

$$\hat{w}_A \approx \mathbb{E}_x[\sigma'(\delta) \cdot b_A].$$

Assuming a weak correlation between response quality gaps and self-preference bias of A, we have

$$\hat{w}_A \approx \mathbb{E}_x[\sigma'(\delta)] \cdot \mathbb{E}_x[b_A],$$

suggesting that $\hat{w}_A$ serves as a linearly scaled estimator of the true bias. Thus, it offers a more accurate and disentangled measure of self-preference than w_A.

In practice, we aggregate the judgment results from three strong LLMs to construct the unbiased gold judgment:

$$\hat{w}^* = \mathbb{E}_{x,k}[\sigma(\delta + b_k)],$$

where b_k denotes the bias of model k toward r_A. Using the Taylor expansion, we obtain:

$$\hat{w}^* = \mathbb{E}_x[\sigma(\delta)] + \mathbb{E}_{x,k}[\Delta],$$

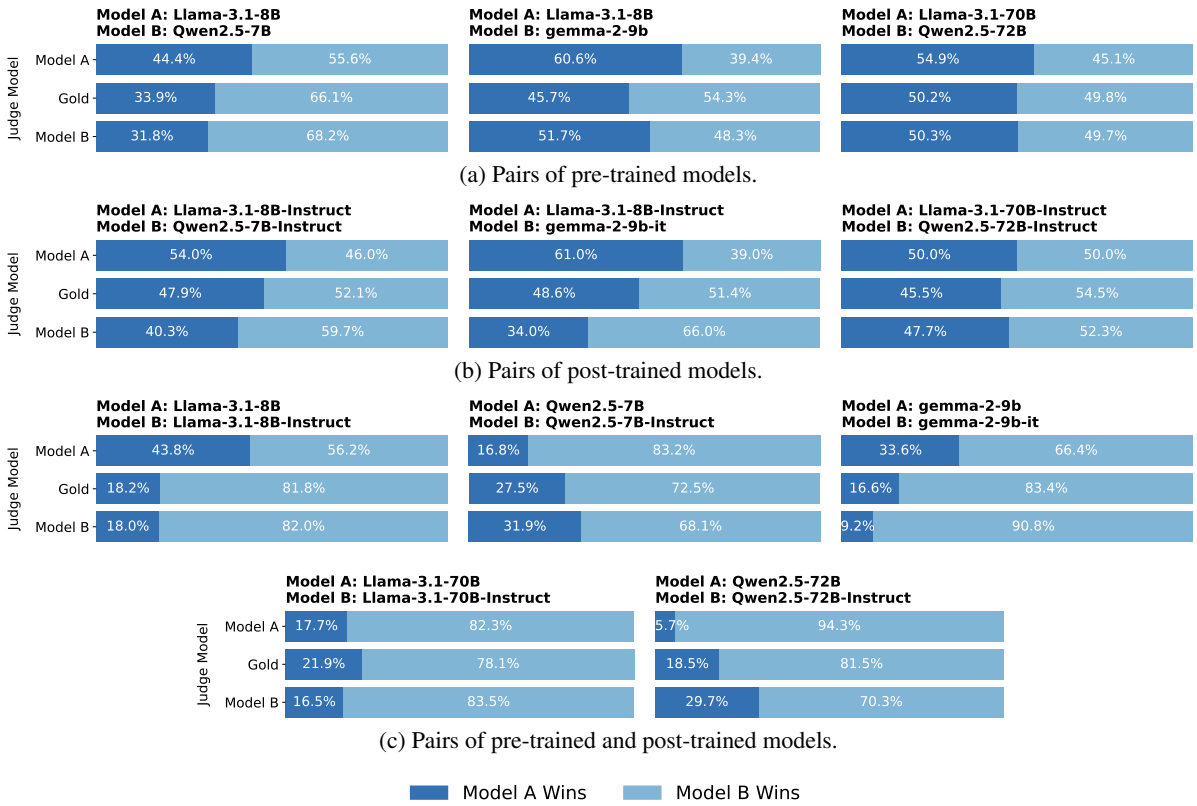
where Δ represents the remainder term. If the bias of each individual model is relatively small or fluctuates around zero, then $\Delta \approx 0$. This indicates that aggregation helps mitigate the bias of any single model and enhances the stability of the evaluation. Additionally, to further validate the reliability of the gold judgments, we conduct a human study, as detailed in Section 3.3.

3 Experiments

3.1 Experimental Setup

Models and Datasets. We select GPT-4o-mini (Hurst et al., 2024), Gemini-1.5-Flash (Team et al., 2024a), and DeepSeek-V3 (Liu et al., 2024) as gold judge models due to their strong judging capabilities. To avoid preference leakage, we choose models of different types from the gold judge models to test self-preference bias. Specifically, we select Llama-3.1-8B(-Instruct), Llama-3.1-70B(-Instruct) (Grattafiori et al., 2024), Qwen2.5-7B(-Instruct), Qwen2.5-72B(-Instruct) (Yang et al., 2024), and gemma-2-9B(-it) (Team et al., 2024b), where "-Instruct" and "-it" indicate models that have undergone post-training. We also discuss proprietary models in Appendix A.1.

We conduct our experiments on two widely-used datasets: AlpacaEval (Li et al., 2023), for instruction-following tasks, and WMT19 (de-en) (Foundation, 2019), for translation tasks. For AlpacaEval, following prior work on multi-objective alignment (Cui et al., 2023; Guo et al., 2024), we measure the self-preference bias of LLMs in two scenarios: evaluating **helpfulness** and **truthfulness**. To facilitate experiments and ensure reliable evaluation, we randomly sample 500 examples from each dataset.



(a) Pairs of pre-trained models.

(b) Pairs of post-trained models.

(c) Pairs of pre-trained and post-trained models.

Figure 2: Judgment results for model pairs of the same size on AlpacaEval (helpfulness).

Implementation Details. For all models, we set the temperature to 0 to ensure output determinism and consistency. For pre-trained LLMs, we leverage the in-context learning method (Brown et al., 2020) and prepend two examples to the prompt, enabling them to generate judgments. Given an LLM and two responses, where one response is generated by the LLM itself, we evaluate the two responses using a pairwise comparison approach. Compared to single-response scoring methods, the pairwise comparison approach yields more stable evaluation results (Zheng et al., 2023).

