Robust Preference Optimization via Dynamic Target Margins

Jie Sun[1] **, Junkang Wu**[1] **, Jiancan Wu**[2] **, Zhibo Zhu**[1] **,**
Xingyu Lu[1] **, Jun Zhou**[1] **, Lintao Ma**[1*] **, Xiang Wang**[3*]
[1] Ant Group [2] Shanghai Key Laboratory of Data Science
[3] National University of Singapore
{kangji.sj, gavin.zzb, lintao.mlt, sing.lxy}@antgroup.com
{jkwu0909, wujcan}@gmail.com
jun.zhoujun@antfin.com xiangwang@u.nus.edu

Abstract

The alignment of Large Language Models (LLMs) is crucial for ensuring their safety and reliability in practical applications. Direct Preference Optimization (DPO) has emerged as an efficient method that directly optimizes models using preference pairs, significantly reducing resource demands. However, the effectiveness of DPO heavily depends on the data quality, which is frequently compromised by noise. In this work, we propose γ-PO, a dynamic target margin preference optimization algorithm that adjust reward margins at the pairwise level. By introducing instance-specific margin calibration, γ-PO strategically prioritizes high-confidence pairs (those demonstrating higher reward margins) while suppressing potential noise from ambiguous pairs. Moreover, γ-PO is a plug-and-play method, compatible with variants of DPO that rely on reward margin between preference pairs. Across benchmarks such as AlpacaEval2 and Arena-Hard, γ-PO achieves an average 4.4% improvement over other baselines, setting new benchmarks for state-of-the-art performance. Additionally, γ-PO requires minimal code changes and has a negligible impact on training efficiency, making it a robust solution for enhancing LLMs alignment. Our codes are available at https://github.com/sunjie279/gammaPO.

1 Introduction

The alignment of Large Language Models (LLMs) with human values remains critical for their safe deployment (Dong et al., 2024). While Reinforcement Learning from Human Feedback (RLHF) (Ouyang et al., 2022a) pioneered model alignment through reward modeling and Proximal Policy Optimization (PPO) (Schulman et al., 2017), its computational intensity and training instability (Zheng et al., 2023b) motivate simpler alternatives. Direct Preference Optimization (DPO) (Rafailov et al., 2023) offers an efficient alternative by directly optimizing preference probabilities with human preference pairs, bypassing the need for explicit reward modeling.

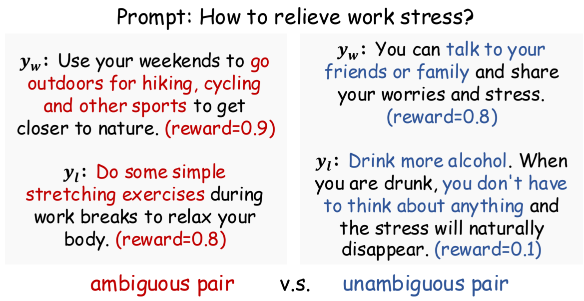

Figure 1: Comparison of ambiguous and unambiguous sample pairs. Ambiguous pairs exhibit narrow reward margins, indicating low confidence in model predictions, whereas unambiguous pairs demonstrate wide reward margins, reflecting high prediction confidence.

However, DPO exhibits significant performance degradation when dealing with data that presents uncertain or ambiguous reward signals (see Figure 1, left panel). In such cases, narrow reward margins reflect inherent ambiguity in response-reward comparisons, which can lead models to overfit to weak or conflicting preference signals rather than capturing consistent human preferences. This challenge is particularly critical as our empirical analysis in Figure 2 reveals that a substantial majority of training pairs are concentrated around zero reward margin, necessitating more attention and methodological improvements to effectively handle such ambiguous instances.

