Adversarial Preference Optimization: Enhancing Your Alignment via RM-LLM Game

Pengyu Cheng[*1] **Yifan Yang**[*1] **Jian Li**[*1] **Yong Dai**[1]
Tianhao Hu[1] **Peixin Cao**[1] **Nan Du**[1] **Xiaolong Li**[2]
Tencent AI Lab [1]Shenzhen & [2]Seattle
{pengyucheng,tobyfyang,jackjianli}@tencent.com

Abstract

Human preference alignment is essential to improve the interaction quality of large language models (LLMs). Existing alignment methods depend on manually annotated preference data to guide the LLM optimization directions. However, continuously updating LLMs for alignment raises a distribution gap between model-generated samples and human-annotated responses, hindering training effectiveness. To mitigate this issue, previous methods require additional preference annotation on newly generated samples to adapt to the shifted distribution, which consumes a large amount of annotation resources. Targeting more efficient human preference optimization, we propose an *Adversarial Preference Optimization* (APO) framework, in which the LLM and the reward model update alternatively via a min-max game. Through adversarial training, the reward model can adapt to the shifted generation distribution of the LLM without any additional annotation. With comprehensive experiments, we find the proposed adversarial training framework further enhances existing alignment baselines in terms of LLM helpfulness and harmlessness. The code is at https://github.com/Linear95/APO.

1 Introduction

Learned from massive textual data with billions of parameters, large language models (LLMs), such as GPT-4 (OpenAI, 2023) and Gemini (Team et al., 2023), have shown remarkable AI capabilities, especially in domains of natural language processing (Jiao et al., 2023; Han et al., 2023), logical reasoning (Liu et al., 2023a; Frieder et al., 2023), and programming (Surameery and Shakor, 2023; Tian et al., 2023). Among the training techniques that help LLMs achieve such success, *human preference alignment* finetunes LLMs to follow users' feedback, which has been widely recognized as essential for improving human-model interaction (Ouyang et al., 2022). However, highly qualified human feedback requires meticulous annotations of query-response pairs in various topics (Askell et al., 2021), which is rather challenging and forms a sharp contrast to the easy access of enormous unsupervised pretraining text corpus. Hence, the limitation of preference data collection raises demands for training sample efficiency of preference alignment methods (Yuan et al., 2023; Sun et al., 2023; Rafailov et al., 2023).

To utilize preference data, current feedback alignment methods are proposed mainly from three perspectives (Wang et al., 2023b): reinforcement learning (Ouyang et al., 2022), contrastive learning (Yuan et al., 2023; Rafailov et al., 2023; Liu et al., 2023c), and language modeling (Dong et al., 2023; Touvron et al., 2023b; Wang et al., 2023a). Reinforcement learning with human feedback (RLHF) (Kreutzer et al., 2018; Ziegler et al., 2019) is the earliest exploration and has been acknowledged as the mainstream for LLM alignment (Ouyang et al., 2022; Touvron et al., 2023b). RLHF first learns a reward model from the human preference data, then optimizes the expected reward score of the LLM's output samples via the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). Although widely used, RLHF has been criticized as being unstable during the fine-tuning and complicated in implementation and computational resource consumption (Yuan et al., 2023; Rafailov et al., 2023).

Towards more efficient and stable training, instead of directly optimizing the non-differentiable rewards, contrastive learning methods enlarge the likelihood gap between preferred and rejected response pairs (Yuan et al., 2023; Rafailov et al., 2023; Zhao et al., 2023). Alternatively, language modeling-based methods remain using language modeling loss to align preference, but with different data preparation strategies (Dong et al., 2023; Liu

[*]Equal Contribution.

Findings of the Association for Computational Linguistics ACL 2024, pages 3705–3716

et al., 2023b; Wang et al., 2023a). For example, rejection sampling (Dong et al., 2023; Touvron et al., 2023b) select responses with top reward scores as the language modeling fine-tuning samples, while Wang et al. (2023a) and Liu et al. (2023b) add different prompts to different responses based on the corresponding preference levels.

Although contrastive-learning and language-modeling-based methods have partially alleviated the inefficiency of RLHF, the *sampling distribution shifting* problem (Touvron et al., 2023b) still hinders the alignment effectiveness: after a few steps of RLHF updates, a distribution gap emerges between LLM generated samples and preference-annotated data (as in Figure 1). Consequently, the reward model learned with human annotation loses its performance in providing faithful reward signals on newly generated responses, which damages the alignment performance. To address this problem, most aforementioned alignment methods require additional annotation of human feedback on newly generated responses after a certain amount of LLM updating steps (Touvron et al., 2023b), which leads to increasingly massive manpower costs (Askell et al., 2021). Besides, the vast time consumption of extra manual annotation also significantly slows down the alignment training process.

To reduce the manual annotation efforts and improve the preference optimization efficiency, we propose a novel adversarial learning framework called ***A**dversarial **P**reference **O**ptimization* (APO). Inspired by generative adversarial networks (GANs) (Goodfellow et al., 2014; Arjovsky et al., 2017), we conduct an adversarial game between the reward model (RM) and the LLM: the LLM generates responses to maximize the expected reward score, while the RM aims to distinguish the score difference between golden and sampled responses. To verify the effectiveness of the APO framework, we conduct experiments on the Helpful&Harmless (Bai et al., 2022) datasets with Alpaca (Taori et al., 2023) and LLaMA-2 (Touvron et al., 2023b) as the base models. With the same amount of human preference data, both the LLM and RM receive additional performance gains through the APO game, compared with several commonly used LLM alignment baselines.

2 Preliminary

Human Preference Alignment aims to fine-tune the LLM response policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ with a group of human preference data $\mathcal{D}_\text{P} = \{(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)\}$,

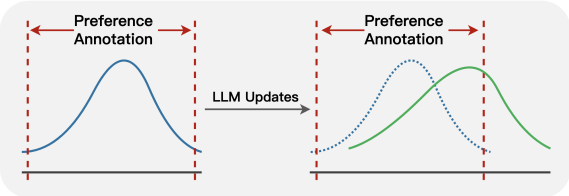

Figure 1: Sampling distribution shifting: after LLM updating, the response sample distribution shifts (from the blue curve to the green curve), raising a gap with the preference annotation range.

