RePO: Understanding Preference Learning Through ReLU-Based Optimization

Junkang Wu[1*] **Kexin Huang**[1] **Xue Wang**[2] **Jinyang Gao**[2] **Bolin Ding**[2]
Jiancan Wu[1,3] **Xiangnan He**[4†] **Xiang Wang**[1†]
[1]University of Science and Technology of China, [2]Alibaba Group
[3]Institute of Dataspace, Hefei Comprehensive National Science Center
[4]MoE Key Lab of BIPC, University of Science and Technology of China
{jkwu0909, xiangwang1223, xiangnanhe}@gmail.com

Abstract

Preference learning has become a common approach in various recent methods for aligning large language models with human values. These methods optimize the preference margin between chosen and rejected responses, subject to certain constraints for avoiding over-optimization. In this paper, we report surprising empirical findings that simple ReLU activation can learn meaningful alignments even using *none* of the following: (i) sigmoid-based gradient constraints, (ii) explicit regularization terms. Our experiments show that over-optimization does exist, but a threshold parameter γ plays an essential role in preventing it by dynamically filtering training examples. We further provide theoretical analysis demonstrating that ReLU-based Preference Optimization (RePO) corresponds to the convex envelope of the 0-1 loss, establishing its fundamental soundness. Our "RePO" method achieves competitive or superior results compared to established preference optimization approaches. We hope this simple baseline will motivate researchers to rethink the fundamental mechanisms behind preference optimization for language model alignment.

1 Introduction

Recent years have witnessed significant advances in aligning large language models (LLMs) with human preferences [1–4]. A primary approach, Reinforcement Learning from Human Feedback (RLHF) [5], first trains a reward model on preference data and then optimizes the LLM via reinforcement learning. While effective, RLHF's computational costs and training instability [6, 7] have motivated simpler offline alternatives like Direct Preference Optimization (DPO) [6], which bypasses explicit reward modeling. Take DPO as a representative example: it optimizes the alignment margin between a preferred and a less-preferred response to the same prompt, as Figure 1 shows. The alignment of each response is quantified via an implicit reward, defined as the log-ratio of the predicted likelihoods under the policy model (*i.e.,* the LLM being optimized) and a reference model (*e.g.,* a fixed supervised fine-tuned (SFT) model).

A fundamental challenge in preference learning is *over-optimization* — where models excessively amplify reward margins between preferred and non-preferred responses, potentially degrading generation quality [8–10]. Several approaches have been developed to mitigate this issue. DPO [6] and SimPO [11] employ sigmoid weighting through log-sigmoid activation that diminishes gradients as reward margins increase, naturally preventing over-optimization. The β parameter controls gradient

[*]Work done at Alibaba Group.
[†]Xiangnan He and Xiang Wang are the corresponding authors.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

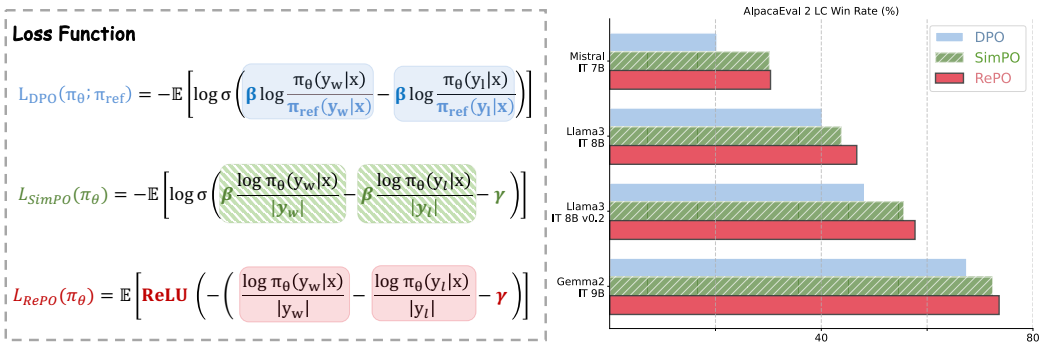

Figure 1: **Comparing preference learning mechanisms.** RePO employs a simpler binary thresholding mechanism than SimPO and DPO, as highlighted in the shaded box. Despite its simplicity, this mechanism achieves competitive results by naturally preventing over-optimization.

distribution sharpness — larger values produce more binary-like gradients, as illustrated in Figure 2. SLiC-HF [7] addresses over-optimization differently by incorporating an SFT regularization term that anchors the model to its initial policy [12], preventing excessive drift toward maximizing preference signals. These mechanisms effectively balance preference optimization with generation quality preservation, forming the foundation of current preference learning approaches.

Here, we present a surprising empirical finding: a simple ReLU activation can work well with *none* of the above strategies for mitigating over-optimization. Our analysis reveals that as parameter β in SimPO approaches infinity, its sigmoid weighting naturally converges to a binary thresholding mechanism — motivating our exploration of **Re**LU-based **P**reference **O**ptimization (**RePO**). This mechanism uses a single ReLU function with only one hyperparameter γ, creating a clear decision boundary that selectively updates sample pairs with insufficient reward margins ($M_\theta < \gamma$) while filtering out well-separated pairs ($M_\theta \geq \gamma$). We illustrate this "RePO" method in Figure 1.

Thanks to its conceptual simplicity, RePO can serve as a hub that relates several existing methods. In essence, our method can be viewed as "SimPO *without* log-sigmoid" or "SLiC-HF *without* SFT regularization term". Interestingly, RePO is related to each method by removing one of its core components. Even so, RePO effectively prevents over-optimization while performing competitively or better (*cf.* Figure 1).

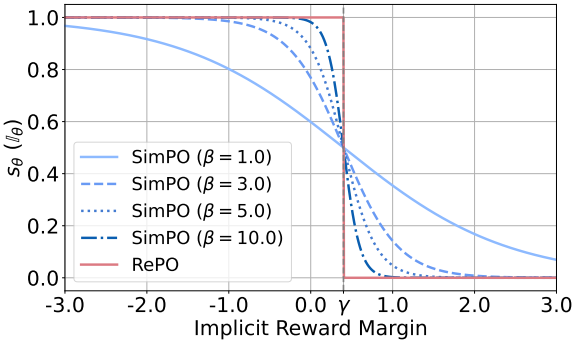

Figure 2: Gradient weighting functions of SimPO (s_θ) and RePO ($\mathbb{I}(M_\theta < \gamma)$). As $\beta \to \infty$, s_θ converges to the binary indicator (red line), establishing RePO as the limit case of SimPO.

We empirically show that over-optimization *do* exist, but ReLU activation with threshold γ is critical to prevent such solutions. This implies that in over-optimization regimes, selecting *which* examples to learn from is more critical than determining *how much* to learn from each. The γ threshold induces an emergent data filtering behavior, focusing dynamically on challenging samples relative to the model's current capability. Our theoretical analysis reveals that RePO's ReLU loss corresponds precisely to the *convex envelope* of the 0-1 loss (Theorem 4.2), explaining why such a simple mechanism is so effective.

Our simple baseline suggests that the ReLU activation with a proper threshold γ can be an essential reason for the common success of related methods. We believe this work's significance lies in revealing how preference learning principles may be simpler than previously thought. By questioning conventional wisdom about necessary components, we hope to motivate researchers to reconsider the fundamental mechanisms behind preference optimization.

2 Preliminaries

Directed Preference Optimization (DPO). DPO [6] stands out as a leading method for offline preference optimization by eliminating the need for an explicit reward model. Instead, it reformulates the reward $r(x, y)$ as a closed-form expression based on policy ratios:

$$r(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x), \tag{1}$$

where $Z(x)$ is a partition function that does not depend on y. This leads to the DPO loss for a given triplet (x, y_w, y_l) as:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}} \left[\log \sigma \left(\beta \left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)} - \log \frac{\pi_{\text{ref}}(y_w \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right) \right], \tag{2}$$

where $\sigma(\cdot)$ denotes the sigmoid function. This loss encourages the policy π_θ to prefer y_w over y_l in alignment with the reference policy.

Sequence Likelihood Calibration (SLiC-HF). SLiC-HF [7] advances preference optimization with two key innovations: (1) it employs a sequence-level calibration loss that contrasts the log-probability difference between preferred and dispreferred responses using a margin γ, and (2) it integrates a regularization term to prevent divergence from the SFT policy, avoiding the need for an explicit KL penalty. The SLiC-HF loss function is defined as:

$$\mathcal{L}_{\text{SLiC-HF}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}} \Big[\text{ReLU}\Big(- \Big(\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x) - \gamma \Big) \Big) - \lambda \log \pi_\theta(y_w \mid x) \Big]. \tag{3}$$

Simple Preference Optimization (SimPO). SimPO [11] advances preference optimization with two key innovations: (1) it normalizes the reward by the length of the response, calculating the average log-probability per token for a response under the policy π_θ, and (2) it incorporates a target reward margin γ to ensure that the reward difference between the preferred and less preferred responses exceeds this margin. The SimPO loss function is defined as:

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}} \left[\log \sigma \left(\beta \left(\frac{\log \pi_\theta(y_w \mid x)}{|y_w|} - \frac{\log \pi_\theta(y_l \mid x)}{|y_l|} - \gamma \right) \right) \right], \tag{4}$$

where $|y|$ denotes the number of tokens in response y, ensuring length-aware scaling of rewards, and γ is the predefined margin that enforces a minimum difference in rewards between y_w and y_l. To align with subsequent discussions, we modify the original SimPO formulation by setting γ to γ/β.

3 Exploring Simple ReLU Activation in Preference Learning

In this section, we explore what makes a simple ReLU activation function effective for preference learning. We first examine the surprising relationship between ReLU activation and sigmoid weighting through empirical experiments. Then, we investigate the key properties that emerge from this simple mechanism, specifically through the lens of gradient behavior, data filtering patterns, and over-optimization control.

