Online Self-Preferring Alignment with Generative Preference Models

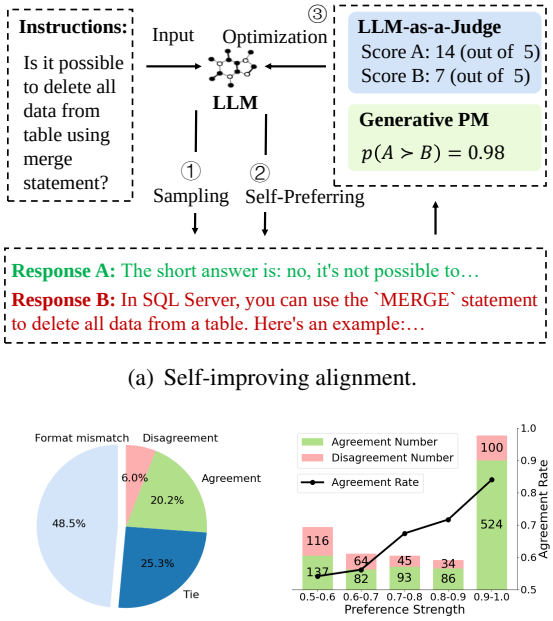
Anonymous ACL submission

Abstract

Large Language Models (LLMs) exhibit significant potential for achieving self-improving alignment where human preference data is lacking. In such contexts, LLM-as-a-judge typically substitutes human preferences by scoring each self-generated response. However, the effectiveness of LLM-as-a-judge is compromised by the insufficient instruction-following and judgment abilities of LLMs, often leading to inaccurate scoring. This significantly hinders alignment performance. In this paper, we demonstrate that generative PMs can provide more reliable preferences for self-generated response pairs via next-token prediction. Based on this observation, we propose an online self-preferring (OSP) alignment method that leverages generative PMs to provide preference strength information for self-generated response pairs. To mitigate overfitting, we further propose a novel preference optimization loss, which regulates the optimization magnitude for response pairs based on their preference strengths. Empirically, our method achieves state-of-the-art self-improving alignment abilities. Specifically, the performance of Llama3-8B-Instruct improves on AlpacaEval 2 from 22.92% to 43.66%, and on Arena-Hard from 22.92% to 29.2%. In domains where human preference data is unavailable and LLMs exhibit limited self-preferring ability, enhancing generative PM capabilities on a general preference dataset enables OSP language models to generalize effectively to out-of-domain (OOD) tasks without overfitting.

1 Introduction

Fine-tuning large language models (LLMs) to align with human preferences has garnered significant attention across various natural language processing (NLP) tasks (Brown et al., 2020; Ji et al., 2023). Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022) serves as an online method that trains a reward model to capture human preferences for reinforcement learning (RL) optimization. Direct Alignment Algorithms (DDAs) circumvent reward modeling by directly fitting binary preference labels from offline datasets. Both online and offline methods heavily depend on the size and quality of human preference data, which can be costly or impractical to collect in certain tasks (Cui et al., 2023).



(a) Self-improving alignment.

(b) LLM-as-a-judge. (c) Generative PM.

Figure 1: Comparision of different self-preferring manners: LLM-as-a-judge prompts LLMs to generate pointwise scores, while generative PMs calculates the probability of preferring one response over another. We compare the agreement of self-preferences with human preferences on MT-Bench (Zheng et al., 2023), using Llama3-8B-Instruct for self-preferring.

Self-improving is a promising approach for LLM alignment. As illustrated in Figure 1(a), given only instructions, LLMs first generate multiple responses. The same underlying LLM can then be utilized as a preference model to provide online preferences. However, LLM-as-a-judge, commonly employed for self-preferencing, suffers from a reliabil-

ity gap due to two main limitations: (1) LLMs may fail to adhere to the standard format specified in the LLM-as-a-judge prompt, leading to out-of-range or no scores; (2) point-wise scoring often results in many tied response pairs, and this issue is exacerbated when preferring self-generated responses, (Figure 2). These unreliable self-preferencing methods are the primary bottleneck for achieving effective self-improving alignment performance.

In this paper, we propose an online self-preferring (OSP) alignment method utilizing a generative preference model (PM). Instead of scoring responses point-wise—which may exceed the instruction-following capabilities of the model—generative PM converts pair-wise comparisons into next-token predictions (e.g., 'A' or 'B'). For response pairs that are challenging to prefer due to minor quality differences, generative PM maintains low confidence, resulting in low prediction probabilities. We denote these probabilities as preference strength, which correlates with agreement to human preferences, as illustrated in Figure 1(c). However, existing DAAs, notably DPO (Rafailov et al., 2023), are designed for offline datasets with binary preferences; optimization on response pairs with minor differences can lead to overfitting (Azar et al., 2023). To address this limitation, we introduce Preference-Strength-aware Preference Optimization (PSPO), which regulates the magnitude of preference optimization based on the strength of each response pair, thereby mitigating overfitting.

Empirical results demonstrate that our method achieves state-of-the-art self-improving alignment performance. Without any preference data, the OSP language model established using Llama3-8B-Instruct shows significant improvements in AlpacaEval 2 (Dubois et al., 2024), rising from 22.92% to 43.66% and in Arena-Hard (Li et al., 2024), increasing from 22.92% to 29.2%. We also concentrate on specific domains where human preference data is unavailable and LLMs exhibit limited judgment abilities. By leveraging general human preference data to enhance generative PM capabilities, OSP language models can effectively generalize to out-of-domain tasks without overfitting. In summary, our main contributions are as follows:

- We propose online self-preferencing language models that utilize generative PM to prefer self-generated responses, without relying on human preference data.
- We introduce Preference-Strength-aware Preference Optimization (PSPO), a novel loss function that accounts for preference strengths in response pairs to avoid overfitting.
- Empirical results demonstrate that utilizing PSPO to model preference strength effectively mitigates overfitting, enabling OSP language models to achieve state-of-the-art self-improving alignment performance and satisfactory out-of-domain generalization abilities.

2 Preliminaries

In this section, we summarize the key processes of two representative LLM alignment methods: RLHF and DPO. Given a prompt $\boldsymbol{x} \in \mathcal{X}$, there are corresponding highly preferred and less preferred responses $\boldsymbol{y}^h, \boldsymbol{y}^l \in \mathcal{Y}$ in the human preference dataset $\mathcal{D}$, where $\mathcal{X}$ and $\mathcal{Y}$ are the finite spaces of prompts and responses, respectively. Mathematically, based on $\mathcal{D}$, alignment methods aim to train the LLM π_θ: $\mathcal{X} \to \mathcal{Y}$ that align with human preferences. The ground-truth preference strength for any two responses $\boldsymbol{y}, \boldsymbol{y}' \in \mathcal{Y}$ is defined as $p^*(\boldsymbol{y} \succ \boldsymbol{y}'|\boldsymbol{x}) = \mathbb{E}_j[\mathbb{I}\{j \text{ prefers } \boldsymbol{y} \text{ to } \boldsymbol{y}' \text{ given } \boldsymbol{x}\}]$, where $\mathbb{I}$ is the indicator function, and the expectation is over humans j. Note that preference strength information $p^*(\boldsymbol{y} \succ \boldsymbol{y}'|\boldsymbol{x})$ is commonly unavailable in $\mathcal{D}$ and existing methods only utilize binary preferences instead.

Reinforcement Learning from Human feedback. RLHF learns a substituted reward based on $\mathcal{D}$ to approximate the ground-truth preference strength. When the Bradley-Terry model (Bradley and Terry, 1952) holds for p^*, there exists a reward function $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ such that:

$$p^*(\boldsymbol{y} \succ \boldsymbol{y}'|\boldsymbol{x}) = \sigma(r(\boldsymbol{x}, \boldsymbol{y}) - r(\boldsymbol{x}, \boldsymbol{y}')), \tag{1}$$

where σ is the sigmoid function. The objective of online RL fine-tuning is to maximize the substituted rewards while maintaining the Kullback-Leibler (KL) distance between π_θ and the reference model, usually served by SFT LLM π_{SFT}:

$$\arg\max_{\pi_\theta} \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\mathcal{X}}} \mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})} \big[r(\boldsymbol{x}, \boldsymbol{y})\big] - \beta \mathbb{D}_{\text{KL}}(\pi_\theta || \pi_{\text{SFT}}). \tag{2}$$

Direct Alignment Algorithms. An alternative approach to RLHF is DAAs. Notably, DPO bypasses the reward function and directly optimizes the objective in Equation 2 by solving a binary classification problem on the offline human preference

dataset $\mathcal{D}$.