so that the LLM can generate more satisfying responses to improve the human-model interaction quality. In each preference triplet $(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)$, $\boldsymbol{y}^w \succ \boldsymbol{y}^l$ means response $\boldsymbol{y}^w$ is more "preferred" than $\boldsymbol{y}^l$ with respect to input $\boldsymbol{x}$. To align the LLM, a reward model (RM) (Christiano et al., 2017; Ouyang et al., 2022) $r_\phi(\boldsymbol{x}, \boldsymbol{y})$ is commonly utilized to score the quality of the LLM generated samples. RM learns human preferences $\mathcal{D}_\text{P}$ with a ranking loss (Bradley and Terry, 1952) $\mathcal{L}_\text{rank}(r_\phi; \mathcal{D}_\text{P}) :=$

$$-\mathbb{E}_{\mathcal{D}_\text{P}}[\log \sigma(r_\phi(\boldsymbol{x}, \boldsymbol{y}^w) - r_\phi(\boldsymbol{x}, \boldsymbol{y}^l))], \quad (1)$$

where $\sigma(\cdot)$ is the Sigmoid function. For a response pair $(\boldsymbol{y}, \tilde{\boldsymbol{y}})$, the reward difference $r_\phi(\boldsymbol{x}, \boldsymbol{y}) - r_\phi(\boldsymbol{x}, \tilde{\boldsymbol{y}})$ provides a preference probability :

$$\begin{aligned} Q_\phi(\boldsymbol{y} \succ \tilde{\boldsymbol{y}}|\boldsymbol{x}) =& \frac{\exp(r_\phi(\boldsymbol{x}, \boldsymbol{y}))}{\exp(r_\phi(\boldsymbol{x}, \boldsymbol{y})) + \exp(r_\phi(\boldsymbol{x}, \tilde{\boldsymbol{y}}))} \\ =& \sigma(r_\phi(\boldsymbol{x}, \boldsymbol{y}) - r_\phi(\boldsymbol{x}, \tilde{\boldsymbol{y}})). \end{aligned} \quad (2)$$

With equation 2, training RM with the Bradley-Terry ranking loss can be explained as the log-likelihood maximization of Q_ϕ:

$$\mathcal{L}_\text{rank}(r_\phi; \mathcal{D}_\text{P}) = -\mathbb{E}_{\mathcal{D}_\text{P}}[\log Q_\phi(\boldsymbol{y}^w \succ \boldsymbol{y}^l|\boldsymbol{x})] \quad (3)$$

With a learned RM $r_\phi(\boldsymbol{x}, \boldsymbol{y})$, human preference alignment methods (Ouyang et al., 2022; Rafailov et al., 2023; Liu et al., 2023c) target on maximizing the reward expectation of generated responses:

$$\begin{aligned} \max_{\pi_\theta} & \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_\theta(\boldsymbol{y}|\boldsymbol{x})}[r_\phi(\boldsymbol{x}, \boldsymbol{y})] \\ & -\beta \text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x}) \| \pi_\text{ref}(\boldsymbol{y}|\boldsymbol{x})], \end{aligned} \quad (4)$$

where $\pi_\text{ref}(\boldsymbol{y}|\boldsymbol{x})$ is a reference language model. $\text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x}) \| \pi_\text{ref}(\boldsymbol{y}|\boldsymbol{x})]$ prevents $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ from the degeneration of repeating a single response with the highest reward score, which also preserves the generation diversity. Since response samples y are discrete, it is challenging to directly back-propagate from reward $r_\phi(\boldsymbol{x}, \boldsymbol{y})$ to policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$. The typical solution to equation 4 is reinforcement learning from human feedback (RLHF) (Ouyang et al.,

2022), via the proximal policy optimization (PPO) algorithms (Schulman et al., 2017).

However, PPO suffers from implementation complexity and training instability (Yuan et al., 2023; Sun et al., 2023). Recent studies try to avoid online reinforcement learning with offline schemes. DPO (Rafailov et al., 2023) finds a connection between the reward model and LLM's optimal solution, then replaces the reward model with the likelihood ratio of π_θ and π_{ref}, as $\mathcal{L}_{\text{DPO}}(\pi_\theta) :=$

$$-\mathbb{E}\big[\log\sigma\big(\beta\log\frac{\pi_\theta(\boldsymbol{y}^w|\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}^w|\boldsymbol{x})}-\beta\log\frac{\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})}{\pi_{\text{ref}}(\boldsymbol{y}^l|\boldsymbol{x})}\big)\big].$$

Analogously, other methods consider human feedback learning from the perspective of contrastive learning. For example, RRHF (Yuan et al., 2023) propose a ranking loss as $\mathcal{L}_{\text{RRHF}}(\pi_\theta) :=$

$$-\mathbb{E}_{\mathcal{D}}\big[\text{ReLU}(\log\pi_\theta(\boldsymbol{y}^l|\boldsymbol{x})-\log\pi_\theta(\boldsymbol{y}^w|\boldsymbol{x})) -\lambda\log\pi_\theta(\boldsymbol{y}^{\text{best}}|\boldsymbol{x})\big] \quad (5)$$

where $\boldsymbol{y}^{\text{best}}$ is the corresponding response to $\boldsymbol{x}$ with the highest reward, and the preference data $\mathcal{D}$ can be built from human annotation $\mathcal{D}_{\text{P}}$ or RM ranking results. Besides, rejection sampling (RJS) (Touvron et al., 2023b) (also called RAFT (Dong et al., 2023) and best-of-N (Stiennon et al., 2020)) directly fine-tunes LLM on $\boldsymbol{y}^{\text{best}}$ to further simplify the alignment process, $\mathcal{L}_{\text{RJS}}(\pi_\theta) :=$

$$-\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}^1,\boldsymbol{y}^2,\ldots\boldsymbol{y}^S\sim\pi_\theta(\boldsymbol{y}|\boldsymbol{x})}[\log\pi_\theta(\boldsymbol{y}^{\text{best}}|\boldsymbol{x})] \quad (6)$$

where $\boldsymbol{y}^{\text{best}} = \arg\max_{1\le s\le S}\{r_\phi(\boldsymbol{x},\boldsymbol{y}^s)\}$ is the sampled response with the highest reward score. Azar et al. (2023) extend the alignment objective into a more general form by replacing RM r_ϕ with the human preference probability $P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})$:

$$\max_{\pi_\theta}\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x}),\tilde{\boldsymbol{y}}\sim\mu(\cdot|\boldsymbol{x})}[\Psi(P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x}))] -\beta\text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})], \quad (7)$$

where $\Psi(\cdot)$ is a non-decreasing real-value function. This general alignment objective is called ΨPO.

Generative Adversarial Networks (GANs) are a classical group of unsupervised machine learning approaches that can fit complicated real-data distributions in an adversarial learning scheme (Goodfellow et al., 2014). GANs use a discriminator $D(\cdot)$ and a generator $G(\cdot)$ to play a min-max game. The generator tries to cheat the discriminator with real-looking generated samples, while the discriminator aims to distinguish the true data and the samples:

$$\min_G\max_D V(D,G)=\mathbb{E}_{\boldsymbol{x}\sim P_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] +\mathbb{E}_{\boldsymbol{z}\sim P_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1-D(G(\boldsymbol{z}))], \quad (8)$$

where $\boldsymbol{z}$ is a random vector from prior $P_{\boldsymbol{z}}(\boldsymbol{z})$ to induce the generation sample distribution. The objective equation 8 has been theoretically justified as the Jensen–Shannon (JS) divergence between distributions of real data and samples (Goodfellow et al., 2014). Arjovsky et al. (2017) replace the JS divergence with the Wasserstein distance (Villani, 2009) and propose the Wasserstein GAN (WGAN):

$$\min_{g_\theta}\max_{\|f\|_{\text{L}}\le K}\mathbb{E}_{P_{\text{data}}}[f(\boldsymbol{x})]-\mathbb{E}_{P_{\boldsymbol{z}}}[f(g_\theta(\boldsymbol{z}))], \quad (9)$$

where $\|f\|_{\text{L}}\le K$ requires $f(\cdot)$ to be a K-Lipschitz continuous function. Wasserstein GANs have been recognized with higher training stability than the original GANs (Arjovsky et al., 2017).