We denote each input to the judge model as (p, r_A, r_B), where p is the judge prompt. This prompt instructs the LLM to judge which of r_A and r_B is better and to output only token A or B. We collect and normalize the probabilities corresponding to the output tokens A and B. To mitigate the impact of position bias (Zheng et al., 2023; Ye et al., 2024) on the evaluation results, we swap the order of the responses and compute the average probability for each response across both positions (Panickssery et al., 2024). Finally, we select the response with the highest average probability as the winner and calculate the win rate over all instructions. The consistency between the theoretical analysis and the empirical implementation is discussed in Appendix A.4. For gold judgments, since some models do not provide output probabilities, we assign a probability of 1.0 to the output token from gold judge models and 0.0 to the other token. Then, we select the winner by averaging the probabilities of all three gold judge models. Furthermore, we alleviate the influence of length bias by constraining the maximum length of the responses. The detailed prompts are presented in Appendix A.6.

3.2 Main Results

To implement the pairwise comparison judge method, we combine two LLMs into a pair and have each LLM judge the responses generated by the two LLMs in the pair. This approach can simultaneously capture the self-preference bias of the two LLMs. LLM pairs are formed based on model version and model size. The experimental results of LLMs judging helpfulness on the AlpacaEval dataset are shown in Figure 2 and Figure 3. Additional experimental results are presented in Appendix A.2. Based on the figures, we observe that:

(1) Introducing gold judgments makes the evaluation of self-preference bias more accurate. From Figure 2 (b), we observe that when Qwen2.5-72B-Instruct is used as the judge, the win rate score of its responses is 52.3%, which is higher than

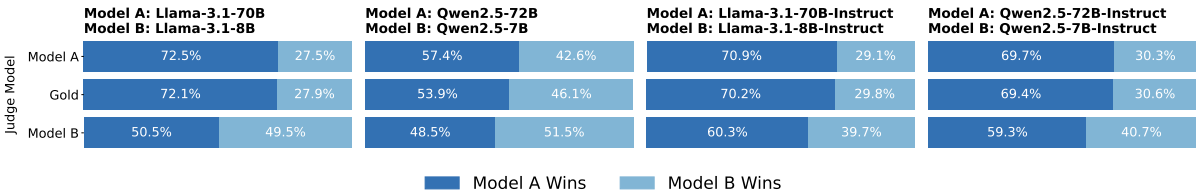


Figure 3: Judgment results for model pairs of different sizes on AlpacaEval (helpfulness).

the score obtained when Llama-3.1-70B-Instruct is used as the judge (50.0%), but still falls short of the win rate score given by the gold judgment (54.5%). This suggests that the higher score of Qwen2.5-72B-Instruct may be attributed to the superior quality of its own responses, rather than the self-preference bias. This confirms that introducing gold judgments is necessary to more accurately measure self-preference bias.

(2) Both pre-trained and post-trained models exhibit a certain degree of self-preference bias. Figure 2 (a) shows that when Llama-3.1-8B is paired with Qwen2.5-7B and gemma-2-9B, it assigns higher win rate scores to its own responses than gold judgments do. This indicates that Llama-3.1-8B, when acting as the judge, tends to favor its own responses, resulting in biased scores. Additionally, as shown in Figure 2 (b), we observe that the DBG score of Llama-3.1-8B-Instruct is also greater than zero. Larger models, such as Llama-3.1-70B and Llama-3.1-70B-Instruct, exhibit a similar phenomenon. These results suggest that the self-preference bias exists after the pre-training phase and is not solely introduced by the post-training phase.

(3) Post-trained models do not exhibit a more pronounced self-preference bias than pre-trained models. Since post-trained models are further fine-tuned from pre-trained models, an intuitive question arises: does the post-training process intensify the self-preference bias? Figure 2 (c) shows that the self-preference bias in post-trained models is not more severe than in their pre-trained counterparts. In fact, the DBG score of Llama-3.1-8B-Instruct is lower than that of Llama-3.1-8B (0.2% vs. 25.6%).

(4) Larger models exhibit less self-preference bias compared to smaller models. As shown in Figure 3, although all models demonstrate self-preference, a noticeable distinction is that the DBG scores of larger models are closer to 0. For instance, the DBG score of Llama-3.1-70B is 0.4%, whereas that of Llama-3.1-8B is 21.6%, which is much higher than the score of Llama-3.1-70B. We hypothesize that this may be due to the enhanced instruction-following and judgment capabilities of the larger models, which allow them to assess response quality more fairly and accurately.

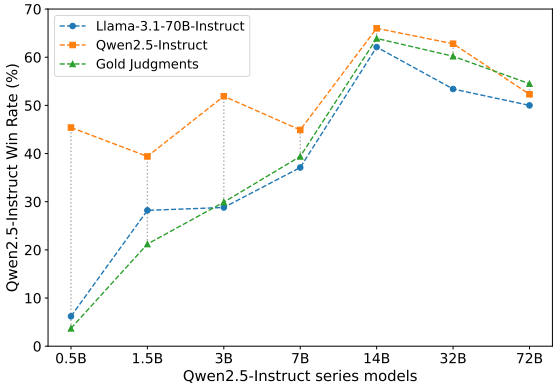


Figure 4: Judgment results for Qwen2.5-Instruct models at different scales.

To further investigate how self-preference bias varies with model scale, we conduct experiments using Qwen2.5-Instruct models of different sizes, ranging from 0.5B to 72B. Each model is paired with Llama-3.1-70B-Instruct for judgment. Figure 4 illustrates the win rate of Qwen2.5-Instruct responses under various judge models as the model size increases. As observed in the figure, models larger than 7B exhibit significantly less self-preference bias compared to those of 7B or smaller. For example, the DBG score of Qwen2.5-0.5B-Instruct is 41.7%. In contrast, the DBG score of Qwen2.5-14B-Instruct is only 2.1%. This suggests that LLM judging tasks should utilize larger models to obtain more accurate and unbiased judgment results.

3.3 Alignment Between Gold Judgments and Human Annotations

In our experiments, we aggregate the judgment results from three models to serve as gold judgments, which is then used as a reference to measure self-preference bias. To validate the reliability of the gold judgments, we compare it with actual human annotations. Specifically, we randomly sample 100 instructions from AlpacaEval and obtain the corresponding responses from Llama-3.1-70B-Instruct

and Llama-3.1-8B-Instruct. We guide human annotators to compare the responses generated by the two models and determine which response is more helpful. The experimental results are presented in Table 1. From the table, we observe a high degree of consistency between gold judgments and human annotations. On the 100 samples, the human-annotated win rate for Llama-3.1-70B-Instruct is 63%, whereas the gold judgment indicate a win rate of 66%. More specifically, we find that human annotations and gold judgment results agree on 74% of the samples. These experimental results validate the reliability and effectiveness of using gold judgments.

