

---

# AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

---

Junkang Wu<sup>1</sup> Xue Wang<sup>2</sup> Zhengyi Yang<sup>1</sup> Jiancan Wu<sup>1</sup> Jinyang Gao<sup>2</sup> Bolin Ding<sup>2</sup> Xiang Wang<sup>1\*</sup>  
Xiangnan He<sup>1\*</sup>

## Abstract

Aligning large language models (LLMs) with human preferences requires balancing policy optimization with computational stability. While recent offline methods like DPO and SimPO bypass reinforcement learning’s complexity, they face critical limitations: DPO relies on static reference models that degrade with policy updates, and SimPO assumes a uniform target reward margin that ignores instance-wise preference strength. We propose AlphaDPO, an adaptive preference optimization framework that dynamically reparameterizes the reference distribution to address these issues. Our key innovation lies in an implicit reference model  $\hat{\pi}_{\text{ref}} \propto U(y|x)(\pi_{\theta}/\pi_{\text{ref}})^{\alpha}$ , which interpolates between policy-driven specialization and uniform exploration while enabling instance-adaptive reward margins. Theoretically, we prove AlphaDPO implicitly controls sequential KL divergence between iterative policy updates, ensuring stability even with poorly calibrated reference models. Empirically, AlphaDPO achieves state-of-the-art performance on AlpacaEval 2 (58.7% LC win rate) and Arena-Hard (35.7% win rate) across Mistral2-7B, Llama3-8B, and Gemma2-9B, demonstrating robust alignment without multi-stage training. Our work establishes adaptive reference reparameterization as a principled mechanism for preference optimization. The code is available at <https://github.com/junkangwu/alpha-DPO>.

## 1. Introduction

Learning from human feedback is essential for aligning large language models (LLMs) with human values and inten-

tions (Leike et al., 2018), ensuring they are helpful, honest, and harmless (Askell et al., 2021). Reinforcement learning from human feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022; Stiennon et al., 2020) is a widely used method for fine-tuning LLMs to achieve this goal. However, RLHF faces challenges, particularly in computational efficiency and training stability due to its multi-stage process. Recently, alternative offline algorithms like DPO (Rafailov et al., 2023) and SimPO (Meng et al., 2024) have been explored. Specifically, DPO reparameterizes the reward function in RLHF to directly learn a policy model ( $\pi_{\theta}$ ) from preference data, removing the need for an explicit reward model. Building on DPO, SimPO removes the reference model requirement but introduces a target reward margin  $\gamma$  to enhance the separation between response pairs, achieving leading performance. This naturally raises the question:

*Do we really need a reference model in the alignment process?*

This question prompts a deeper analysis of SimPO’s underlying mechanism: it can be viewed as a variant of DPO where the original reference model  $\pi_{\text{ref}}$  is replaced by an *implicit reference model*  $\hat{\pi}_{\text{ref}}$ . In SimPO, the target reward margin  $\gamma$  actually reflects a constant difference between the log likelihoods of a selected response and a rejected one, i.e.,  $(\log \hat{\pi}_{\text{ref}}(y_w|x) - \log \hat{\pi}_{\text{ref}}(y_l|x))$ . As the constant difference  $\gamma$  is independent of arbitrary responses, this implicitly assumes a uniform reference distribution (cf. Figure 1). By tuning  $\gamma$ , SimPO effectively finds an “ideal” uniform *implicit reference model*, yielding substantial performance improvements over standard DPO, particularly when the original reference model  $\pi_{\text{ref}}$  is suboptimal (Hong et al., 2024). While conceptually appealing with empirical improvements, SimPO has two inherent limitations: (1) Applying the same target reward margin to all pairwise comparisons ignores the variability in the data (Yang et al., 2024; Wu et al., 2024), potentially compromising decision quality in some cases; and (2) The implicit assumption of a uniform reference model somehow lacks a solid theoretical foundation. These limitations could hinder the model’s ability to achieve alignment across varied training data, especially in domains with diverse preferences or complex reward structures (Morimura et al., 2024; He et al., 2024).

<sup>\*</sup>Equal contribution <sup>1</sup>MoE Key Lab of BIPC, University of Science and Technology of China <sup>2</sup>Alibaba Group. Correspondence to: Xiang Wang <xiangwang1223@gmail.com>, Xiangnan He <hexn@ustc.edu.cn>.

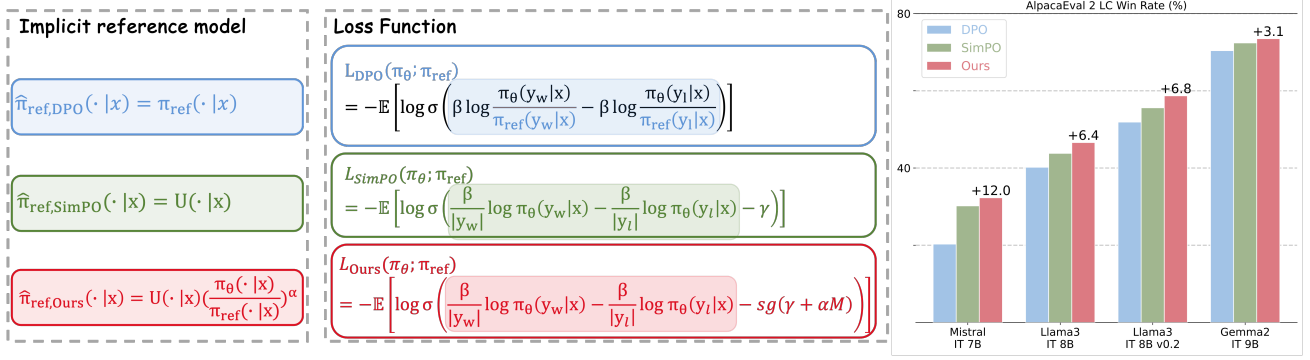


Figure 1. DPO, SimPO and AlphaDPO mainly differ in their *implicit reference model*, as indicated in the shaded box, leading to variations in their respective loss functions. AlphaDPO outperforms DPO and SimPO across a wide range of settings on AlpacaEval 2.

In response to these challenges, we propose an adaptive preference distribution, which gives rise to an adaptive reward margin for different response pairs. We term this simple yet effective preference optimization algorithm AlphaDPO. Specifically, the adaptive preference distribution is heuristically set as:  $\hat{\pi}_{\text{ref}}(y|x) = U(y|x) (\pi_{\theta}(y|x)/\pi_{\text{ref}}(y|x))^{\alpha}$ . Here,  $U(y|x)$ , inspired by SimPO, employs a uniform distribution to establish an initial target reward margin, while the term  $(\pi_{\theta}(y|x)/\pi_{\text{ref}}(y|x))^{\alpha}$  adjusts the balance between the policy model  $\pi_{\theta}$  and the reference model  $\pi_{\text{ref}}$  to achieve a personalized reward margin. When  $\alpha = 0$ , AlphaDPO reduces to SimPO; as  $\alpha$  increases, the ratio between  $\pi_{\theta}$  and  $\pi_{\text{ref}}$  becomes dominant, enabling a personalized, dynamic target. More important, AlphaDPO offers several intriguing theoretical insights: We demonstrate that AlphaDPO balances alignment and diversity via KL divergence control. By approximating the sequential KL divergence between the policy and the reference model, AlphaDPO achieves computational efficiency and robustness, particularly when the reference model is not well-calibrated at the token level.

Extensive analysis indicates that AlphaDPO leverages preference data more effectively by assigning personalized margins to each pair, resulting in an improved policy model. As demonstrated in Figure 1, our method consistently outperforms DPO and SimPO across three base model settings (Mistral2-7B, Llama3-8B, and Gemma2-9B) on AlpacaEval 2 and Arena-Hard (*cf.* Section 5). Notably, we achieve a 58.7 length-controlled win rate on AlpacaEval 2, and a 35.7 win rate on Arena-Hard, establishing it as the strongest 8B open-source model to date.

## 2. Preliminaries

**Offline Alignment.** In the offline alignment problem, we have access to a dataset  $\mathcal{D} = \{(x, y_w, y_l)\}$  comprising prompts  $x$  and labeled response pairs  $(y_w, y_l)$  obtained from a reference policy  $\pi_{\text{ref}}$ . Here,  $y_w$  is the preferred (winning) response and  $y_l$  is the less preferred (losing) response. Al-

though the underlying latent reward function  $r^*(x, y)$  that governs these preferences is not directly observable, the Bradley-Terry (BT) model (Bradley & Terry, 1952) provides a framework for modeling pairwise comparisons:

$$\mathbb{P}(y_w \succ y_l|x) = \frac{\exp(r^*(x, y_w))}{\exp(r^*(x, y_w)) + \exp(r^*(x, y_l))}, \quad (1)$$

where  $r^*(x, y)$  assigns a latent reward to each response  $y$  given prompt  $x$ . The goal of offline alignment is to learn a policy  $\pi_{\theta}$  that approximates  $r^*(x, y)$  using  $\mathcal{D}$ .

**Reinforcement Learning from Human Feedback (RLHF).** Classical offline alignment algorithms employ reinforcement learning with a KL-regularized reward objective (Bai et al., 2022; Ziegler et al., 2019; Ouyang et al., 2022), defined for a regularization parameter  $\eta > 0$ :

$$\max_{\pi_{\theta}} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_{\theta}(y|x)} [r_{\phi}(x, y)] - \beta \mathbb{D}_{\text{KL}}[\pi_{\theta}(y|x) || \pi_{\text{ref}}(y|x)], \quad (2)$$

where  $r_{\phi}(x, y)$  is the reward function learned using the BT model on the preference dataset,  $\pi_{\theta}$  is the policy model being optimized,  $\pi_{\text{ref}}$  is the fixed reference policy, typically obtained via supervised fine-tuning. The KL-divergence regularizes the policy to remain close to the reference model.