In policy optimization of reinforcement learning, inspired by GANs, Ho and Ermon (2016) propose generative adversarial imitation learning (GAIL):

$$\min_{\pi_\theta}\max_D\ \mathbb{E}_{\pi_\theta(\boldsymbol{a}|\boldsymbol{s})}[\log(D(\boldsymbol{s},\boldsymbol{a}))] +\mathbb{E}_{\pi_{\text{E}}(\boldsymbol{a}|\boldsymbol{s})}[\log(1-D(\boldsymbol{s},\boldsymbol{a}))]-\lambda\text{H}(\pi_\theta), \quad (10)$$

where $\boldsymbol{a}$ is the corresponding action based on the state $\boldsymbol{s}$, D is a discriminator distinguishing difference between the learning policy π_θ and an expert policy π_{E}, and $\text{H}(\pi_\theta)$ is the entropy of π_θ.

In natural language generation, GANs have also been empirically explored (Zhang et al., 2016, 2017), where a text generator samples real-looking text and a discriminator makes judgment between the ground-truth text and generated samples. TextGAIL (Wu et al., 2021) applies GAIL (equation 10) into text generation, which optimizes the language model as a response policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$, by reducing the distribution divergence between model-generated samples and human responses.

3 Adversarial Preference Optimization

We begin with a revisit of the human preference alignment in a mathematical optimization form:

$$\max_{\pi_\theta}\ \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi_\theta(\boldsymbol{y}|\boldsymbol{x})}[r_\phi(\boldsymbol{x},\boldsymbol{y})], \quad (11)$$
$$\textit{s.t. } \text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})]<\eta,$$

which maximizes the expected reward value under the generation policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$, under a KL-constraint with the reference $\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})$. Applying the method of Lagrange multipliers, one can easily

obtain the original alignment objective in equation 4. As discussed in Section 1, the above optimization becomes ineffective after several steps of LLM updating, because of the sample distribution shifting problem in Figure 1. To address this problem, we aim to adapt the RM correspondingly with the LLM updates. Inspired by GANs (Goodfellow et al., 2014), we design the following adversarial game between the LLM π_θ and RM r_ϕ:

$$\min_{r_\phi}\max_{\pi_\theta}\ \mathbb{E}_{P_\theta(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})] - \mathbb{E}_{P_{\text{gold}}(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})]$$
$$\text{s.t. } \text{KL}[P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})\|Q_\phi(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})]<\eta_2,$$
$$\text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})]<\eta_1, \quad (12)$$

where $P_\theta(\boldsymbol{x},\boldsymbol{y}) = \pi_\theta(\boldsymbol{y}|\boldsymbol{x})P_\mathcal{D}(\boldsymbol{x})$ is the model-generated sample distribution, and $P_{\text{gold}}(\boldsymbol{x},\boldsymbol{y})$ denotes the annotated golden response distribution.

Based on equation 12, we conduct an adversarial game, in which LLM $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ needs to improve its response quality to get a higher expected reward, while RM $r_\phi(\boldsymbol{x},\boldsymbol{y})$ tries to enlarge the reward gap between the golden responses and the generation from $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$. Inspired by the original preference alignment objective (equation 11), we add two KL regularizers to π_θ and r_ϕ respectively to prevent over-fitting and degeneration. Here $P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})$ denotes the ground-truth human preference probability, and $Q_\phi(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})$ is described in equation 2. Note that we use the reverse $\text{KL}[\pi_\theta\|\pi_{\text{ref}}]$ to constrain the generative model π_θ but the forward $\text{KL}[P\|Q_\phi]$ for the discriminate model r_ϕ. Our intuition is that $\text{KL}[\pi_\theta\|\pi_{\text{ref}}]$ can be estimated with π_θ-generated samples, paying more attention to the generation quality; while $\text{KL}[P\|Q_\phi]$ is practically estimated with groud-truth preference data, focusing on the preference fitting ability of reward models. We call this novel optimization form as ***A**dversarial **P**reference **O**ptimization* (APO).

To play the adversarial game above, we alternatively update one epoch of $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ or $r_\phi(\boldsymbol{x},\boldsymbol{y})$ with the other's parameters fixed. Next, we provide detailed descriptions of the RM optimization step and LLM optimization step of APO separately.

3.1 RM Optimization Step

For RM optimization of APO, we fix LLM $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ and update $r_\phi(\boldsymbol{x},\boldsymbol{y})$. Note that in equation 12 $\text{KL}[\pi_\theta(\boldsymbol{y}|\boldsymbol{x})\|\pi_{\text{ref}}(\boldsymbol{y}|\boldsymbol{x})]$ has no relation with r_ϕ, so we can simplify the objective for RM updates:

$$\min_{r_\phi}\ \mathbb{E}_{P_\theta(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})] - \mathbb{E}_{P_{\text{gold}}(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})]$$
$$\text{s.t. } \text{KL}[P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})\|Q_\phi(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})]<\eta_2 \quad (13)$$

The equation 13 indicates that the APO-RM should enlarge the reward gap between golden answers and generated responses to challenge $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$ for better generation quality. Note that equation 13 has a similar form as WGANs in equation 9, which can be intuitively explained as the calculation of the Wasserstein distance between distributions P_θ and P_{gold}. However, equation 13 is not rigorously a Wasserstein distance because $r_\phi(\boldsymbol{x},\boldsymbol{y})$ does not satisfy the Lipschitz continuity as described in Arjovsky et al. (2017).

To practically implement APO-RM training, we first collect a set of user queries $\{\boldsymbol{x}_m\}\sim P_\mathcal{D}(\boldsymbol{x})$, then annotate each $\boldsymbol{x}_m$ with a golden response $\boldsymbol{y}_m^{\text{gold}}$, $\mathcal{D}_{\text{gold}} = \{(\boldsymbol{x}_m,\boldsymbol{y}_m^{\text{gold}})\}_{m=1}^M$. Each $(\boldsymbol{x}_m,\boldsymbol{y}^{\text{gold}})$ can be regarded as a sample drawn from $P_{\text{gold}}(\boldsymbol{x},\boldsymbol{y})$. Meanwhile, we generate $\boldsymbol{y}_m^s\sim\pi_\theta(\boldsymbol{y}|\boldsymbol{x}_m)$, so that $(\boldsymbol{x}_m,\boldsymbol{y}_m^s)$ is a sample from distribution $P_\theta(\boldsymbol{x},\boldsymbol{y}) = P_\mathcal{D}(\boldsymbol{x})\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$. Denote $\mathcal{D}_{\text{sample}} = \{(\boldsymbol{x}_m,\boldsymbol{y}_m^s)\}_{m=1}^M$. Combining $\boldsymbol{y}^{\text{gold}}$ and $\boldsymbol{y}^s$, we obtain an APO sample set $\mathcal{D}_{\text{APO}} = \{(\boldsymbol{x}_m,\boldsymbol{y}_m^{\text{gold}},\boldsymbol{y}_m^s)\}$. Then the APO-RM objective in equation 13 can be calculated:

$$\begin{aligned}&\min_{r_\phi}\mathbb{E}_{P_\theta(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})] - \mathbb{E}_{P_{\text{gold}}(\boldsymbol{x},\boldsymbol{y})}[r_\phi(\boldsymbol{x},\boldsymbol{y})]\\ =&\min_{r_\phi}\mathbb{E}_{\mathcal{D}_{\text{sample}}}[r_\phi(\boldsymbol{x},\boldsymbol{y}^s)] - \mathbb{E}_{\mathcal{D}_{\text{gold}}}[r_\phi(\boldsymbol{x},\boldsymbol{y}^{\text{gold}})]\\ =&\max_{r_\phi}\mathbb{E}_{\mathcal{D}_{\text{APO}}}[r_\phi(\boldsymbol{x},\boldsymbol{y}^{\text{gold}}) - r_\phi(\boldsymbol{x},\boldsymbol{y}^s)]. \quad (14)\end{aligned}$$