3.1 Examining ReLU-based Preference Optimization

Simplification exploration. Our exploration began by questioning whether log-sigmoid activation or SFT regularization are truly necessary for mitigating over-optimization. We simplified the SimPO loss function through two key modifications: (i) removing the hyperparameter β, and (ii) replacing the log-sigmoid function with a ReLU activation.

We adopt the length-normalized *implicit reward margin* M_θ (as introduced in SimPO [11]):

$$M_\theta = \frac{\log \pi_\theta(y_w \mid x)}{|y_w|} - \frac{\log \pi_\theta(y_l \mid x)}{|y_l|}, \tag{5}$$

which quantifies the policy's preference between responses. Using M_θ, we examine a loss function with the following form:

$$\mathcal{L}_{\text{RePO}}(\pi_\theta) = \mathbb{E}_{(x,y_w,y_l)\in\mathcal{D}} \left[\text{ReLU}\left(-(M_\theta - \gamma) \right) \right], \tag{6}$$

where $\gamma \in [0, 1]$ is the sole hyperparameter representing the *target reward margin*.

Gradient behavior investigation. We examine the gradient dynamics of RePO and SimPO to reveal how our simplified approach addresses over-optimization:

$$\nabla_\theta \mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\beta \mathbb{E}_{\mathcal{D}} \left[s_\theta \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right], \tag{7}$$

$$\nabla_\theta \mathcal{L}_{\text{RePO}}(\pi_\theta) = -\mathbb{E}_{\mathcal{D}} \left[\mathbb{I}(M_\theta < \gamma) \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right], \tag{8}$$

where $s_\theta = \sigma(\beta(-M_\theta + \gamma))$ is SimPO's sigmoid weighting function. The terms $\nabla_{\theta, y_w} = \frac{1}{|y_w|} \nabla_\theta \log \pi_\theta(y_w \mid x)$ and $\nabla_{\theta, y_l} = \frac{1}{|y_l|} \nabla_\theta \log \pi_\theta(y_l \mid x)$ correspond to the gradients that increase the probability of the "winning" response y_w and decrease the probability of the "losing" response y_l, respectively. The scaling factor β in Equation 7 linearly amplifies gradient magnitudes but does not alter the relative update directions in adaptive optimizers like Adam [13], as the momentum terms automatically normalize scale variations. We therefore omit β in Figure 2 for clearer visualization of the weighting function shapes.

The key insight is that RePO's ReLU-based gradient (Equation 8) applies uniform updates only to pairs with $M_\theta < \gamma$, while SimPO's gradient (Equation 7) uses continuous β-scaled weights. Figure 2 visualizes this difference, showing RePO as the limiting case of SimPO as $\beta \to \infty$.

Lemma 3.1 (Gradient Equivalence in the SimPO-to-RePO Limit)**.** *Under the same M_θ and γ definitions, the SimPO gradient converges pointwise to the RePO gradient as $\beta \to \infty$:*

$$\lim_{\beta \to \infty} \nabla_\theta \mathcal{L}_{SimPO} = \nabla_\theta \mathcal{L}_{RePO}. \tag{9}$$

Sketch. The convergence follows from the pointwise limit of the sigmoid weighting:

$$\lim_{\beta \to \infty} s_\theta = \lim_{\beta \to \infty} \sigma(\beta(-M_\theta + \gamma)) = \mathbb{I}(M_\theta < \gamma).$$

Substituting this into Equation 7 yields Equation 8. □

Remark 3.2. Please check Appendix for all proofs. Lemma 3.1 establishes RePO as the asymptotic limit of SimPO with large β, explaining two key advantages we will demonstrate in Section 3.2: comparable performance without β tuning complexity, and an effective binary thresholding mechanism that induces implicit data filtering for controlling over-optimization.

3.2 Empirical Study

The previous section analyzes the relationship between SimPO and RePO from the perspective of gradient behavior. In this section, we compare their performance from an empirical standpoint.

Experimental setup. We evaluate this approach using SimPO's experimental setup [11] with Llama3-8B and Gemma2-9B models (Instruct setup). For consistency, we use the same training datasets as SimPO: princeton-nlp/llama3-ultrafeedback-armorm for Llama3-8B and princeton-nlp/gemma2-ultrafeedback-armorm for Gemma2-9B. For all SimPO experiments, we set $\beta = 10.0$ and $\gamma = 0.4$ for Gemma2-9B and $\beta = 10.0$ and $\gamma = 0.3$ for Llama3-8B, unless otherwise specified. We track optimization progress using two reward margin metrics:

$$m_{\text{batch}} = \mathbb{E}_{(x, y_w, y_l) \in \mathcal{B}}[M_\theta], \quad m_{\mathcal{D}} = \mathbb{E}_{(x, y_w, y_l) \in \mathcal{D}}[M_\theta], \tag{10}$$

measuring response separation within each batch (m_{batch}) and across the entire training set ($m_{\mathcal{D}}$).

Evaluation benchmarks. We evaluate on two established benchmarks for open-ended generation: AlpacaEval 2 [14] (measuring instruction-following quality against GPT-4) and Arena-Hard [15] (testing complex reasoning). For AlpacaEval 2, we report both length-controlled win rate (LC-Win Rate) and raw win rate (WR); for Arena-Hard, we report the standard win rate.

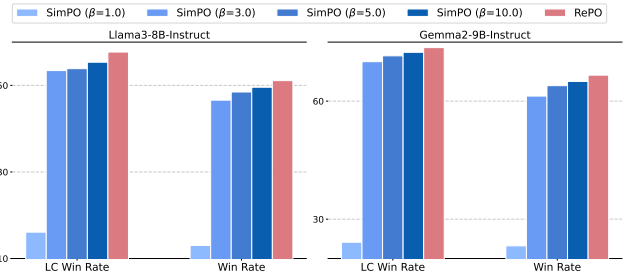

Figure 3: Performance of SimPO with varying β and RePO on AlpacaEval2 benchmark.

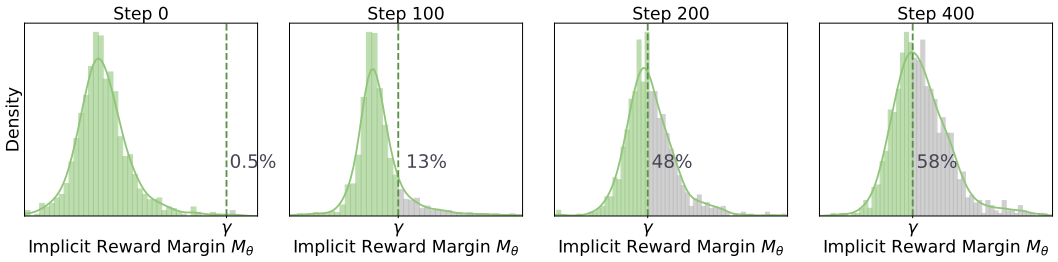

Figure 4: Implicit reward margin M_θ distribution across training steps (total: 467) for RePO at $\gamma = 0.4$. Dashed line: $\gamma = 0.4$. Green: samples below γ (gradient descent); gray: samples above γ (zero gradient). Numbers: fraction of samples above γ.

Observation 1: Large β enhances SimPO's performance when paired with appropriate γ. We systematically evaluate SimPO across varying values of $\beta \in \{1.0, 3.0, 5.0, 10.0\}$, while maintaining fixed γ values that we empirically determined to be suitable for each model architecture ($\gamma = 0.4$ for Gemma2-9B and $\gamma = 0.3$ for Llama3-8B). As shown in Figure 3, increasing β leads to consistent performance improvements across all evaluation metrics, with diminishing returns observed beyond $\beta = 5.0$. These findings align with observations in the SimPO paper[3].

Observation 2: RePO matches high-β SimPO. RePO achieves performance comparable to SimPO with a large β. As shown in Figure 3, RePO achieves win rates of 51.1% on Llama3-8B and 66.6% on Gemma2-9B, comparable to SimPO's performance. This aligns with Lemma 3.1, which establishes that RePO can be interpreted as a limiting case of SimPO as $\beta \to \infty$.

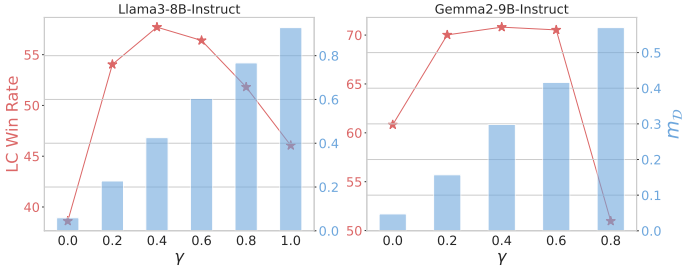

Figure 5: Line plot of RePO performance (AlpacaEval 2 LC Win Rate) and bar chart of mean reward margins ($m_\mathcal{D}$) across varying γ values. See Appendix D.3 for details.

Observation 3: γ threshold creates a natural alignment-optimization tradeoff. Our experiments track the mean implicit reward margin $m_\mathcal{D}$ (*cf.* Equation 10) across training pairs. As Figure 5 illustrates, increasing γ directly elevates $m_\mathcal{D}$ while performance follows an inverted U-shaped pattern — improving initially but declining beyond a critical threshold. In RePO, gradients vanish when the implicit reward margin exceeds γ, effectively filtering out well-separated pairs from updates. This mechanism creates a fundamental tradeoff: small γ values retain excessive zero-gradient samples causing under-filtering, while large γ values force updates on most samples, potentially leading to over-optimization [8] and ultimately degrading performance.