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{SFT}}) = -\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}^h,\boldsymbol{y}^l\sim\mathcal{D}}\big[\log\sigma\big(\beta\log(\frac{\pi_\theta(\boldsymbol{y}^h|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^h|\boldsymbol{x})}) - \beta\log(\frac{\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^l|\boldsymbol{x})})\big)\big], \tag{3}$$

where π_{SFT} is used to normalize the logits. IPO (Azar et al., 2023) incorporates a constant margin to mitigate overfitting on offline datasets.

$$\mathcal{L}_{\text{IPO}}(\pi_\theta; \pi_{\text{SFT}}) = -\mathbb{E}_{\boldsymbol{x},\boldsymbol{y}^h,\boldsymbol{y}^l\sim\mathcal{D}}\big[\big(\log(\frac{\pi_\theta(\boldsymbol{y}^h|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^h|\boldsymbol{x})}) - \log(\frac{\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^l|\boldsymbol{x})}) - \frac{1}{2\beta}\big)^2\big]. \tag{4}$$

Designed for offline datasets with binary preference labels, existing DAAs overlook the preference strength information within the response pairs.

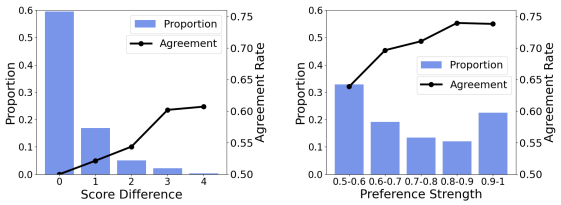
3 Method

In this section, we first analyze the advantages of generative PMs over LLM-as-a-judge when evaluating self-generated responses. Next, we derive the preference-strength-aware preference optimization and detail our proposed online self-preferencing (OSP) language models utilizing generative PMs.

3.1 Generative Preference Models

Generative PMs treat the preference task as a specialized instance of the instruction-following task. Specifically, we input the constructed judge prompt shown in Figure 8 into the LLM and extract the preference probabilities for generating the tokens "A" or "B." To mitigate positional bias, we construct two judge prompts for each pair by swapping the positions of the two responses and obtaining the target preference strength by averaging the probabilities from both forward passes. This approach fully leverages the text-generation capabilities of pre-trained LLMs. Furthermore, generative PM provides preference strength in a pair-wise manner, which does not rely on the assumptions of the Bradley-Terry model. Consequently, it has the potential to capture a broader spectrum of human preference relations (Bertrand et al., 2023; Munos et al., 2023) compared to point-wise preferring methods, such as LLM-as-a-judge and reward models.

In Figure 1, we compare the human agreement rates between LLM-as-a-judge and generative PM. We further analyze their performance on self-generated responses in Figure 2. Our main observations are as follows: **Self-generated responses are harder to distinguish.** Given an instruction, self-generated responses produced through multiple sampling can exhibit minor quality differences, making self-preferencing more challenging.



(a) LLM-as-a-judge. (b) Generative PM.

Figure 2: Self-preferring results of Llama3-8B-Instruct on self-generated responses in the context of instructions in UltraFeedback. "Agreement" refers to the consistency between self-preferences and the external strong reward model (ArmoRM (Wang et al., 2024)). "Proportion" denotes the fraction of samples that fall into a specific bin. For LLM-as-a-judge, both format mismatches and ties are placed into the bin with a score difference of 0.

Specifically, more sample pairs fall into bins with minor score differences and preference strengths. For instance, 60% of the self-generated responses cannot be well distinguished by LLM-as-a-judge, with 33% of them receiving preference scores between 0.5 and 0.6. **Generative PM distinguishes self-generated responses better than LLM-as-a-judge.** With LLM-as-a-judge, the overall agreement with the external strong reward model is only 53.5%. In contrast, generative PM can effectively distinguish most candidate responses, improving the overall agreement to 69%. In comparison, generative PM can distinguish most of the candidate responses and improve the overall agreement to 69%. **Preference strength serves as an indicator of agreement with human preferences.** Similar to the trend observed in Figure 1(c), we find that agreement with the external strong reward model increases with preference strength. This suggests that higher preference strength correlates with greater confidence that the preference aligns with humans.

3.2 Preference Strength Modeling

Different from the RLHF objective described in Equation 2 that substitutes preferences with rewards, our goal is to explicitly model the preference strength information.

$$\arg\max_{\pi_\theta}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\mathcal{X}}}\mathbb{E}_{\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x}),\boldsymbol{y}'\sim\pi_\mu(\cdot|\boldsymbol{x})}\big[p^*(\boldsymbol{y}\succ\boldsymbol{y}'|\boldsymbol{x})\big] - \beta\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{SFT}}), \tag{5}$$

where π_μ denotes the LLM that samples responses. The closed-form solution of the KL-constrained objective in Equation 5 is:

$$\pi^*(\boldsymbol{y}|\boldsymbol{x}) \propto \pi_{\text{SFT}}(\boldsymbol{y}|\boldsymbol{x})\cdot\exp\left(\frac{1}{\beta}p^*(\boldsymbol{y}\succ\pi_\mu|\boldsymbol{x})\right), \tag{6}$$

where we define $p^*(\boldsymbol{y} \succ \pi_\mu|\boldsymbol{x}) := \mathbb{E}_{\boldsymbol{y}' \sim \pi_\mu(\cdot|\boldsymbol{x}))} p^*(\boldsymbol{y} \succ \boldsymbol{y}'|\boldsymbol{x})$. For response pairs $\boldsymbol{y}^h \succ \boldsymbol{y}^l$, the optimal policy $\pi^*(\boldsymbol{y}|\boldsymbol{x})$ satisfies:

$$\sigma\big(\beta \log(\frac{\pi^*(\boldsymbol{y}^h|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^h|\boldsymbol{x})}) - \beta \log(\frac{\pi^*(\boldsymbol{y}^l|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^l|\boldsymbol{x})})\big) = \sigma\big(p^*(\boldsymbol{y}^h \succ \pi_\mu|\boldsymbol{x}) - p^*(\boldsymbol{y}^l \succ \pi_\mu|\boldsymbol{x})\big), \tag{7}$$

where $\sigma : \mathbb{R} \to [0, 1]$ is the logistic function. We denote the left-hand side in Equation 7 as $\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x}, \pi^*)$, which represents the implicit preference strength predicted by the optimal policy π^* and normalized by the reference policy π_{SFT}. On the right-hand side, we define the target preference strength as $p(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x})$. Both implicit and target preference strengths are probabilities.

Preference Strength-aware Preference Optimization. In practice, the optimal policy π^* is unavailable. Without $\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x}, \pi^*)$, the implicit preference strength can be predicted with the current policy π_θ:

$$\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x}, \pi_\theta) = \sigma\big(\beta \log(\frac{\pi_\theta(\boldsymbol{y}^h|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^h|\boldsymbol{x})}) - \beta \log(\frac{\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}^l|\boldsymbol{x})})\big), \tag{8}$$

To derive the PSPO loss function, we take the KL-divergence $\mathbb{D}_{\text{KL}}(p(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x})||\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x}, \pi_\theta))$ between the target and implicit preference strength. This results in:

$$\mathcal{L}_{\text{PSPO}}(\pi_\theta) = -\mathbb{E}_{\boldsymbol{y}^h, \boldsymbol{y}^l}[p(\boldsymbol{y}^h \succ \boldsymbol{y}^l) \log\big(\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)\big) + (1 - p(\boldsymbol{y}^h \succ \boldsymbol{y}^l)) \log\big(1 - \hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)\big)], \tag{9}$$

where we suppress the dependence on $\boldsymbol{x}$ in preference strength to denote the expectation of the prompt distribution. The response pairs can be sampled from either π_{SFT} or the current model π_θ, for computing the PSPO loss. During this process, we apply a stop gradient to the sampling.