**Directed Preference Optimization (DPO).** DPO (Rafailov et al., 2023) is a leading offline preference optimization method. Instead of learning an explicit reward model, DPO reparameterizes the reward function  $r(x, y)$  using a closed-form expression involving the optimal policy:

$$r(x, y) = \beta \log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x), \quad (3)$$

where  $Z(x)$  is the partition function independent of  $y$ . This leads to the DPO loss for any triplet  $(x, y_w, y_l)$ :

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\theta}(y_l|x)} - \beta \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right]. \quad (4)$$

where  $\sigma(\cdot)$  denotes the sigmoid function.

**Simple Preference Optimization (SimPO).** SimPO (Meng et al., 2024) introduces two key contributions: (1) a length-normalized reward, calculated as the average log-probability per token of a response under the policy model  $\pi_\theta$ , and (2) a target reward margin  $\gamma$  to ensure the reward difference between winning and losing responses exceeds this margin. The SimPO loss is formulated as:

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right], \quad (5)$$

where  $|y|$  denotes the length (*i.e.*, number of tokens) of response  $y$ , normalizing the reward by response lengths, and  $\gamma$  is the target reward margin.

### 3. Method

In this section, we establish a unified framework that connects DPO and SimPO (Section 3.1), highlighting the critical role of the reference model in preference optimization. We then introduce AlphaDPO (Section 3.2), a new preference optimization algorithm that synergizes the strengths of both DPO and SimPO.

#### 3.1. A Common Framework for DPO and SimPO

A key insight in our work is that SimPO implicitly adopts a uniform distribution over the vocabulary as its reference model, whereas DPO employs the SFT model as the reference. By examining the role of the reference model in both methods, we derive the following result:

**Theorem 3.1.** *Let  $U(y|x)$  denote a uniform distribution over the vocabulary for a given input  $x$ , replacing  $\pi_{\text{ref}}(y|x)$  in the DPO loss function. Then, the DPO loss function simplifies to:*

$$\mathcal{L}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma (\beta (\log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x)) - \gamma)], \quad (6)$$

where  $\gamma = \beta (\log U(y_w|x) - \log U(y_l|x))$  is a constant. Under a length-normalized reward formulation, this loss function becomes:

$$\mathcal{L}_{LN}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right]. \quad (7)$$

Therefore, SimPO can be interpreted as a special case of DPO where the reference model is a uniform distribution.

**Remark 3.2. Why do winning and losing responses have different probabilities under a uniform policy?** Although both winning and losing responses are sampled from the

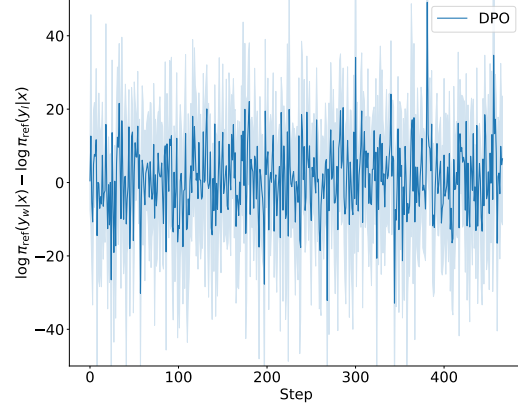


Figure 2.  $\log \pi_{\text{ref}}(y_w|x) - \log \pi_{\text{ref}}(y_l|x)$  along the training steps. The random fluctuations suggest  $\pi_{\text{ref}}$  struggles to distinguish between the preferred and less preferred responses.

same policy model, their selection probabilities diverge through the reward model’s scoring mechanism. Specifically, winning response ( $y_w$ ) are those assigned higher scores, while losing response ( $y_l$ ) receive lower scores. Even under a uniform policy, these selections remain distinct because their selection probabilities are tied to their relative scores, not to the underlying uniform distribution over the response space.

Theorem 3.1 establishes a unified framework for DPO and SimPO by showing that replacing the reference model  $\pi_{\text{ref}}$  in DPO with a uniform distribution  $U$  reduces the DPO loss to the SimPO loss, up to a constant term  $\gamma$ . This reveals that SimPO is essentially DPO with a uniform reference model. Consequently, the term  $\beta (\log \pi_{\text{ref}}(y_w|x) - \log \pi_{\text{ref}}(y_l|x))$  collapses to a constant, reflecting the difference in selection probabilities induced by the scoring mechanism. This emphasizes the pivotal role of the implicit reference model in preference optimization.

**Limitations of DPO:** As depicted in Figure 2, the reference model  $\pi_{\text{ref}}$  in DPO may not effectively distinguish between the preferred ( $y_w$ ) and less preferred ( $y_l$ ) responses, as its outputs do not inherently reflect the preference information. In contrast, using a uniform distribution as in SimPO results in a reward margin  $\gamma$ , ensuring that the reward difference between the preferred and less preferred responses is entirely governed by the policy model  $\pi_\theta$ .

**Limitations of SimPO:** While SimPO simplifies the loss function by using a constant offset  $\gamma$ , this one-size-fits-all approach overlooks the variability inherent in different data instances. The fixed  $\gamma$  across all training samples could lead to suboptimal performance, especially in the presence of noise or inconsistencies in the data. Moreover, completely discarding reference model  $\pi_{\text{ref}}$  may eliminate potentially useful prior knowledge about language structure and semantic relationships that could help discriminate between similar response pairs.

### 3.2. Proposed Method: AlphaDPO

Our analysis highlights the significant impact of the reference model in preference optimization. To overcome the limitations identified in both DPO and SimPO, we propose the following principles:

**Principle 1:** *The reference model should contribute to differentiating between preferred and less preferred responses.*

**Principle 2:** *The reference model should adapt to discrepancies between response pairs to capture instance-specific nuances.*

Principle 1 addresses the shortcoming in DPO, where the reference model may inadequately distinguish between  $y_w$  and  $y_l$ , introducing uncertainty without a guaranteed margin. Principle 2 rectifies the oversimplification in SimPO, where the absence of a reference model fails to account for variability across different instances.

**Deriving the AlphaDPO Objective.** Starting from the standard Reinforcement Learning (RL) objective for preference optimization, we redefine the reference model  $\pi_{\text{ref}}$  as an *implicit reference model*  $\hat{\pi}_{\text{ref}}$ , formulated as:

$$\hat{\pi}_{\text{ref}}(y|x) \propto U(y|x) \left( \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)} \right)^{\alpha}, \quad (8)$$

where  $\alpha$  is a hyperparameter that determines the extent to which the policy model  $\pi_{\theta}$  modifies the reference model  $\pi_{\text{ref}}$ , and  $U(y|x)$  represents a uniform distribution providing a stable baseline. This formulation shares structural similarities with the weak-to-strong preference optimization framework (Zhu et al., 2025), where the aligned strong model  $\pi_r^{\text{strong}}$  is expressed as:

$$\pi_r^{\text{strong}}(y|x) \propto \pi_{\text{ref}}^{\text{strong}}(y|x) \left( \frac{\pi_r^{\text{weak}}(y|x)}{\pi_{\text{ref}}^{\text{weak}}(y|x)} \right)^{\alpha}. \quad (9)$$

Both formulations employ model ratios to dynamically adjust the reference distribution —  $\frac{\pi_{\theta}}{\pi_{\text{ref}}}$  in Equation 8 and  $\frac{\pi_r^{\text{weak}}}{\pi_{\text{ref}}^{\text{weak}}}$  in Equation 9. Specifically,  $\hat{\pi}_{\text{ref}}$  interpolates between existing methods, controlled by the smoothness parameter  $\alpha$ : i) When  $\alpha = 0$ ,  $\hat{\pi}_{\text{ref}}$  reduces to the uniform distribution  $U(y|x)$ , equivalent to the assumption in SimPO. ii) When  $\alpha = 1$ ,  $\hat{\pi}_{\text{ref}}$  fully incorporates the proportionality  $\frac{\pi_{\theta}}{\pi_{\text{ref}}}$ , aligning with DPO.

The motivation behind this redefinition is to overcome the potential suboptimality of both SimPO and DPO by learning instance-specific adjustments while preserving useful prior knowledge. Rather than relying on either a fixed reference model or a constant margin,  $\hat{\pi}_{\text{ref}}$  creates a flexible framework that can better capture varying degrees of preference strength across different instances. Further details are provided in Appendix C.

Substituting  $\hat{\pi}_{\text{ref}}$  into the original DPO loss function, we

obtain the AlphaDPO objective:

$$\begin{aligned} \mathcal{L}_{\text{AlphaDPO}}(\pi_{\theta}, \pi_{\text{ref}}) &= -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\theta}(y_l|x)} - \beta \log \frac{\hat{\pi}_{\text{ref}}(y_w|x)}{\hat{\pi}_{\text{ref}}(y_l|x)} \right) \right] \\ &= -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \left( \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\theta}(y_l|x)} \right) - [\gamma + \alpha M(x, y_w, y_l)] \right) \right], \end{aligned} \quad (10)$$

where  $\gamma = \beta \left( \log \frac{U(y_w|x)}{U(y_l|x)} \right)$  is a constant offset as before, and  $M(x, y_w, y_l)$  is defined as:

$$M(x, y_w, y_l) = \beta \left( \log \frac{\pi_{\theta}(y_w|x) \pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_w|x) \pi_{\theta}(y_l|x)} \right). \quad (11)$$

The term  $M(x, y_w, y_l)$  measures the divergence between the policy model  $\pi_{\theta}$  and the reference model  $\pi_{\text{ref}}$  over the response pairs, effectively capturing instance-specific discrepancies as described in Principle 2.