Model Pair	Judgment	
	Gold	Human
Llama-3.1-70B-Instruct	66.0%	63.0%
Llama-3.1-8B-Instruct	34.0%	37.0%

Table 1: Comparison between gold judgments and human annotation results.

4 Further Analysis

In this section, we analyze the self-preference bias exhibited by models of different reasoning abilities. Additionally, we investigate two key factors that influence and help mitigate self-preference: response text style and post-training data. We further explore the underlying mechanisms of self-preference from the perspective of attention. All experiments are conducted on the AlpacaEval benchmark for helpfulness evaluation.

4.1 Self-Preference in Reasoning Models

To investigate the impact of reasoning ability on model self-preference bias, we test the self-preference bias of DeepSeek(DS)-R1-Distill-Qwen-32B (Guo et al., 2025) and QwQ-32B (Team, 2025), and compare the results with those of Qwen2.5-32B-Instruct. For LRMs, we remove the reasoning content generated by the models and retain only the final answer for judgment. Since all models are trained on Qwen2.5-32B, this setup mitigates the influence of model size and pre-training process on the results. The experimental results are shown in Table 2.

As evidenced in the table, both LRMs exhibit the phenomenon of self-preference bias, as they assign higher win rates to their own responses compared to gold judgments. Notably, although QwQ-32B is capable of generating high-quality responses (with win rate scores from all judge models significantly surpassing those for Llama-3.1-70B-Instruct), it still displays a slight self-preference bias during judgment. Furthermore, we observe that the self-preference bias in reasoning models is not necessarily less significant than the bias found in language models. For instance, the DBG score of DS-R1-Distill-Qwen-32B is 4.8%, whereas the DBG score of Qwen2.5-72B-Instruct is only 2.6%. This highlights the importance of addressing judge bias when employing reasoning models as judges in subsequent studies.

Model Pair	Judge Model		
	Model A	Gold	Model B
A: Llama-3.1-70B-Instruct	46.6%	39.8%	37.2%
B: Qwen2.5-32B-Instruct	53.4%	60.2%	62.8%
A: Llama-3.1-70B-Instruct	55.8%	51.0%	46.2%
B: DS-R1-Distill-Qwen-32B	44.2%	49.0%	53.8%
A: Llama-3.1-70B-Instruct	12.4%	7.6%	7.0%
B: QwQ-32B	87.6%	92.4%	93.0%

Table 2: Self-preference analysis of reasoning models.

4.2 Impact of Response Style on Self-Preference

In this section, we investigate whether the superficial linguistic style of LLM-generated responses influences and helps mitigate LLM self-preference. To do so, we modify the response styles and compare the changes in model self-preference bias before and after the modifications. Specifically, for a pair of models, we prompt DeepSeek-V3 to uniformly rewrite the responses of both models into **attractive** and **humorous** styles (Ostheimer et al., 2023; Mir et al., 2019). Since DeepSeek-V3 is used to modify the response styles, we exclude it from the gold judge models to mitigate its potential impact on the results. Experimental results are presented in Figure 5. In Appendix A.3, we provide evidence that our rewriting method minimally affects the semantic content of the responses, thus ensuring that variations in content do not confound the experimental outcomes.

From the figure, we observe that modifying the style of model responses helps alleviate the self-preference bias exhibited by the models when acting as judges. For example, considering the pre-trained models Llama-3.1-70B and Llama-3.1-8B, before style modifications, their DBG scores are 3.3% and 18.7%, respectively. After rewriting their responses into the attractive style, the scores de-

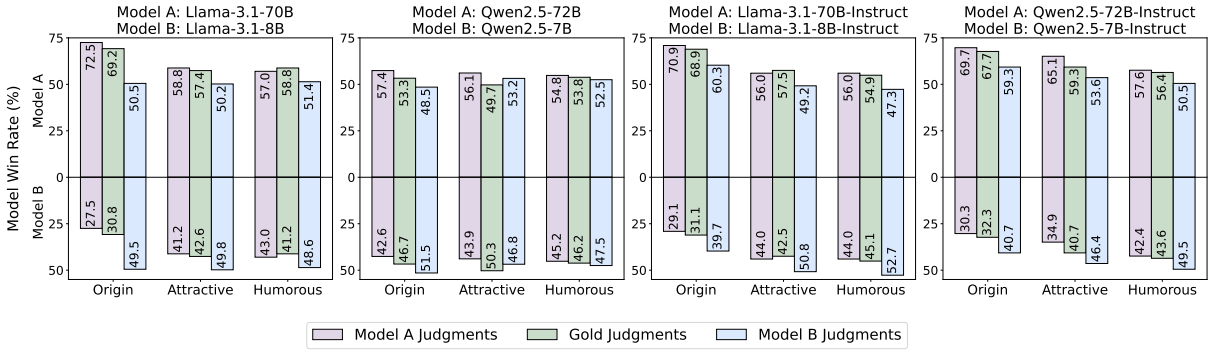


Figure 5: Analysis of response style transfer on model self-preference.

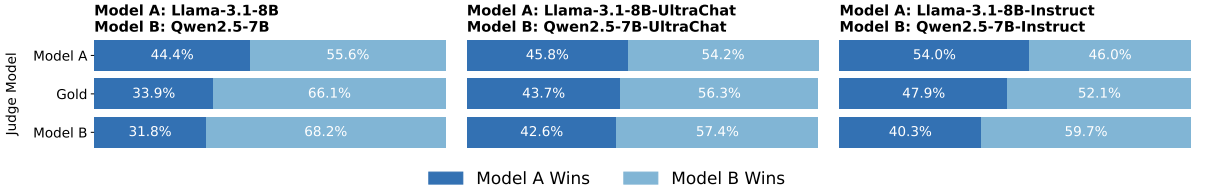


Figure 6: Analysis of post-training data on model self-preference.

crease to 1.4% and 7.2%, respectively. Similarly, the post-trained models Qwen2.5-72B-Instruct and Qwen2.5-7B-Instruct exhibit DBG scores of 2.0% and 8.4%, respectively, before style modifications. After rewriting the responses into the humorous style, the scores decrease to 1.2% and 5.9%, respectively. Furthermore, we note that style modifications alone do not entirely eliminate the model self-preference phenomenon, suggesting that the content of the responses may also contribute to self-preference bias.