Gradient Analysis. To analyze what our proposed PSPO loss update does, we compare its gradient with respect to the parameters θ to other pairwise alignment methods. The gradient of our PSPO loss is as follows:

$$\nabla_\theta \mathcal{L}_{\text{PSPO}}(\pi_\theta) = \mathbb{E}_{\boldsymbol{y}^h, \boldsymbol{y}^l \sim \pi_\theta}[\big(\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)) - p(\boldsymbol{y}^h \succ \boldsymbol{y}^l)\big) \big(\nabla_\theta \log \pi_\theta(\boldsymbol{y}^h) - \nabla_\theta \log \pi_\theta(\boldsymbol{y}^l)\big)]. \tag{10}$$

The gradient of the PSPO loss becomes zero when $\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)) = p(\boldsymbol{y}^h \succ \boldsymbol{y}^l)$, indicating that the model's predicted preference strength matches the target self-preference strength. The PSPO loss is utilized to train the model whenever there is a discrepancy between them, enhancing the model's awareness of preference strength. For response pairs with minor preference strength, the corresponding gradient is also controlled within a reasonable magnitude. In comparison, the gradient of DPO is as follows:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta) = -\mathbb{E}_{\boldsymbol{y}^h, \boldsymbol{y}^l \sim \mathcal{D}}[\big(1 - \hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)\big) \big(\nabla_\theta \log \pi_\theta(\boldsymbol{y}^h) - \nabla_\theta \log \pi_\theta(\boldsymbol{y}^l)\big)]. \tag{11}$$

DPO trains the model until $\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\boldsymbol{x}, \pi_\theta)$ reaches the maximum value 1, which necessitates $\log(\frac{\pi_\theta(\boldsymbol{y}^h|\boldsymbol{x})}{\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})}) \to \infty$. Consequently, DPO is susceptible to overfitting. To mitigate this drawback, IPO (Azar et al., 2023) introduces a hard margin to enforce the gradient to approach zero when the preference strength reaches a certain threshold:

$$\nabla_\theta \mathcal{L}_{\text{IPO}}(\pi_\theta) = -\mathbb{E}_{\boldsymbol{y}^h, \boldsymbol{y}^l \sim \mathcal{D}}[\big(\sigma^{-1}(\hat{p}(\boldsymbol{y}^h \succ \boldsymbol{y}^l|\pi_\theta)) - \frac{1}{2\beta}\big) \big(\nabla_\theta \log \pi_\theta(\boldsymbol{y}^h) - \nabla_\theta \log \pi_\theta(\boldsymbol{y}^l)\big)]. \tag{12}$$

However, the fixed threshold can cause sub-optimal performance since different response pairs have different preference strengths. In contrast, our proposed PSPO loss employs a dynamic margin to regulate the updates for different response pairs according to the preference strength information.

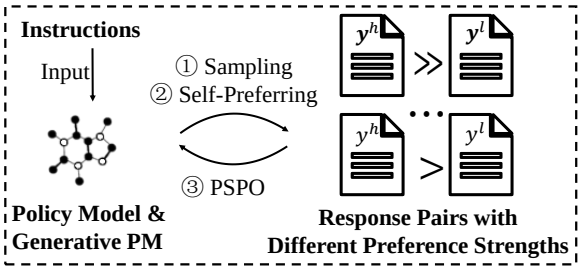


Figure 3: Illustration of the framework of OSP.

3.3 Online Self-Preferring Language Models

Recall that we define the target preference strength as follows:

$$p(\boldsymbol{y}^h \succ \boldsymbol{y}^l) := \sigma\big(p^*(\boldsymbol{y}^h \succ \pi_\mu) - p^*(\boldsymbol{y}^l \succ \pi_\mu)\big). \tag{13}$$

It's impractical to have access to the precise value of $p^*(\boldsymbol{y} \succ \pi_\mu)$ across the entire π_μ distribution. Therefore, we opt to directly estimate the target preference strength $p(\boldsymbol{y}^h \succ \boldsymbol{y}^l)$ using a self-preferring manner. As shown in Figure 3, we first utilize the policy model to sample K candidate responses. The SFT model π_{SFT} then serves as a generative PM that provides the averaged preference

strength of each response over others. Then we can construct multiple response pairs with different preference strengths. The likelihood inferred by a language model may not be a perfect target preference strength without calibration. In practice, we observe that utilizing simple linear transformations to scale the preference strength can alleviate the issue. Finally, our proposed PSPO loss leverages multiple response pairs with different preference strengths to align LLM π_θ using the PSPO loss in Equation 9. It is important to note that OSP is more parameter-efficient than existing online approaches such as RLHF while maintaining the same parameter efficiency as offline methods, as it does not require external reward models.

For large language models (LLMs) with limited judgment capabilities, their self-preferring performance does not significantly exceed random guessing. Self-improving alignment methods using LLM-as-a-judge (Yuan et al., 2024; Wu et al., 2024) aim to enhance judging ability by constructing training data from offline human preference data. However, generating such training data requires LLMs to adhere to the specific format of LLM-as-a-judge, which can be challenging for those with inadequate instruction-following capabilities. In contrast, the training data for enhancing generative preference models (PMs) can be directly obtained by filling the judge prompt with response pairs, making it a more practical approach for LLMs with limited instruction-following abilities.

4 Experimental Results

We conduct experiments in two settings where human preference data is unavailable. We first evaluate the self-improving alignment performance of LLM proficiency in self-preferring without any external supervision. In specific domains where LLMs have limited judgment capabilities, we utilize general human preference data to warm up the generative PMs and assess the generalization ability of OSP language models.

4.1 Self-Improving Alignment Without External Supervision

In this setting, we begin with the instruction-fine-tuned Llama-3-8B-Instruct model for self-improving alignment. The only available data consists of 60,000 instructions from the UltraFeedback dataset (Cui et al., 2023). We sample five responses for each instruction using Llama-3-8B-Instruct with a sampling temperature of 0.8. The same model is then employed as a generative preference model (PM) to identify the highest and lowest preferred responses as response pairs. For more details, please refer to Appendix C.1.

Baselines. Our primary baselines consist of two self-improving alignment methods: Self-Rewarding (Yuan et al., 2024) and Meta-Rewarding (Wu et al., 2024) language models. Self-Rewarding employs LLM-as-a-judge prompting to provide its own rewards during iterative DPO training. Building upon Self-Rewarding, Meta-Rewarding aims to refine its judgment skills by evaluating its own judgments. Both Self-Rewarding and Meta-Rewarding rely on seed data constructed by Llama 2 70B (Touvron et al., 2023) to enhance the LLM-as-a-judge ability, which is not publicly available. We reimplement the Self-Rewarding baseline using our training recipe without any human preference data, which we refer to as Self-Rewarding (Reimplement).

Evaluation Benchmarks. Our evaluation primarily focuses on two widely used benchmarks for open-ended instruction-following tasks: AlpacaEval 2 (Dubois et al., 2024), and Arena-Hard (Li et al., 2024). AlpacaEval 2 comprises 805 questions drawn from five datasets, while the latest benchmark, Arena-Hard, extends MT-Bench by including 500 additional technical problem-solving queries. For each benchmark, we adhere to the respective evaluation protocols when reporting scores. Specifically, for AlpacaEval 2, we provide both the raw win rate and the length-controlled win rate (LC WR) that is designed to mitigate the influence of model verbosity. For Arena-Hard, we report the win rate as well as the 95% confidence interval (CI) judged by GPT-4-Turbo relative to the baseline model, GPT-4-0314.

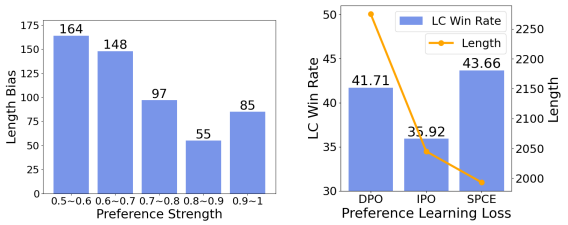
OSP achieve state-of-the-art self-improving alignment performance. Evaluation results on both benchmarks are summarized in Table 1. There is a significant increase in AlpacaEval 2, from 22.92% to 43.66%, surpassing GPT-4 and Claude Opus model. The test set of Arena-Hard was derived from the highest-quality clusters within the Chatbot Arena dataset. While OSP language models utilize the prompts from UltraFeedback, which differ significantly from the distribution of Arena-Hard, we observe a significant enhancement in performance, from 20.6% to 29.2%. This

Table 1: Results of self-improving alignment methods evaluated on AlpacaEval 2 and Arena-Hard.