Existing robust alignment approaches typically adopt two distinct paradigms: (1) applying soft labels through label smoothing techniques to reduce overconfidence in potentially noisy annotations (Chowdhury et al., 2024), and (2) implementing rule-based data filtering mechanisms to eliminate ambiguous preference pairs (Wu et al., 2024). Both categories face critical limitations:

*Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2025, pages 5399–5416
July 27 - August 1, 2025

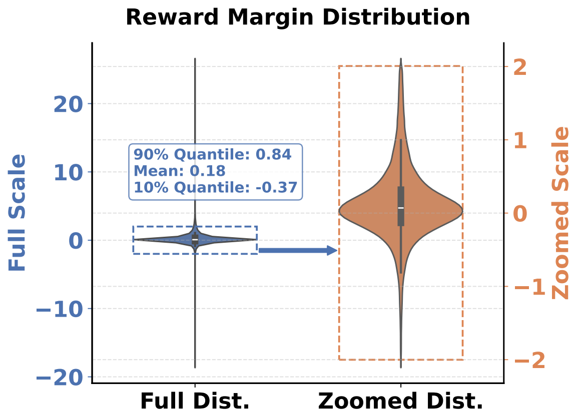

Figure 2: Distribution of the reward margin for the Mistral model using the SimPO objective on the Ultrafeedback Binarized dataset. The **blue** violin plot represents the full range of the original distribution, while the **brown** plot provides a zoomed-in view of the distribution, focusing on the central values close to zero.

- **One-Size-Fits-All Objective:** Current methods (Chowdhury et al., 2024; Meng et al., 2024; Amini et al., 2024; Rafailov et al., 2023) employ uniform loss functions that treat all preference pairs equally, disregarding varying confidence scores across samples.
- **Strong Theoretical Assumptions:** Prior work requires either known label flip rates (Chowdhury et al., 2024) or specific reward margin distributions (Wu et al., 2024), coupled with complex auxiliary mechanisms such as β-guided filtering that significantly limit practical adoption.

For modeling preference pair ambiguity, the Bradley-Terry model (Bradley and Terry, 1952) provides a probabilistic foundation through $P(y_w \succ y_l) = \sigma(r_w - r_l)$, where smaller reward margin indicates higher uncertainty. Building on this insight, recent work SimPO (Meng et al., 2024) introduces a fixed target margin γ_0 to reshape the reward landscape via $\sigma(r_w - r_l - \gamma_0)$. While effective in clean data scenarios, this rigid margin fails to adapt to the spectrum of uncertainty inherent in real-world preference pairs. This limitation motivates our key research question:

Can adaptive target margin adjustment based on reward margin enhance alignment robustness?

We propose γ-PO, a plug-and-play method that dynamically adjusts target margins γ_i for each preference pair. Our approach employs confidence-aware margin scaling: for unambiguous pairs with a significant difference in rewards, we increase the target margins to strengthen gradient signals during training. Conversely, for ambiguous pairs with small reward margins, we reduce the target margins to mitigate the influence of uncertain label. This dual mechanism provides two complementary benefits: (1) Algorithm-agnostic design compatible with any reward margin-based method (*e.g.,* DPO, SimPO), and (2) Theoretical connections to dynamic label smoothing (Su et al., 2022). We derive that γ-PO inherently implements *dynamic label smoothing* through its dynamic target margins. Specifically, smaller reward margins induce larger ε (*i.e.,* smoother labels), effectively smoothing labels for ambiguous pairs. Conversely, larger margins reduce ε (*i.e.,* sharper labels), sharpening supervision for unambiguous samples.

Experiments across four base models (LLaMA-3-8B-Instruct, Mistral-7B-Instruct, Gemma-2-9B-Instruct, and Qwen-2.5-7B-Instruct) show γ-PO achieves 4.4% average gains on AlpacaEval2 (Li and Zhang, 2023), Arena-Hard (Li et al., 2024), and MT-Bench (Zheng et al., 2023a), with minimal computational overhead. Additionally, our plug-and-play approach requires only a few lines of code and has a negligible impact on training efficiency, yet significantly enhances performance.

2 Preliminaries

2.1 Reinforcement Learning from Human Feedback

While LLMs are pre-trained on extensive datasets, those undergoing only supervised fine-tuning may output harmful or helpless content. To better align LLMs with human preferences, a widely adopted method is Reinforcement Learning from Human Feedback (Ouyang et al., 2022a), which involves three main steps: supervised fine-tuning (Zhou et al., 2023; Ding et al., 2023), reward model training (Chen et al., 2024; Gao et al., 2024), and policy model optimization. Specifically, the policy model is trained using Proximal Policy Optimization (PPO) (Anthony et al., 2017) that maximizes response rewards with Kullback-Leibler (KL) regularization (Kullback and Leibler, 1951) between the policy model π_θ and a reference model π_{ref}:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)} \left[r_\phi(x,y) - \beta\mathbb{D}_{\text{KL}}[\pi_\theta(y|x)||\pi_{\text{ref}}(y|x)]\right], \tag{1}$$