**Stop Gradient on  $\hat{\pi}_{\text{ref}}$ :** Although  $\hat{\pi}_{\text{ref}}$  depends on  $\pi_{\theta}$  and  $\pi_{\text{ref}}$ , it is intended to serve as a fixed reference during optimization. To prevent gradients from backpropagating through  $\hat{\pi}_{\text{ref}}$  to  $\pi_{\theta}$ , we apply a stop-gradient operation, denoted as  $\text{sg}[\cdot]$ , ensuring that  $\hat{\pi}_{\text{ref}}$  remains constant during the policy updates.

**Normalization of  $M(x, y_w, y_l)$ :** To stabilize training and avoid  $M(x, y_w, y_l)$  dominating the loss due to scale variations, we apply Z-score normalization (Patro & Sahu, 2015) to  $M$ :

$$M^*(x, y_w, y_l) = \frac{M(x, y_w, y_l) - \mu_M}{\sigma_M}, \quad (12)$$

where  $\mu_M$  and  $\sigma_M$  are the mean and standard deviation of  $M$  computed over the training dataset.

**Length-normalized Reward Formulation:** Inspired by SimPO, we incorporate length normalization into our method. This adjustment ensures that rewards are scaled appropriately with respect to the length of the sequences, thereby stabilizing the training process. As demonstrated in our experiments (cf. Appendix D.2), we also confirmed that even without length normalization, our method remains effective and continues to show performance improvements.

**Final Objective:** Incorporating the above considerations, the final AlphaDPO loss function becomes:

$$\begin{aligned} \mathcal{L}_{\text{AlphaDPO}}(\pi_{\theta}, \pi_{\text{ref}}) &= -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( u(x, y_w, y_l) - \text{sg}[\gamma + \alpha M^*(x, y_w, y_l)] \right) \right], \end{aligned} \quad (13)$$

where  $u(x, y_w, y_l) = \frac{\beta}{|y_w|} \log \pi_{\theta}(y_w|x) - \frac{\beta}{|y_l|} \log \pi_{\theta}(y_l|x)$ . This formulation ensures balanced influence between the



policy and reference models, aligning with Principles 1 and 2. By incorporating the normalized discrepancy term  $M^*(x, y_w, y_l)$ , AlphaDPO adaptively adjusts the margin between preferred and less preferred responses based on instance-specific differences, enhancing learning.

#### 4. Theoretical Analysis of AlphaDPO

Balancing alignment performance with response diversity is crucial in recent alignment methods (Zeng et al., 2024; Wang et al., 2024a; Ji et al., 2024a). A popular approach is the Token-Level Direct Preference Optimization (TDPO) method (Zeng et al., 2024), which introduces fine-grained control of the KL divergence at the token level. Given a prompt  $x$  and preceding tokens  $y^{<t}$ , the policy  $\pi_\theta$  generates the next token  $z$  by sampling from  $\pi_\theta(z|x, y^{<t})$ .

By mapping the reward model to a token-level format, the TDPO loss is defined as:

$$\mathcal{L}_{\text{TDPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \delta(x, y_w, y_l) \right) \right], \quad (14)$$

where the margin term  $\delta(x, y_w, y_l)$  is defined as:

$$\begin{aligned} \delta(x, y_w, y_l) &= \beta \mathbb{D}_{\text{SeqKL}}[x, y_l; \pi_{\text{ref}} || \pi_\theta] - \beta \mathbb{D}_{\text{SeqKL}}[x, y_w; \pi_{\text{ref}} || \pi_\theta], \\ &= \beta \sum_{t=1}^{|y_l|} \mathbb{E}_z \left[ \log \frac{\pi_{\text{ref}}(z|[x, y_l^{<t}])}{\pi_\theta(z|[x, y_l^{<t}])} \right] - \beta \sum_{t=1}^{|y_w|} \mathbb{E}_z \left[ \log \frac{\pi_{\text{ref}}(z|[x, y_w^{<t}])}{\pi_\theta(z|[x, y_w^{<t}])} \right]. \end{aligned} \quad (15)$$

Here,  $z \sim \pi_{\text{ref}}$  and  $\mathbb{D}_{\text{SeqKL}}[x, y; \pi_{\text{ref}} || \pi_\theta]$  denotes the sequential KL divergence between  $\pi_{\text{ref}}$  and  $\pi_\theta$  along the sequence  $y$  given  $x$ .

Below we present a lemma establishing theoretical connections between TDPO and AlphaDPO w.r.t. the margin terms.

**Lemma 4.1** (Equivalence of Margin Terms). *Consider the margin term  $\delta(x, y_w, y_l)$  that represents the difference in sequential KL divergences between a reference policy  $\pi_{\text{ref}}$  and policy  $\pi_\theta$  for preferred sequence  $y_w$  and rejected sequence  $y_l$ , conditioned on input  $x$ :*

$$\begin{aligned} \delta(x, y_w, y_l) &= \beta \mathbb{D}_{\text{SeqKL}}[x, y_l; \pi_{\text{ref}} || \pi_\theta] \\ &\quad - \beta \mathbb{D}_{\text{SeqKL}}[x, y_w; \pi_{\text{ref}} || \pi_\theta], \end{aligned} \quad (17)$$

*Under the assumption that the sequential KL divergence can be approximated by the log-likelihood ratio of complete sequences, the margin term  $\delta(x, y_w, y_l)$  admits the following approximation:*

$$\begin{aligned} \delta(x, y_w, y_l) &\approx \beta \left( \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \\ &= M(x, y_w, y_l), \end{aligned}$$

where  $M(x, y_w, y_l)$  is the margin term in the AlphaDPO objective.

The proof follows from the sequence-level approximation of the KL divergence between  $\pi_{\text{ref}}$  and  $\pi_\theta$  along a sequence  $y$ , where:

$$\begin{aligned} \mathbb{D}_{\text{SeqKL}}[x, y; \pi_{\text{ref}} || \pi_\theta] &= \sum_{t=1}^{|y|} \mathbb{E}_{z \sim \pi_{\text{ref}}} \left[ \log \frac{\pi_{\text{ref}}(z|x, y^{<t})}{\pi_\theta(z|x, y^{<t})} \right] \\ &\approx \log \frac{\pi_{\text{ref}}(y|x)}{\pi_\theta(y|x)}. \end{aligned}$$

Applying this approximation to both  $y_w$  and  $y_l$ , the difference  $\delta(x, y_w, y_l)$  simplifies to the difference of log-probability ratios, thereby establishing the equivalence with the margin term in AlphaDPO.

Lemma 4.1 highlights that the margin term  $\delta(x, y_w, y_l)$ , which represents the sequential KL divergence difference between preferred and rejected responses, can be directly mapped to the term  $M(x, y_w, y_l)$  in AlphaDPO. This mapping underscores the theoretical connection between the two approaches in terms of alignment control. While TDPO operates at the token level and provides fine-grained control, AlphaDPO offers greater computational efficiency by operating at the sequence level without sacrificing performance (cf. Appendix Table 6). Moreover, the sequence-level approximation enhances robustness to token-level noise in  $\pi_{\text{ref}}$ , making AlphaDPO particularly suited for scenarios where the reference policy may not be perfectly aligned.

**Advantages of the margin term  $\delta(x, y_w, y_l)$ .** The core contribution of TDPO lies in introducing the margin term  $r(x, y_w) - r(x, y_l) - \delta(x, y_w, y_l)$ , which is similar to *DPO with an offset* (Amini et al., 2024) and helps control the KL divergence. In contrast, AlphaDPO generalizes this approach by replacing  $\delta(x, y_w, y_l)$  with  $M(x, y_w, y_l)$ , a more flexible margin term inspired by SimPO. Appendix Table 6 supports this with performance comparisons between TDPO and AlphaDPO. These findings illustrate: i) Adding an offset to DPO and its variants is a robust strategy, applicable to both TDPO and AlphaDPO. ii) The choice of offset is critical — the sequence-level margin term  $M(x, y_w, y_l)$  is particularly effective when applied to potentially unreliable reference models.

#### 5. Experiments

In this section, we present the main results of our experiments, highlighting the superior performance of AlphaDPO over existing methods on various benchmarks and ablation studies to analyze the impact of different components of AlphaDPO.

Table 1. AlpacaEval 2, Arena-Hard results across four settings. “WR” denotes the raw win rate, “LC” the length-controlled win rate, and “SC” the style-controlled win rate. The best results are highlighted in bold, while the second-best are underlined.

Method	Llama3-Instruct (8B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
SFT	24.0	23.6	22.1	22.2	22.4	19.0	15.4	18.3	13.2	12.9
DPO	40.2	<b>38.1</b>	31.9	32.0	31.2	20.3	17.9	18.9	13.7	13.4
IPO	35.9	34.4	29.2	29.9	30.2	22.3	18.6	22.4	16.6	16.2
CPO	29.6	34.4	26.3	28.1	29.4	26.2	31.7	<u>26.6</u>	<u>21.4</u>	<b>23.8</b>
KTO	38.3	34.1	30.3	30.6	30.3	19.4	20.3	21.5	16.0	16.8
ORPO	31.6	29.8	26.6	26.6	26.3	24.0	23.0	24.4	18.5	18.6
R-DPO	40.3	37.3	33.1	32.9	<u>32.9</u>	21.4	22.2	18.7	14.0	13.8
SimPO	<u>43.8</u>	<u>38.0</u>	<u>33.5</u>	<u>33.5</u>	32.6	<u>30.2</u>	<u>32.1</u>	25.6	19.8	20.1
AlphaDPO	<b>46.6</b>	<b>38.1</b>	<b>34.1</b>	<b>34.2</b>	<b>33.3</b>	<b>32.3</b>	<b>32.6</b>	<b>27.2</b>	<b>21.5</b>	<u>21.5</u>