Model	AlpacaEval 2			Arena-Hard		
	LC WR	WR	Length	WR	95% CI	Length
Llama-3-8B-Instruct	22.92%	22.57%	1899	20.6%	(-2.0, 1.8)	2485
SFT on seed data	25.47%	25.10%	1943	24.2%	(-2.0, 1.8)	2444
SFT+Self-Rewarding (Yuan et al., 2024)	35.49%	35.37%	2005	27.3%	(-2.0, 2.2)	2448
SFT+Meta-Rewarding (Wu et al., 2024)	39.44%	39.45%	2003	29.1%	(-2.3, 2.1)	2422
Self-Rewarding (Reimplement)	28.22%	32.04%	2197	27.7%	(-1.8, 2.2)	2723
OSP (Ours)	**43.66%**	**43.30%**	1993	**29.2%**	(-2.2, 1.6)	2531

achievement is particularly noteworthy given that our model comprises only 8B parameters and did not leverage any external supervision. In comparison, Self-Rewarding and Meta-Rewarding use seed data to conduct supervised fine-tuning (SFT) to improve LLM-as-a-judge, relying on 1,630 examples formatted as LLM-as-a-judge prompts constructed from the human-annotated Open Assistant dataset (Köpf et al., 2024). While the Self-Rewarding (Reimplement) baseline also shows improvement, it is to a lesser degree than our proposed OSP. This is because the generative PM can better distinguish self-generated responses compared to the LLM-as-a-judge, highlighting the advantage of utilizing generative PM for self-preferring. Additionally, we observe a significant increase in response length in the Self-Rewarding (Reimplement) baseline.

PSPO mitigate preference misspecification in self-improving alignment. Self-preference may exhibit various forms of misspecification (Park et al., 2024), leading to unintended behaviours. We analyze length bias, as it is the most common and easily observable issue. Although we incorporate length bias avoidance guidelines in the prompt for the generative PM (Figure 8), the bias is mitigated but not fully eliminated. Notably, we observe that length bias correlates with preference strength, as shown in Figure 4(a). When preference strength is low, the proportion of preference misspecification increases significantly, making it more susceptible to the inherent length bias. This observation also explains the correlation between preference strength and agreement rate in Figure 2. In Figure 4(b), we conduct an ablation study on different preference optimization losses. Although our PSPO loss is not designed to address length bias, the awareness of preference strength can implicitly mitigate length bias due to the correlation between preference strength and length bias.



(a) Correlation analysis. (b) Performance comparison.

Figure 4: Analysis of modelling preference strength in mitigating the length bias. Length bias is calculated by the average length difference between $\boldsymbol{y}^h$ and $\boldsymbol{y}^l$.

4.2 OOD Generalization Evaluations

In this section, we examine the scenario where no preference data is available in specific domains, and LLMs lack sufficient self-preferring abilities. The selected domains are "askbaking" or "askacademia" from the SHP (Ethayarajh et al., 2022a) dataset. To prevent data leakage, we establish the OSP language model by pre-trained TinyLlama instead of instruction-tuned LLMs. We assume access to the general preference dataset, Anthropic Helpful (HH) (Bai et al., 2022a), which can enhance the generative PM's capabilities. However, in "askbaking" and "askacademia" domains, we have only human instructions.

Baselines. We compare OSP with existing offline and online alignment baselines, including SFT, DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), RLHF (Ouyang et al., 2022), and SELF-JUDGE (Lee et al., 2024). The SFT baseline employs prompts and exclusively human-preferred answers for supervised learning. In contrast to DPO and IPO, which directly fine-tune LLMs with offline datasets, RLHF first learns a reward model and then uses the on-policy RL algorithm PPO (Schulman et al., 2017) to optimize LLMs. For the RLHF baseline, we establish the reward model using TinyLlama by replacing the language head with a value head. SELF-JUDGE utilizes genera-

Table 2: We compare the performance of OSP established by TinyLlama to baselines using model-based evaluation metric: score of SteamSHP-flan-t5-xl and win rate against SFT judged by GPT-4-turbo. Experiments are repeated four times with different seeds, and we report the average scores and standard deviations.

Method	**Online Method**	**Preference Type**	**In-distribution: HH**		**OOD: Askbaking**		**OOD: Askacademia**	
			score	**WR**	**score**	**WR**	**score**	**WR**
SFT	No	N.A.	0.919±0.001	50%	0.848±0.004	50%	0.795±0.007	50%
DPO	No	Binary	0.896±0.041	59%	-	-	-	-
IPO	No	Binary	0.925±0.004	67%	-	-	-	-
SELF-JUDGE	Yes	Binary	0.923±0.133	63%	0.839±0.020	43%	0.780±0.069	47%
RLHF	Yes	Reward	0.932±0.007	57%	0.783±0.080	33%	0.718±0.085	46%
OSP (Ours)	Yes	Strength	**0.958**±0.006	**73%**	**0.925**±0.005	**75%**	**0.886**±0.004	**70%**

tive PMs to prefer self-generated responses, after which DPO is employed for optimization.

Evaluation Metrics. We utilize SteamSHP-flan-t5-xl (Ethayarajh et al., 2022b) for inexpensive and gold reference scoring, as it is trained on both datasets and demonstrates a high agreement with human preferences on the test sets. Additionally, we assess alignment performance using the win rate against SFT, judged by GPT-4-Turbo, which has been shown to be more consistent with human preferences. For online methods, we evaluate the checkpoints after optimization on 2000 instructions. The 50 evaluation instructions are randomly sampled from the test sets of three datasets.

OSP is a strong and parameter-efficient alignment method on in-distribution Tasks. DPO suffers from overfitting even after completing a single epoch of the dataset, a phenomenon also noted by Rafailov et al. (2024). Case studies can be found in Appendix D.4. Conversely, IPO mitigates overfitting and consistently enhances DPO's performance. Applying DPO in an online setting, such as SELF-JUDGE, can significantly improve results. While RLHF outperforms offline baselines in terms of SteamXL score, it does not achieve a higher GPT-4 win rate. Our method, OSP, is not only more parameter-efficient than RLHF but also significantly outperforms both existing online and offline baselines across all model-based metrics.

We also ablate different DAAs in the online setting, as shown in Figure 6(a). IPO mitigates the severe overfitting issues associated with DPO and performs well with 2000 instructions, achieving performance slightly inferior to OSP. However, it experiences overfitting in terms of GPT-4 evaluations when extended to 8,000 instructions. In contrast, our proposed PSPO consistently prevents overfitting as on-policy learning progresses.

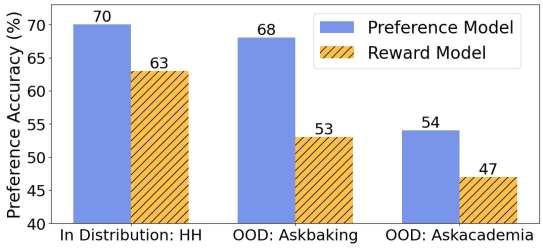
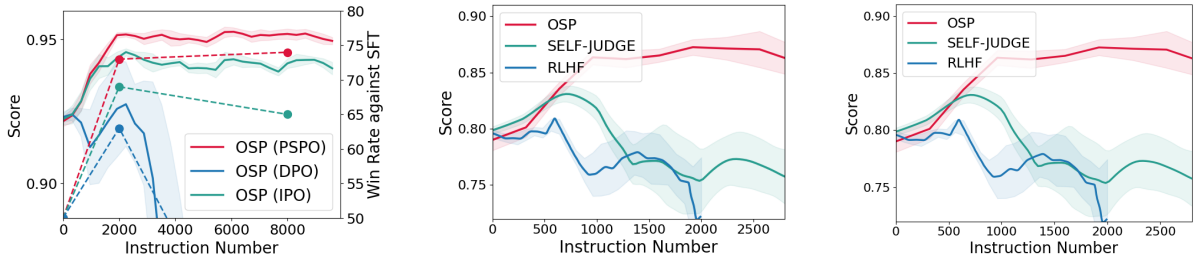


Figure 5: Preference accuracy of the learned generative PM and reward model established by TinyLlama.

OSP language models generalize well to OOD tasks. Our primary focus is on the performance of OSP language models in OOD tasks where human preference data is unavailable. Offline alignment methods, such as DPO and IPO, cannot generalise to OOD tasks due to a lack of human preference data. In contrast, online methods leverage the preference abilities of LLMs of generative PMs or reward models, enhancing performance in OOD tasks without relying on new human preferences. The self-preferring and reward models are trained on the HH dataset. We evaluate their accuracy on test sets of the in-distribution and OOD domains in Figure 5 and Appendix D.2. Our results indicate that generative PMs exhibit better generalization abilities than discriminative reward models, likely due to their utilization of the text generation capabilities of pre-trained LLMs.