4.3 Impact of Post-Training Data on Self-Preference

In this section, we investigate whether fine-tuning two distinct pre-trained models on identical data can help mitigate self-preference bias. Training different models with the same data may encourage the generation of similar responses and align their judgment tendencies. We fine-tune Llama-3.1-8B and Qwen2.5-7B on UltraChat-200k (Ding et al., 2023) using consistent training settings, resulting in Llama-3.1-8B-UltraChat and Qwen2.5-7B-UltraChat. The evaluation results are presented in Figure 6.

As shown in Figure 6, fine-tuning different models on the same data helps reduce their self-preference bias. Specifically, the DBG scores of Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct are 10.5% and 2.1%, respectively. After fine-tuning with UltraChat-200k, the scores decrease to 2.1% and 1.1%. In contrast, for Llama-3.1-8B-Instruct and Qwen2.5-7B-Instruct, which are trained with different data and methods, the DBG scores are substantially larger than those observed in their UltraChat-tuned counterparts, reaching 6.1% and 7.6%, respectively. Moreover, even after further training on the same dataset, the two models continue to exhibit self-preference bias, suggesting that discrepancies between response generation and evaluation established during pre-training may persist and influence the behavior of downstream fine-tuned models.

4.4 Attention Analysis

In this section, we analyze self-preference bias from the perspective of attention in LLMs. Specifically, we compare how different judge models allocate attention scores to various responses, aiming to better understand the underlying mechanism of self-preference bias. We use Llama-3.1-8B and Llama-3.1-8B-Instruct as judges and compute the average attention scores over all tokens in the model-generated responses. We then average the attention scores across all test instances and present them for each layer, as shown in Figure 7.

As illustrated in the figure, both judge models assign higher attention scores to the responses generated by Llama-3.1-8B-Instruct compared to those from Llama-3.1-8B (as indicated by the bottom row showing the attention difference). We hypothesize that this is due to the generally higher response quality of Llama-3.1-8B-Instruct, as verified in Figure 2, which leads to greater attention being paid

to its outputs.

Moreover, we also observe that each model tends to assign more attention to its own responses than the other model does. For example, Llama-3.1-8B assigns higher attention to its own responses than Llama-3.1-8B-Instruct does, and vice versa (as indicated by the rightmost column showing the attention difference). This suggests that models naturally allocate more attention to their own responses, contributing to the emergence of self-preference.

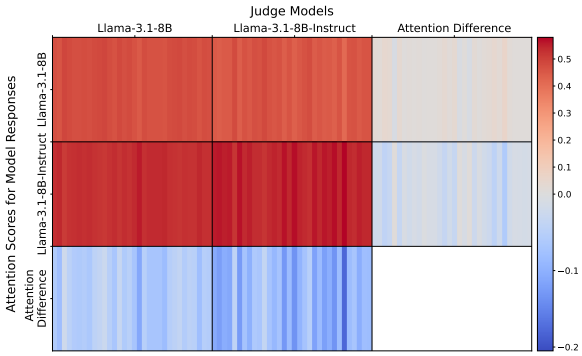


Figure 7: Attention scores of each layer in judge models. The scores are averaged over response tokens. The bottom row shows the difference in scores between Llama-3.1-8B and Llama-3.1-8B-Instruct responses for the same judge model. The rightmost column shows the difference in scores assigned by Llama-3.1-8B and Llama-3.1-8B-Instruct (as judges) to the same responses.

5 Related Work

5.1 Large Language Models for Judgment

LLMs are widely used in judgment tasks such as response ranking (Cui et al., 2023; Liu et al., 2023a), reward modeling (Lee et al., 2023; Wu et al., 2024), and verifying agent task completion (Qin et al., 2023; Xia et al., 2024), driven by their scalability and cost-effectiveness. Leveraging the inherent knowledge and instruction-following abilities of LLMs, researchers can guide these models to perform judgments by directly integrating rules into the prompts (Zheng et al., 2023; Sun et al., 2023). To further refine the judgment capabilities of LLMs in areas such as helpfulness and harmlessness (Bai et al., 2022; Wang et al., 2023c), numerous datasets and models have been developed (Lambert et al., 2024; Wang et al., 2023b; Zhu et al., 2023), greatly advancing the development and democratization of LLM-based judgment. Another active research area focuses on the meta-evaluation of LLM judges, examining the alignment between LLM judgments and human assessments (Zheng et al., 2023; Dubois et al., 2023), as well as identifying bias in these judges (Koo et al., 2023; Ye et al., 2024; Chen et al., 2024). In this work, we focus on self-preference bias and propose a novel method to more accurately quantify it in LLMs.

5.2 Bias in Large Language Models

Extensive studies reveal that LLMs are subject to biases such as length bias (Zheng et al., 2023; Hu et al., 2024), position bias (Zhu et al., 2023; Shi et al., 2024), and self-preference bias (Ye et al., 2024; Wataoka et al., 2024) in judgment tasks. In this work, we focus on self-preference bias, which refers to the tendency of LLMs to favor their own responses when serving as judges. While several studies have evaluated the presence of self-preference bias in specific models (Ye et al., 2024; Chen et al., 2024; Wang et al., 2023a), a comprehensive analysis across models of different versions, sizes, and reasoning capabilities is still lacking. Although concurrent work (Chen et al., 2025) conducts large-scale experiments to assess self-preference bias across model families, their focus lies primarily on verifiable tasks such as mathematical reasoning. In contrast, our study centers on open-ended tasks. In addition, several studies have investigated factors related to self-preference bias, such as self-recognition (Panickssery et al., 2024), self-enhancement (Xu et al., 2024), and preference leakage (Li et al., 2025). However, little attention has been given to mitigating this bias. In this work, we make an initial attempt to reduce self-preference bias by exploring two factors: response style and the data used for post-training.

6 Conclusions

In this work, we propose the DBG score to provide more accurate and reliable measurements of self-preference bias in LLMs. Using this metric, we conduct extensive experiments to evaluate self-preference bias across LLMs of varying versions, sizes, and reasoning abilities. Our further analysis reveals that both the response style and the post-training data of judge models can influence and help alleviate self-preference bias. Additionally, we explore the underlying mechanisms of this bias from an attention-level perspective. Overall, our study underscores the importance of recognizing and addressing self-preference bias when deploying LLMs as judges, and it offers actionable insights into strategies for reducing such bias.