Method	Llama3-Instruct v0.2 (8B)					Gemma2-Instruct (9B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
SFT	24.0	23.6	22.1	22.2	22.4	48.7	36.5	32.0	42.2	42.1
DPO	51.9	<u>50.8</u>	26.1	31.5	33.9	70.4	<b>66.9</b>	43.9	55.6	<u>58.8</u>
IPO	40.6	39.6	<b>31.1</b>	34.2	34.9	62.6	58.4	41.1	51.9	53.5
CPO	36.5	40.8	29.4	32.8	34.2	56.4	53.4	42.4	53.3	55.2
KTO	41.4	36.4	27.1	29.5	28.9	61.7	55.5	41.7	52.3	53.8
ORPO	36.5	33.1	28.8	30.8	30.4	56.2	46.7	35.1	45.3	46.2
R-DPO	51.6	50.7	29.2	34.3	<u>35.0</u>	68.3	<b>66.9</b>	45.1	55.9	57.9
SimPO	<u>55.6</u>	49.6	28.5	34.0	33.6	<u>72.4</u>	65.0	<u>45.0</u>	<u>56.1</u>	57.8
AlphaDPO	<b>58.7</b>	<b>51.1</b>	<u>30.8</u>	<b>36.3</b>	<b>35.7</b>	<b>73.4</b>	<u>66.1</u>	<b>48.6</b>	<b>59.3</b>	<b>60.8</b>

## 5.1. Experiments Setup

**Models and training settings.** We optimize preferences using three model families: Llama3-8B (AI@Meta, 2024), Mistral2-7B (Jiang et al., 2023), and Gemma2-9B (Rivière et al., 2024), all in the Instruct setup. Following Meng et al. (2024), we utilize pre-trained instruction-tuned models (meta-llama/Meta-Llama-3-8B-Instruct, mistralai/Mistral-7B-Instruct-v0.2, google/gemma-2-9b-it) as SFT models. For a fair comparison, we use the same training data as SimPO: princeton-nlp/llama3-ultrafeedback-armorm<sup>1</sup>, princeton-nlp/mistral-instruct-ultrafeedback<sup>2</sup>, and princeton-nlp/gemma2-ultrafeedback-armorm<sup>3</sup> for Llama3-8B, Mistral2-7B, and Gemma2-9B, respectively. Additionally, the v0.2 Llama3-Instruct setup uses RLHFlow/ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024b)

<sup>1</sup><https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback-armorm>

<sup>2</sup><https://huggingface.co/datasets/princeton-nlp/mistral-instruct-ultrafeedback>

<sup>3</sup><https://huggingface.co/datasets/princeton-nlp/gemma2-ultrafeedback-armorm>

as the reward model for ranking generated data, significantly enhancing performance. These configurations represent state-of-the-art methods, positioning our models among the top performers on various leaderboards.

**Evaluation benchmarks.** We evaluate our models using two widely recognized open-ended instruction-following benchmarks: AlpacaEval 2 (Li et al., 2023) and Arena-Hard (Li et al., 2024). These benchmarks assess the models’ conversational abilities across a diverse range of queries and are extensively used by the research community. For AlpacaEval 2, we report the length-controlled win rate (LC) and raw win rate (WR). For Arena-Hard, we provide the win rate (WR), length-controlled win rate (LC), and style-controlled win rate (SC) compared to baseline models. Note that style significantly impacts performance on these leaderboards.

**Baselines.** We compare AlphaDPO with several state-of-the-art preference optimization methods: DPO (Rafailov et al., 2023), SimPO (Meng et al., 2024), IPO (Azar et al., 2024), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), and R-DPO (Park et al., 2024).

Table 2. Ablation studies under Llama3-Instruct v0.2 and Mistral-Instruct settings. We ablate each key design of AlphaDPO and explore variants of the implicit reference model  $\hat{\pi}_{\text{ref}}$ .

Method	Llama3-Instruct v0.2 (8B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
$U(\cdot x)$	55.6	49.6	28.5	34.0	33.6	30.2	32.1	25.6	19.8	20.1
$U(\cdot x) (\pi_{\theta}(\cdot x)/\pi_{\text{ref}}(\cdot x))^{\alpha}$	58.7	51.1	30.8	36.3	35.7	32.3	32.6	27.2	21.5	21.5
w/o Normalization	56.5	49.7	23.1	28.4	27.7	32.1	33.1	25.2	19.7	19.6
w/o sg	2.7	3.7	7.7	5.4	6.3	27.2	27.7	25.8	20.3	20.7
$\gamma = 0$	51.2	44.9	30.0	34.5	33.3	31.9	31.3	24.2	19.6	19.3
$U(\cdot x) (\pi_{\theta}(\cdot x))^{\alpha}$	57.2	50.4	27.6	33.5	32.9	31.6	34.1	26.9	21.3	21.5
$U(\cdot x) (\pi_{\text{ref}}(\cdot x))^{\alpha}$	56.3	49.5	29.0	34.3	33.5	28.6	30.9	25.5	20.1	20.3
$U(\cdot x) (1/\pi_{\text{ref}}(\cdot x))^{\alpha}$	56.3	49.2	29.0	34.4	33.8	32.2	33.1	26.0	20.7	20.6

We also include the SFT model as a baseline. We thoroughly tune the hyperparameters for each baseline and report the best performance. Further details can be found in Appendix D.1.

## 5.2. Main Results

**AlphaDPO consistently outperforms existing preference optimization methods.** As shown in Table 1, while all preference optimization algorithms improve over the SFT baseline, AlphaDPO achieves superior performance compared to existing methods specifically on the AlpacaEval 2 LC metric. These significant improvements highlight the robustness and effectiveness of AlphaDPO. Specifically, AlphaDPO outperforms the best baseline by an average of 3 percentage points in AlpacaEval 2 LC win rate. Furthermore, on benchmarks such as Arena-Hard, AlphaDPO achieves state-of-the-art or second-best results, demonstrating its competitiveness across different evaluation settings.

**Impact of Metrics on Leaderboard Rankings.** While both benchmarks are widely used, the standard win rate (WR) metric shows poor separability among different methods, making it challenging to distinguish their relative performance. Minor differences in WR may stem from biases towards generating detailed or aesthetically pleasing responses, aligning with observations by Dubois et al. (2024) and Chen et al. (2024a). In contrast, the length-controlled (LC) and style-controlled (SC) win rates offer more reliable and interpretable metrics, as they reduce the influence of verbosity and stylistic biases, thereby better reflecting true performance.

**The importance of the design on the implicit reference model.** As the core contribution of this work is to propose a novel reference model  $\hat{\pi}_{\text{ref}}(y|x) = U(y|x) (\pi_{\theta}(y|x)/\pi_{\text{ref}}(y|x))^{\alpha}$ , we also evaluate other variants of the reference model. Specifically, we compare AlphaDPO with three variants: (1)  $\hat{\pi}_{\text{ref}}(y|x) =$

$U(y|x) (\pi_{\theta}(y|x))^{\alpha}$ , (2)  $\hat{\pi}_{\text{ref}}(y|x) = U(y|x) (\pi_{\text{ref}}(y|x))^{\alpha}$ , and (3)  $\hat{\pi}_{\text{ref}}(y|x) = U(y|x) (1/\pi_{\text{ref}}(y|x))^{\alpha}$ . As shown in Table 2, most of the variants perform better than SimPO ( $\hat{\pi}_{\text{ref}}(y|x) = U(y|x)$ ), which demonstrates the importance of adaptive margin between pairs. Besides, our proposed reference model consistently outperforms other variants, indicating the effectiveness of the proposed design.

**All key designs in AlphaDPO are crucial.** To further analyze the impact of different components in AlphaDPO, we conduct ablation studies by removing key components from AlphaDPO. As shown in Table 2, removing normalization, stop gradient, or setting  $\gamma = 0$  all lead to significant performance drops, highlighting the importance of these components in AlphaDPO.

## 5.3. KL divergence control in AlphaDPO

**Outstanding Performance and Lower KL.** As noted in Rafailov et al. (2023); Zeng et al. (2024), it is crucial to consider both performance and KL divergence when comparing algorithms. A slightly higher win rate accompanied by a significantly higher KL divergence is often not desirable. In line with the design principles of TDPO, we implemented SimPO and AlphaDPO. Figure 3a 3b presents the KL divergence curves. The results indicate that as  $\alpha$  increases, the KL divergence of AlphaDPO remains stable or even decreases slightly when compared to SimPO. This demonstrates that AlphaDPO not only achieves superior performance but also maintains a lower KL divergence, indicating a better balance between alignment and control of KL divergence during the training process.

**Mitigating Over-Optimization.** Over-optimization, as described by Gao et al. (2023) and Rafailov et al. (2024), refers to a phenomenon where model performance exhibits a hump-shaped pattern across different targets: beyond an optimal point, further increasing the KL budget results in diminishing returns. To investigate this, we evaluate

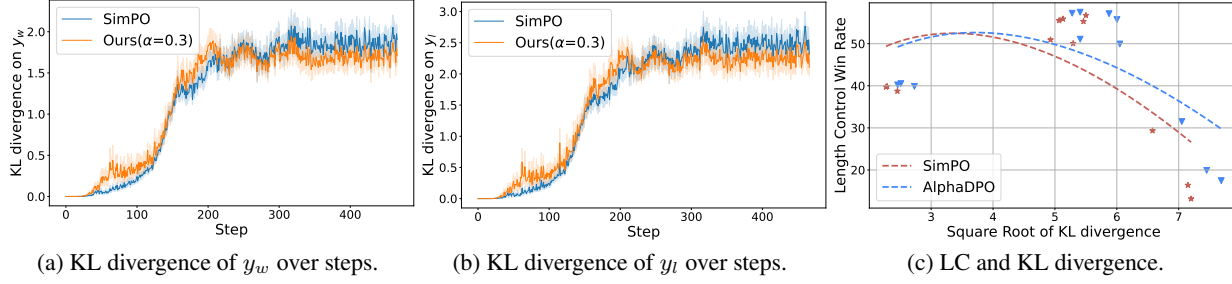


Figure 3. Analysis of KL divergence and LC trade-off. (a) KL divergence for chosen samples ( $y_w$ ), (b) KL divergence for rejected samples ( $y_l$ ), and (c) relationship between LC and KL divergence.