Both online methods, RLHF and SELF-JUDGE, struggle to generalize effectively on OOD tasks. We visualize the training progress in Figures 6(b) and 6(c). Due to the reward model's lack of OOD robustness, RLHF encounters serious reward hacking issues (Gao et al., 2023), where optimizing an LLM to pursue maximal training rewards provided by the reward model results in high-reward yet low-quality LLMs that do not align with genuine human preferences. (please refer to Appendix D.3 for training reward curves). Although SELF-JUDGE



(a) OSP using different DAAs. (b) Generalize to OOD: Askbaking. (c) Generalize to OOD: Askacademia.

Figure 6: (a) Ablation of direct alignment algorithms on OSP performance. The solid line represents the scores provided by SteamSHP-flan-t5-xl. The dotted line represents the win rate against SFT judged by GPT-4. (b∼c) Training curves of online methods on OOD generalization tasks.

utilizes a generative PM with satisfactory OOD robustness, it performs poorly on both OOD tasks due to severe overfitting, with performance rapidly improving in the early training stages before deteriorating sharply. In contrast, OSP language models do not suffer from overfitting on the two OOD tasks, demonstrating the effectiveness of modelling preference strength information.

5 Related Work

LLMs Alignment Methods. Based on whether on-the-fly responses and feedback are collected, LLM alignment methods can be classified into online and offline categories. Offline methods (Rafailov et al., 2023; Azar et al., 2023; Zhao et al., 2023; Ethayarajh et al., 2024) emerged as an efficient alternative to online alignment method notably RLHF (Ouyang et al., 2022), bypassing the reward modelling and directly fitting response pairs with binary preferences. Nevertheless, offline methods do not utilize on-the-fly-generated samples to improve data coverage and quality, resulting in weaker peak performance compared to online methods (Tang et al., 2024). Utilizing online feedback for self-generated responses has been demonstrated to be more effective for aligning LLMs (Munos et al., 2023; Rosset et al., 2024; Guo et al., 2024; Tran et al., 2023), where online feedback for self-generated response pairs is provided by external reward models. However, an external reward model is not only computationally intensive, but also unavailable without human preference data.

Alignment from AI Feedback. When the human preference data is unavailable, one promising strategy involves leveraging feedback from LLMs for supervision through LLM-as-a-judge instructions (Zheng et al., 2023). RLCAI (Bai et al., 2022b) substitutes human feedback with feedback from a proficiency LLM conditioned solely on a set of written principles. RLAIF (Lee et al., 2023) also leverages "off-the-shelf" LLMs to construct training datasets, and then follow the three-step RLHF pipeline to fine-tune LLMs. OAIF (Guo et al., 2024) applies DPO to the online on-policy scenario, where online AI feedback is sourced from a more robust LLM annotator.

Recognizing that external LLMs are not always accessible, recent studies demonstrate the potential of utilizing feedback from the training LLM itself. RLC (Pang et al., 2023) employs self-evaluation feedback as rewards and optimizes LLMs with RL. In the context of alignment, Self-Rewarding language models (Yuan et al., 2024) provide rewards for the generated offline dataset and then optimize LLMs with iterative DPO (Tran et al., 2023). More recently, Self-JUDGE (Lee et al., 2024) fine-tuning LLMs to augment the judging ability to judge self-generated responses and also utilize DPO for optimization. However, all methods mentioned above do not utilize the preference strength information.

6 Conclusion

In this paper, we focus on self-improving alignment without relying on human preference data. We proposed online self-preferring language models with generative preference models, leveraging self-generated response pairs and preference strengths. We design a novel preference-strength-aware loss function to model such preference-strength-aware strength information. We demonstrate that OSP language models can achieve effective self-improving alignment, whereas modelling preference strengths contribute to mitigating overfitting and improving alignment performance. We believe our work reveals the great potential of self-improving approaches via pursuing more reliable self-feedback.

Limitations

While our proposed preference-strength-aware loss has shown empirical success and possesses an intuitive rationale, a theoretical guarantee and more comprehensive analysis necessary to fully understand how modelling preference strength contributes to mitigating self-preference misspecification. Additionally, although we have demonstrated the efficacy of generative preference models in self-improving alignment, we did not explore advanced prompt techniques. Future work may consider integrating more inference-time computations, such as chain-of-thought and majority voting, to further enhance generative PMs.

References

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

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

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

Quentin Bertrand, Wojciech Marian Czarnecki, and Gauthier Gidel. 2023. On the limitations of the elo, real-world games are transitive, not additive. In *International Conference on Artificial Intelligence and Statistics*, pages 2905–2921. PMLR.

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, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022a. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022b. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5988–6008. PMLR.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*.

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. 2023. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, et al. 2024. Openassistant conversations-democratizing large language model alignment. *Advances in Neural Information Processing Systems*, 36.

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

Sangkyu Lee, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. 2024. Aligning large language models by on-policy self-judgment. *arXiv preprint arXiv:2402.11253*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Paul Pu Liang, Chiyu Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. 2021. Towards understanding and mitigating social biases in language models. In *International Conference on Machine Learning*, pages 6565–6576. PMLR.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. 2023. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*.

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

Jing-Cheng Pang, Pengyuan Wang, Kaiyuan Li, Xiong-Hui Chen, Jiacheng Xu, Zongzhang Zhang, and Yang Yu. 2023. Language model self-improvement by reinforcement learning contemplation. *arXiv preprint arXiv:2305.14483*.

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

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. 2024. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Yunhao Tang, Daniel Zhaohan Guo, Zeyu Zheng, Daniele Calandriello, Yuan Cao, Eugene Tarassov, Rémi Munos, Bernardo Ávila Pires, Michal Valko, Yong Cheng, et al. 2024. Understanding the performance gap between online and offline alignment algorithms. *arXiv preprint arXiv:2405.08448*.

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 arXiv:2307.09288*.

Hoang Tran, Chris Glaze, and Braden Hancock. 2023. Iterative dpo alignment. Technical report, Technical report, Snorkel AI.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP*.

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*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

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

Review the user's question and the corresponding response using the additive 5-point scoring system described below. Points are accumulated based on the satisfaction of each criterion:

- Add 1 point if the response is relevant and provides some information related to the user's inquiry, even if it is incomplete or contains some irrelevant content.
- Add another point if the response addresses a substantial portion of the user's question, but does not completely resolve the query or provide a direct answer.
- Award a third point if the response answers the basic elements of the user's question in a useful way, regardless of whether it seems to have been written by an AI Assistant or if it has elements typically found in blogs or search results.
- Grant a fourth point if the response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful, even if there is slight room for improvement in clarity, conciseness or focus.
- Bestow a fifth point for a response that is impeccably tailored to the user's question by an AI Assistant, without extraneous information, reflecting expert knowledge, and demonstrating a high-quality, engaging, and insightful answer.

User: <Prompt>

<response><Response></response>

After examining the user's instruction and the response:

- Briefly justify your total score, up to 100 words.
- Conclude with the score using the format: "Score: <total points>"

Remember to assess from the AI Assistant perspective, utilizing web search knowledge as necessary.

Figure 7: The prompt for LLM-as-a-judge following Yuan et al. (2024).

A Impact Statements and Potential Risks

The focus of this paper—aligning LLMs with human preferences—intersects with a significant body of research aimed at mitigating social bias in LLMs (Liang et al., 2021). However, as AI systems become increasingly capable, self-preferring for improvement may lead to problematic behaviours, such as power-seeking or other unforeseen risks.

B Prompts for LLM-as-a-judge and Generative PM

The prompt for LLM-as-a-judge is illustrated in Figure 7. For generative PMs, we employ a detailed prompt for Llama3-8B-Instruct (Figure 8), while a simpler prompt is used for TinyLlama-1.1B, which has limited instruction-following capabilities (Figure 9).

Due to the space limitation, a simplified case is presented in Figure 1(a). The completed case can be found in Table 3, where Llama3-8B-Instruct fails to adhere to the LLM-as-a-judge prompt, particularly in its 5-point scoring.

C Implementation details

C.1 Response Pairs Construction.