Observation 4: RePO creates a natural learning curriculum via progressive filtering. Figure 4 reveals an unexpected pattern in how the distribution of implicit reward margins M_θ evolves throughout training. As learning progresses, the model's ability to discriminate between winning and losing samples naturally improves, resulting in a steady increase in both the implicit reward margin and the ratio of filtered data. Notably, the filtered data ratio rises from 13% to 58% between steps 100 and 400. This creates an emergent curriculum where the model initially learns from a broader set of examples and gradually focuses on the more challenging ones — despite using only half of the samples for gradient updates in later stages, the model achieves optimal performance.

3.3 Over-Optimization Analysis

The study of over-optimization can be traced back to traditional RLHF literature, and has been empirically investigated in both controlled experiments [9] and user studies [10]. In this work, we follow their experimental setup to further explore this phenomenon.

[3] In their official repository, the authors note: *"SimPO requires a much larger β than DPO... In many cases, an even larger* (e.g., *10) could yield better results."*

Model Over-Optimization: Building on Rafailov et al. [16], we investigate over-optimization in RePO, by evaluating six different values of γ $(0.0, 0.2, 0.4, 0.6, 0.8, 1.0)$, each corresponding to varying levels of data filtering. Across all cases, we observe a distinct hump-shaped performance pattern: while moderate filtering improves alignment, excessive filtering causes performance to degrade, highlighting the over-optimization effect.

Scaling Law Fits. Previous work [9, 16] has established scaling laws for reward model scores as a function of the KL divergence between the initial and optimized policies. In contrast, we eliminate the reference model and the associated computational cost of calculating KL divergence. Instead, we use the mean implicit reward margin during training as a proxy metric. The reward function $R(d)$ is given by:

$$R(d) = d(\alpha - \beta \log d), \qquad (11)$$

where α and β are constants dependent on the reward model's dataset size and parameter count, and $d = m_{\text{batch}}$. Without training a proxy reward model, we substitute GPT-4 win rates over dataset completions for the gold reward. Interestingly, we find that this scaling law provides an accurate relationship between d and win rates for RePO.

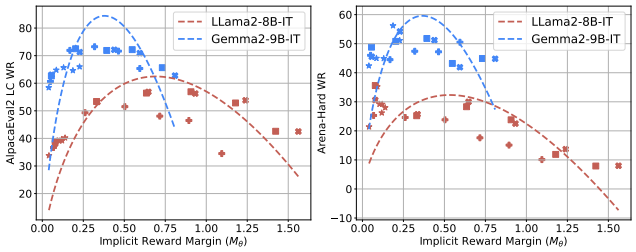

Figure 6: Over-optimization patterns for RePO on Llama3-8B-IT and Gemma2-9B-IT, using AlpacaEval2 LC win rates and Arena-Hard raw win rates. Dotted curves represent theoretical fits based on Gao et al. [9]'s scaling laws, using GPT-4 win rates instead of standard reward model scores.

3.4 RePO++: Exploring Extensions of ReLU-based Filtering

While exploring ReLU's thresholding behavior, we observed an interesting limitation: for cases where the implicit reward margin is smaller than γ, their gradient weights become uniform, not differentiating between samples of varying difficulty.

This observation naturally led us to wonder: could we preserve the effective filtering mechanism while reintroducing some degree of weighting? To explore this question, we experimented with combining ReLU's binary filtering with SimPO's continuous weighting:

$$\mathcal{L}_{\text{RePO++}}(\pi_\theta) = -\mathbb{E}_{\mathcal{D}}\left[\log \sigma\left(-\text{ReLU}\left(-\beta\left(M_\theta - \gamma\right)\right)\right)\right], \qquad (12)$$

This exploration was a natural follow-up to our main discovery about ReLU's effectiveness, rather than our primary contribution. We were curious to see whether combining the best aspects of both approaches might yield additional insights about preference learning mechanisms.

What does this combined approach reveal? To understand the behavior of this extension, we examined its gradient with respect to the parameters θ:

$$\nabla_\theta \mathcal{L}_{\text{RePO++}}(\pi_\theta) = -\beta \mathbb{E}_{\mathcal{D}}\left[s_\theta \cdot \mathbb{I}_\theta \cdot \left(\nabla_{\theta, y_w} - \nabla_{\theta, y_l}\right)\right], \qquad (13)$$

where $s_\theta = \sigma\left(\beta\left(-M_\theta + \gamma\right)\right)$ and $\mathbb{I}_\theta$ is an indicator function that is 1 if $M_\theta < \gamma$ and 0 otherwise.

We observed that this gradient combines properties we discovered in both approaches: it scales updates by s_θ (similar to SimPO) and filters them using $\mathbb{I}_\theta$ (the key discovery in our ReLU exploration), focusing the model on less-separated pairs while giving higher weights to smaller separations.

Adaptation with RePO++. The core contribution of RePO++ lies in leveraging ReLU to mitigate over-optimization while preserving the standard workflow of preference optimization. This makes RePO++ easily adaptable to existing DPO-like methods. For instance, as shown in Equation 12, replacing M_θ with $\log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)} - \log \frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_l|x)}$ seamlessly integrates RePO++ into DPO, forming a ReLU-enhanced version of DPO.

4 Theoretical Analysis: ReLU's Optimality in Preference Learning

Next, we establish a surprising theoretical connection between preference optimization and binary classification, revealing why our simple ReLU-based approach achieves superior performance.

Following Tang et al. [17], preference learning can be reformulated as binary classification. Given pairs (z, l) where $z \in \mathbb{R}^k$ and $l \in \{-1, 1\}$, we aim to learn a predictor $\hat{\ell}(z)$ whose sign matches l. The classification accuracy is: $\frac{1}{2}\mathbb{E}[\text{sign}(\hat{\ell}(z) \cdot l)] + \frac{1}{2}$. This corresponds to minimizing the 0-1 loss:

$$\mathcal{L}_{0\text{-}1}(\hat{\ell}) := \mathbb{E}\left[1 - \text{sign}(\hat{\ell}(z) \cdot l)\right] \tag{14}$$

For preference data (y_w, y_l) where $y_w \succ y_l$, we set $l = 1$ and parameterize $\hat{\ell}(y_w, y_l) = r_\phi(y_w) - r_\phi(y_l)$, yielding the objective:

$$\mathcal{L}_f(\hat{\ell}) := \mathbb{E}\left[f(r_\phi(y_w) - r_\phi(y_l))\right] \tag{15}$$

Where f determines the surrogate loss: $f(x) = \mathbb{I}(x < 0)$ gives the 0-1 loss, $f(x) = -\log \sigma(x)$ yields SimPO's logistic loss, and $f(x) = \text{ReLU}(-x)$ gives our method's loss.

Our key insight comes from analyzing the convex envelope of the 0-1 loss:

Definition 4.1. The convex envelope of $\mathcal{L}_{0\text{-}1}$ over a closed convex set $D \subseteq \mathbb{R}$ is:

$$\text{conv}_D\mathcal{L}_{0\text{-}1}(x) := \sup\{h(x) \mid h \text{ is convex}, h \leq \mathcal{L}_{0\text{-}1}\, \forall x \in D\} \tag{16}$$

Theorem 4.2 (ReLU as Convex Envelope)**.** *For $D = [-a, b]$ with $a, b > 0$, the convex envelope of $\mathcal{L}_{0\text{-}1}(x) = \mathbb{I}(x < 0)$ is:*

$$\text{conv}_D\mathcal{L}_{0\text{-}1}(x) = \frac{1}{a}\text{ReLU}(-x) \tag{17}$$

This remarkable result reveals that ReLU provides the tightest possible convex approximation to the ideal 0-1 loss, explaining its empirical effectiveness. Furthermore:

Corollary 4.3 (Optimality Preservation)**.** *Let $D \subseteq \mathbb{R}$ be convex. Then:*

$$\arg\min_{\hat{\ell}} \mathcal{L}_{0\text{-}1}(\hat{\ell}) = \arg\min_{\hat{\ell}} \text{conv}_D\mathcal{L}_{0\text{-}1}(\hat{\ell}) \tag{18}$$

And for $D = [-a, b]$:

$$\arg\min_{x \in D} \mathcal{L}_{0\text{-}1}(x) = \arg\min_{x \in D} \frac{1}{a}\text{ReLU}(-x) \tag{19}$$

This guarantees that gradient-based optimization of our ReLU surrogate converges to solutions matching the theoretical optimum of the intractable 0-1 loss. Importantly:

Corollary 4.4 (Logistic Loss Suboptimality)**.** *The logistic loss $f_{\log-\text{sigmoid}}(x) = -\log \sigma(x)$ is not the convex envelope of $\mathcal{L}_{0\text{-}1}$.*

This theoretical foundation explains why our simple ReLU-based approach consistently outperforms more complex mechanisms like SimPO's sigmoid weighting — ReLU provides optimality guarantees that logistic loss cannot match, while being computationally more efficient.

5 Experiments

In this section, we examine how our simplified ReLU-based approach behaves across different models and settings. Rather than focusing solely on performance gains, we explore patterns that help explain why such a simple mechanism works effectively in practice.

5.1 Experimental Setup

The core experimental configuration extends our investigation from Section 3.2 to include Mistral2-7B [18] alongside previously examined models. For the Llama3-Instruct v0.2 experiments, we employed the RLHFlow/ArmoRM-Llama3-8B-v0.1 [19] reward model for ranking generated data. We benchmark our approach against established preference optimization methods: DPO [6], SimPO [11], IPO [20], CPO [21], KTO [22], ORPO [23], and R-DPO [24], with SFT models serving as baselines. Implementation details are provided in Appendix D.1. We also evaluate on downstream tasks from the Huggingface Open Leaderboard benchmarks [25], with additional details in in Appendix D.2. The code is available at https://github.com/junkangwu/RePO.