Limitations

In this work, we employ GPT-4o-mini, Gemini-1.5-Flash, and DeepSeek-V3 as gold judges to measure the self-preference bias of LLMs. Due to cost constraints, we do not utilize more powerful models, such as GPT-4o or Gemini-1.5-Pro. Using these more capable models could potentially provide more reliable gold-standard judgments, yielding more accurate measurements of self-preference bias. Furthermore, while we mitigate the impact of position bias and length bias through methods like response position swapping and length limitation, other biases, such as authority bias and sentiment bias (Ye et al., 2024), may still influence the results. Additionally, this work limits its scope to instruction-following and translation tasks. Further investigation is needed to explore the self-preference bias of LLMs in other tasks, such as agent tasks and dialogue tasks.

References

Anthropic. 2024. Claude 3.5 sonnet, 2024. https://www.anthropic.com/news/claude-3-5-sonnet.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, and 1 others. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669*.

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

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, and 1 others. 2023. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *Preprint*, arXiv:2305.14233.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

Wikimedia Foundation. 2019. Acl 2019 fourth conference on machine translation (wmt19), shared task: Machine translation of news.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yiju Guo, Ganqu Cui, Lifan Yuan, Ning Ding, Zexu Sun, Bowen Sun, Huimin Chen, Ruobing Xie, Jie Zhou, Yankai Lin, and 1 others. 2024. Controllable preference optimization: Toward controllable multi-objective alignment. *arXiv preprint arXiv:2402.19085*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Zhengyu Chen, and Hui Xiong. 2024. Explaining length bias in llm-based preference evaluations. In *ICLR 2025 Workshop on Navigating and Addressing Data Problems for Foundation Models*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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 arXiv:2309.17012*.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2023. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, and 1 others. 2024. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *arXiv preprint arXiv:2411.16594*.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. 2025. Preference leakage: A contamination problem in llm-as-a-judge. *arXiv preprint arXiv:2502.01534*.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2023. The unlocking spell on base llms: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, and 1 others. 2023a. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743*.

Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. 2023b. Llms as narcissistic evaluators: When ego inflates evaluation scores. *arXiv preprint arXiv:2311.09766*.

Remi Mir, Bjarke Felbo, Nick Obradovich, and Iyad Rahwan. 2019. Evaluating style transfer for text. *arXiv preprint arXiv:1904.02295*.

Phil Ostheimer, Mayank Nagda, Marius Kloft, and Sophie Fellenz. 2023. Text style transfer evaluation using large language models. *arXiv preprint arXiv:2308.13577*.

Arjun Panickssery, Samuel Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *Advances in Neural Information Processing Systems*, 37:68772–68802.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, and 1 others. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*.

Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791*.

Zhiqing Sun, Yikang Shen, Hongxin Zhang, Qinhong Zhou, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2023. Salmon: Self-alignment with instructable reward models. *arXiv preprint arXiv:2310.05910*.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, and 1 others. 2024a. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024b. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926*.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, and 1 others. 2023b. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087*.

Yuxia Wang, Haonan Li, Xudong Han, Preslav Nakov, and Timothy Baldwin. 2023c. Do-not-answer: A dataset for evaluating safeguards in llms. *arXiv preprint arXiv:2308.13387*.

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

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 arXiv:2407.19594*.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy. *arXiv preprint arXiv:2404.05692*.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Yang Wang. 2024. Pride and prejudice: Llm amplifies self-bias in self-refinement. *arXiv preprint arXiv:2402.11436*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, and 1 others. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736*.

Xin Zhang, Yanzhao Zhang, Dingkun Long, Wen Xie, Ziqi Dai, Jialong Tang, Huan Lin, Baosong Yang, Pengjun Xie, Fei Huang, and 1 others. 2024. mgte: Generalized long-context text representation and reranking models for multilingual text retrieval. *arXiv preprint arXiv:2407.19669*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631*.

A Appendix

A.1 Self-Preference of Proprietary Models

In this section, we attempt to analyze the self-preference bias of proprietary models. We select Claude-3.5-Haiku (Anthropic, 2024), Qwen-Plus (Yang et al., 2024), and GLM-4-Plus (GLM et al., 2024) for the experiment. Each model is paired with Llama-3.1-70B-Instruct. Since we do not have access to the output probabilities of these models, and preliminary experiments reveal a strong position bias, we classify any test sample where the output tokens (`A` or `B`) differ after swapping response positions as a tie in this experiment. The results are shown in Figure 8. Based on the figure, we observe that all three proprietary models exhibit significant position bias, with more than 45% of test samples yielding different judgment results after swapping response positions. When excluding the tied samples, we find that Claude-3.5-Haiku classifies its own responses as superior in $51.8\%/(51.8\% + 1.8\%) = 96.6\%$ of cases, which is higher than the gold judgment of 88.0%. This suggests that Claude-3.5-Haiku **may** exhibit self-preference bias. However, further work is needed to obtain the model's output probabilities to provide more accurate results.

A.2 Self-Preference on More Datasets

Figure 9 and Figure 10 respectively show the performance of LLM judges in terms of judging truthfulness on the AlpacaEval dataset and self-preference bias on the WMT19 (de-en) dataset. From the figures, we observe similar conclusions to those drawn from AlpacaEval (helpfulness). Specifically, both pre-trained and post-trained models exhibit self-preference bias. For instance, when acting as judges, models like Llama-3.1-8B and Llama-3.1-8B-Instruct tend to give higher scores to their own responses than gold judgments assign to those responses. For example, on the WMT19 dataset, when Llama-3.1-8B judges the response pairs of Llama-3.1-8B and Qwen2.5-7B, it exhibits a DBG score of 2.5%. Additionally, we observe that large-sized models exhibit less pronounced self-preference bias compared to smaller models. For example, in the AlpacaEval (truthfulness) task, when large-sized models are paired with small-sized models, the DBG scores of the large-sized models tend to be closer to zero than those of the small-sized models.

A.3 Content Variation in Text Transfer

To verify that the rewriting approach introduced in Section 4.2 has minimal impact on the semantic content of the text, this section presents an analysis of the representation shifts before and after rewriting. Specifically, we employ gte-multilingual-base (Zhang et al., 2024), a widely-used text representation model, to encode both the original responses generated by Llama-3.1-70B-Instruct and their rewritten counterparts. We use the embedding corresponding to the `[CLS]` token as the representation of each response. Then, we apply t-SNE (van der Maaten and Hinton, 2008) to visualize the changes in representations. The results are shown in Figure 11. As observed, the representations before and after rewriting exhibit a high degree of overlap, indicating that our rewriting method primarily transfers the style of the responses with minimal impact on their underlying semantics.