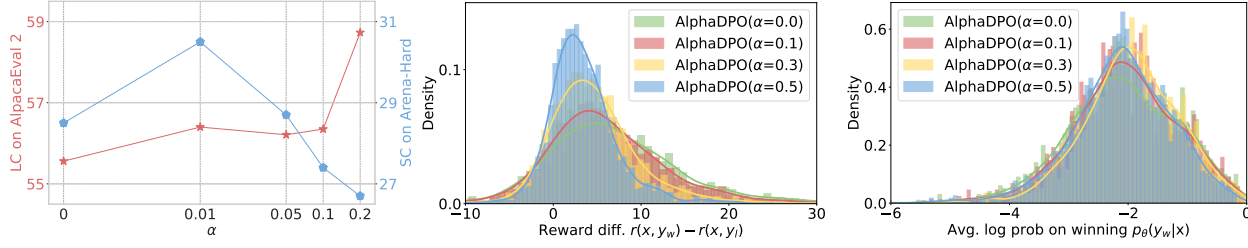


Figure 4. Impact of  $\alpha$  on (a) LC and SC win rate, (b) reward difference distribution, and (c) log-likelihood distribution of chosen responses in AlphaDPO.

SimPO and AlphaDPO at four intermediate checkpoints, corresponding to different KL budgets. As illustrated in Figure 3c, it is intriguing that while the performance of our approach does decrease with increasing KL budget, the decline is relatively modest. This indicates that our method effectively mitigates the issue of over-optimization.

#### 5.4. The Impact of $\alpha$ in AlphaDPO

**Effect of  $\alpha$  on Performance.** We investigated how the parameter  $\alpha$  in AlphaDPO impacts the win rate on AlpacaEval 2 and Arena-Hard. The results, as shown in Figure 4 (a), indicate that the style-control win rate on Arena-Hard initially increases and then decreases with increasing  $\alpha$ . In contrast, the length-control win rate on AlpacaEval 2 exhibits a consistently increasing trend. This suggests that the optimal value of  $\alpha$  varies depending on the evaluation benchmarks. Further experiments refer to Appendix D.3.

**Impact of  $\alpha$  on the reward distribution.** We visualize the distribution of the learned reward margin  $r(x, y_w) - r(x, y_l)$  and the log likelihood of the chosen response  $\log \pi_\theta(y_w|x)$  under different  $\alpha$  values in Figure 4 (b,c). Decreasing  $\alpha$  results in a flatter reward margin, while the log likelihood distribution remains relatively unchanged. Conversely, in SimPO (cf. Figure 6), increasing  $\gamma$  yields a flatter reward margin distribution but at the cost of also flattening the log likelihood distribution, which undesirably lowers the log likelihood of positive samples. This indicates that AlphaDPO can better balance the relationship between the

reward margin and log likelihood.

## 6. Discussion

**Conclusion.** We proposed AlphaDPO, an adaptive preference optimization method that improves LLM alignment by introducing a dynamic reward margin based on instance-specific differences. AlphaDPO addresses limitations in previous methods like DPO and SimPO by balancing alignment and diversity through KL divergence control. Our theoretical guarantees and empirical results show that AlphaDPO consistently outperforms baselines on benchmarks like AlpacaEval 2 and Arena-Hard, with significant improvements in win rates, establishing it as a robust solution for LLM fine-tuning.

**Limitations and Future Work.** While AlphaDPO enhances performance, it introduces an additional hyperparameter,  $\alpha$ , requiring manual tuning. Future work could focus on developing an adaptive approach to automatically adjust this parameter. Additionally, although we show AlphaDPO’s theoretical equivalence to online methods, it remains an offline approach. Extending it to online learning would allow real-time adaptation, broadening its application in interactive environments. Lastly, we observed that different benchmarks, such as AlpacaEval 2 and Arena-Hard, require distinct parameter settings for optimal performance. Investigating a more generalized approach that adapts effectively across multiple benchmarks would further improve the model’s versatility.



## Impact Statement

This work introduces AlphaDPO, an adaptive preference optimization framework that improves Large Language Model (LLM) alignment with human preferences. By establishing a more principled and effective mechanism for preference optimization, AlphaDPO contributes to developing more robustly aligned LLMs, a crucial step towards safer and more beneficial AI systems.

## Acknowledgment

This research is supported by the National Science and Technology Major Project (2023ZD0121102), and the National Natural Science Foundation of China (62302321, U24B20180). This research was also supported by the advanced computing resources provided by the Supercomputing Center of the USTC.

## References

- AI@Meta. Llama 3 model card. 2024. URL [https://github.com/meta-llama/llama3/blob/main/MODEL\\_CARD.md](https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md).
- Amini, A., Vieira, T., and Cotterell, R. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.
- Anthony, T., Tian, Z., and Barber, D. Thinking fast and slow with deep learning and tree search. *NeurIPS*, 2017.
- Askell, A., Bai, Y., Chen, A., Drain, D., Ganguli, D., Henighan, T., Jones, A., Joseph, N., Mann, B., Das-Sarma, N., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Kernion, J., Ndousse, K., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., and Kaplan, J. A general language assistant as a laboratory for alignment. *CoRR*, abs/2112.00861, 2021.
- Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. In *AISTATS*, 2024.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Das-Sarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., Showk, S. E., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T. B., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv*, abs/2204.05862, 2022.
- Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952.
- Chen, G. H., Chen, S., Liu, Z., Jiang, F., and Wang, B. Humans or llms as the judge? A study on judgement biases. *Arxiv*, abs/2402.10669, 2024a.
- Chen, L., Zhu, C., Soselia, D., Chen, J., Zhou, T., Goldstein, T., Huang, H., Shoenybi, M., and Catanzaro, B. ODIN: Disentangled reward mitigates hacking in RLHF. In *ICML*, 2024b.
- Christiano, P. F., Leike, J., Brown, T. B., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *NIPS*, pp. 4299–4307, 2017.
- Conover, M., Hayes, M., Mathur, A., Xie, J., Wan, J., Shah, S., Ghodsi, A., Wendell, P., Zaharia, M., and Xin, R. Free dolly: Introducing the world’s first truly open instruction-tuned LLM, 2023. URL <https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm>.
- Ding, N., Chen, Y., Xu, B., Qin, Y., Zheng, Z., Hu, S., Liu, Z., Sun, M., and Zhou, B. Enhancing chat language models by scaling high-quality instructional conversations. In *EMNLP*, 2023.
- Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. RAFT: reward ranked finetuning for generative foundation model alignment. *Trans. Mach. Learn. Res.*, 2023.
- Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. Length-controlled alpacaEval: A simple way to debias automatic evaluators. In *COLM*, 2024.
- Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. KTO: Model alignment as prospect theoretic optimization. In *ICML*, 2024.
- Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *ICML*, 2023.
- Geng, X., Gudibande, A., Liu, H., Wallace, E., Abbeel, P., Levine, S., and Song, D. Koala: A dialogue model for academic research. *Blog post, April*, 1:6, 2023.
- Gorbatovski, A., Shaposhnikov, B., Malakhov, A., Surnachev, N., Aksenov, Y., Maksimov, I., Balagansky, N., and Gavrilov, D. Learn your reference model for real good alignment. In *ICLR*, 2025.
- Guo, S., Zhang, B., Liu, T., Liu, T., Khalman, M., Llinares, F., Ramé, A., Mesnard, T., Zhao, Y., Piot, B., Ferret, J., and Blondel, M. Direct language model alignment from online AI feedback. *Arxiv*, abs/2402.04792, 2024.

- Havrilla, A., Raparthy, S., Nalmpantis, C., Dwivedi-Yu, J., Zhuravinskiy, M., Hambro, E., and Railneau, R. GLoRe: When, where, and how to improve LLM reasoning via global and local refinements. In *ICML*, 2024.
- He, L., Xia, M., and Henderson, P. What’s in your” safe” data?: Identifying benign data that breaks safety. *COLM*, 2024.
- Hong, J., Lee, N., and Thorne, J. ORPO: Monolithic preference optimization without reference model. In *EMNLP*, 2024.
- Ji, H., Lu, C., Niu, Y., Ke, P., Wang, H., Zhu, J., Tang, J., and Huang, M. Towards efficient exact optimization of language model alignment. In *ICML*, 2024a.
- Ji, K., He, J., and Gu, Q. Reinforcement learning from human feedback with active queries. *CoRR*, abs/2402.09401, 2024b.
- Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., de Las Casas, D., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., Lavaud, L. R., Lachaux, M.-A., Stock, P., Scao, T. L., Lavril, T., Wang, T., Lacroix, T., and Sayed, W. E. Mistral 7B. *ArXiv*, abs/2310.06825, 2023.
- Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Köpf, A., Kilcher, Y., von Rütte, D., Anagnostidis, S., Tam, Z. R., Stevens, K., Barhoum, A., Nguyen, D. M., Stanley, O., Nagyfi, R., et al. Openassistant conversations-democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.
- Lambert, N., Pyatkin, V., Morrison, J. D., Miranda, L. J. V., Lin, B. Y., Chandu, K. R., Dziri, N., Kumar, S., Zick, T., Choi, Y., Smith, N. A., and Hajishirzi, H. RewardBench: Evaluating reward models for language modeling. In *NAACL*, 2025.
- Langley, P. Crafting papers on machine learning. In *ICML*, 2000.
- Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., and Legg, S. Scalable agent alignment via reward modeling: a research direction. *CoRR*, abs/1811.07871, 2018.
- Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Zhu, B., Gonzalez, J. E., and Stoica, I. From live data to high-quality benchmarks: The Arena-Hard pipeline, April 2024. URL <https://lmsys.org/blog/2024-04-19-arena-hard/>.
- Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. AlpacaEval: An automatic evaluator of instruction-following models. [https://github.com/tatsu-lab/alpaca\\_eval](https://github.com/tatsu-lab/alpaca_eval), 2023.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let’s verify step by step. In *ICLR*, 2024.
- Liu, T., Qin, Z., Wu, J., Shen, J., Khalman, M., Joshi, R., Zhao, Y., Saleh, M., Baumgartner, S., Liu, J., Liu, P. J., and Wang, X. Lipo: Listwise preference optimization through learning-to-rank. In *NAACL*, 2025a.
- Liu, Y., Liu, P., and Cohan, A. Understanding reference policies in direct preference optimization. In *NAACL*, 2025b.
- Luo, H., Sun, Q., Xu, C., Zhao, P., Lou, J., Tao, C., Geng, X., Lin, Q., Chen, S., and Zhang, D. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *ICLR*, 2025.
- Meng, Y., Xia, M., and Chen, D. Simpo: Simple preference optimization with a reference-free reward. In *NeurIPS*, 2024.
- Morimura, T., Sakamoto, M., Jinnai, Y., Abe, K., and Ariu, K. Filtered direct preference optimization. In *EMNLP*, 2024.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P. F., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.
- Park, R., Rafailov, R., Ermon, S., and Finn, C. Disentangling length from quality in direct preference optimization. In *ACL*, 2024.
- Patro, S. G. K. and Sahu, K. K. Normalization: A preprocessing stage. *Arxiv*, abs/1503.06462, 2015.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.
- Rafailov, R., Chittepudi, Y., Park, R., Sikchi, H., Hejna, J., Knox, W. B., Finn, C., and Niekum, S. Scaling laws for reward model overoptimization in direct alignment algorithms. In *NeurIPS*, 2024.