where response y is generated from policy π_θ based on the prompt x in dataset $\mathcal{D}$, and r_ϕ is the reward model. To estimate the labeling uncertainty of tuple (x, y_w, y_l), Bradley-Terry (BT) model (Bradley and Terry, 1952) defines the success probability:

$$\mathbb{P}(y_w \succ y_l \mid x) = \sigma(r_\phi(x, y_w) - r_\phi(x, y_l)), \tag{2}$$

where σ is the sigmoid function.

2.2 Direct Preference Optimization

DPO (Rafailov et al., 2023) is a prominent alignment method that directly optimizes the policy model, eliminating the need for training an explicit reward model and significantly reducing the resources required for alignment:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right], \tag{3}$$

where y_w is the prefered response over y_l, and β is the hyperparameter.

2.2.1 Preference Optimization with Margin

Recent studies such as ODPO (Amini et al., 2024) and SimPO (Meng et al., 2024) propose to modulate the reward difference between preferred and less preferred samples by introducing a target margin γ_0. We express the generalized preference optimization with a target margin through a unified loss function:

$$\mathcal{L}_{\text{margin PO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma \left(r_w - r_l - \gamma_0 \right) \right]. \tag{4}$$

The definition of r_w and r_l in each method are summarized in Table 1.

2.2.2 Robust Preference Optimization

To address the challenge of noisy preference data, Robust DPO (rDPO) (Chowdhury et al., 2024) has been proposed. rDPO enhances robustness by explicitly incorporating the label flip rate into its formulation, thereby reducing sensitivity to erroneous preference flips. The loss function for rDPO is defined as follows:

$$\mathcal{L}_{\text{rDPO}} = (1 - \varepsilon)\mathcal{L}_{\text{DPO}}(y_w \succ y_l) + \varepsilon \mathcal{L}_{\text{DPO}}(y_l \succ y_w), \tag{5}$$

where ε represents the flip rate of labels. This formulation ensures that when there is no label flip ($\varepsilon = 0$), the rDPO loss reduces to the standard DPO loss. rDPO shares similarities with label smoothing techniques, which introduce regularization to soften predictions' confidence. In rDPO, the parameter ε can be seen as a smoothing factor that controls the degree of confidence for the predictions. This provides a mechanism analogous to label smoothing, specifically tailored for handling noisy preferences optimization.

Methods	r_w	r_l
ODPO	$\beta \log \frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)}$	$\beta \log \frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}$
SimPO	$\frac{\beta}{\|y_w\|} \log \pi_\theta(y_w\|x)$	$\frac{\beta}{\|y_l\|} \log \pi_\theta(y_l\|x)$

Table 1: Definitions of r_w and r_l for alignment method ODPO and SimPO.

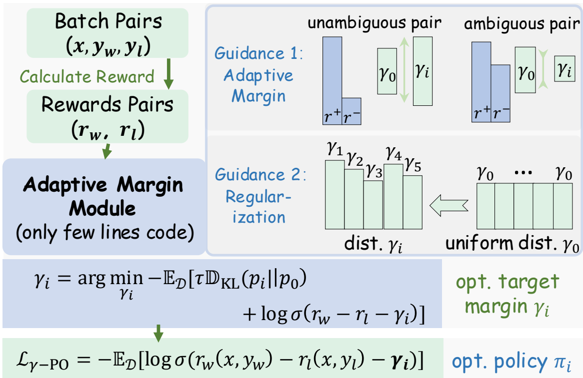

Figure 3: The Dynamic Target Margin Module dynamically adjusts the adaptive target margin (γ_i) through reward-driven optimization, guided by the dual mechanisms described in Section 3.1. The optimized γ_i subsequently replaces the static margin (γ_0) in the policy optimization loss function, enabling adaptive margin adjustment throughout the training process.

3 Method

In this section, we first present the methodology for deriving the adaptive target margin. Next, we present our proposed method, γ-PO, along with its two variants, γ-DPO and γ-SimPO. Finally, we conduct a comprehensive comparative analysis with rDPO (Chowdhury et al., 2024), highlighting its superior robustness and alignment capabilities.