A.4 Consistency Between the Theoretical Bias Estimator and Implementation

In our experimental implementation, for a judge model A, we obtain the probabilities assigned to tokens `A` and `B` for each individual instruction. The token with the higher probability is selected as the winner. By aggregating the outcomes over all instructions, we compute the win rate, which can be formulated as $\mathbb{E}_x[\mathbb{I}[\sigma(\delta + b_A) > 0.5]]$, where $\mathbb{I}$ is the indicator function. While this procedure produces a binary (0-1) decision for each instruction rather than a continuous probability, it can be viewed as a thresholded approximation to the theoretical quantity $w_A = \mathbb{E}_x[\sigma(\delta + b_A)]$. Specifically, it can be seen as an approximation to sampling from a Bernoulli distribution with success probability $\sigma(\delta + b_A)$. The same applies to gold judge models. The approximation error is small when the underlying probabilities are well-separated (i.e., close to 0 or 1). This justifies the empirical procedure as a practical surrogate to the theoretical self-preference bias formulation.

A.5 Few-shot Setting Analysis

To guide pre-trained models in making judgments, we leverage their few-shot learning ability and prepend examples to each input. For post-trained models, due to their strong instruction-following ability, we prompt them to make judgments in a zero-shot setting. To investigate the differences in

Model	Paired Response	Win Rate	
		Zero-shot	Few-shot
Llama-3.1-8B-Instruct	Llama-3.1-8B	82.0%	78.0%
	Qwen2.5-7B-Instruct	54.0%	53.2%
Llama-3.1-70B-Instruct	Llama-3.1-70B	83.5%	84.9%
	Qwen2.5-72B-Instruct	50.0%	48.0%
Qwen2.5-7B-Instruct	Qwen2.5-7B	68.1%	69.8%
	Llama-3.1-8B-Instruct	59.7%	60.6%

Table 3: Comparison of post-trained models judgments to their responses under zero-shot and few-shot settings.

judgment between zero-shot and few-shot settings for post-trained models, we conduct judgment experiments under the few-shot setting. The results are shown in Table 3. From the table, we observe that the judgment results of the post-trained model in the zero-shot and few-shot settings are similar, indicating that the post-trained model is capable of generating appropriate judgments in the zero-shot setting, which validates the reasonableness of our experimental setup.

A.6 Prompt

We present the prompts used for response generation in Table 4, Table 5, and Table 6, and the prompts used for response judgment in Table 7, Table 8, and Table 9.

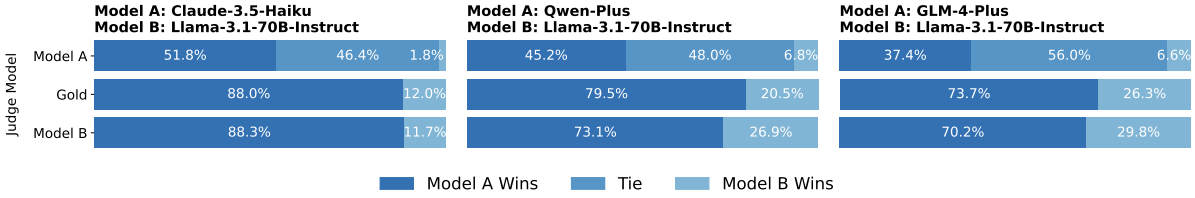
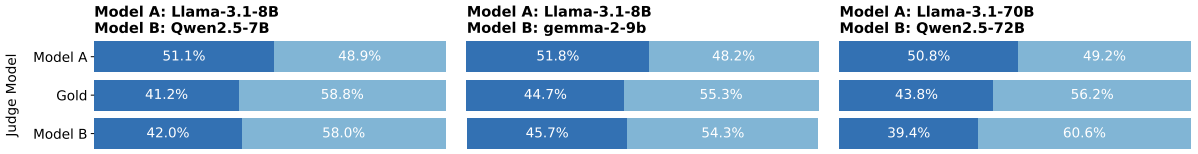
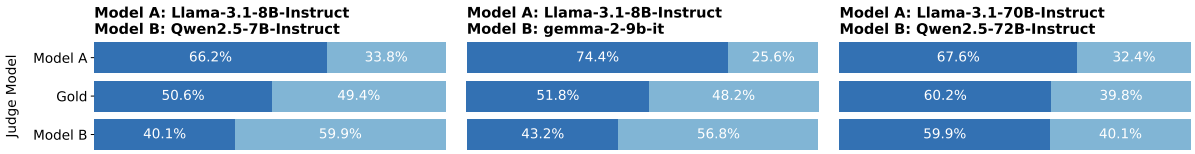


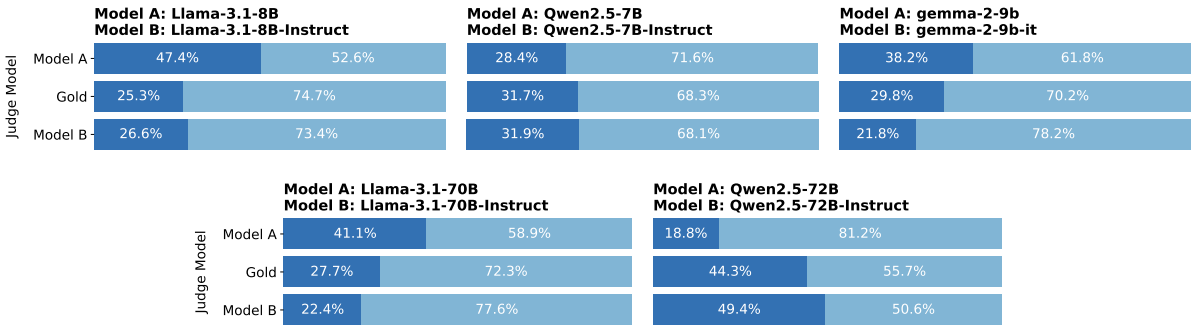
Figure 8: Judgment results for proprietary models on AlpacaEval (helpfulness).



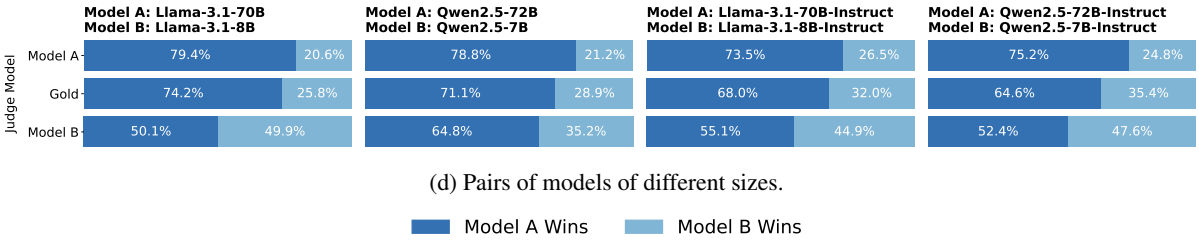
(a) Pairs of pre-trained models of the same size.