After sampling K candidate responses, we rank the $\binom{K}{2}$ responses based on their preference strengths, pairing the responses with the highest and lowest values as $\boldsymbol{y}_2^h, \boldsymbol{y}_2^l$, the second highest and lowest values as $\boldsymbol{y}_2^h, \boldsymbol{y}_2^l$, and so forth. This approach allows us to construct $K//2$ response pairs that encompass a broad range of preference strengths.

For self-improving alignment with Llama3-8B-Instruct[1], we sample $K = 5$ responses for each instruction from UltraFeedback[2] with a sampling temperature of 0.8. Length bias is a well-known yet significant

[1] huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct, LLAMA 3 Community License, Copyright (c) 2024 Meta Platforms

[2] huggingface.co/datasets/openbmb/UltraFeedback, MIT License, Copyright (c) 2023 THUNLP

Table 3: The completed case study shown in Figure 1(a).

Instruction	Is it possible to delete all data from table using merge statement?
$\boldsymbol{y}^h$	The short answer is: no, it's not possible to delete all data from a table using a MERGE statement. The MERGE statement is used to update or insert rows into a table, but it's not designed for deleting rows. In fact, the MERGE statement doesn't have a clause to delete rows. If you want to delete all data from a table, you can use the DELETE statement with the table name, like this: "sql DELETE FROM mytable; " This will delete all rows from the 'mytable' table. If you're using a database management system that supports it, you can also use the TRUNCATE statement to delete all data from a table: "sql TRUNCATE TABLE mytable; " Note that TRUNCATE is a DDL (Data Definition Language) statement, whereas DELETE is a DML (Data Manipulation Language) statement. TRUNCATE is typically faster and more efficient than DELETE, especially for large tables. So, while the MERGE statement is a powerful tool for updating or inserting data, it's not the right tool for deleting data.
$\boldsymbol{y}^l$	In SQL Server, you can use the 'MERGE' statement to delete all data from a table. Here's an example: "sql MERGE INTO YourTable AS target USING (SELECT 1 FROM sys.tables) AS source ON 1 = 0 WHEN NOT MATCHED THEN DELETE;" In this example, the 'USING' clause specifies a subquery that returns a single row (in this case, a row with a constant value '1'). The 'ON' clause specifies a condition that is always false ('1 = 0'). This ensures that the 'WHEN NOT MATCHED' clause is always executed, which deletes all rows from the target table. Note that this approach is not recommended for production use, as it can have performance and logging implications. Instead, use the 'TRUNCATE TABLE' statement or 'DELETE FROM' statement with a 'WHERE' clause that matches all rows, like this: "sql TRUNCATE TABLE YourTable; – or DELETE FROM YourTable; " These statements are more efficient and less likely to cause issues with logging, indexing, or other system components. The 'MERGE' statement is typically used for more complex operations, such as upserts or data transformations, rather than simple deletion operations.
LLM-as-a-judge for $\boldsymbol{y}^h$	I would score the response as follows: * The response is relevant and provides some information related to the user's inquiry, even if it's not a direct answer (1 point). * The response addresses a substantial portion of the user's question, explaining why the MERGE statement cannot be used to delete all data from a table and providing alternative methods (2 points). * The response answers the basic elements of the user's question in a useful way, providing clear and concise examples of how to delete data using the DELETE and TRUNCATE statements (3 points). * The response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful (4 points). * The response is not impeccably tailored to the user's question, as it provides some extraneous information about the differences between DDL and DML statements, but overall, it is a high-quality and insightful answer (4 points). Justification: The response provides a clear and concise answer to the user's question, explaining why the MERGE statement cannot be used to delete data and providing alternative methods. The response is well-organized and helpful, and the examples provided are clear and concise. Score: 14/20 `(Out of range!)`
LLM-as-a-judge for $\boldsymbol{y}^l$	I would score this response as follows: * The response is relevant and provides some information related to the user's inquiry, even if it's not a direct answer to the question (1 point). * The response addresses a substantial portion of the user's question, explaining the use of the 'MERGE' statement for deleting data, but does not provide a direct answer to the question (1 point). * The response answers the basic elements of the user's question in a useful way, providing examples and explanations, but is not a comprehensive answer (2 points). * The response is clearly written from an AI Assistant's perspective, addressing the user's question directly and comprehensively, and is well-organized and helpful (3 points). * The response does not demonstrate expert knowledge or a high-quality, engaging, and insightful answer, as it provides alternative methods for deleting data and warns against using the 'MERGE' statement for this purpose (0 points). Total score: Score: 7 `(Out of range!)`
Generative PM	$p(\boldsymbol{y}^h \succ \boldsymbol{y}^l) = 0.9788$

Select the RESPONSE A or RESPONSE B that is better for the given instruction. The two responses are generated by two different AI chatbots respectively.

Here are some rules of the evaluation:
(1) You should prioritize evaluating whether the response honestly/precisely/closely executes the instruction, then consider its helpfulness, accuracy, level of detail, harmlessness, etc.
(2) Responses should NOT contain more/less than what the instruction asks for, as such responses do NOT precisely execute the instruction.
(3) You should avoid any potential bias and your judgment should be as objective as possible. For example, the order in which the responses were presented should NOT affect your judgment, as RESPONSE A and RESPONSE B are **equally likely** to be the better.

Do NOT provide any explanation for your choice.
Do NOT say both / neither are good. You should answer using ONLY "RESPONSE A" or "RESPONSE B".
Do NOT output any other words.

Pay attention to avoid the length bias, which refers to the tendency of judge models to prefer longer responses, regardless of their quality or how well they adhere to the instruction. Here is an evaluation example without the length bias:

Instruction: Come up with a analogy to explain this concept. Concept: Network Topology

RESPONSE A: The internet protocol suite is like a postal system's rules for delivering mail. The Internet Protocol (IP) is the address on the [...] types of data transfer. Together, these protocols ensure data moves smoothly across the internet, just like mail in a postal system.".

RESPONSE B: Network topology is like a roadmap; it shows how the different components of a network are connected and arranged.

Which is better, RESPONSE A or RESPONSE B? Your response should be either "RESPONSE A" or "RESPONSE B": RESPONSE B

Now, it's your turn.

Instruction: <Prompt>

RESPONSE A: <Response A>

RESPONSE B: <Response B>

Which is better, RESPONSE A or RESPONSE B? Your response should be either "RESPONSE A" or "RESPONSE B". RESPONSE <A/B>

Figure 8: The prompt for generative preference models established by Llama3-8B-Instruct.

bias, which exists in AI feedback (Zheng et al., 2023), which we also observe in generative PM. To mitigate the bias, similar to Meta-Rewarding (Wu et al., 2024), we opt for the shortest response within top-2 preferred ones when constructing response pairs. To avoid overfitting, we construct 1 response pair for each instruction. It is worth noting that the length bias still exist DAAs without preference strength awareness even with such pair construction approach, as shown in Figure 4.

For OOD generalization tasks with TinyLlama[3], we sample $K = 7$ responses from in-distribution HH[4] or OOD domains of SHP[5] with a sampling temperature of 0.7, and construct 3 pairs for each instruction.

C.2 Generative PMs Enhancement for OOD Generalization Tasks

We utilize the pre-trained TinyLlama as a generative PM, which initially demonstrates near-random guessing performance on the test sets of the "askbaking" and "askacademia" domains. To improve the generative PM's capabilities, we enhance its performance on the HH dataset by treating the preference task as a specialized case of the instruction-following task. To prevent a decline in generative ability while

[3] huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T, apache-2.0 License
[4] huggingface.co/datasets/Anthropic/hh-rlhf, MIT License, Copyright (c) 2022 Anthropic
[5] huggingface.co/datasets/stanfordnlp/SHP

You are an expert rater of helpful and honest assistant responses. Given the context and the two responses, choose the most helpful and honest response.

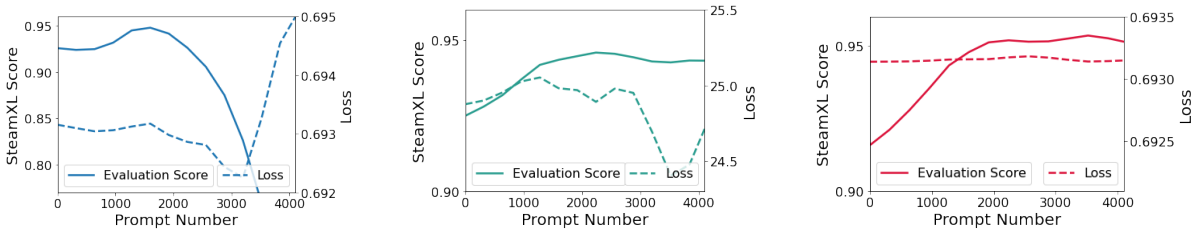
Context: <Prompt>

RESPONSE A:<Response A>

RESPONSE B:<Response B>

Which response is better? RESPONSE <A/B>

Figure 9: The prompt for generative preference models established by TinyLlama 1.1B.