Table 1: **AlpacaEval 2 (AE2), Arena-Hard (AH) results across four settings.** "WR" denotes the raw win rate,"LC" the length-controlled win rate. The best results are highlighted in bold, while the second-best are underlined.

Method	Llama3-Instruct (8B)			Mistral-Instruct (7B)			Llama3-Instruct v0.2 (8B)			Gemma2-Instruct (9B)		
	AE 2		AH	AE 2		AH	AE 2		AH	AE 2		AH
	LC	WR	WR	LC	WR	WR	LC	WR	WR	LC	WR	WR
SFT	24.0	23.6	22.4	19.0	15.4	12.9	24.0	23.6	22.4	48.7	36.5	42.1
SLiC-HF	26.9	27.5	26.2	24.1	24.6	18.9	33.9	32.5	29.3	65.1	60.5	53.7
DPO	40.2	38.1	31.2	20.3	17.9	13.4	48.2	47.5	**35.2**	70.4	**66.9**	58.8
IPO	35.9	34.4	30.2	22.3	18.6	16.2	40.6	39.6	34.9	62.6	58.4	53.5
CPO	29.6	34.4	29.4	26.2	31.7	**23.8**	36.5	40.8	34.2	56.4	53.4	55.2
KTO	38.3	34.1	30.3	19.4	20.3	16.8	41.4	36.4	28.9	61.7	55.5	53.8
ORPO	31.6	29.8	26.3	24.0	23.0	18.6	36.5	33.1	30.4	56.2	46.7	46.2
R-DPO	40.3	37.3	32.9	21.4	22.2	13.8	51.6	50.7	35.0	68.3	**66.9**	57.9
SimPO	43.8	38.0	32.6	30.2	32.1	20.1	55.6	49.6	33.6	72.4	65.0	57.8
RePO	**46.7**	**41.1**	**33.3**	**30.4**	**33.6**	20.3	**57.7**	**51.1**	**35.2**	**73.6**	66.6	**59.1**

5.2 Result Comparisons

Observation: Simple ReLU thresholding exhibits surprising effectiveness. Table 1 reveals an unexpected pattern: despite removing components previously thought essential, the simple ReLU-based approach consistently performs well across all evaluated models and benchmarks. This finding aligns with our theoretical analysis showing that binary thresholding directly approximates the convex envelope of the 0-1 loss. On AlpacaEval 2, we observe improvements of 0.2-2.8 points in LC win rates across different configurations compared to the strongest baselines.

5.3 Methodology Comparisons

Beyond alignment, we also compare the methodologies of these preference learning methods. Our method plays a hub to connect these methods.

Relation to SimPO. SimPO employs sigmoid weighting via log-sigmoid activation to attenuate gradients as reward margins increase, mitigating over-optimization. RePO can be viewed as "SimPO without log-sigmoid," replacing this continuous scaling with binary filtering. To validate this relationship, we integrated a ReLU-based filtering mechanism into SimPO (*cf.* RePO++ Equation 12). Table 2 confirms that ReLU's filtering mechanism enhances performance. As demonstrated in Section 3.4, RePO++ directly addresses over-optimization while retaining the benefits of some weighting.

Relation to SLiC-HF. RePO can be characterized as "SLiC-HF without SFT regularization". To ensure a fair comparison (while disregarding differences in length normalization), we investigated the impact of SFT regularization by varying its coefficient, λ. The results, presented in Appendix Table 6 and further details in Appendix D.5, indicate that this additional regularization term offers no discernible improvement. This suggests that SFT regularization targets a different optimization challenge, distinct from the direct over-optimization problem RePO addresses.

Relation to DPO. Mathematically, DPO is equivalent to SimPO when the margin γ is defined as $\log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x)$ (ignoring length normalization). However, directly substituting the log-sigmoid function with ReLU in DPO's formulation leads to a significant performance degradation (see Appendix Table 7 and Appendix D.6). This underscores the critical role of the threshold γ in determining the effectiveness of over-optimization prevention. As identified by Wu et al. [26], reference model based reward margins are often unreliable as target margins, which explains why SimPO's explicit γ parameter is effective for preference learning.

5.4 Effect of ReLU Filtering Across Methods

Having observed the effectiveness of binary thresholding, we naturally questioned whether this mechanism might enhance other preference learning approaches. Table 2 shows that integrating ReLU filtering consistently improved performance across both DPO and SimPO frameworks, suggesting that selective gradient application based on margin thresholds provides benefits beyond our specific implementation. Our experiments with the combined approach (*cf.* RePO++ in Section 3.4) revealed

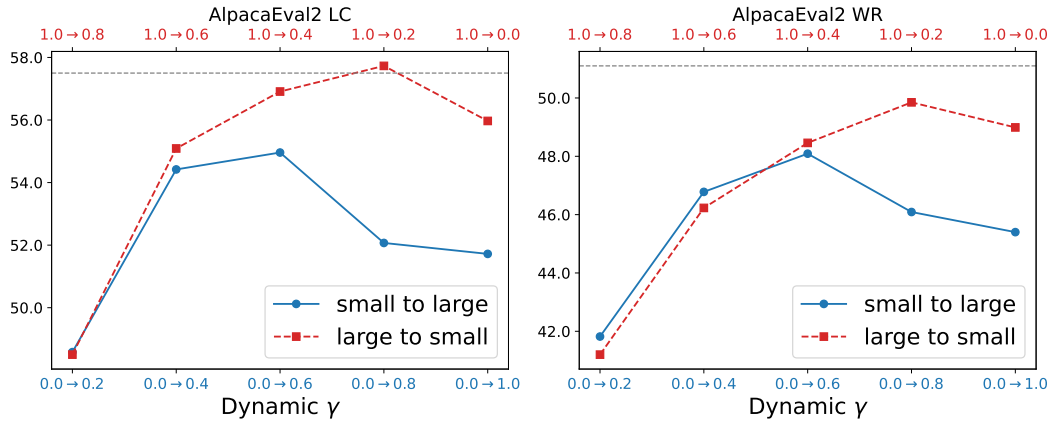

Figure 7: Exploration of dynamic γ scheduling on Llama3-Instruct v0.2 (8B). The dashed line represents performance with a fixed γ. We observed that decreasing γ from an initially larger value creates a natural curriculum that enhances performance.

particularly strong improvements when applied to DPO (5%–12% gains), with notable performance on Arena-Hard (reaching 65.7). This design effectively mitigates over-optimization while preserving the benefits of the original scheme.

5.5 Dynamic Margin Scheduling and Curriculum Learning

Our investigation into the role of target reward margin γ led us to an unexpected discovery about curriculum learning. We experimented with dynamic scheduling of γ throughout training, implementing two strategies: (i) increasing γ from small to large, and (ii) decreasing γ from large to small. Figure 7 reveals a striking pattern: starting with a moderately large value of γ and gradually decreasing it ($1.0 \rightarrow 0.2$) naturally creates an effective curriculum that improves model performance. In contrast, both excessively large values ($1.0 \rightarrow 0.8$) and small values ($0.0 \rightarrow 0.2$) led to suboptimal outcomes.

Table 2: **Performance improvements of RePO and RePO++ over DPO and SimPO.** Results are present on AlpacaEval 2 (AE 2) and Arena-Hard (AH) with LC (%) and WR (%). Red numbers indicate relative improvements.

Method	Llama3-Instruct v0.2 (8B)			Gemma2-Instruct (9B)		
	AE 2		AH	AE 2		AH
	LC	WR	WR	LC	WR	WR
DPO	48.2	47.5	35.2	70.4	66.9	58.8
w. RePO	$50.3^{+4.4\%}$	$51.8^{+9.1\%}$	$38.2^{+8.5\%}$	$73.8^{+4.8\%}$	$71.0^{+6.1\%}$	$64.2^{+9.2\%}$
w. RePO++	$50.8^{+5.4\%}$	$52.2^{+9.9\%}$	$37.2^{+5.7\%}$	$71.8^{+2.0\%}$	$69.5^{+3.9\%}$	$65.7^{+11.7\%}$
SimPO	55.6	49.6	33.6	72.4	65.0	57.8
w. RePO	$57.7^{+3.8\%}$	$51.1^{+3.0\%}$	$35.2^{+4.8\%}$	$73.6^{+1.7\%}$	$66.6^{+2.5\%}$	$59.1^{+2.2\%}$
w. RePO++	$56.1^{+0.9\%}$	$50.1^{+1.0\%}$	$35.9^{+6.8\%}$	$74.1^{+2.3\%}$	$66.5^{+2.3\%}$	$59.8^{+3.5\%}$

This observation reveals an intriguing self-regulating property: early in training when the model is underfitting, a larger γ permits more aggressive updates across more examples. As training progresses, the decreasing γ naturally focuses learning on increasingly challenging examples, effectively preventing over-optimization. This emergent curriculum behavior, arising from a simple parameter schedule, suggests that binary thresholding captures fundamental learning dynamics that more complex mechanisms might obscure.

6 Discussion

Conclusion Our exploration of simple ReLU activation in preference learning has revealed several key insights. We found that binary thresholding, implemented through a straightforward ReLU function, provides an effective mechanism for preventing over-optimization in language model alignment. Our theoretical analysis showed that this seemingly simple approach is, in fact, the convex envelope of the ideal 0-1 loss function, explaining its surprising effectiveness. Rather than developing yet another complex preference optimization method, our work uncovered how fundamental properties like data selection and implicit curriculum learning emerge naturally from basic principles.