3.1 Derivation of Adaptive Target Margin

The core objective of preference optimization with margin is to enforce $r_w > r_l + \gamma_0$ where γ_0 is a fixed margin. Our framework extends this by introducing instance-specific margins to address ambiguous reward signal in preference optimization, ensuring that:

$$r_w - r_l > \gamma_i \iff r_l - r_w + \gamma_i < 0. \tag{6}$$

To optimize γ_i, we focus on instances that violate this condition (where $r_l - r_w + \gamma_i \geq 0$). This leads to the following optimization objective:

$$\min_{\gamma_i} \max\left\{0, r_l - r_w + \gamma_i\right\}. \tag{7}$$

For computational efficiency, we reformulate this objective using the logarithmic approximation (*i.e.,* $\max\{0, y-x\} \approx -\log \frac{\exp(x)}{\exp(x)+\exp(y)}$), yielding:

$$\begin{aligned} &\min_{\gamma_i} -\log \frac{\exp(r_w)}{\exp(r_w)+\exp(r_l)\exp(\gamma_i)} \\ =&\min_{\gamma_i} -\log \sigma(r_w - r_l - \gamma_i). \end{aligned} \tag{8}$$

This logarithmic formulation enhances numerical stability during gradient computation while maintaining the theoretical guarantees.

Our dynamic target margin γ_i are guided by two principles: **(1) Adaptive Margin:** The preference pairs with larger reward margins represent higher-confidence annotations and should receive stronger learning signals, while ambiguous pairs require more conservative treatment, and **(2) Regularization:** To mitigate the risk of margin collapse (*i.e.,* $\gamma_i \to -\infty$), the adaptive target margin distribution must preserve a reasonable scale relative to a baseline target margin γ_0. To achieve this, we regularize the target margin distribution by minimizing the KL divergence with respect to a uniform prior. This dual-objective formulation is expressed as:

$$\begin{aligned} \gamma_i = \arg\min_{\gamma_i} -\mathbb{E}_{\mathcal{D}}[&\log \sigma(r_w - r_l - \gamma_i) \\ &+ \tau\, \mathbb{D}_{\text{KL}}(p_i \| p_0)], \end{aligned} \tag{9}$$

where $p_i = \gamma_i / \sum_j \gamma_j$ and p_0 represent the normalized γ_i distribution and the uniform distribution, respectively. Regularization coefficient τ controls the trade-off between adaptation flexibility and stability. The complete derivation and computational details of γ_i are provided in Appendix C. For clarity, we summarize the key notations used in this paper in Table 9.

3.2 Generalizability of γ-PO

Building upon the dynamic target margin γ_i derived in Section 2.2.1, we extend the conventional fixed margin loss in Sec 2.2.1 to a dynamic, instance-specific formulation. Consequently, we propose γ-PO, a method that adapts the margin for each instance. The final policy optimization objective is consequently defined as:

$$\mathcal{L}_{\gamma\text{-PO}} = -\mathbb{E}_{\mathcal{D}}[\log \sigma(r_w - r_l - \gamma_i)]. \tag{10}$$

Algorithm 1 γ-PO Algorithm

Input: SFT model π_{SFT}, dataset $\mathcal{D}$, total number of iterations T, learning rate α,
Initial policy model $\pi_\theta = \pi_{\text{SFT}}$,
for $t = 0$ to T **do**
    Sample a batch of tuples (x, y_w, y_l) from $\mathcal{D}$,
    Calculate the rewards (r_w, r_l) via Table 1,
    Calculate the adaptive margin γ_i using Equation (9),
    Update the policy parameters θ for π_θ by applying gradient descent to Equation (10): $\theta_{t+1} = \theta_t - \alpha \nabla_{\theta_t} \mathcal{L}_{\gamma\text{-PO}}$.
end for