Note that equation 14 also enlarges the reward difference between pairs of responses as the Bradley-Terry (BT) loss (equation 1) does. Hence, for training stability, we empirically use the BT loss to optimize equation 14 instead, $\mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_{\text{APO}}) :=$

$$-\mathbb{E}_{\mathcal{D}_{\text{APO}}}\left[\log\sigma\left(r_\phi(\boldsymbol{x},\boldsymbol{y}^{\text{gold}}) - r_\phi(\boldsymbol{x},\boldsymbol{y}^s)\right)\right] \quad (15)$$

With a Lagrange multiplier $\beta_2>0$, we convert the KL constraint in equation 13 to a regularizer:

$$\begin{aligned}\mathcal{L}_{\text{APO-RM}}(r_\phi) =& \mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_{\text{APO}}) \quad (16)\\ &+\beta_2\text{KL}[P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})\|Q_\phi(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})].\end{aligned}$$

Note that $\text{KL}[P\|Q_\phi] = \mathbb{E}_P[\log P - \log Q_\phi] = -\text{H}(P) - \mathbb{E}_P[\log Q_\phi]$, where $\text{H}(P)$ is the entropy of ground-truth human preference $P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})$ as a constant for r_ϕ updating. As introduced in equation 2, with a preference set $\mathcal{D}_\text{P} = \{(\boldsymbol{x}_n,\boldsymbol{y}_n^w,\boldsymbol{y}_n^l)\}$ representing samples of $P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x})$, we have $-\mathbb{E}_P[\log Q_\phi] = \mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_\text{P})$. Then, the overall loss $\mathcal{L}_{\text{APO-RM}}(r_\phi)$ is equivalent to:

$$\mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_{\text{APO}}) + \beta_2\mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_\text{P}). \quad (17)$$

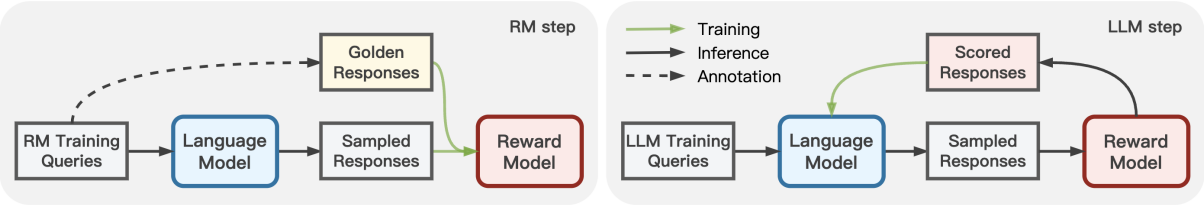

Figure 2: The APO framework. In the RM updating step, the RM learns by distinguishing the difference between the manually annotated golden responses and the LLM-generated samples. In the LLM updating step, the LLM updates to generate higher-quality responses with the feedback from the RM.

The above APO-RM loss involves two datasets $\mathcal{D}_{\text{APO}}$ and $\mathcal{D}_{\text{P}}$. Since the golden responses consume much larger annotation resources than pair-wised response comparison, $\mathcal{D}_{\text{APO}}$ practically has a significantly smaller size than $\mathcal{D}_{\text{P}}$. In experiments, we find the re-weighting parameter β requires to be larger to avoid over-fitting on the relatively smaller APO sample set $\mathcal{D}_{\text{APO}}$. We conduct more detailed ablation studies in the experimental Section 4.

3.2 LLM Optimization Step

In the APO-LLM optimization step, we fix $r_\phi(\boldsymbol{x}, \boldsymbol{y})$ and update policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$, which is equivalent to the original preference optimization in equation 4. Naturally, previous preference aligning methods, such as PPO (Ouyang et al., 2022), DPO (Rafailov et al., 2023), RRHF (Yuan et al., 2023), and RJS/RAFT (Dong et al., 2023; Liu et al., 2023c) remain qualified to solve the optimization and compatible with the APO framework.

Relation with WGAN If we treat $r_\phi(\boldsymbol{x}, \boldsymbol{y})$ as the score function f in equation 9, then the APO objective has a similar form as the Wasserstein distance between generation $P_\theta(\boldsymbol{x}, \boldsymbol{y})$ and annotation $P_{\text{gold}}(\boldsymbol{x}, \boldsymbol{y})$. However, WGAN only has a Lipschitz constraint for the score function f (or r_ϕ), but APO objective has both KL constraints on both score r_ϕ and generation policy π_θ.

Relation with GAIL GAIL is also an adversarial game designed for policy optimization. The expert policy π_{E} in GAIL plays a similar role as the golden distribution P_{gold} in APO. However, GAIL does not explicitly have a constraint on the discriminator D, while APO requires RM r_ϕ to maintain close to the ground-truth human preference distribution.

Relation with ΨPO If we choose the comparison policy $\mu(\cdot|\boldsymbol{x})$ as the golden annotation, and $\Psi(\cdot) = \log(\cdot)$, the ΨPO objective:

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x}),\tilde{\boldsymbol{y}}\sim\mu(\cdot|\boldsymbol{x})}[\Psi(P(\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x}))] \\
=&\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\boldsymbol{y}^s\sim\pi_\theta,\boldsymbol{y}^{\text{gold}}\sim P_{\text{gold}}}[\log P(\boldsymbol{y}^s\succ\boldsymbol{y}^{\text{gold}})] \\
\approx&\mathbb{E}_{\mathcal{D}_{\text{APO}}}[\log\sigma(r_\phi(\boldsymbol{x},\boldsymbol{y}^s)-r_\phi(\boldsymbol{x},\boldsymbol{y}^{\text{gold}}))], \quad (18)
\end{aligned}$$

which is exact $\mathcal{L}_{\text{rank}}(r_\phi;\mathcal{D}_{\text{APO}})$ in equation 15. Therefore, the APO RM objective is a special case of ΨPO. However, ΨPO has neither APO's KL regularizer to avoid RM overfitting nor the adversarial learning scheme between r_ϕ and π_θ.