Limitations and future directions. Our current exploration is limited to offline preference learning settings. Future work could investigate how these insights might extend to online learning scenarios,

where preferences are gathered interactively. Additionally, while we found that a fixed margin threshold works well in practice, exploring adaptive or context-aware thresholds might further improve performance in highly dynamic environments. The relationship between binary filtering and self-play scenarios [27] — where the model generates its own feedback — is another promising direction that could lead to more scalable alignment techniques.

Beyond alignment, our work connects to LLM reasoning research [28, 29]. Future work should investigate how KL penalties and gradient clipping in GRPO [30] and PPO [5] balance preventing over-optimization against preserving reasoning capabilities — a critical consideration for advancing alignment methodologies.

Acknowledgments and Disclosure of Funding

This research was supported by the National Science and Technology Major Project (2023ZD0121102), the National Natural Science Foundation of China (U24B20180, 62302321), and the Fundamental Research Funds for the Central Universities (WK2100250065). This research also benefited from the advanced computing resources provided by the Supercomputing Center of the USTC.

References

[1] Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul Ronald Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, and et al. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023.

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023.

[3] OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023.

[4] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Túlio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *CoRR*, abs/2303.12712, 2023.

[5] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

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

[7] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. Slic-hf: Sequence likelihood calibration with human feedback. *CoRR*, abs/2305.10425, 2023.

[8] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016.

[9] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *ICML*, volume 202 of *Proceedings of Machine Learning Research*, pages 10835–10866. PMLR, 2023.

[10] Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In *NeurIPS*, 2023.

[11] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *NeurIPS*, 2024.

[12] Paria Rashidinejad and Yuandong Tian. Sail into the headwind: Alignment via robust rewards and dynamic labels against reward hacking. *ICLR*, 2025.

[13] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[15] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The Arena-Hard pipeline, April 2024. URL https://lmsys.org/blog/2024-04-19-arena-hard/.

[16] Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. *CoRR*, abs/2406.02900, 2024.

[17] Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. In *ICML*. OpenReview.net, 2024.

[18] Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *ArXiv*, abs/2310.06825, 2023.

[19] Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *ArXiv*, abs/2406.12845, 2024.

[20] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *ArXiv*, abs/2310.12036, 2023.

[21] Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. *ArXiv*, abs/2401.08417, 2024.

[22] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306, 2024.

[23] Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. *ArXiv*, abs/2403.07691, 2024.

[24] Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *ArXiv*, abs/2403.19159, 2024.

[25] Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. Open LLM leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard, 2023.

[26] Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. α-dpo: Adaptive reward margin is what direct preference optimization needs. *CoRR*, abs/2410.10148, 2024.

[27] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *ICML*. OpenReview.net, 2024.

[28] OpenAI. Learning to reason with llms, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

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

[30] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[31] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *NIPS*, pages 4299–4307, 2017.

[32] Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul F. Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *ArXiv*, abs/1909.08593, 2019.

[33] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022.

[34] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. LIMA: Less is more for alignment. *NeurIPS*, 2023.

[35] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model, 2023.

[36] Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. Koala: A dialogue model for academic research. *Blog post, April*, 1:6, 2023.

[37] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 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.

[38] Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Minh Nguyen, Oliver Stanley, Richárd Nagyfi, et al. Openassistant conversations-democratizing large language model alignment. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023.

[39] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In *EMNLP*, 2023.

[40] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[41] Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. ODIN: Disentangled reward mitigates hacking in RLHF. *arXiv preprint arXiv:2402.07319*, 2024.

[42] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[43] Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Railneau. GLoRe: When, where, and how to improve LLM reasoning via global and local refinements. *arXiv preprint arXiv:2402.10963*, 2024.

[44] Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. RewardBench: Evaluating reward models for language modeling. *ArXiv*, abs/2403.13787, 2024.

[45] Thomas Anthony, Zheng Tian, and David Barber. Thinking fast and slow with deep learning and tree search. *Advances in neural information processing systems*, 30, 2017.

[46] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *ACL (1)*, pages 12248–12267. Association for Computational Linguistics, 2024.

[47] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024.

[48] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β. *CoRR*, abs/2407.08639, 2024.

[49] Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust DPO: aligning language models with noisy feedback. In *ICML*. OpenReview.net, 2024.

[50] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. *CoRR*, abs/2407.07880, 2024.

[51] Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse KL: generalizing direct preference optimization with diverse divergence constraints. In *ICLR*. OpenReview.net, 2024.

[52] Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D. Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J. Foster. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *CoRR*, abs/2407.13399, 2024.

[53] Saeed Khaki, JinJin Li, Lan Ma, Liu Yang, and Prathap Ramachandra. Rs-dpo: A hybrid rejection sampling and direct preference optimization method for alignment of large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1665–1680, 2024.

[54] Ziyu Ye, Rishabh Agarwal, Tianqi Liu, Rishabh Joshi, Sarmishta Velury, Quoc V Le, Qijun Tan, and Yuan Liu. Evolving alignment via asymmetric self-play. *arXiv preprint arXiv:2411.00062*, 2024.

[55] Shiqi Wang, Zhengze Zhang, Rui Zhao, Fei Tan, and Nguyen Cam-Tu. Reward difference optimization for sample reweighting in offline rlhf. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2109–2123, 2024.

[56] William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. In *Forty-first International Conference on Machine Learning*, 2024.

[57] Sen Yang, Leyang Cui, Deng Cai, Xinting Huang, Shuming Shi, and Wai Lam. Not all preference pairs are created equal: A recipe for annotation-efficient iterative preference learning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6549–6561, 2024.

[58] Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. RLHF workflow: From reward modeling to online RLHF. *CoRR*, abs/2405.07863, 2024.

[59] Dahyun Kim, Yungi Kim, Wonho Song, Hyeonwoo Kim, Yunsu Kim, Sanghoon Kim, and Chanjun Park. sdpo: Don't use your data all at once. *CoRR*, abs/2403.19270, 2024.

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

[61] Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under kl-constraint. In *ICML*. OpenReview.net, 2024.

[62] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *ICML*. OpenReview.net, 2024.

[63] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *CoRR*, abs/2405.00675, 2024.

[64] Zhaolin Gao, Jonathan D. Chang, Wenhao Zhan, Owen Oertell, Gokul Swamy, Kianté Brantley, Thorsten Joachims, J. Andrew Bagnell, Jason D. Lee, and Wen Sun. REBEL: reinforcement learning via regressing relative rewards. *CoRR*, abs/2404.16767, 2024.

[65] Daniele Calandriello, Zhaohan Daniel Guo, Rémi Munos, Mark Rowland, Yunhao Tang, Bernardo Ávila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, Rishabh Joshi, Zeyu Zheng, and Bilal Piot. Human alignment of large language models through online preference optimisation. In *ICML*. OpenReview.net, 2024.

[66] Princeton University. ORF 523 Lecture 8: Online Convex Optimization, Accessed 2025. URL https://www.princeton.edu/~aaa/Public/Teaching/ORF523/ORF523_Lec8.pdf. Accessed: 31 Jan 2025.

[67] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2020.

[68] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018.

[69] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

[70] Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, pages 3214–3252, 2022.

[71] Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

[72] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

A Related Works

Reinforcement learning from human feedback. RLHF is a technique designed to align large language models with human preferences and values [31–33, 20]. Traditional RLHF is typically structured in three stages: supervised fine-tuning [34–39], reward modeling [9, 40–44], and policy optimization [5, 45]. In the third stage, Proximal Policy Optimization (PPO) is widely adopted. In contrast, RLOO [46] reduces the GPU memory footprint of RLHF by eliminating the Critic model and leverages a Leave-One-Out strategy to achieve superior performance. GRPO [47], another variant of PPO, improves mathematical reasoning abilities while optimizing memory usage by replacing the Leave-One-Out method with a direct subtraction of the mean of all samples for a given prompt.

Offline preference optimization. In addition to DPO, several alternative preference optimization objectives have been proposed. IPO [20] addresses overfitting issues inherent in DPO. ORPO [23] and SimPO [11] aim to remove reliance on a reference model. R-DPO [24] targets the reduction of exploitation due to sequence length, while KTO [22] handles preference optimization in the absence of pairwise data. CPO [21] and β-DPO [48] focus on improving the quality of preference data. Another research direction addresses noise in offline alignment, which arises from the need to construct pairwise data. rDPO [49], a variant of DPO, mitigates preference noise and enhances policy robustness, while DrDPO [50] applies distributed robust optimization to tackle this issue. Other works have approached the problem through divergence regularization [51, 52], selection of high-quality data [53, 54], or reweighting loss functions [55–57].

Iterative Preference Optimization. Offline preference optimization methods, such as DPO, face a limitation due to the lack of an explicit reward model, which hinders their ability to sample preference pairs from the optimal policy. To address this, iterative preference optimization techniques have been proposed. These methods iteratively update the reference model using the most recent policy model or generate new preference pairs in each iteration [58–62, 27, 63, 64]. For instance, SPIN [27] employs a self-play framework to fine-tune the model in a supervised manner, while Yuan et al. [62] annotate preferences throughout the iterative process. REBEL improves sample quality by regressing the relative reward. Additionally, [65] generates data using a mixture policy, similar to the Nash-MD algorithm [60].

B Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here

C Proofs

C.1 Proof of Lemma 3.1

Lemma 3.1 (Gradient Equivalence in the SimPO-to-RePO Limit)**.** *Under the same M_θ and γ definitions, the SimPO gradient converges pointwise to the RePO gradient as $\beta \to \infty$:*

$$\lim_{\beta \to \infty} \nabla_\theta \mathcal{L}_{SimPO} = \nabla_\theta \mathcal{L}_{RePO}. \tag{9}$$

Proof. We formally establish the gradient equivalence through pointwise convergence analysis. Let $\mathcal{D}$ be the data distribution and θ denote model parameters. Recall the gradient expressions:

SimPO Gradient:

$$\nabla_\theta \mathcal{L}_{\text{SimPO}} = -\beta \mathbb{E}_{\mathcal{D}} \left[\sigma\big(\beta(-M_\theta + \gamma)\big) \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right] \tag{20}$$

RePO Gradient:

$$\nabla_\theta \mathcal{L}_{\text{RePO}} = -\mathbb{E}_{\mathcal{D}} \left[\mathbb{I}(M_\theta < \gamma) \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right] \tag{21}$$

where $\sigma(\cdot)$ is the sigmoid function. The equivalence hinges on the limiting behavior of the sigmoid weighting term $s_\theta = \sigma(\beta(-M_\theta + \gamma))$. We analyze three cases:

Case 1: $M_\theta < \gamma$ Here, $-M_\theta + \gamma > 0$. As $\beta \to \infty$,

$$\lim_{\beta \to \infty} \sigma\big(\beta(-M_\theta + \gamma)\big) = \lim_{z \to \infty} \sigma(z) = 1 = \mathbb{I}(M_\theta < \gamma).$$

Case 2: $M_\theta > \gamma$ Here, $-M_\theta + \gamma < 0$. As $\beta \to \infty$,

$$\lim_{\beta \to \infty} \sigma\big(\beta(-M_\theta + \gamma)\big) = \lim_{z \to -\infty} \sigma(z) = 0 = \mathbb{I}(M_\theta < \gamma).$$

Case 3: $M_\theta = \gamma$ This occurs on a measure-zero set under continuous distributions. The limit becomes:

$$\lim_{\beta \to \infty} \sigma(0) = \frac{1}{2} \neq \mathbb{I}(M_\theta < \gamma),$$

which is negligible in expectation.

Thus, $\lim_{\beta \to \infty} s_\theta = \mathbb{I}(M_\theta < \gamma)$ almost everywhere. Substituting this into the SimPO gradient:

$$\lim_{\beta \to \infty} \nabla_\theta \mathcal{L}_{\text{SimPO}} = -\lim_{\beta \to \infty} \beta \mathbb{E}_{\mathcal{D}} \left[s_\theta \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right] \tag{22}$$

$$= -\mathbb{E}_{\mathcal{D}} \left[\lim_{\beta \to \infty} \beta s_\theta \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right] \quad \text{(Dominated Convergence Theorem)} \tag{23}$$

To resolve the β scaling, observe that for $M_\theta \neq \gamma$:

$$\lim_{\beta \to \infty} \beta s_\theta = \begin{cases} \lim_{\beta \to \infty} \beta \cdot 1 = \infty & \text{if } M_\theta < \gamma \\ \lim_{\beta \to \infty} \beta \cdot 0 = 0 & \text{if } M_\theta > \gamma \end{cases} \tag{24}$$

The divergence when $M_\theta < \gamma$ is mitigated by adaptive optimizers like Adam, which normalize gradient magnitudes through momentum terms. Formally, let $g_\theta = \nabla_{\theta, y_w} - \nabla_{\theta, y_l}$. Under Adam's update rule:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon},$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected momentum estimates. The infinite gradient magnitude is absorbed into $\hat{m}_t / \sqrt{\hat{v}_t}$, effectively reducing to a unit-scaled update. Hence, in normalized update space:

$$\lim_{\beta \to \infty} \beta s_\theta \cdot g_\theta \propto \mathbb{I}(M_\theta < \gamma) \cdot g_\theta.$$

Combining these results:

$$\lim_{\beta \to \infty} \nabla_\theta \mathcal{L}_{\text{SimPO}} = -\mathbb{E}_{\mathcal{D}} \left[\mathbb{I}(M_\theta < \gamma) \cdot (\nabla_{\theta, y_w} - \nabla_{\theta, y_l}) \right] = \nabla_\theta \mathcal{L}_{\text{RePO}}, \tag{25}$$

which completes the proof. □

C.2 Proof of Theorem 4.2

Theorem 4.2 (ReLU as Convex Envelope)**.** *For $D = [-a, b]$ with $a, b > 0$, the convex envelope of $\mathcal{L}_{0\text{-}1}(x) = \mathbb{I}(x < 0)$ is:*

$$\text{conv}_D \mathcal{L}_{0\text{-}1}(x) = \frac{1}{a}\text{ReLU}(-x) \tag{17}$$

Proof. We demonstrate that $h(x) = \frac{1}{a}\text{ReLU}(-x)$ satisfies the convex envelope definition through three sequential arguments.

1. Convexity and Underestimation: The ReLU function is convex as the pointwise maximum of affine functions (Rule 3). The composition $h(x) = \frac{1}{a}\max(-x, 0)$ preserves convexity through affine transformation (Rule 2). For all $x \in D$:

- When $x < 0$: $h(x) = -\frac{x}{a} \leq 1 = \mathcal{L}_{0\text{-}1}(x)$, since $x \geq -a \implies -\frac{x}{a} \leq 1$
- When $x \geq 0$: $h(x) = 0 = \mathcal{L}_{0\text{-}1}(x)$

Thus $h(x) \leq \mathcal{L}_{0\text{-}1}(x)$ over D.

2. Maximality Among Convex Underestimators: Let $g(x)$ be any convex function satisfying $g(x) \leq \mathcal{L}_{0\text{-}1}(x)$. For $x \in [-a, 0)$, convexity implies:

$$g(x) \leq \frac{-x}{a}g(-a) + \left(1 + \frac{x}{a}\right)g(0) \leq \frac{-x}{a}$$

since $g(-a) \leq 1$ and $g(0) \leq 0$. For $x \geq 0$, $g(x) \leq 0$. Hence $g(x) \leq h(x)$ for all $x \in D$.

3. Epigraph Characterization: The epigraph $\text{epi}(h)$ coincides with the convex hull of $\text{epi}(\mathcal{L}_{0\text{-}1}) \cap (D \times \mathbb{R})$. The affine segment $h(x) = -\frac{x}{a}$ on $[-a, 0)$ connects the points $(-a, 1)$ and $(0, 0)$, forming the tightest convex fit to the 0-1 loss's discontinuity. By Theorem 1 in [66], this construction achieves the convex envelope. □

C.3 Proof of Corollary 4.3

Corollary 4.3 (Optimality Preservation)**.** *Let $D \subseteq \mathbb{R}$ be convex. Then:*

$$\arg\min_{\hat{\ell}} \mathcal{L}_{0\text{-}1}(\hat{\ell}) = \arg\min_{\hat{\ell}} \text{conv}_D \mathcal{L}_{0\text{-}1}(\hat{\ell}) \tag{18}$$

And for $D = [-a, b]$:

$$\arg\min_{x \in D} \mathcal{L}_{0\text{-}1}(x) = \arg\min_{x \in D} \frac{1}{a}\text{ReLU}(-x) \tag{19}$$

Proof. **Part 1:** By Theorem 1 in the lecture notes (Page 5), for any function f and convex set S:

$$\min_{x \in S} f(x) = \min_{x \in S} \text{conv}_S f(x).$$

Let $S = D$ and $f = \mathcal{L}_{0\text{-}1}$. The equality of minima implies:

$$\{x^* \in D \mid \mathcal{L}_{0\text{-}1}(x^*) = \min \mathcal{L}_{0\text{-}1}\} \subseteq \{x^* \in D \mid \text{conv}_D \mathcal{L}_{0\text{-}1}(x^*) = \min \text{conv}_D \mathcal{L}_{0\text{-}1}\}.$$

To show reverse inclusion, suppose $x^* \in \arg\min \text{conv}_D \mathcal{L}_{0\text{-}1}$. Since $\text{conv}_D \mathcal{L}_{0\text{-}1}(x^*) \leq \mathcal{L}_{0\text{-}1}(x^*)$ and $\text{conv}_D \mathcal{L}_{0\text{-}1}$ attains its minimum at the same points as $\mathcal{L}_{0\text{-}1}$, x^* must also minimize $\mathcal{L}_{0\text{-}1}$.

Part 2: For $D = [-a, b]$, both $\mathcal{L}_{0\text{-}1}(x)$ and $\frac{1}{a}\text{ReLU}(-x)$ attain their minimum value 0 on $[0, b]$. For $x \in [-a, 0)$, $\frac{1}{a}\text{ReLU}(-x)$ is strictly decreasing, achieving its minimum at $x = 0$. Thus:

$$\arg\min_{x \in D} \mathcal{L}_{0\text{-}1}(x) = \arg\min_{x \in D} \frac{1}{a}\text{ReLU}(-x) = [0, b].$$

□

C.4 Proof of Corollary 4.4

Corollary 4.4 (Logistic Loss Suboptimality)**.** *The logistic loss* $f_{\log-\text{sigmoid}}(x) = -\log\sigma(x)$ *is not the convex envelope of* $\mathcal{L}_{0\text{-}1}$.

Proof. We demonstrate violation of the convex envelope's defining property. Consider $D = [-1, 1]$:

1. **Underestimation Failure:** For $x > 0$:

$$-\log\sigma(x) = -\log\left(\frac{1}{1+e^{-x}}\right) = \log(1+e^{-x}) > 0 = \mathcal{L}_{0\text{-}1}(x)$$

Thus $f_{\log-\text{sigmoid}} \not\leq \mathcal{L}_{0\text{-}1}$ over D, violating the envelope requirement.

2. **Non-Maximality:** Even if scaled, the logistic loss's curvature differs from the ReLU envelope. For $x \in (-1, 0)$, $\frac{d^2}{dx^2}(-\log\sigma(x)) = \sigma(x)(1-\sigma(x)) > 0$, making it strictly convex – incompatible with the affine structure of $\text{conv}_D\mathcal{L}_{0\text{-}1}$.