The proposed γ-PO framework demonstrates remarkable generalizability as an algorithm-agnostic method. It can be seamlessly integrated with existing approaches, yielding enhanced variants such as γ-DPO:

$$\begin{aligned} &\mathcal{L}_{\gamma\text{-DPO}} = -\mathbb{E}_{\mathcal{D}}[\log \sigma(r_w - r_l - \gamma_i)], \quad \text{where} \\ &r_w = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)},\ r_l = \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}, \end{aligned} \tag{11}$$

and γ-SimPO:

$$\begin{aligned} &\mathcal{L}_{\gamma\text{-SimPO}} = -\mathbb{E}_{\mathcal{D}}[\log \sigma(r_w - r_l - \gamma_i)], \quad \text{where} \\ &r_w = \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x),\ r_l = \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x). \end{aligned} \tag{12}$$

3.3 Relation of γ-PO and rDPO

To bridge γ-PO with existing robust methods, we analyze its connection to rDPO through their loss formulations. The rDPO objective applies label smoothing to the standard DPO loss:

$$\begin{aligned} \mathcal{L}_{\text{rDPO}} = &-\varepsilon\, \mathbb{E}_{\mathcal{D}}[\log \sigma(r_l - r_w - \gamma_0)] \\ &-(1-\varepsilon)\, \mathbb{E}_{\mathcal{D}}[\log \sigma(r_w - r_l - \gamma_0)], \end{aligned} \tag{13}$$

where ε controls label smoothing intensity. Our key insight is that γ-PO implicitly implements *adaptive label smoothing* through dynamic margins. We formalize this connection via:

Theorem 3.1. *Let $\delta = \gamma_i - \gamma_0$ and $m = r_w - r_l$. When $|\delta| \ll |m|$, equating $\mathcal{L}_{rDPO}$ and $\mathcal{L}_{\gamma\text{-}PO}$ yields the approximation:*

$$\varepsilon \approx \frac{\delta[1 - \sigma(m - \gamma_0)]}{\log(1 + e^{m+\gamma_0}) - \log(1 + e^{\gamma_0 - m})}. \tag{14}$$

This analysis demonstrates that γ-PO inherently implements *adaptive label smoothing* through its dynamic margins. The derived relationship between δ (margin adjustment) and ε (smoothing parameter) reveals that smaller reward margin $m = r_w - r_l$ automatically induce larger ε values, effectively smoothing labels for ambiguous pairs. Conversely, a larger margin reduces ε, sharpening supervision for high-confidence samples. Crucially, the negative gradient $\partial\varepsilon/\partial d < 0$ formalizes this inverse relationship, enabling γ-PO to self-regulate label certainty based on reward margin magnitudes. By contrast, rDPO's static ε requires pre-defined label flip rates and lacks this instance-level adaptation. This perspective provides new insights into the interpretation of γ-PO. The detailed proof of this theorem is provided in Appendix A.

4 Experiments

This section starts with an overview of the experimental setup, including various base models, training datasets, evaluation methods, and baselines. We then present the experimental results and visualize the dynamic target reward margin along with the hyperparameter analysis.

4.1 Experiment Setup

Base Model. Our approach is tested on four different base models: LLaMA-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), Gemma-2-9B-Instruct (Rivière et al., 2024), and Qwen-2.5-7B-Instruct (Yang et al., 2024). The diversity of these base models provides a solid foundation for our experiments.

Dataset. Following the approach in SimPO (Meng et al., 2024), a strong reward model is used to re-annotate the preference datasets. In this work, five different responses are generated for each prompt from the Ultrafeedback Binarized dataset[*] using the base models. These responses are then re-annotated using a more powerful reward model, RLHFlow/ArmoRM-Llama3-8B-v0.1[*]. Then, the responses with the highest and lowest scores are selected to combine the preference data.

Evaluation. We use three well-established metrics, AlpacaEval2, Arena-Hard, and MT-Bench to assess LLMs alignment with human preferences. AlpacaEval2 (Dubois et al., 2024) is an enhanced automated metric that evaluates LLMs performance using Win Rate (WR) and Length-Controlled Win Rate (LC) with a dataset of 805 instructions. Both the reference model and judge model are GPT-4-1106-preview (OpenAI, 2023). The length control improves robustness to length manipulation and better aligns with human evaluations. Arena-Hard (Li et al., 2024) assesses LLMs by applying 500 prompts derived from the Chatbot Arena dataset. It benchmarks the LLMs against GPT-4-0314 (OpenAI, 2023) as the baseline, and GPT-4-1106-preview evaluates the responses. Each response is rated using a five-point Likert scale (Liang et al., 2020), which facilitates a fair comparison and emphasizes performance disparities between models. MT-bench (Zheng et al., 2023a) evaluates chatbots by using 80 multi-turn dialogue prompts that are specifically designed to assess their abilities in multi-round interactions and following instructions. It benchmarks the chatbots against GPT-4, which serves as both the baseline and the evaluator for the responses. Each response is scored based on specific criteria, allowing for objective comparisons and highlighting the differences in how well various models handle complex conversational tasks.