- Rivière, M., Pathak, S., Sessa, P. G., Hardin, C., Bhupatiraju, S., Hussenot, L., Mesnard, T., Shahriari, B., Ramé, A., Ferret, J., Liu, P., Tafti, P., Friesen, A., Casbon, M., Ramos, S., Kumar, R., Lan, C. L., Jerome, S., Tsitsulin, A., Vieillard, N., Stanczyk, P., Girgin, S., Momchev, N., Hoffman, M., Thakoor, S., Grill, J., Neyshabur, B., Bachem, O., Walton, A., Severyn, A., Parrish, A., Ahmad, A., Hutchison, A., Abdagici, A., Carl, A., Shen, A., Brock, A., Coenen, A., Laforge, A., Paterson, A., Bastian, B., Piot, B., Wu, B., Royal, B., Chen, C., Kumar, C., Perry, C., Welty, C., Choquette-Choo, C. A., Sinopalnikov, D., Weinberger, D., Vijaykumar, D., Rogozinska, D., Herbison, D., Bandy, E., Wang, E., Noland, E., Moreira, E., Senter, E., Eltyshv, E., Visin, F., Rasskin, G., Wei, G., Cameron, G., Martins, G., Hashemi, H., Klimczak-Plucinska, H., Batra, H., Dhand, H., Nardini, I., Mein, J., Zhou, J., Svensson, J., Stanway, J., Chan, J., Zhou, J. P., Carrasqueira, J., Iljazi, J., Becker, J., Fernandez, J., van Amersfoort, J., Gordon, J., Lipschultz, J., Newlan, J., Ji, J., Mohamed, K., Badola, K., Black, K., Millican, K., McDonnell, K., Nguyen, K., Sodhia, K., Greene, K., Sjöstrand, L. L., Usui, L., Sifre, L., Heuermann, L., Lago, L., and McNealus, L. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. Preference ranking optimization for human alignment. In *AAAI*, 2024.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *NeurIPS*, 2020.
- Swamy, G., Dann, C., Kidambi, R., Wu, S., and Agarwal, A. A minimaximalist approach to reinforcement learning from human feedback. In *ICML*, 2024.
- Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model, 2023.
- Wang, C., Jiang, Y., Yang, C., Liu, H., and Chen, Y. Beyond reverse KL: generalizing direct preference optimization with diverse divergence constraints. In *ICLR*, 2024a.
- Wang, H., Xiong, W., Xie, T., Zhao, H., and Zhang, T. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *EMNLP*, 2024b.
- Wu, J., Xie, Y., Yang, Z., Wu, J., Gao, J., Ding, B., Wang, X., and He, X.  $\beta$ -dpo: Direct preference optimization with dynamic  $\beta$ . In *NeurIPS*, 2024.
- Xiong, W., Dong, H., Ye, C., Zhong, H., Jiang, N., and Zhang, T. Gibbs sampling from human feedback: A provable KL- constrained framework for RLHF. *CoRR*, abs/2312.11456, 2023.
- Xu, H., Sharaf, A., Chen, Y., Tan, W., Shen, L., Durme, B. V., Murray, K., and Kim, Y. J. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *ICML*, 2024.
- Yang, S., Cui, L., Cai, D., Huang, X., Shi, S., and Lam, W. Not all preference pairs are created equal: A recipe for annotation-efficient iterative preference learning. In *ACL*, 2024.
- Ye, C., Xiong, W., Zhang, Y., Jiang, N., and Zhang, T. A theoretical analysis of nash learning from human feedback under general kl-regularized preference. *CoRR*, abs/2402.07314, 2024.
- Yuan, H., Yuan, Z., Tan, C., Wang, W., Huang, S., and Huang, F. RRHF: rank responses to align language models with human feedback. In *NeurIPS*, 2023.
- Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. In *ICML*, 2024.
- Zeng, Y., Liu, G., Ma, W., Yang, N., Zhang, H., and Wang, J. Token-level direct preference optimization. In *ICML*, 2024.
- Zhang, S., Yu, D., Sharma, H., Yang, Z., Wang, S., Hassan, H., and Wang, Z. Self-exploring language models: Active preference elicitation for online alignment. *TMLR*, 2025.
- Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. LIMA: Less is more for alignment. *NeurIPS*, 2023.
- Zhu, W., He, Z., Wang, X., Liu, P., and Wang, R. Weak-to-strong preference optimization: Stealing reward from weak aligned model. In *ICLR*, 2025.
- Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P. F., and Irving, G. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593, 2019.

## A. Related Work

**Reinforcement learning from human feedback.** RLHF is a technique that aligns large language models with human preferences and values (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Azar et al., 2024). Traditional RLHF can be divided into three stages: supervised fine-tuning (Zhou et al., 2023; Taori et al., 2023; Geng et al., 2023; Conover et al., 2023; Köpf et al., 2023; Ding et al., 2023), reward modeling (Gao et al., 2023; Luo et al., 2025; Chen et al., 2024b; Lightman et al., 2024; Havrilla et al., 2024; Lambert et al., 2025), and policy optimization (Schulman et al., 2017; Anthony et al., 2017). In the third stage, Proximal Policy Optimization (PPO) is a widely used algorithm. Additionally, Xiong et al. (2023) proposed efficient algorithms for the reverse-KL regularized contextual bandit framework in RLHF. Ye et al. (2024) introduced provably efficient algorithms for KL-regularized Nash-Learning from Human Feedback (NLHF). Furthermore, Ji et al. (2024b) developed an active-query-based PPO algorithm with specific regret bounds and query complexity.

**Offline direct preference optimization.** Several alternative preference optimization objectives have been proposed in addition to DPO (Rafailov et al., 2023). IPO (Azar et al., 2024) addresses the overfitting issues associated with DPO. ORPO (Hong et al., 2024) and SimPO (Meng et al., 2024) aim to eliminate the dependence on a reference model. R-DPO (Park et al., 2024) focuses on mitigating exploitation based on sequence length. KTO (Ethayarajh et al., 2024) deals with preference optimization when data are not pairwise. CPO (Xu et al., 2024) and  $\beta$ -DPO (Wu et al., 2024) emphasize the quality of preference data. Another line of research explores comparisons among more than two instances (Dong et al., 2023; Liu et al., 2025a; Song et al., 2024; Yuan et al., 2023).

**Online direct preference optimization.** Offline direct preference optimization methods are simple but rely on preference data collected offline. RLHF methods interact online with the language model being aligned but require policy gradients. In contrast, online direct preference optimization methods combine the advantages of both approaches. Yuan et al. (2024) proposed a “self-rewarding” approach in which the policy being aligned provides online feedback to itself. Alternatively, OAIF (Guo et al., 2024) is a novel online preference optimization method that can leverage feedback from any LLM, including those stronger than the LLM being aligned. Swamy et al. (2024) also concurrently investigate the importance of online preference but still rely on reward models (RMs). SELMA (Zhang et al., 2025) improves exploration efficiency by selectively favoring responses with high potential rewards rather than indiscriminately sampling unseen responses.

## B. Proofs

### B.1. Proof of Theorem 3.1

**Theorem 3.1.** *Let  $U(y|x)$  denote a uniform distribution over the vocabulary for a given input  $x$ , replacing  $\pi_{\text{ref}}(y|x)$  in the DPO loss function. Then, the DPO loss function simplifies to:*

$$\mathcal{L}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(\beta (\log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x)) - \gamma)], \quad (6)$$

where  $\gamma = \beta (\log U(y_w|x) - \log U(y_l|x))$  is a constant. Under a length-normalized reward formulation, this loss function becomes:

$$\mathcal{L}_{LN}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right]. \quad (7)$$

Therefore, SimPO can be interpreted as a special case of DPO where the reference model is a uniform distribution.