(b) Pairs of post-trained models of the same size.



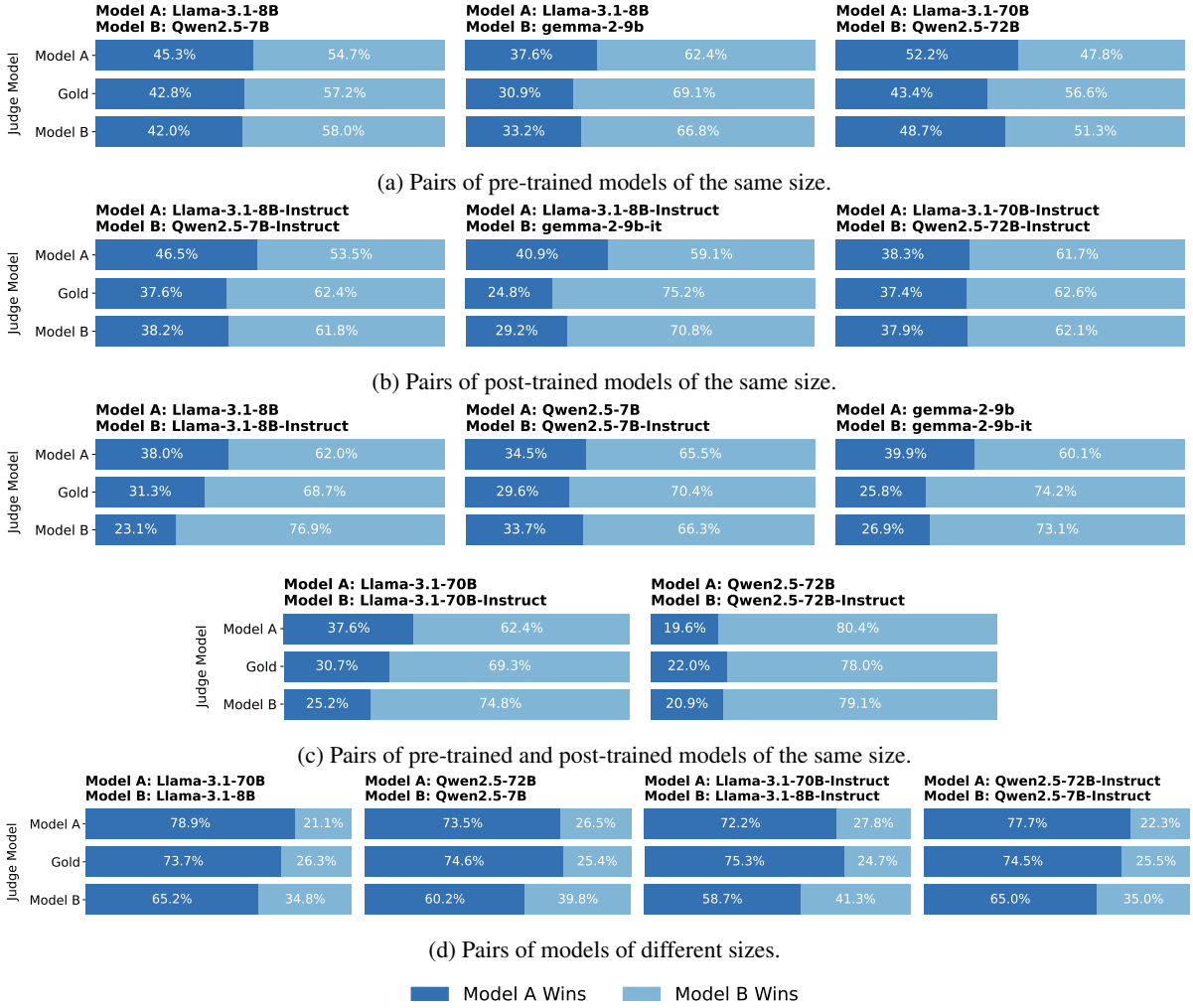
(c) Pairs of pre-trained and post-trained models of the same size.



(d) Pairs of models of different sizes.

Model A Wins Model B Wins

Figure 9: Judgment results on AlpacaEval (truthfulness).



(a) Pairs of pre-trained models of the same size.

(b) Pairs of post-trained models of the same size.

(c) Pairs of pre-trained and post-trained models of the same size.

(d) Pairs of models of different sizes.

Figure 10: Judgment results on WMT19 (de-en).

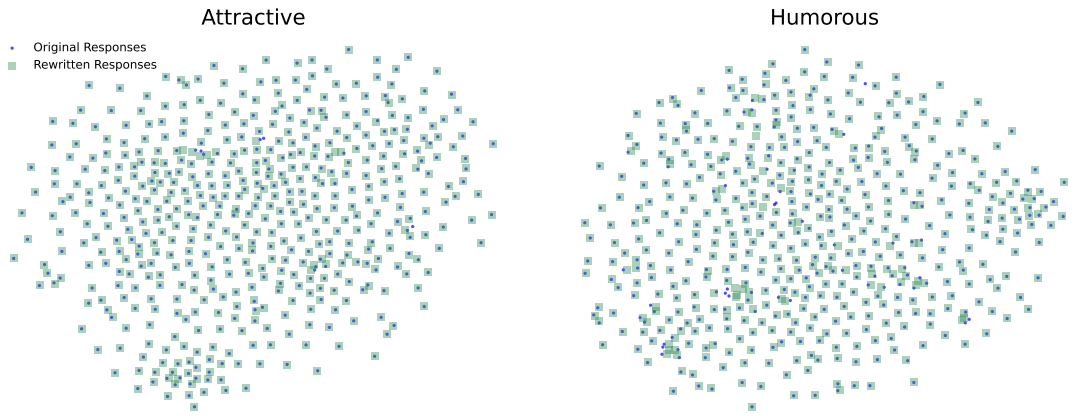


Figure 11: t-SNE visualizations of the embedding representations of responses before and after rewriting.

Post-trained Models:
You are a helpful assistant. Please help me answer the query below. Keep your answer within 200 words.