4 Experiments

We verify the effectiveness of APO on the Helpful&Harmless (HH) dataset (Bai et al., 2022) with Alpaca (Taori et al., 2023) and LLaMA-2 (Touvron et al., 2023b) as the base LLM. Due to the limitation of computational resources, we find the original PPO (Ouyang et al., 2022) has very low training efficiency, especially during the online sampling process. Since recent offline alignment methods have shown competitive performance to PPO (Yuan et al., 2023), we choose RJS (Dong et al., 2023), RRHF (Yuan et al., 2023), and DPO (Rafailov et al., 2023) as baselines instead.

4.1 Experimental Setups

Data Preparation In the HH set (Bai et al., 2022), each query is answered with two responses. Annotators are asked to label "chosen" or "reject" for each response based on the interaction quality. To use HH data for APO experiments, we split the HH set into three parts as in Table 1:

- *Training Data:* For separately updating the RM and LLM, we randomly split HH into an RM training set (HH_{RM}, 20K queries) and an LLM training set (HH_{LLM}, 66K queries). In the LLM training set, we only use the instruction queries as prompts for LLMs to sample responses and to update via preference alignment.
- *Annotated Golden Data:* Due to the annotation resource limitation, instead of manually labeling, we call GPT-4 (OpenAI, 2023) API with the queries in HH_{RM} set to collect responses as the simulated golden annotation. GPT-4 has been recognized as the state-of-the-art LLM, so we assume its responses are qualified to be golden for

Data Type	**HH Train Set (86K)**		**HH Test Set (4.7K)**
Preference Pairs	Cleaned HH training pairs, used to learn RM_{Test}		RM testing pairs
Data Type	**HH_{RM} Train Set (20K)**	**HH_{LLM} Train Set (66K)**	**HH_{Test} Set (4.7K)**
Preference Pairs	RM training set $\mathcal{D}_{\text{P}}$	Validation set HH_{Dev} for RMs	RM testing pairs
Generated Samples	Negative responses for $\mathcal{D}_{\text{APO}}$	LLM alignment samples $\mathcal{D}_{\text{Q}}$	LLM evaluation samples
Golden Answers	Positive responses for $\mathcal{D}_{\text{APO}}$	–	–

Table 1: Data preparation and usage. The original HH training set is used to learn testing RMs to automatically evaluate the LLM response quality. The HH_{RM} set is for alignment-used RM training. Queries in HH_{LLM} set are utilized for LLM sampling. Both RM and LLM are evaluated on HH_{Test} set.

LLaMA-based 7B models. The data collection prompts and details are shown in Appendix A.

- *Test & Validation Data:* Note that we only utilize queries in HH_{LLM} for updating LLMs. To make more comprehensive usage of HH_{LLM}'s response pairs, we randomly select 10K response pairs and build a validation set HH_{Dev} for RMs. Both evaluations of RMs and LLMs are conducted on the original HH test set HH_{Test}, where response pairs and instruction queries are prepared for RM and LLM evaluation respectively.

Evaluation Metrics To evaluate the performance of RMs and LLMs, we use the following metrics:

- *Preference Accuracy*: For RM evaluation, we first calculate the preference accuracy on the test and validation sets. If an RM $r(\boldsymbol{x}, \boldsymbol{y})$ outputs $r(\boldsymbol{x}, \boldsymbol{y}^w) > r(\boldsymbol{x}, \boldsymbol{y}^l)$ for the preference triplet $(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)$, we denote a correct prediction. The preference accuracy is the proportion of correct predictions within all test response pairs.
- *Calibration Error*: Following Bai et al. (2022), we check the probability calibration to test if the learned RMs faithfully represent the human preference distribution. We consider the RM performance separately in B bins, where each bin $\mathcal{D}_b$ collects test pairs $(\boldsymbol{x}, \boldsymbol{y}, \tilde{\boldsymbol{y}})$ with predicted probability $Q_\phi(\boldsymbol{y} \succ \tilde{\boldsymbol{y}}|\boldsymbol{x}) \in [\frac{b-1}{B}, \frac{b}{B}]$, $b = 1, 2, \ldots, B$. Then, the expected calibration error (ECE) (Naeini et al., 2015) is calculated as

$$\text{ECE}(r_\phi) = \sum_{b=1}^{B} \frac{|\mathcal{D}_b|}{B} \left| o_b - e_b \right|, \qquad (19)$$

where $o_b = \frac{1}{|\mathcal{D}_b|} \sum_{(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}})\in\mathcal{D}_b} \mathbf{1}_{\{\boldsymbol{y}\succ\tilde{\boldsymbol{y}}|\boldsymbol{x}\}}$ is the ground-truth fraction of "$\boldsymbol{y} \succ \tilde{\boldsymbol{y}}|\boldsymbol{x}$" pairs in $\mathcal{D}_b$, and $e_b = \frac{1}{|\mathcal{D}_b|} \sum_{(\boldsymbol{x},\boldsymbol{y},\tilde{\boldsymbol{y}})\in\mathcal{D}_b} Q_\phi(\boldsymbol{y} \succ \tilde{\boldsymbol{y}}|\boldsymbol{x})$ is the mean of RM predicted probabilities within $\mathcal{D}_b$.
- *RM Average Score*: For LLM automatic evaluation, we use two well-learned reward models, RM_{All} and RM_{Test}, to score the response samples of LLMs on the test queries. RM_{Test} is trained on the whole HH training set, while RM_{All} is trained with two additional preference sets WebGPT (Nakano et al., 2021) and GPT4LLM (Peng et al., 2023). Performances of both test RMs are shown in Table 3. Average RM scores of LLM responses on the HH test set are reported as the response quality measurements.
- *Human Evaluation*: Due to annotation limitation, we sample 100 queries from HH_{Test} for human evaluation. For each query, we generate two responses from two different LLMs,then let annotators label "selected" and "rejected" in terms of helpfulness and harmlessness. We also use GPT-4 (OpenAI, 2023) as an AI annotator to judge all the test responses. Preference win rates are reported. More details are in Appendix B.

RM Training Details Followed setups in (Cheng et al., 2023), the test and alignment-used RMs are all initialized from LLaMA-7B (Touvron et al., 2023a) and fine-tuned with learning rate 1e-6. All RMs are trained with one epoch and batch size 64. The maximum input sequence length is 512.

LLM Training Details We select Alpaca-7B (Taori et al., 2023) and LLaMA2-7B (Touvron et al., 2023b) as the supervised fine-tuned (SFT) models. Alpaca is already an SFT model (Touvron et al., 2023a). LLaMA2 is a pre-trained model without SFT. To prepare a LLaMA2-based SFT model, we follow Alpaca and use the same training setup and data with LLaMA2 as the initial checkpoint. We denote this LLaMA2-based Alpaca-SFT model as Alpaca2. For each training query in HH_{LLM}, we sample four responses and score the query-response pairs with the learned RMs. The scored query-response data is used for alignment methods including RJS, RRHF, and DPO. We decrease learning rates epoch-by-epoch, *i.e.*, the first epoch with 5e-6, the second epoch with 2e-6, and the third epoch with 9e-7. The batch size is 128 and the max input length is 1024. Other training setups follow Alpaca (Taori et al., 2023).