Baselines. We compare γ-PO with several state-of-the-art preference optimization methods, including DPO (Rafailov et al., 2023), IPO (Azar et al., 2024a), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), R-DPO (Park et al., 2024), rDPO (Chowdhury et al., 2024), β-DPO (Wu et al., 2024), and SimPO (Meng et al., 2024). We also use SFT model as a baseline. Further information regarding the baselines and training details can be found in Appendix B.

4.2 Experiment Results

γ-PO consistently outperforms the baselines. We apply our method to one of the leading baseline preference optimization approaches, SimPO, resulting in γ-SimPO. The results in Table 2 show that γ-SimPO consistently outperforms other methods in both AlpacaEval2 and Arena-Hard, demonstrating stable performance improvements. For instance, in AlpacaEval2, it boosts the performance of Qwen2.5-Instruct from 34.1% to 39.0% of LC, and from 33.3% to 34.3% of WR. In the Mistral-Instruct setting, γ-SimPO outperforms the baseline SimPO by 6.5% (LC) and 7.1% (WR) in AlpacaEval2, and enhances WR of Arena-Hard from 20.7% to 22.9%. On average, our method leads to a 4.4% improvement. Additionally, we also evaluate our

[*] HuggingFaceH4/ultrafeedback_binarized
[*] RLHFlow/ArmoRM-Llama3-8B-v0.1

Method	Llama3-Instruct (8B)						Mistral-Instruct (7B)					
	AlpacaEval 2			Arena-Hard		Average	AlpacaEval 2			Arena-Hard		Average
	LC(%)	WR(%)	err(%)	WR(%)	95%CI		LC(%)	WR(%)	err(%)	WR(%)	95%CI	
SFT	28.2	27.4	1.38	24.3	(-1.7, 1.9)	26.6	21.3	16.0	1.15	13.1	(-1.5, 1.1)	16.8
IPO	38.3	35.9	1.48	31.4	(-2.3, 2.5)	35.2	24.0	19.3	1.21	16.3	(-1.6, 1.5)	19.9
KTO	34.9	32.5	1.45	27.8	(-2.8, 3.5)	31.7	18.9	20.2	1.23	14.4	(-1.6, 1.6)	17.8
CPO	27.9	26.8	1.36	25.4	(-2.3, 2.9)	26.7	25.3	22.6	1.30	18.4	(-1.7, 1.9)	22.1
ORPO	32.0	30.3	1.41	25.9	(-1.9, 2.4)	29.4	24.0	21.8	1.27	20.4	(-2.1, 2.0)	22.1
R-DPO	47.9	43.5	1.48	32.9	(-2.5, 2.7)	41.4	24.8	22.0	1.27	17.1	(-1.6, 1.7)	21.3
rDPO	47.8	43.7	1.48	34.4	(-2.2, 2.4)	42.0	23.7	20.9	1.25	16.3	(-1.5, 1.7)	20.3
DPO	46.8	42.7	1.48	32.7	(-2.6, 2.7)	40.7	22.5	20.2	1.20	17.1	(-2.0, 1.6)	19.9
β-DPO	48.2	44.0	1.48	36.4	(-2.0, 2.1)	42.9	23.3	20.6	1.22	17.5	(-1.4, 1.6)	20.5
SimPO	54.3	46.6	1.53	31.5	(-2.4, 3.0)	44.1	29.3	30.8	1.38	20.7	(-2.0, 1.8)	26.9
γ-SimPO	**55.4**	**48.7**	1.48	**37.3**	(-2.6, 2.8)	**47.1**	**31.2**	**33.0**	1.39	**22.9**	(-1.7, 2.2)	**29.0**
Improve	+2.0%	+4.5%	-	+6.8%	+2.5%	-	+6.5%	+7.1%	-	+10.6%	-	+7.8%