Hence $f_{\log-\text{sigmoid}}$ cannot be the convex envelope. □

D Experiments

D.1 Implementation Details

Empirical observations indicate significant performance sensitivity to model parameter initialization and learning rate selection across compared methods. To establish rigorous comparison benchmarks, we conducted systematic hyperparameter searches adhering to the specifications in each method's original publication. The complete search space configuration is documented in Table 3. Notably, substantial architecture updates to both Llama3-8B and Instruct-7B necessitated re-implementation of the SimPO method, as the original implementation became incompatible with the revised model interfaces.

Training Protocol All experiments employed standardized training configurations to ensure comparability:

- Batch size: 128 (consistent across methods)
- Learning rate: Searched in {3e-7, 5e-7, 8e-7, 1e-6}
- Training duration: Single epoch with cosine annealing schedule
- Warmup: 10% of total training steps
- Optimizer: Adam [13] ($\beta_1 = 0.9$, $\beta_2 = 0.999$)
- Sequence length: 2048 tokens (fixed for all inputs)

The learning rate schedule follows a triangular policy with amplitude decay, selected through cross-validation on held-out development sets. All implementations utilize full-precision floating-point arithmetic to prevent gradient quantization artifacts.

Hyperparameters in RePO. Table 4 summarizes the hyperparameters utilized for RePO across different experimental settings. Our methodology only involves one hyperparameter: γ. Based on empirical evidence, we recommend setting γ to a default value of 0.5, as this configuration has consistently demonstrated reliability.

Decoding Hyperparameters. The decoding hyperparameters employed in this study align with those used in SimPO[4]. We express our gratitude to the SimPO team for their generosity in sharing their insights and configurations, which have been instrumental in our work.

Computation Environment. All training experiments described in this paper were conducted using 8×A100 GPUs. The experimental setup follows the guidelines provided in the alignment-handbook repository[5], ensuring reproducibility and consistency with established practices.

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

Method	Objective	Hyperparameter
SLiC-HF [7]	$\max(0, \delta - \log \pi_\theta(y_w\|x) + \log \pi_\theta(y_l\|x)) - \lambda \log \pi_\theta(y_w\|x)$	$\lambda \in [0.1, 0.5, 1.0, 10.0]$ $\delta \in [0.1, 0.5, 1.0, 2.0]$
DPO [6]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta \log \frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$	$\beta \in [0.01, 0.05, 0.1]$
IPO [20]	$\left(\log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log \frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - \frac{1}{2\tau}\right)^2$	$\tau \in [0.01, 0.1, 0.5, 1.0]$
CPO [21]	$-\log \sigma\left(\beta \log \pi_\theta(y_w\|x) - \beta \log \pi_\theta(y_l\|x)\right) - \lambda \log \pi_\theta(y_w\|x)$	$\alpha = 1.0,\ \beta \in [0.01, 0.05, 0.1]$
KTO [22]	$-\lambda_w \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - z_{\text{ref}}\right) + \lambda_l \sigma\left(z_{\text{ref}} - \beta \log \frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$, where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta \text{KL}\left(\pi_\theta(y\|x)\|\pi_{\text{ref}}(y\|x)\right)\right]$	$\lambda_l = \lambda_w = 1.0$ $\beta \in [0.01, 0.05, 0.1]$
ORPO [23]	$-\log p_\theta(y_w\|x) - \lambda \log \sigma\left(\log \frac{p_\theta(y_w\|x)}{1-p_\theta(y_w\|x)} - \log \frac{p_\theta(y_l\|x)}{1-p_\theta(y_l\|x)}\right)$, 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 [24]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta \log \frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - (\alpha\|y_w\| - \alpha\|y_l\|)\right)$	$\alpha \in [0.05, 0.1, 0.5, 1.0]$ $\beta \in [0.01, 0.05, 0.1]$
SimPO [11]	$-\log \sigma\left(\frac{\beta}{\|y_w\|} \log \pi_\theta(y_w\|x) - \frac{\beta}{\|y_l\|} \log \pi_\theta(y_l\|x) - \gamma\right)$	$\beta \in [2.0, 4.0, 6.0, 8.0]$ $\gamma \in [0.3, 0.5, 1.0, 1.2, 1.4, 1.6]$
RePO	$\text{ReLU}[-(\frac{1}{\|y_w\|} \log \pi_\theta(y_w\|x) - \frac{1}{\|y_l\|} \log \pi_\theta(y_l\|x) - \gamma)]$	$\gamma \in [0.2, 0.4, 0.5, 0.6, 0.8]$

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

Setting	γ	Learning rate
Mistral-Instruct	0.4	6e-7
Llama3-Instruct	0.6	1e-6
Llama3-Instruct-v0.2	0.6	1e-6
Gemma2-Instruct	0.4	8e-7

D.2 Downstream Task Evaluation

To assess the impact of RePO on downstream task performance, we evaluate models trained with different preference optimization methods on a diverse set of tasks from the Huggingface Open Leaderboard [25]. The tasks include MMLU [67], ARC [68], HellaSwag [69], TruthfulQA [70], Winogrand [71], and GSM8K [72]. We adhere to standard evaluation protocols and present the results for all models in Table 5.

Overall Performance. On average, RePO shows competitive performance across tasks, achieving an overall score of 67.49 on the Llama3-Instruct model and 70.58 on the Llama3-Instruct v0.2 model. The performance is generally close to that of other preference optimization methods, but it is worth noting that in some cases, it slightly lags behind models like SimPO or DPO, particularly on tasks such as ARC, HellaSwag, and TruthfulQA. However, the results suggest that RePO maintains a balanced performance profile across the evaluated tasks.

General Knowledge and Reasoning. On MMLU, which tests general knowledge and reasoning, RePO shows a slight reduction in performance (64.95 for Llama3-Instruct and 65.00 for Llama3-Instruct v0.2) compared to models such as RRHF and SimPO. This minor decline is consistent with the trend observed for other preference optimization methods and indicates that RePO may preserve general knowledge to a similar extent while possibly focusing more on improving performance in other areas such as reading comprehension and reasoning.

Reading Comprehension and Commonsense Reasoning. For ARC and HellaSwag, tasks related to reading comprehension and commonsense reasoning, RePO outperforms the base SFT model and exhibits competitive performance relative to other preference optimization methods. The Llama3-Instruct v0.2 model with RePO achieves a score of 80.50 on HellaSwag, which is comparable to the best-performing methods. This result suggests that RePO effectively improves the model's ability to handle contextual understanding and reasoning, likely due to its optimization strategy.

[4] https://github.com/princeton-nlp/SimPO/tree/main/eval
[5] https://github.com/huggingface/alignment-handbook

Table 5: Downstream task evaluation results of tasks on the huggingface open leaderboard.

	MMLU (5)	ARC (25)	HellaSwag (10)	TruthfulQA (0)	Winograd (5)	GSM8K (5)	Average
Llama3-Instruct							
SFT	67.06	61.01	78.57	51.66	74.35	68.69	66.89
RRHF	67.20	61.52	79.54	53.76	74.19	66.11	67.05
SLiC-HF	66.41	61.26	78.80	53.23	76.16	66.57	67.07
DPO	66.88	63.99	80.78	59.01	74.66	49.81	65.86
IPO	66.52	61.95	77.90	54.64	73.09	58.23	65.39
CPO	67.05	62.29	78.73	54.01	73.72	67.40	67.20
KTO	66.38	63.57	79.51	58.15	73.40	57.01	66.34
ORPO	66.41	61.01	79.38	54.37	75.77	64.59	66.92
R-DPO	66.74	64.33	80.97	60.32	74.82	43.90	65.18
SimPO	65.63	62.80	78.33	60.70	73.32	50.72	65.25
RePO	64.95	62.03	77.58	60.96	72.93	66.49	67.49
Llama3-Instruct v0.2							
SFT	67.06	61.01	78.57	51.66	74.35	68.69	66.89
RRHF	66.60	63.74	80.98	59.40	76.32	58.68	67.62
SLiC-HF	66.91	61.77	79.17	56.36	76.40	68.23	68.14
DPO	65.57	65.87	79.66	63.08	74.51	73.01	70.28
IPO	66.06	64.85	81.02	57.29	76.72	76.12	70.34
CPO	65.67	62.12	79.63	56.34	77.98	75.28	69.50
KTO	65.99	62.88	79.02	54.66	74.66	76.42	68.94
ORPO	65.75	63.99	79.91	57.02	78.06	75.13	69.98
R-DPO	66.17	65.36	79.98	57.94	75.06	75.36	69.98
SimPO	65.18	67.15	78.04	64.92	73.88	71.34	70.08
RePO	65.00	68.09	80.50	64.38	76.16	69.37	70.58

Truthfulness. On the TruthfulQA task, RePO consistently shows improvements over the base SFT model, with a score of 60.96 for Llama3-Instruct and 64.38 for Llama3-Instruct v0.2. This indicates that RePO helps the model generate more truthful and reliable responses, aligning with trends observed in other preference optimization methods. The improvement in this area is especially notable given the inherent difficulty of this task, which tests the model's ability to avoid generating false information.

Math Performance. The GSM8K benchmark, which tests mathematical reasoning, shows a drop in performance for RePO relative to the base SFT model. Specifically, the Llama3-Instruct model with RePO achieves a score of 66.49, which is lower than other methods such as SimPO or R-DPO, which focus more on improving mathematical reasoning. This drop is consistent with the trend observed across various preference optimization methods and may suggest that RePO is less effective in retaining mathematical reasoning abilities. Further investigation into this issue could provide insights into potential strategies for addressing this gap.