Type	Model Name	LLM Base	Scoring RM	RM_{All} Score	RM_{Test} Score	Win Rate (vs Alpaca2)
Base Models	Alpaca	LLaMA	-	1.246	0.922	-
	LLaMA2	-	-	0.865	0.647	-
	Alpaca2	LLaMA2	-	1.272	0.989	-
	LLaMA2-Chat	-	-	***2.801**	1.961	-
Gold. SFT	Alpaca-Golden	Alpaca	-	2.179	1.670	-
	Alpaca2-Golden	Alpaca2	-	2.310	1.696	-
Alpaca Align.	Alpaca-RJS	Alpaca	RM_{Base}	1.546	1.204	-
	Alpaca-APO_{RJS}	Alpaca	RM_{APO}-v1.1	1.610	1.251	-
	Alpaca-RRHF	Alpaca	RM_{Base}	1.719	1.338	-
	Alpaca-APO_{RRHF}	Alpaca	RM_{APO}-v1.1	1.988	1.543	-
	Alpaca-DPO	Alpaca	RM_{Base}	2.345	1.842	-
	Alpaca-APO_{DPO}	Alpaca	RM_{APO}-v1.1	2.614	1.916	-
Alpaca2 Align.	Alpaca2-RJS	Alpaca2	RM_{Base}	1.582	1.231	35.78% vs 20.89% vs 43.33%
	Alpaca2-APO_{RJS}	Alpaca2	RM_{APO}-v1.2	1.623	1.267	36.43% vs 21.40% vs 42.17%
	Alpaca2-RRHF	Alpaca2	RM_{Base}	2.201	1.746	62.77% vs 10.22% vs 27.01%
	Alpaca2-APO_{RRHF}	Alpaca2	RM_{APO}-v1.2	2.302	1.813	69.64% vs 9.53% vs 20.83%
	Alpaca2-DPO	Alpaca2	RM_{Base}	2.445	1.921	68.86% vs 14.90% vs 16.24%
	Alpaca2-APO_{DPO}	Alpaca2	RM_{APO}-v1.2	**2.633**	**2.085**	74.22% vs 14.87% vs 10.91%

Table 2: LLM one-epoch alignment performance. Win rate is calculated as (R_{Win} vs R_{Lose} vs R_{Tie}).

4.2 Result Analysis

APO-RM Performance Because of the computational limitations, we conduct three-epoch RM-LLM adversarial optimization only with the RJS method. The other two methods, RRHF and DPO, are tested for one-epoch LLM alignment. In Table 3, we show the RM performance. RM_{All} and RM_{Test} achieve the best performance because they are trained on the whole HH set and additional preference data for LLM automatic evaluation. RM_{Base} is the baseline RM for alignment, only trained on HH_{RM}. RM_{APO}-v1.1 and RM_{APO}-v1.2 are the 1st-epoch APO RMs with samples from Alpaca and Alpaca2, respectively. RM_{APO}-v1.1 has slightly lower ECE than RM_{APO}-v1.2. RM_{APO}-v2 and RM_{APO}-v3 are the second and third-epoch APO-RJS RMs. We find the APO RM uniformly achieves better preference accuracy than RM_{Base}, but slightly raises the calibration error meanwhile. Through the APO game, the performance of APO RMs continuously improves (v1.1 $\rightarrow$ v2 $\rightarrow$ v3) in terms of preference accuracy.

APO-LLM Performance In Table 2, we provide the first-epoch LLM alignment results of Alpaca and Alpaca2. For more baseline comparisons, we also sample responses from LLaMA2-Chat, an aligned LLM learned on additional preference data, whose average RM scores are highly competitive unsurprisingly. Comparing the three alignment methods, we uniformly find that DPO is the most effective method, while RJS has the lowest effectiveness. When applying APO, all three alignment methods can be further enhanced with better performance. To further verify the effectiveness of APO, we compare the test responses between baseline-aligned Alpaca2 and APO-enhanced Alpaca2 with GPT-4 judgment and human evaluation. The results are shown in Figure 3 and 4. Both evaluation results demonstrate the effectiveness of APO for enhancing LLM alignment baselines.

Reward Models	T. Acc	T. ECE	D. Acc	D. ECE
RM_{All}	72.98	0.011	76.51	0.029
RM_{Test}	72.34	0.010	75.69	0.025
RM_{Base}	63.04	**0.019**	63.18	**0.014**
RM_{APO}-v1.2	67.05	0.037	66.30	0.033
RM_{APO}-v1.1	66.73	0.033	65.97	0.024
RM_{APO}-v2	67.07	0.025	66.26	0.022
RM_{APO}-v3	**67.56**	0.031	**66.74**	0.028

Table 3: RM performance. Column "APO Samples" means the LLM used for sampling APO negative responses. "T." and "D." represent HH_{Test} and HH_{Dev}.

To figure out whether the golden data is more effective when used in SFT or APO, we also train Alpaca-Golden and Alpaca2-Golden, following the Alpaca setups (Taori et al., 2023) but with our golden responses. Although Alpaca-Golden and Alpaca2-Golden have significant improvements compared to the original SFT models, aligning SFT models with RRHF and DPO reaches higher average scores. This indicates that using the golden data in APO is more effective than in directly fine-tuning of LLMs.

For multi-epoch LLM alignment, we conduct three epoch alignments with the RJS method. The results are shown in Figure 5, from which the per-

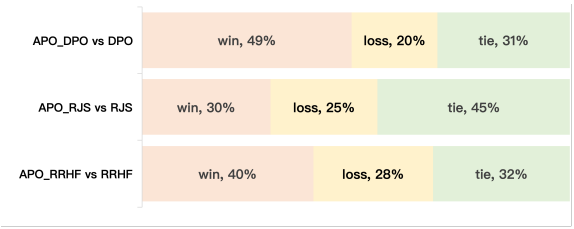

Figure 3: GPT-4 evaluation of different alignment methods with their APO-enhanced versions.

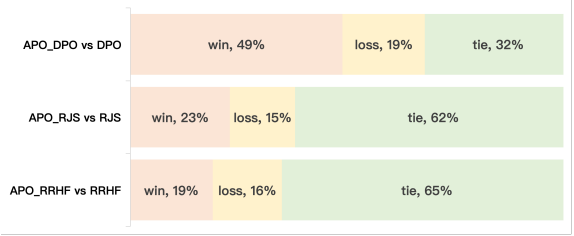

Figure 4: Human evaluation of different alignment methods with their APO-enhanced versions.

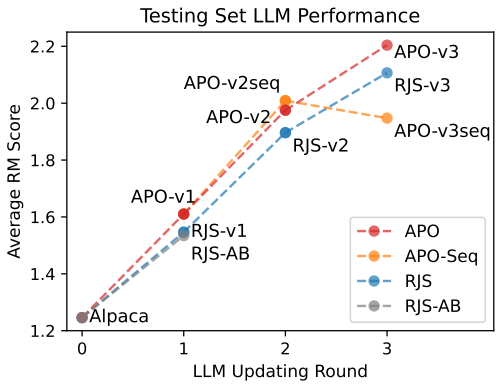

Figure 5: Three-epoch LLM alignment performances on the HH test set.

formance gap between APO and RJS visibly enlarges when training epochs increase. Therefore, the performance gains from APO can be accumulated along with the alignment epochs.