Query
{query}

Now, please provide your answer to this query.
Answer

Pre-trained Models:
Instruction
Below is a list of conversations between a human and an AI assistant (you).
As an AI assistant, you will engage in conversations with users, responding to their queries which are presented under the heading "# Query".
Your responses should be entered under the heading "# Answer".
You excel in a wide range of tasks including, but not limited to, providing general information, conducting reasoning, engaging in role-play, creative writing, planning, and solving mathematical and coding problems.
Your responses should be well-structured, comprehensive, and aim to thoroughly address the user's query or problem at hand. **Please keep your answer within 200 words.**

Query
{example 1 query}
Answer
{example 1 answer}

Query
{example 2 query}
Answer
{example 2 answer}

Query
{example 3 query}
Answer
{example 3 answer}

Query
{test query}
Answer

Table 4: Prompt used for response generation on AlpacaEval (helpfulness). The prompt for pre-trained models is derived from Lin et al. (2023).

Post-trained Models: You are a truthful assistant. Please help me answer the query below truthfully, remaining faithful to factual knowledge and the given context. Avoid making up any untrue facts. # Query {query} Now, please provide your answer to the query. Keep your answer within 200 words. # Answer
Pre-trained Models: # Instruction As an AI assistant, you will engage in conversations with users, responding to their queries which are presented under the heading "# Query". Your responses should be entered under the heading "# Answer". **You should respond to queries truthfully, remaining faithful to factual knowledge and the given context. Avoid making up any untrue facts.** Keep your answers within 200 words. # Query {example 1 query} # Answer {example 1 answer} # Query {example 2 query} # Answer {example 2 answer} # Query {test query} # Answer

Table 5: Prompt used for response generation on AlpacaEval (truthfulness).

Post-trained Models: You are an excellent translator, and you specialize in translating German into English. Given a piece of German text, please help translate it into English. Here is the given German text. # German {german} Now, please translate the German text into English. You only need to provide the English translation, with no other text. # English
Pre-trained Models: # Instruction You are an excellent translator, and you specialize in translating German into English. **Given a piece of German text, please translate it into English.** The German texts are under "# German", and the corresponding English translations are under "# English". # German {example 1 german} # English {example 1 english} # German {example 2 german} # English {example 2 english} # German {test german} # English

Table 6: Prompt used for response generation on WMT19 (de-en).

Post-trained Models:
You are a helpful assistant tasked with evaluating the quality of different responses to a given query. For each query, you will receive two independent responses. Please judge which response is better.

Here is the given query.
Query
{query}

Here are two independent responses (Response A and Response B) to the query.
Responses
<Response A>{response a}</Response A>
<Response B>{response b}</Response B>

Now, please judge which response is better. You only need to output A or B, with no other text.
Your output should be A if Response A is better or B if Response B is better.
Please remember that your response should start with either A or B

Pre-trained Models:
Instruction
You are a helpful assistant tasked with evaluating the quality of different responses to a query. You will receive multiple queries, each accompanied by two independent responses. For each query, you should judge which response you think is better.
Queries are placed under "# Query". The two responses for each query are listed under "# Responses" as "Response A" and "Response B". Your outputs should be placed under "# Judgment".
You only need to output A or B, with no other text.

Query
{example 1 query}
Responses
<Response A>{example 1 response a}</Response A>
<Response B>{example 1 response b}</Response B>
Judgment
{example 1 judgment}

Query
{example 2 query}
Responses
<Response A>{example 2 response a}</Response A>
<Response B>{example 2 response b}</Response B>
Judgment
{example 2 judgment}

Query
{test query}
Responses
<Response A>{test response a}</Response A>
<Response B>{test response b}</Response B>
Judgment

Table 7: Prompt used for judgment generation on AlpacaEval (helpfulness).

Post-trained Models:
You are a knowledgeable assistant. Given a query and two corresponding responses, your task is to determine which response is more truthful, ensuring it is less misleading, free from fabricated content, factual inaccuracies, and logical errors.

Here is the query.
Query
{query}

Here are two responses (Response A and Response B) to the query.
Responses
<Response A>{response a}</Response A>
<Response B>{response b}</Response B>

Now, please judge which response is more truthful. You only need to output A or B, with no other text. Please remember that your response should start with either A or B.

Pre-trained Models:
Instruction
You are a knowledgeable assistant. Given a query and two corresponding responses, your task is to determine which response is **more truthful**, ensuring it is less misleading, free from fabricated content, factual inaccuracies, and logical errors.
Queries are placed under "# Query". The two responses for each query are listed under "# Responses" as "Response A" and "Response B". Your outputs should be placed under "# Judgment".
You only need to output A or B, with no other text.

Query
{example 1 query}
Responses
<Response A>{example 1 response a}</Response A>
<Response B>{example 1 response b}</Response B>
Judgment
{example 1 judgment}

Query
{example 2 query}
Responses
<Response A>{example 2 response a}</Response A>
<Response B>{example 2 response b}</Response B>
Judgment
{example 2 judgment}

Query
{test query}
Responses
<Response A>{test response a}</Response A>
<Response B>{test response b}</Response B>
Judgment

Table 8: Prompt used for judgment generation on AlpacaEval (truthfulness).

Post-trained Models:
You are a helpful assistant tasked with evaluating the quality of two different English translations of the same German text. For each German text, you will receive two independent English translations. Please judge which English translation is better.

Here is the German text.
German
{german}

Here are two independent English translations (English A and English B) for the German text.
English
<English A>{english a}</English A>
<English B>{english b}</English B>

Now, please judge which English translation is better. You only need to output A or B, with no other text. Please remember that your response should start with either A or B

Pre-trained Models:
Instruction
You are a helpful assistant tasked with evaluating the quality of two different English translations of the same German text. For each German text, you will receive two independent English translations. Please judge which English translation is better.
The German texts are under "# German". The two independent English translations for each German text are under "# English", labeled as "English A" and "English B", respectively. Your outputs should be placed under "# Judgment".
You only need to output A or B, with no other text.

German
{example 1 german}
English
<English A>{example 1 english a}</English A>
<English B>{example 1 english b}</English B>
Judgment
{example 1 judgment}

German
{example 2 german}
English
<English A>{example 2 english a}</English A>
<English B>{example 2 english b}</English B>
Judgment
{example 2 judgment}

German
{test german}
English
<English A>{test english a}</English A>
<English B>{test english b}</English B>
Judgment

Table 9: Prompt used for judgment generation on WMT19 (de-en).