*Proof.* Let  $U(y|x)$  denote a uniform distribution over the vocabulary  $\mathcal{V}$  for a given input  $x$ . Specifically, for any sequence  $y$ , the uniform distribution is defined as:

$$U(y|x) = \prod_{t=1}^{|y|} \frac{1}{|\mathcal{V}|} = \left( \frac{1}{|\mathcal{V}|} \right)^{|y|}.$$

Consider the DPO loss function:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)} - \beta \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right].$$



By substituting  $\pi_{\text{ref}} = U$ , the term involving the reference policy simplifies to:

$$\beta \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)} = \beta (\log U(y_w|x) - \log U(y_l|x)) = \gamma,$$

where  $\gamma$  is a constant. This constancy arises because  $y_w$  and  $y_l$  are chosen from distinct subsets of the vocabulary, ensuring that  $\log U(y_w|x) - \log U(y_l|x)$  does not depend on the lengths of the sequences but is instead determined by the fixed probabilities of the respective subsets. Consequently,  $\gamma$  remains fixed across all samples in  $\mathcal{D}$ .

Substituting back into the DPO loss function, we obtain:

$$\mathcal{L}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma (\beta (\log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x)) - \gamma)].$$

Under the length-normalized reward formulation, the rewards are adjusted by the lengths of the sequences  $y_w$  and  $y_l$ . This normalization yields:

$$\mathcal{L}_{\text{LN}}(\pi_\theta; U) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right].$$

Here,  $\gamma$  remains a fixed constant since it is derived from the uniform distribution over distinct vocabulary subsets corresponding to  $y_w$  and  $y_l$ .

Comparing this with the SimPO loss function:

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x) - \gamma \right) \right],$$

it is evident that:

$$\mathcal{L}_{\text{LN}}(\pi_\theta; U) = \mathcal{L}_{\text{SimPO}}(\pi_\theta).$$

Thus, when the reference policy  $\pi_{\text{ref}}$  is a uniform distribution over distinct vocabulary subsets for  $y_w$  and  $y_l$ , the DPO loss function simplifies to the SimPO loss function with  $\gamma$  being a fixed constant. This establishes that SimPO is a special case of DPO under the specified conditions.  $\square$

## B.2. Proof of Lemma 4.1

**Lemma B.1** (Equivalence of Margin Terms). *Consider the margin term  $\delta(x, y_w, y_l)$  that represents the difference in sequential KL divergences between a reference policy  $\pi_{\text{ref}}$  and policy  $\pi_\theta$  for preferred sequence  $y_w$  and rejected sequence  $y_l$ , conditioned on input  $x$ :*

$$\begin{aligned} \delta(x, y_w, y_l) &= \beta \mathbb{D}_{\text{SeqKL}}[x, y_l; \pi_{\text{ref}} || \pi_\theta] \\ &\quad - \beta \mathbb{D}_{\text{SeqKL}}[x, y_w; \pi_{\text{ref}} || \pi_\theta], \end{aligned} \tag{17}$$

*Under the assumption that the sequential KL divergence can be approximated by the log-likelihood ratio of complete sequences, the margin term  $\delta(x, y_w, y_l)$  admits the following approximation:*

$$\begin{aligned} \delta(x, y_w, y_l) &\approx \beta \left( \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \\ &= M(x, y_w, y_l), \end{aligned}$$

where  $M(x, y_w, y_l)$  is the margin term in the AlphaDPO objective.

*Proof of Lemma 4.1.* We begin by expanding the definition of  $\delta(x, y_w, y_l)$ :

$$\delta(x, y_w, y_l) = \beta \mathbb{D}_{\text{SeqKL}}[x, y_l; \pi_{\text{ref}} || \pi_\theta] - \beta \mathbb{D}_{\text{SeqKL}}[x, y_w; \pi_{\text{ref}} || \pi_\theta]$$

Expanding each sequential KL divergence, we have:

$$\mathbb{D}_{\text{SeqKL}}[x, y; \pi_{\text{ref}} \parallel \pi_{\theta}] = \sum_{t=1}^{|y|} \mathbb{E}_{z \sim \pi_{\text{ref}}} \left[ \log \frac{\pi_{\text{ref}}(z \mid [x, y^{<t}])}{\pi_{\theta}(z \mid [x, y^{<t}])} \right]$$

Substituting this into the expression for  $\delta$ , we obtain:

$$\delta(x, y_w, y_l) = \beta \sum_{t=1}^{|y_l|} \mathbb{E}_{z \sim \pi_{\text{ref}}} \left[ \log \frac{\pi_{\text{ref}}(z \mid [x, y_l^{<t}])}{\pi_{\theta}(z \mid [x, y_l^{<t}])} \right] - \beta \sum_{t=1}^{|y_w|} \mathbb{E}_{z \sim \pi_{\text{ref}}} \left[ \log \frac{\pi_{\text{ref}}(z \mid [x, y_w^{<t}])}{\pi_{\theta}(z \mid [x, y_w^{<t}])} \right]$$

Under the assumption that the reference policy  $\pi_{\text{ref}}$  has large errors, we approximate the expectation  $\mathbb{E}_{z \sim \pi_{\text{ref}}}$  with a uniform distribution. This approximation simplifies each expectation term as follows:

$$\sum_{t=1}^{|y|} \mathbb{E}_{z \sim \pi_{\text{ref}}} \left[ \log \frac{\pi_{\text{ref}}(z \mid [x, y^{<t}])}{\pi_{\theta}(z \mid [x, y^{<t}])} \right] \approx \log \frac{\pi_{\theta}(y \mid x)}{\pi_{\text{ref}}(y \mid x)}$$

Applying this approximation to both sequential KL divergence terms, we obtain:

$$\delta(x, y_w, y_l) \approx \beta \left( \log \frac{\pi_{\theta}(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} - \log \frac{\pi_{\theta}(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} \right)$$

This expression can be rewritten as:

$$\delta(x, y_w, y_l) \approx \beta \left( \log \frac{\pi_{\theta}(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_{\theta}(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) = M(x, y_w, y_l)$$

where  $M(x, y_w, y_l)$  is the margin term defined in the AlphaDPO objective. Thus, we have shown that:

$$\delta(x, y_w, y_l) \approx M(x, y_w, y_l)$$

This completes the proof. □

### C. The motivation for the proposed $\hat{\pi}_{\text{ref}}(y|x)$

The motivation for the proposed reference policy  $\hat{\pi}_{\text{ref}}(y|x)$  can be clarified as follows:

- **Utility Theory Perspective:** The proposed  $\hat{\pi}_{\text{ref}}(y|x)$  is designed with the uniform distribution  $U(y|x)$  as a baseline. The term  $\left( \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)} \right)^{\alpha}$  dynamically adjusts the reward margin by balancing contributions from the policy and reference models. This mechanism can be interpreted through the lens of utility theory as relative attractiveness, enabling adaptive instance-specific reward modeling.
- **Gradient Perspective** By introducing  $\hat{\pi}_{\text{ref}}(y|x)$ , the framework mitigates the label flipping issues found in DPO or SimPO. In the SimPO framework, the gradient is expressed as:

$$\nabla_{\theta} \mathcal{L}_{\text{SimPO}}(\pi_{\theta}) = -\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ s_{\theta} \left( \frac{1}{|y_w|} \nabla_{\theta} \log \pi_{\theta}(y_w|x) - \frac{1}{|y_l|} \nabla_{\theta} \log \pi_{\theta}(y_l|x) \right) \right],$$

where  $s_{\theta} = \sigma \left( \frac{\beta}{|y_l|} \log \pi_{\theta}(y_l|x) - \frac{\beta}{|y_w|} \log \pi_{\theta}(y_w|x) + \gamma \right)$ .

This formulation may amplify weights when the reward estimate is incorrect. By contrast, under AlphaDPO:

$$s_{\theta} = \sigma \left( \frac{\beta}{|y_l|} \log \pi_{\theta}(y_l|x) - \frac{\beta}{|y_w|} \log \pi_{\theta}(y_w|x) + \gamma + \alpha M(x, y_w, y_l) \right),$$

the additional  $\alpha M(x, y_w, y_l)$  component increases weight when the reward estimate is accurate, ensuring a more robust reward signal.

- **Motivational Core** The central goal of the proposed AlphaDPO is to address the unreliability of the reference policy, as outlined in Section 3.1. By integrating the policy model into the reference model design, the quality of the reference model is enhanced, improving fine-tuning performance. Similar concepts have been explored in recent works (Gorbatovski et al., 2025; Liu et al., 2025b).

## D. Experiments

### D.1. Implementation Details

We observed that the performance of various methods is highly sensitive to model parameters and learning rates. To ensure a fair comparison, we conducted a hyperparameter search as specified in the respective papers. The specific search ranges are detailed in Table 3. Furthermore, due to recent updates to both Llama3-8b and Instruct-7b models, we had to re-implement SimPO as the original results were no longer directly applicable.

**Training hyperparameters.** For other parameters, we used a consistent batch size of 128 across all methods. The learning rate was searched within the range of [3e-7, 5e-7, 8e-7, 1e-6], and all models were trained for a single epoch with a cosine learning rate schedule and a 10% warmup phase. Adam was used as the optimizer (Kingma & Ba, 2014). Additionally, the maximum sequence length was set to 2048.