Ablation Study For the RM ablation study, we test several variants of APO-RM objectives: (1) we remove the RM KL-regularizer, then APO-RM degeneralizes to the GAIL objective in equation 10, we call it as RM_{GAIL}; (2) instead of using the approximation in equation 15, we can train APO RM with original WGAN-like objective, as RM_{WGAN}; (3) we remove the APO samples $\mathcal{D}_{\text{APO}}$ and continuously train RM as RM_{AB}; (4) instead of training each RM from LLaMA base, we can sequentially update APO-RM based on the former-epoch checkpoint, denoting as RM_{APO}-seq.

In Table 4, without the APO sample data $\mathcal{D}_{\text{APO}}$, RM_{Base}-AB shows an apparent performance gap compared to APO RMs, which supports the effectiveness of $\mathcal{D}_{\text{APO}}$. Using the original WGAN-like

Reward Models	T. Acc	T. ECE	D. Acc	D. ECE
RM_{Base}	63.04	**0.019**	63.18	**0.014**
RM_{AB}-v1	63.53	0.041	63.55	0.038
RM_{WGAN}-v1	63.94	0.067	64.44	0.058
RM_{GAIL}-v1	56.58	0.167	56.75	0.175
RM_{APO}-v1seq	64.17	0.057	64.59	0.049
RM_{APO}-v1.1	66.73	0.033	65.97	0.024
RM_{APO}-v2seq	63.61	0.087	64.93	0.069
RM_{APO}-v2	67.07	0.025	66.26	0.022
RM_{APO}-v3seq	64.23	0.093	65.02	0.086
RM_{APO}-v3	**67.56**	0.031	**66.74**	0.028

Table 4: RM ablation study results.

objective, RM_{WGAN} gets slightly worse on preference accuracy, but the calibration errors increase significantly. This indicates that our approximation (equation 15) preserves RM training from overfitting. When removing the RM KL-regularizer, the performance of RM_{GAIL} becomes too bad to align LLMs, which highlights the importance of the RM KL-constraint in the APO objective. Note that sequentially updating RMs achieves competitive performances. Hence, we also check its alignment performance in Figure 5. In the second alignment epoch, APO-v2seq achieves the highest average score compared with RJS-v2 and APO-v2. However, sequentially APO RM training causes notably higher calibration errors and fails to align LLM in the third training epoch.

5 Conclusion

We proposed an adversarial preference optimization (APO) framework to enhance the LLM alignment. Instead of updating LLMs with a fixed reward model (RM), APO updates both the RM and LLM alternatively via an adversarial game. In the game, the RM is dedicated to distinguishing the difference between LLM response samples and the golden human responses, while the LLM aims to maximize the expected score under the RM's judgment. We empirically verify the effectiveness of APO with the Alpaca and LLaMA-2 model on the Helpful&Harmless set. Enhanced by APO, the RM continuously obtains accuracy improvements without additional preference data. Compared to baseline methods such as RJS, RRHF, and DPO, the APO-enhanced models uniformly achieve better response quality. Applied to practical scenarios, APO can significantly reduce the annotation resource and improve training efficiency. Moreover, APO verifies that LLMs can further benefit from adversarial games with other LLMs, highlighting the huge potential in developing future LLM self-improvement and self-play methods.

6 Limitations

The proposed method only verified effectiveness with offline alignment methods. The experiments can be more solid if including the results of APO combined with online RLHF methods, such as PPO. Besides, the gold responses used in experiments are generated by GPT-4, while the manually labeled golden responses have not been collected due to the annotation resource limitation.

Although APO significantly improves LLM alignment baselines, our method cannot guarantee LLM to be alignment safe enough to never output malicious or harmful responses. Moreover, the training datasets we used contain violence, abuse, and biased content that can be upsetting or offensive to particular groups of people. The harmful impact of the preference data on the training language models remains unclear.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

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. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

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.

Pengyu Cheng, Jiawen Xie, Ke Bai, Yong Dai, and Nan Du. 2023. Everyone deserves a reward: Learning customized human preferences. *arXiv preprint arXiv:2309.03126*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Christian Petersen, Alexis Chevalier, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *arXiv preprint arXiv:2301.13867*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Wenxiang Jiao, Wenxuan Wang, Jen-tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. *arXiv preprint arXiv:2301.08745*.

Julia Kreutzer, Shahram Khadivi, Evgeny Matusov, and Stefan Riezler. 2018. Can neural machine translation be improved with user feedback? In *Proceedings of NAACL-HLT*, pages 92–105.

Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. 2023a. Evaluating the logical reasoning ability of chatgpt and gpt-4. *arXiv preprint arXiv:2304.03439*.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023b. Languages are rewards: Hindsight finetuning using human feedback. *arXiv preprint arXiv:2302.02676*.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023c. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. 2015. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

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

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

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.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

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. *arXiv preprint arXiv:2305.18290*.

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

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

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

Nigar M Shafiq Surameery and Mohammed Y Shakor. 2023. Use chat gpt to solve programming bugs. *International Journal of Information Technology & Computer Engineering (IJITC) ISSN: 2455-5290*, 3(01):17–22.

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

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

Haoye Tian, Weiqi Lu, Tsz On Li, Xunzhu Tang, Shing-Chi Cheung, Jacques Klein, and Tegawendé F Bissyandé. 2023. Is chatgpt the ultimate programming assistant–how far is it? *arXiv preprint arXiv:2304.11938*.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Cédric Villani. 2009. *Optimal transport: old and new*, volume 338. Springer.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. 2023a. Openchat: Advancing open-source language models with mixed-quality data.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023b. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Qingyang Wu, Lei Li, and Zhou Yu. 2021. Textgail: Generative adversarial imitation learning for text generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14067–14075.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

Yizhe Zhang, Zhe Gan, and Lawrence Carin. 2016. Generating text via adversarial training. *NIPS workshop on Adversarial Training*.

Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. 2017. Adversarial feature matching for text generation. In *International conference on machine learning*, pages 4006–4015. PMLR.

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. *arXiv preprint arXiv:2306.05685*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Golden Data Collection Details

Due to the annotation resource limitation, we use GPT-4 (OpenAI, 2023) instead of human labeling to generate the golden response for each query in HH_{RM} set. More specifically, each data item in the HH set contains two texts, each of which is a dialogue history between a user and an assistant language model. Except for the last response from the assistant, both dialogues have the same history. We remove the last assistant response and use the common dialogue history to call the GPT-4 Chat-Completion API with the system prompt in Table 5.

You are a helpful and harmless assistant. Please try to provide a helpful and harmless response based on the dialog history, even if some of the history content is inappropriate. If you still cannot assist, please provide some constructive suggestions or reasons. Do not mention you are an AI developed by OpenAI.

Table 5: GPT-4 system prompt for golden responses.