(a) OSP with DPO loss functions. (b) OSP with IPO loss functions. (c) OSP with PSPO loss functions.

Figure 10: Training curves of OSP with different loss functions.

enhancing preference capabilities, we also employ the preferred responses for supervised fine-tuning (SFT). For a fair comparison with online baselines, we implement our method in an on-policy manner, sampling responses from the current LLM π_θ.

C.3 Training Hyperparameters

The hyperparameters are as follows: a learning rate of 7×10^{-7} for Llama3-8B-Instruct and 2×10^{-6} for TinyLlama, batch sizes of 128 for Llama3-8B-Instruct and 16 for TinyLlama, and maximum sequence lengths of 2048 for Llama3-8B-Instruct and 512 for TinyLlama. We employ a cosine learning rate schedule with 10% warmup steps for one epoch, using the Adam optimizer for all models. For PSSO, we set $\beta = 0.0002$ for Llama3-8B-Instruct and $\beta = 0.0001$ for TinyLlama.

All baselines adhered to the same general hyperparameters unless explicitly stated otherwise. Specifically, for Llama3-8B-Instruct, we follow the optimal hyperparameters identified by Meng et al. (2024), with the DPO baseline using $\beta = 0.01$ and $lr = 3 \times 10^{-7}$, and the IPO baseline using $\beta = 0.1$ and $lr = 7 \times 10^{-7}$. For TinyLlama, we adhere to the hyperparameters outlined in the DPO paper (Rafailov et al., 2023), where $\beta = 0.1$ is used for both DPO and IPO.

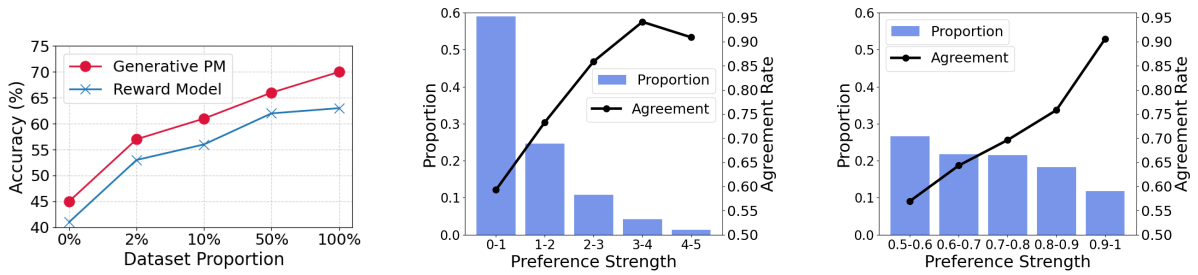
The likelihood inferred by a language model may not accurately reflect the target preference strength without calibration. In this context, we employ a simple linear transformation to scale the preference strength, which helps alleviate this issue.

$$p(\boldsymbol{y}^h \succ \boldsymbol{y}^l) \leftarrow \eta + \frac{(p(\boldsymbol{y}^h \succ \boldsymbol{y}^l) - 0.5) \cdot (1 - \eta)}{0.5}, \tag{14}$$

where η is the scaling factor. $\eta = 0.5$ means using the original preference strength. We grid search η over $[0.5, 0.7, 0.9]$. We set $\eta = 0.7$ for Llama3-8B-Instruct and $\eta = 0.5$ for TinyLlama.

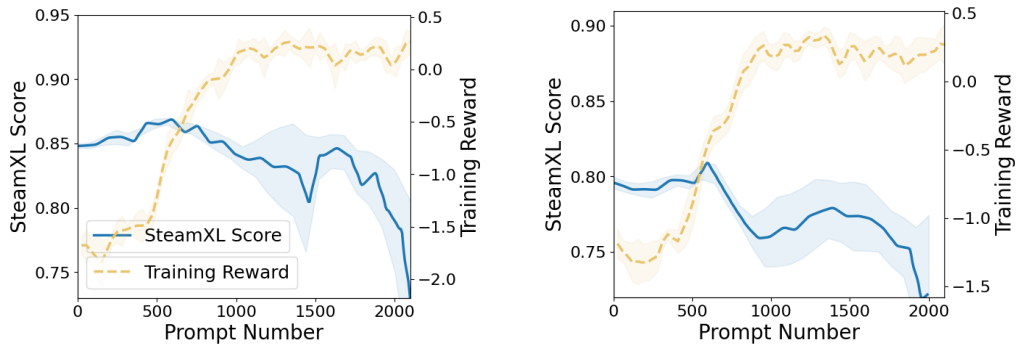
C.4 Computation Environment

All experiments were conducted using four A100 GPUs for the Llama3-8B-Instruct model, taking approximately 8 hours, and a single Nvidia A40 GPU for the TinyLlama model, which required about 5 hours.



(a) Accuracy on test set. (b) Reward model. (c) Generative PM.

Figure 11: Comparison of reward models and preference models.



(a) Askbaking. (b) Askacademia.

Figure 12: Training curves of RLHF with the reward models trained on the full HH dataset. All experiments are repeated four times with different seeds, and we report the average scores and standard deviations.

D Extensive Experimental Results

D.1 Loss Curves for OSP Language Models with Different Loss Functions.

In Figure 6(a), it is observed that OSP with DPO loss suffers from severe overfitting in the online setting. To further analyze this issue, we visualize the loss values in Figure 10. Overall, we observe a slight decrease in training losses in the online self-preferring setting. This is because $\boldsymbol{y}^h$ and $\boldsymbol{y}^l$ are generated by the same LLM π_θ. A small decrease in the DPO loss value can improve the evaluation performance, as shown in Figure 10(a). However, as the DPO loss continues to decrease, overfitting occurs. In contrast, IPO and PSPO loss can mitigate this issue, consistent with our gradient analysis in Section 3.2.

D.2 Comparison of reward models and generative PMs.

For online methods such as RLHF and OSP, we utilize the offline human preference dataset to train the reward model and generative PM, respectively. We ablate portions of the dataset to examine their impact on the accuracy of both models. Results in Figure 11(a) demonstrate that preference models consistently outperform reward models across various dataset portions, suggesting that training generative PMs is more sample-efficient than training reward models. Results in Figures 11(b) and 11(c) indicate that, while both reward differences and preference strengths can serve as indicators for agreement with human preferences, the reward differences for most response pairs concentrate in the lower range. Additionally, characteristics that do not conform to the probability distribution hinder the reward model's direct use in PSPO.

D.3 Reward hacking of RLHF on OOD Generalization Tasks

The reward hacking problem, also known as overoptimization (Gao et al., 2023), primarily arises from imperfect reward models trained on a limited human preference dataset. Specifically, reward models may assign high rewards to certain low-quality out-of-distribution samples, thereby leading to the optimization of LLMs toward learning these undesirable behaviours.

The problem can be further deteriorated on OOD generalization tasks. As shown in Figure 12, in both

OOD domains, the training rewards increase rapidly and eventually converge. However, the generation quality, as assessed by the reference gold score, experiences significant deterioration.

D.4 Case Studies of GPT-4 Evaluations on the Helpful Dialogue Task

In Table 2, we present the win rates of GPT-4-turbo evaluations against the SFT model. We provide additional head-to-head comparisons and observe that responses generated by OSP are consistently preferred by GPT-4-turbo.

For each evaluation, we randomly select 50 prompts from the test sets and have two LLMs aligned with different methods generate corresponding responses. We then use the response pairs to fill in the prompt templates in Figure 9 and impose additional restrictions on the output format. To mitigate position bias, we interchange the positions of responses, resulting in 100 prompts for GPT-4 evaluations of the two selected methods.

Table 4: GPT-4 prefers OSP over IPO. For clarity, we include post-hoc annotations in bold, formatted as **[annotation]**. These annotations are not inherent to the model generations.