Method	Gemma2-Instruct (9B)						Qwen2.5-Instruct (7B)					
	AlpacaEval 2			Arena-Hard		Average	AlpacaEval 2			Arena-Hard		Average
	LC (%)	WR (%)	err (%)	WR (%)	95%CI (%)		LC (%)	WR (%)	err (%)	WR (%)	95%CI (%)	
SFT	50.2	38.1	1.44	40.8	(-2.3, 2.1)	43.0	23.2	22.2	1.29	35.9	(-3.1, 3.1)	27.1
IPO	61.3	57.2	1.45	56.7	(-2.8, 2.6)	58.4	25.1	25.4	1.34	43.7	(-4.4, 2.5)	31.4
KTO	60.7	54.5	1.46	51.8	(-3.3, 2.8)	55.7	32.6	30.5	1.38	46.6	(-3.6, 3.0)	36.6
CPO	50.3	38.8	1.45	41.8	(-2.4, 2.1)	43.6	23.3	19.5	1.29	38.3	(-3.1, 3.1)	27.0
ORPO	54.9	46.4	1.48	44.5	(-2.3, 2.2)	48.6	26.4	23.0	1.29	38.3	(-3.2, 2.7)	29.2
R-DPO	66.9	**65.9**	1.37	60.9	(-2.3, 2.8)	64.6	33.2	31.7	1.39	53.1	(-3.6, 3.5)	39.3
rDPO	68.2	**65.9**	1.39	59.1	(-2.4, 2.9)	64.4	34.1	33.3	1.42	**65.2**	(-2.0, 2.6)	44.2
DPO	67.8	65.8	1.39	60.3	(-2.0, 2.3)	64.6	32.3	31.8	1.40	54.3	(-3.7, 2.4)	39.5
β-DPO	68.0	**65.9**	1.37	61.0	(-2.2, 2.2)	65.0	32.4	31.4	1.39	61.0	(-2.2, 2.2)	41.6
SimPO	70.9	64.5	1.43	55.3	(-2.4, 2.0)	63.6	33.3	28.2	1.37	59.0	(-2.9, 3.0)	40.2
γ-SimPO	**72.0**	**65.9**	1.42	**62.0**	(-2.5, 1.8)	**66.6**	**39.0**	**34.3**	1.46	64.8	(-2.1, 2.3)	**46.0**
Improve	+1.6%	+0.0%	-	+1.6%	-	+2.5%	+14.4%	+3.0%	-	-0.6%	-	+4.1%

Table 2: The results from AlpacaEval and Arena-Hard under four different settings are shown. "LC" and "WR" represent the Length-Controlled and Raw Win Rates, respectively. **Bold** indicates the best performance for each metric, while underlined values represent the best performance excluding our methods (our method is γ-SimPO, *i.e.*, our plug-and-play modules applied to SimPO). "err" refers to the standard error, while "95% CI" stands for the 95% confidence interval. "Improve" denotes the percentage improvement of the bold value over the underlined one. Each metric, except for "err" and "95% CI", is considered better when it has a higher value.

methods on MT-Bench, as shown in Appendix F. The results presented in Table 10 demonstrate that our proposed methods, γ-DPO and γ-SimPO, consistently achieve superior performance compared to baseline methods across all models. These results highlight the effectiveness of our approach in better aligning LLMs with human preferences.

γ-PO outperforms other plug-and-play methods. We compared γ-PO with other plug-and-play techniques, such as rDPO and β-DPO. Specifically, we applied the β-DPO and rDPO to SimPO, resulting in β-SimPO and rSimPO, respectively. Additionally, we apply the γ-PO approach to both DPO and SimPO, leading to the development of γ-DPO and γ-SimPO. The experimental results in Table 3 show that our method significantly outperforms both rDPO and β-DPO in improving model performance. For instance, in the Qwen2.5-Instruct scenario, γ-SimPO exceeds the baseline SimPO by 6.1 in AlpacaEval2 and 5.8 in Arena-Hard. Moreover, both γ-DPO and γ-SimPO achieved the highest average scores across all metrics and settings, with just one exception. Overall, γ-PO demonstrates a more effective enhancement of alignment performance compared to other plug-and-play methods.