Task-Specific Variability. Overall, RePO exhibits varied performance across tasks. While it performs well in certain areas, such as commonsense reasoning and truthfulness, it lags behind in others, particularly in general knowledge (MMLU) and mathematical reasoning (GSM8K). This variability is in line with the performance trends observed for other preference optimization methods, which often show task-dependent improvements and declines. This suggests that RePO has strengths in some domains, but it may benefit from further refinement to improve performance across all tasks.

D.3 RePO with varying γ

Figure 8 illustrates the effect of the hyperparameter γ on model performance across two evaluation metrics: LC Win Rate and Raw Win Rate. The analysis is conducted on two models, Llama3-8B-Instruct (left) and Gemma2-9B-Instruct (right). The LC Win Rate, shown in red (left y-axis), represents the model's alignment with learned preferences, whereas the Raw Win Rate, shown in blue (right y-axis), evaluates overall ranking performance based on human preference comparisons.

Moderate γ values lead to optimal performance. Moderate values of γ (0.4–0.6) yield the best balance between preference alignment and generalization. Both models achieve their highest LC Win Rate in this range, indicating that preference optimization is most effective when applied at an intermediate level. As γ increases beyond 0.6, LC Win Rate starts to decline, likely due to overfitting,

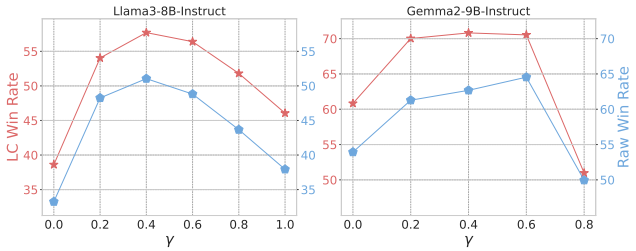

Figure 8: Impact of the hyperparameter γ on LC Win Rate and Raw Win Rate for Llama3-8B-Instruct (left) and Gemma2-9B-Instruct (right).

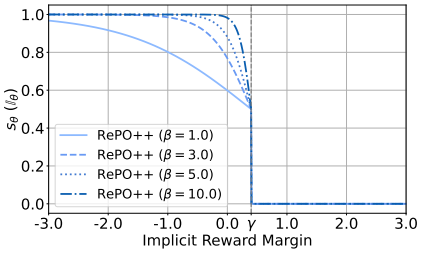

Figure 9: Gradient weighting functions of RePO++ ($s_\theta \cdot \mathbb{I}(M_\theta < \gamma)$).

where the model overly aligns with preference data at the expense of generalization. Conversely, at $\gamma = 0.0$, where no preference optimization is applied, the LC Win Rate remains low, emphasizing the necessity of preference tuning.

Raw Win Rate trends reveal model robustness differences. The Raw Win Rate follows a similar trend but highlights differences in robustness across models. For Llama3-8B-Instruct, the Raw Win Rate peaks at $\gamma = 0.4$ before declining, suggesting that excessive preference optimization ($\gamma > 0.6$) negatively impacts the model's ability to generalize. In contrast, Gemma2-9B-Instruct exhibits a more stable Raw Win Rate across a wider range of γ, reaching its highest performance at $\gamma = 0.6$ before experiencing a sharp decline at $\gamma = 0.8$. This suggests that Gemma2-9B-Instruct maintains better robustness to preference optimization compared to Llama3-8B-Instruct.

Gemma2-9B-Instruct outperforms Llama3-8B-Instruct. Gemma2-9B-Instruct consistently outperforms Llama3-8B-Instruct in both LC Win Rate and Raw Win Rate. This observation indicates that Gemma2-9B-Instruct not only aligns more effectively with learned preferences but also retains superior generalization capability. The results highlight the importance of carefully selecting γ to avoid performance degradation at extreme values. Future work could explore adaptive strategies for dynamically tuning γ, ensuring that preference optimization enhances alignment without compromising generalization.

D.4 Analsis on RePO++

Figure 9 illustrates the relationship between the gradient and the implicit reward margin. As shown in the figure, when the implicit reward margin is greater than γ, the gradient becomes zero. In this case, the model can stop updating for well-separated pairs, thus preventing overfitting. On the other hand, when the implicit reward margin is less than γ, the model continues to increase the weight for less-separated pairs. Furthermore, the harder the pair is to distinguish, the larger the gradient becomes, eventually converging to 1.0. This behavior is reminiscent of curriculum learning, where more difficult samples are assigned higher weights.

D.5 Analysis of the Relationship Between SLiC-HF and RePO

To provide deeper insights into the relationship between SLiC-HF and RePO, we conducted additional experiments examining the effect of SFT regularization—a core component of SLiC-HF that is absent

in our method. As demonstrated in Table 6, we systematically evaluated performance across varying values of the regularization coefficient λ.

Mathematical Comparison. From a formulation perspective, SLiC-HF combines a hinge loss term with an SFT regularization component that penalizes deviation from the reference model. Specifically, the SFT regularization is controlled by parameter λ, which balances preference optimization against model drift. By contrast, RePO eliminates this regularization entirely, relying solely on its binary threshold mechanism to control optimization.

Impact of SFT Regularization. Our experimental results reveal a consistent trend: as λ increases from 0.0 to 5.0, performance on AlpacaEval2 LC steadily declines from 34.1% to 27.8%. This progressive degradation suggests that stronger regularization toward the initial SFT model actually hinders effective preference learning in this context. The optimal performance occurs at $\lambda = 0.0$, which effectively transforms SLiC-HF into a variant of RePO.

Different Optimization Challenges. These findings suggest that SFT regularization and RePO's threshold-based filtering address fundamentally different optimization challenges. While SFT regularization was originally introduced to mitigate catastrophic forgetting and preserve general capabilities, our results indicate that for direct preference optimization, such regularization is unnecessary and potentially counterproductive. Instead, RePO's selective gradient application through its threshold mechanism appears sufficient to prevent over-optimization while maintaining effective preference learning.

This analysis complements our main findings and further supports our hypothesis that carefully designed filtering mechanisms can effectively replace more complex regularization schemes in preference optimization.

Table 6: The hyperparameter λ in SliC-HF used for each Llama3-Instruct v0.2.

SLiC-HF	$\lambda = 0.0$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1.0$	$\lambda = 3.0$	$\lambda = 5.0$
AlpacaEval2 LC	34.1	33.9	32.8	30.8	28.6	27.8

D.6 Detailed Analysis of the Relationship Between DPO and RePO

To further investigate the relationship between DPO and RePO, we conducted additional experiments examining how different formulation components affect performance. As shown in Table 7, we systematically evaluated five variants that decompose the key elements of each approach.

DPO as a Special Case of SimPO. From a mathematical perspective, DPO can be viewed as a specific instance of SimPO where $\gamma = \log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x)$ (ignoring length normalization for equivalence). This connection highlights how DPO implicitly defines its target margin based on reference model probabilities rather than using an explicit hyperparameter.

Impact of ReLU Without Explicit Margin. The second row of Table 7 shows that directly replacing log-sigmoid with ReLU while maintaining DPO's implicit margin definition leads to catastrophic performance degradation (-44% on AlpacaEval LC, -47% on AlpacaEval WR, and -26% on Arena-Hard). This dramatic decline reveals that the binary threshold mechanism of ReLU is only effective when paired with an appropriate explicit margin parameter.

The Critical Role of γ. Rows 3-5 demonstrate that adding an explicit γ parameter consistently improves performance across all metrics regardless of whether log-sigmoid, ReLU, or their combination is used. The most substantial gains appear when ReLU and γ are combined (+4.4% LC, +9.1% WR on AlpacaEval), supporting our hypothesis that explicit threshold-based filtering effectively controls over-optimization.

Complementary Mechanisms. Interestingly, the combination of both mechanisms (row 5) yields the highest overall performance, suggesting that while RePO's binary filtering mechanism addresses the core over-optimization challenge, the continuous weighting from log-sigmoid may provide complementary benefits for fine-grained preference learning.

Table 7: Impact Analysis of γ Scaling and ReLU Mechanisms in DPO Training. Benchmark results on AlpacaEval 2.0 (AE2) and Arena-Hard (AH) demonstrate percentage point changes in Length-Controlled Win Rate (LC-WR) and Base Win Rate (WR) for Llama3-Instruct-v0.2 (8B). Values represent relative performance deltas (%) compared to standard DPO baseline.

Method	γ	**ReLU**	$\log\sigma$	**Llama3-Instruct v0.2 (8B)**		
				AE 2		**AH**
				LC	**WR**	**WR**
$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}-\beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$	×	×	✓	48.2	47.5	35.2
$\text{ReLU}\left(-\left(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}-\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)\right)$	×	✓	×	$26.9^{-44\%}$	$25.1^{-47\%}$	$26.2^{-26\%}$
$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}-\beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}-\gamma\right)$	✓	×	✓	$50.0^{+3.7\%}$	$50.7^{+6.7\%}$	$36.8^{+4.5\%}$
$\text{ReLU}\left(-\left(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}-\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}-\gamma\right)\right)$	✓	✓	×	$50.3^{+4.4\%}$	$51.8^{+9.1\%}$	$38.2^{+8.5\%}$
$-\log\sigma\left(-\text{ReLU}\left(-\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}-\beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}-\gamma\right)\right)\right)$	✓	✓	✓	$50.8^{+5.4\%}$	$52.2^{+9.9\%}$	$37.2^{+5.7\%}$

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: see abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: see Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: see Appendix C.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: see Appendix D.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: https://github.com/junkangwu/RePO.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: see Appendix D.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: The results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We carried out all computational tasks on a suite of four 80GB A100 GPUs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research has been conducted with strict adherence to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: see Appendix B.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The LLM is used only for writing, editing.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.