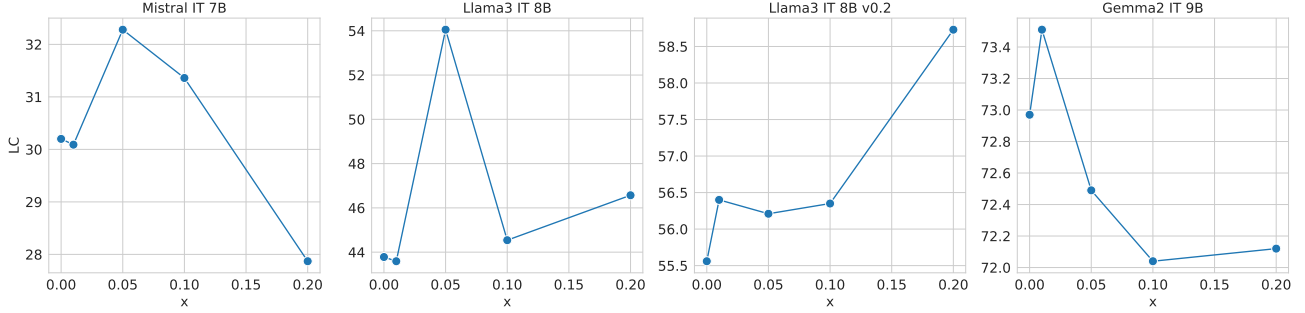
Table 3. Various preference optimization objectives and hyperparameter search range.

Method	Objective	Hyperparameter
DPO (Rafailov et al., 2023)	$-\log \sigma \left( \beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} \right)$	$\beta \in [0.01, 0.05, 0.1]$
IPO (Azar et al., 2024)	$\left( \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} - \frac{1}{2\tau} \right)^2$	$\tau \in [0.01, 0.1, 0.5, 1.0]$
CPO (Xu et al., 2024)	$-\log \sigma \left( \beta \log \pi_\theta(y_w x) - \beta \log \pi_\theta(y_l x) \right) - \lambda \log \pi_\theta(y_w x)$	$\alpha = 1.0, \beta \in [0.01, 0.05, 0.1]$
KTO (Ethayarajh et al., 2024)	$-\lambda_w \sigma \left( \beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - z_{\text{ref}} \right) + \lambda_l \sigma \left( z_{\text{ref}} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} \right)$ , where $z_{\text{ref}} = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\beta \text{KL}(\pi_\theta(y x)    \pi_{\text{ref}}(y x))]$	$\lambda_l = \lambda_w = 1.0$ $\beta \in [0.01, 0.05, 0.1]$
ORPO (Hong et al., 2024)	$-\log p_\theta(y_w x) - \lambda \log \sigma \left( \log \frac{p_\theta(y_w x)}{1-p_\theta(y_w x)} - \log \frac{p_\theta(y_l x)}{1-p_\theta(y_l x)} \right)$ , where $p_\theta(y x) = \exp \left( \frac{1}{ y } \log \pi_\theta(y x) \right)$	$\lambda \in [0.1, 0.5, 1.0, 2.0]$
R-DPO (Park et al., 2024)	$-\log \sigma \left( \beta \log \frac{\pi_\theta(y_w x)}{\pi_{\text{ref}}(y_w x)} - \beta \log \frac{\pi_\theta(y_l x)}{\pi_{\text{ref}}(y_l x)} - (\alpha y_w  - \alpha y_l ) \right)$	$\alpha \in [0.05, 0.1, 0.5, 1.0]$ $\beta \in [0.01, 0.05, 0.1]$
SimPO (Meng et al., 2024)	$-\log \sigma \left( \frac{\beta}{ y_w } \log \pi_\theta(y_w x) - \frac{\beta}{ y_l } \log \pi_\theta(y_l x) - \gamma \right)$	$\beta \in [2.0, 4.0, 6.0, 8.0]$ $\gamma \in [0.3, 0.5, 1.0, 1.2, 1.4, 1.6]$
AlphaDPO	$-\log \sigma (u(x, y_w, y_l) - \text{sg}[\gamma + \alpha M^*(x, y_w, y_l)])$ where $u(x, y_w, y_l) = \frac{\beta}{ y_w } \log \pi_\theta(y_w x) - \frac{\beta}{ y_l } \log \pi_\theta(y_l x)$	$\beta \in [2.5, 10.0], \gamma \in [0.1, 0.3, 0.5]$ $\alpha \in [1e-2, 5e-2, 0.1, 0.2]$

Table 4. The hyperparameter values in AlphaDPO used for each training setting.

Setting	$\beta$	$\gamma$	$\alpha$	Learning rate
Mistral-Instruct	2.5	0.15	5e-2	6e-7
Llama3-Instruct	2.5	0.6	0.2	1e-6
Llama3-Instruct-v0.2	10.0	0.4	0.2	1e-6
Gemma2-Instruct	10.0	0.4	5e-2	8e-7

**Hyperparameter in AlphaDPO.** Table 4 outlines the hyperparameters used for AlphaDPO under various settings. It’s important to note that while our approach involves three key parameters, we have found through experience that  $\beta$  can be reliably set to 10.0 by default. Among these parameters,  $\gamma$  typically requires more careful tuning. As for  $\alpha$ , we have observed consistent performance improvements when set to 5e-2 by default. If you are already familiar with the parameter settings for SimPO, you can focus your search primarily on  $\alpha$  or simply adopt the default setting of  $\alpha = 5e-2$ .

Figure 5. AlphaDPO LC on AlpacaEval 2 with different  $\alpha$  values.

**Decoding hyperparameters.** The decoding hyperparameters used in this study are the same as those employed by SimPO<sup>4</sup>. We extend our sincere gratitude to the SimPO team for sharing their invaluable insights.

**Computation environment.** All training experiments presented in this paper were conducted using 8xA100 GPUs, as per the procedures detailed in the alignment-handbook repository.<sup>5</sup>

## D.2. AlphaDPO without length-normalized

In this paper, we consider length-normalized training as a stability technique and not as a primary contribution of this work. Existing research (Meng et al., 2024) has demonstrated that length normalization can indeed enhance model performance, particularly with respect to the length control win rate. However, to validate the general applicability of AlphaDPO—specifically, its stability and performance without length normalization—we conducted experiments across several models: meta-llama/Meta-Llama-3-8B-Instruct, mistralai/Mistral-7B-Instruct-v0.2, and google/gemma-2-9b-it.

We evaluated DPO, SimPO without length normalization, and AlphaDPO without length normalization. The experimental results, as shown in Table 5, demonstrate that AlphaDPO consistently achieves performance improvements even without the use of length normalization. This indicates the robustness and general effectiveness of AlphaDPO.

Table 5. Performance comparison without length-normalization on AlpacaEval2. “LC” denotes the length-controlled win rate, and “WR” represents the raw win rate.

Method	Llama3-Instruct (8B)		Mistral-Instruct (7B)		Llama3-Instruct v0.2 (8B)		Gemma2-Instruct (9B)	
	LC (%)	WR (%)	LC (%)	WR (%)	LC (%)	WR (%)	LC (%)	WR (%)
DPO	40.2	38.1	20.3	18.0	51.1	53.3	70.2	66.9
SimPO w/o LN	42.4	40.4	30.5	38.2	49.2	52.6	71.2	69.9
AlphaDPO w/o LN	<b>44.4</b>	<b>42.6</b>	<b>32.0</b>	<b>38.4</b>	<b>51.1</b>	<b>54.0</b>	<b>72.7</b>	<b>70.5</b>

## D.3. AlphaDPO with different $\alpha$

To analyze the impact of  $\alpha$  on the model, we adjust its value for four different models. The results are illustrated in Figure 5. When  $\alpha$  is set to 0, the model degenerates to SimPO. As  $\alpha$  increases, performance improves across all models, although the optimal value of  $\alpha$  varies among them. This highlights the significance of  $\alpha$ .

It is noteworthy that within the parameter tuning range [1e-2, 5e-2, 0.1, 0.2], the optimal  $\alpha$  values are consistently around 0.1 or even closer to 5e-2.

<sup>4</sup><https://github.com/princeton-nlp/SimPO/tree/main/eval>

<sup>5</sup><https://github.com/huggingface/alignment-handbook>



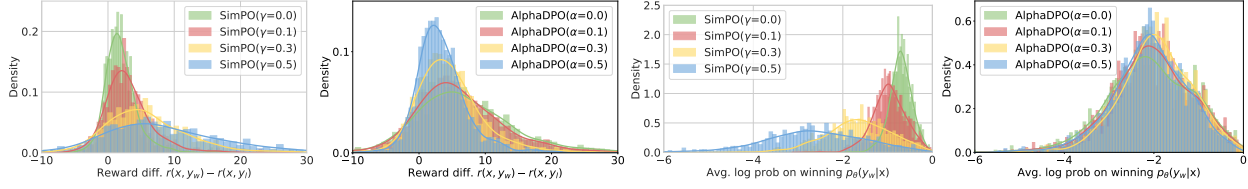


Figure 6. (a) SimPO: Reward difference distribution under different  $\gamma$  values. (b) AlphaDPO: Reward difference distribution under different  $\alpha$  values. (c) SimPO: Log likelihood distribution on chosen responses under different  $\gamma$  values. (d) AlphaDPO: Log likelihood distribution on chosen responses under different  $\alpha$  values.

#### D.4. Comparison With TDPO

To investigate the relationship between TDPO and AlphaDPO, we conducted the experiments, with the results outlined below.

Table 6. Performance comparison between TDPO and AlphaDPO.

Method	Llama3-Instruct (8B)	
	LC (%)	WR (%)
TDPO	52.8	45.9
AlphaDPO w/ $\delta(x, y_w, y_l)$	56.9	50.4
AlphaDPO w/ $M(x, y_w, y_l)$	58.7	51.1

In its original form, TDPO did not perform well on Llama3-8B. By applying Lemma 4.1, we modified the expression  $M(x, y_w, y_l)$  in AlphaDPO to use TDPO’s  $\delta(x, y_w, y_l)$ , converting our sentence-level estimations to a token-level calculation. This adjustment resulted in a noticeable performance improvement, which we attribute to the length-normalization,  $\gamma$  and z-score normalization of  $\delta(x, y_w, y_l)$ . Nevertheless, the modified TDPO still underperformed compared to AlphaDPO. This indicates that, when the  $\pi_{\text{ref}}$  is suboptimal, token-level calculations are prone to significant errors.