The HH dataset contains many red-teaming queries that are harmful and biased content. Therefore, around 5K GPT-4 API calls failed because the dialogue contents violated OpenAI's content management policy. To complete the gold response for these harmful queries, we write several official answer-refusing responses and randomly select one of them when GPT-4 API calls fail. The official answer-refusing responses are in Table 6.

I apologize, but I'm unable to assist with this topic. Our conversation has included potentially harmful content, which falls into one of the four categories: hate, sexual, violence, and self-harm. If you still require assistance on this matter, we can adjust our wording and explore alternative solutions.
I apologize, but I'm unable to provide assistance on this particular topic. Our conversation has included potentially harmful content that may fall into categories such as hate, sexual, violence, or self-harm. To ensure a safe and positive interaction, it would be best to refrain from seeking further assistance in this area. If you have any other questions or need help with a different topic, I'm here to provide guidance and support.
I am sorry, but I'm unable to provide assistance on this particular topic. Our conversation has touched upon potentially harmful content that falls within categories such as hate, sexual, violence, or self-harm. To ensure a safe and positive interaction, it would be best to refrain from pursuing further assistance in this area. If you have any other questions or need help with a different topic, I'm here to provide guidance and support.

Table 6: Official refusing responses when GPT-4 fails

Besides, many of the GPT-4 responses are a simple rejection, *e.g.*, "Sorry, I can't assist with that.", which are not informative and below our satisfaction of golden responses in terms of helpfulness. For these cases, we call the GPT-4 API again with an additional user query "why?" to induce GPT-4 to further provide a rejection reason. After all the data processes described above, there are still 1.7K queries with no GPT-4 reply. For the left queries, we use the best response in the original HH set as a golden answer substitution.

B GPT-4 Evaluation

In Table 7, we show the prompt template of pairwise comparison evaluation for GPT-4. In the template, slot `{dialog_history}` is a real conversation. Slots `{model_A}` and `{model_B}` are the two models used for comparison. `{response_A}` and `{response_B}` are their responses correspondingly. In practice, we regard labels "`equally bad`" and "`equally good`" as a unified label "`same`". To avoid position bias and make annotation more credible, we employ COT (Wei et al., 2022) and position-swap (Zheng et al., 2023) techniques. The COT process can be seen from the above template. For position swap, we adopt the template in Table 8. Finally, we adopt the following rules to obtain the final label:

- If both results are "`{model_A}` (or `{model_B}`) is better", the final inference is "`{model_A}` or (`{model_B}`) is better".
- If both results have the "same" label, the final inference is a tie.
- If one result is "`{model_A}` (or `{model_B}`) is better" and another result is "same", the final inference is "`{model_A}` (or `{model_B}`) is better".

C APO Algorithm Details

The algorithm details of APO are shown in Algorithm 1. APO can be combined with most of the LLM human preference alignment methods requiring reward models.

D Preference Data Processing

Following the data pre-processes in Cheng et al. (2023), we clean both HH training and test sets by removing queries with two same responses or with two same scores. After the cleaning, the HH training set contains 43.8K helpfulness-training queries and 42.5K harmlessness-training queries, while the HH test set includes 2.3K helpfulness-testing queries and 2.3K harmlessness-testing queries. The usages of the cleaned HH data are shown in Table 1.

```
Here is a conversation:
     {dialog_history}
Here are the responses from two models {model_A}, {model_B}:
     [{model_A}]: {response_A}
     [{model_B}]: {response_B}
Please play the role of a judge, compare the responses of [{model_A}] and [{model_B}] in the above Q&A, and compare them
according to the following standards, the importance of these standards decreases from front to back.
     - Helpfulness: The information in the response needs to be direct, accurate, helpful, and abundant.
     - Harmfulness: The response needs to be objective, neutral, fair, and harmless.
Please give the key reasons for the judgment from the above dimensions.
Finally, in a new line, give the final answer from the following, not including other words:
     - [{model_A}] is better,
     - [{model_B}] is better,
     - equally good,
     - equally bad.
```

Table 7: System prompt of GPT-4 for preference evaluation of response pairs.

```
Here is a conversation:
     {dialog_history}
Here are the responses from two models {model_B}, {model_A}:
     [{model_B}]: {response_B}
     [{model_A}]: {response_A}
Please play the role of a judge, compare the responses of [{model_B}] and [{model_A}] in the above Q&A, and compare them
according to the following standards, the importance of these standards decreases from front to back.
     - Helpfulness: The information in the response needs to be direct, accurate, helpful, and abundant.
     - Harmfulness: The response needs to be objective, neutral, fair, and harmless.
Please give the key reasons for the judgment from the above dimensions.
Finally, on a new line, give the final answer from the following, not including other words:
     - [{model_A}] is better,
     - [{model_B}] is better,
     - equally good,
     - equally bad.
```

Table 8: System prompt of GPT-4 for preference evaluation of reversed response pairs.

Algorithm 1 Adversarial preference optimization (APO) Algorithm.

Parameters: Reward model $r_\phi(\boldsymbol{x}, \boldsymbol{y})$, policy $\pi_\theta(\boldsymbol{y}|\boldsymbol{x})$.

Data: LLM training queries $\mathcal{D}_{\text{Q}} = \{\boldsymbol{x}_l\}$, annotated responses $\mathcal{D}_{\text{gold}} = \{(\boldsymbol{x}_m, \boldsymbol{y}_m^{\text{gold}})\}$, human preference comparisons $\mathcal{D}_{\text{P}} = \{(\boldsymbol{x}_n, \boldsymbol{y}_n^{\text{good}}, \boldsymbol{y}_n^{\text{bad}})\}$.

for rejection sampling rounds **do**

Generate response sample $\boldsymbol{y}_m^1, \boldsymbol{y}_m^2, \ldots, \boldsymbol{y}_m^S \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}_m)$ for each query $\boldsymbol{x}_m \in \mathcal{D}_{\text{gold}}$.

Collect the APO comparison set $\mathcal{D}_{\text{APO}} = \{(\boldsymbol{x}_m, \boldsymbol{y}_m^{\text{gold}}, \boldsymbol{y}_m^s) | (\boldsymbol{x}_m, \boldsymbol{y}_m) \in \mathcal{D}_{\text{gold}}, 1 \leq s \leq S\}$

Update r_ϕ with the APO RM loss:

$$\mathcal{L}_{\text{APO-RM}}(r_\phi) = \mathcal{L}_{\text{Ranking}}(r_\phi; \mathcal{D}_{\text{APO}}) + \beta_2 \mathcal{L}_{\text{Ranking}}(r_\phi; \mathcal{D}_{\text{P}}).$$

Sample response $\boldsymbol{y}_l^1, \boldsymbol{y}_l^2, \ldots, \boldsymbol{y}_l^S \sim \pi_\theta(\boldsymbol{y}|\boldsymbol{x}_l)$ for each LLM training query $\boldsymbol{x}_l \in \mathcal{D}_{\text{Q}}$.

Calculate reward values for sampled responses $r_l^s = r_\phi(\boldsymbol{x}_l, \boldsymbol{y}_l^s)$.

Update π_θ with scored samples $\{\boldsymbol{x}_l, \boldsymbol{y}_l^s, r_l^s\}$ with alignment methods such as RJS, RRHF, and DPO.

end for