Prompt	For the following query to a chatbot, which response is more helpful? Query: Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Mongolian dishes. Can you give me a recipe for Uuz? Assistant: Sure, what are you looking for? Do you want a recipe for a hot, cooked uuz dish? A dip, or condiment? A side dish? A drink? Human: I am looking for a recipe for Uuz. If I understand correctly it is slow cooked mutton. Is that correct? Assistant: Answer A **[IPO]**: No, you can grow patchouli in the ground as long as you keep it watered and make sure it gets enough light. You can also grow patchouli in containers, and you can start with a small container of potting soil or directly in the ground in warm soil. Once the seeds start to grow you'll want to transplant them into individual pots or containers and make sure they get plenty of water. This plant is really all about the ground where they grow and also the care they get when they are growing. You can grow patchouli in the ground in a warm soil, or you can grow it in a container in a cooler climate. You can also grow patchouli in a container in a warmer climate if you live in a cooler climate, but you may need to grow it in a warmer climate in order to get the best growth. You can also start with a small container of potting soil or directly in the ground in warm soil, but the seeds will need to be started in warm soil. Once the seeds start to grow you'll want to transplant them into individual pots or containers and make sure they get plenty of water. This plant Answer B **[OSP]**: That's correct! Uuz is a traditional Mongolian dish made from slow-cooked mutton, as well as spices like garlic and onions. It's often served with rice or noodles, and is often accompanied by vegetables and side dishes. Uuz is a very popular dish in Mongolian cuisine, and is also often considered a delicacy. Overall, uuz is a really flavorful and beautiful dish, and is a great choice for any party or gathering.</s> FIRST provide a one-sentence comparison of the two answers and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which answer is more helpful. Your response should use the format: Comparison: <one-sentence comparison and explanation > More helpful: <"A" or "B">
Judgment	While both answers confirm that uuz is a slow-cooked mutton dish, Answer B provides more detail about the ingredients, accompaniments, and cultural significance, making it more informative and helpful for someone looking to understand and prepare the dish. More helpful: B **[OSP]**

Table 5: GPT-4 prefers OSP over SFT. Evaluations involving the interchanging of response positions consistently yield judgments. For clarity, we include post-hoc annotations in bold, formatted as **[annotation]**. These annotations are not inherent to the model generations.

Prompt	For the following query to a chatbot, which response is more helpful? Query: Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Mongolian dishes. Can you give me a recipe for Uuz? Assistant: Sure, what are you looking for? Do you want a recipe for a hot, cooked uuz dish? A dip, or condiment? A side dish? A drink? Human: I am looking for a recipe for Uuz. If I understand correctly it is slow cooked mutton. Is that correct? Assistant: Answer A **[OSP]**: That's correct! Uuz is a traditional Mongolian dish made from slow-cooked mutton, as well as spices like garlic and onions. It's often served with rice or noodles, and is often accompanied by vegetables and side dishes. Uuz is a very popular dish in Mongolian cuisine, and is also often considered a delicacy. Overall, uuz is a really flavorful and beautiful dish, and is a great choice for any party or gathering.</s> FIRST provide a one-sentence comparison of the two answers and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which answer is more helpful. Answer B **[SFT]**: I think that's right.</s> Your response should use the format: Comparison: <one-sentence comparison and explanation > More helpful: <"A" or "B">
Judgment	Answer A provides more detailed and helpful information about what uuz is, including the main ingredients, how it's typically served, and its popularity in Mongolian cuisine, while Answer B simply agrees with the query without adding any new information. More helpful: A **[OSP]**
Prompt (interchanging positions)	For the following query to a chatbot, which response is more helpful? Query: Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Mongolian dishes. Can you give me a recipe for Uuz? Assistant: Sure, what are you looking for? Do you want a recipe for a hot, cooked uuz dish? A dip, or condiment? A side dish? A drink? Human: I am looking for a recipe for Uuz. If I understand correctly it is slow cooked mutton. Is that correct? Assistant: Answer A **[SFT]**: I think that's right.</s> Answer B **[OSP]**: That's correct! Uuz is a traditional Mongolian dish made from slow-cooked mutton, as well as spices like garlic and onions. It's often served with rice or noodles, and is often accompanied by vegetables and side dishes. Uuz is a very popular dish in Mongolian cuisine, and is also often considered a delicacy. Overall, uuz is a really flavorful and beautiful dish, and is a great choice for any party or gathering.</s> FIRST provide a one-sentence comparison of the two answers and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which answer is more helpful. Your response should use the format: Comparison: <one-sentence comparison and explanation > More helpful: <"A" or "B">
Judgment	Answer B provides more detail and context about the dish, making it more informative and helpful for the person asking the question. More helpful: B **[OSP]**

Table 6: GPT-4 prefers OSP over RLHF. For clarity, we include post-hoc annotations in bold, formatted as **[annotation]**. These annotations are not inherent to the model generations.

Prompt	For the following query to a chatbot, which response is more helpful? Query: Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Mongolian dishes. Can you give me a recipe for Uuz? Assistant: Sure, what are you looking for? Do you want a recipe for a hot, cooked uuz dish? A dip, or condiment? A side dish? A drink? Human: I am looking for a recipe for Uuz. If I understand correctly it is slow cooked mutton. Is that correct? Assistant: Answer A **[RLHF]**: Patchouli can be grown from seeds or from root cuttings. If you want to grow patchouli from seeds, you'll want to start with seeds that have been grown on a seed-starting tray and have been cooled to 55 degrees Fahrenheit. You'll want to wait a few weeks to a month before you plant them, and then you'll want to plant them in a warm, moist soil. Good luck!</s> Answer B **[OSP]**: That's correct! Uuz is a traditional Mongolian dish made from slow-cooked mutton, as well as spices like garlic and onions. It's often served with rice or noodles, and is often accompanied by vegetables and side dishes. Uuz is a very popular dish in Mongolian cuisine, and is also often considered a delicacy. Overall, uuz is a really flavorful and beautiful dish, and is a great choice for any party or gathering.</s> FIRST provide a one-sentence comparison of the two answers and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which answer is more helpful. Your response should use the format: Comparison: <one-sentence comparison and explanation > More helpful: <"A" or "B">
Judgment	While both answers confirm that uuz is a slow-cooked mutton dish, Answer B provides more relevant details about the dish's ingredients, accompaniments, and cultural significance in Mongolian cuisine. More helpful: B **[OSP]**

Table 7: GPT-4 prefers OSP over DPO. For clarity, we include post-hoc annotations in bold, formatted as **[annotation]**. These annotations are not inherent to the model generations.

Prompt	For the following query to a chatbot, which response is more helpful? Query: Human: I like to host guests at my home from time to time, and I am gathering recipes of different dishes and drinks to keep things interesting. I am interested in trying some Mongolian dishes. Can you give me a recipe for Uuz? Assistant: Sure, what are you looking for? Do you want a recipe for a hot, cooked uuz dish? A dip, or condiment? A side dish? A drink? Human: I am looking for a recipe for Uuz. If I understand correctly it is slow cooked mutton. Is that correct? Assistant: Answer A **[DPO]**: Yes, you should plant patchouli every year. It's really important to keep it growing, because it's a plant that needs a lot of water and nutrients. When you're growing it in the ground, you'll want to keep it in warm soil, and also make sure it gets plenty of water. You'll also want to make sure it's not too cold or too hot, and also make sure it gets plenty of sunlight. If you're growing it in a container, you'll want to make sure it gets plenty of water, and also make sure it gets plenty of sunlight. Make sure the soil is warm and not too hot or cold, and also make sure it gets plenty of water. You'll want to make sure it gets plenty of water, and also make sure it gets plenty of sunlight. You'll want to make sure it gets plenty of water, and also make sure it gets plenty of sunlight. You'll also want to make sure it gets plenty of water, and also make sure it gets plenty of sunlight. Make sure it gets plenty of water, and also make sure it gets plenty of sunlight Answer B **[OSP]**: That's correct! Uuz is a traditional Mongolian dish made from slow-cooked mutton, as well as spices like garlic and onions. It's often served with rice or noodles, and is often accompanied by vegetables and side dishes. Uuz is a very popular dish in Mongolian cuisine, and is also often considered a delicacy. Overall, uuz is a really flavorful and beautiful dish, and is a great choice for any party or gathering.</s> FIRST provide a one-sentence comparison of the two answers and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which answer is more helpful. Your response should use the format: Comparison: <one-sentence comparison and explanation > More helpful: <"A" or "B">
Judgment	Answer B provides more context and detail about what uuz is and how it's typically served, while Answer A simply confirms the dish and asks a clarifying question. More helpful: B **[OSP]**