γ-PO handles flipped data better. To evaluate the resilience of the γ-SimPO method in the presence of label uncertainty, we deliberately introduce label noise into the training dataset by flipping labels at random rates of 10% and 20%. Building on the optimal baseline provided by the SimPO method, we conducted comparative experiments us-

Method	Llama3-Instruct (8B)				Mistral-Instruct (7B)			
	AlpacaEval 2		Arena-Hard	Average	AlpacaEval 2		Arena-Hard	Average
	LC(%)	WR(%)	WR(%)	Avg. (rank)	LC(%)	WR(%)	WR(%)	Avg. (rank)
DPO	$46.8^{+0.0}$	$42.7^{+0.0}$	$32.7^{+0.0}$	40.7 (4)	$22.5^{+0.0}$	$20.2^{+0.0}$	$17.1^{+0.0}$	19.9 (4)
rDPO	$47.8^{+1.0}$	$43.7^{+1.0}$	$34.4^{+1.7}$	42.0 (2)	$23.7^{+1.2}$	$20.9^{+0.7}$	**17.5**$^{+0.4}$	20.7 (2)
β-DPO	**48.2**$^{+1.4}$	**44.0**$^{+1.3}$	**36.4**$^{+3.7}$	**42.9 (1)**	$23.3^{+0.8}$	$20.6^{+0.4}$	**17.5**$^{+0.4}$	20.5 (3)
γ-DPO	$47.8^{+1.0}$	$43.0^{+0.3}$	$35.1^{+2.4}$	42.0 (2)	**23.8**$^{+1.3}$	**21.7**$^{+1.5}$	**17.5**$^{+0.4}$	**21.0 (1)**
SimPO	$54.3^{+0.0}$	$46.6^{+0.0}$	$31.5^{+0.0}$	44.1 (4)	$29.3^{+0.0}$	$30.8^{+0.0}$	$20.7^{+0.0}$	26.9 (4)
rSimPO	$53.0^{-1.3}$	$45.5^{-1.1}$	$36.5^{+5.0}$	45.0 (3)	$28.9^{-0.4}$	$30.3^{-0.5}$	$22.2^{+1.5}$	27.1 (3)
β-SimPO	$54.8^{+0.5}$	$48.1^{+1.5}$	$32.6^{+1.1}$	45.2 (2)	$29.4^{+0.1}$	$31.1^{+0.3}$	$22.6^{+1.9}$	27.7 (2)
γ-SimPO	**55.4**$^{+1.1}$	**48.7**$^{+2.1}$	**37.3**$^{+5.8}$	**47.1 (1)**	**31.2**$^{+1.9}$	**33.0**$^{+2.2}$	**22.9**$^{+2.2}$	**29.0 (1)**
Method	**Gemma2-Instruct (9B)**				**Qwen2.5-Instruct (7B)**			
	AlpacaEval 2		**Arena-Hard**	**Average**	**AlpacaEval 2**		**Arena-Hard**	**Average**
	LC(%)	WR(%)	WR(%)	Avg. (rank)	LC(%)	WR(%)	WR(%)	Avg. (rank)
DPO	$67.8^{+0.0}$	$65.8^{+0.0}$	$60.3^{+0.0}$	64.6 (3)	$32.3^{+0.0}$	$31.8^{+0.0}$	$54.3^{+0.0}$	39.5 (4)
rDPO	$68.2^{+0.4}$	$65.9^{+0.1}$	$59.1^{-1.2}$	64.4 (4)	$34.1^{+1.8}$	$33.3^{+1.5}$	$65.2^{+10.9}$	44.2 (2)
β-DPO	$68.0^{+0.2}$	$65.9^{+0.1}$	$61.0^{+0.7}$	65.0 (2)	$32.4^{+0.1}$	$31.4^{-0.4}$	$61.0^{+6.7}$	41.6 (3)
γ-DPO	**70.6**$^{+2.8}$	**68.0**$^{+2.2}$	**64.1**$^{+3.8}$	**67.6 (1)**	**34.4**$^{+2.1}$	**34.3**$^{+2.5}$	**68.4**$^{+14.1}$	**45.7 (1)**
SimPO	$70.9^{+0.0}$	$64.5^{+0.0}$	$55.3^{+0.0}$	63.6 (4)	$33.3^{+0.0}$	$28.2^{+0.0}$	$59.0^{+0.0}$	40.2 (4)
rSimPO	$71.6^{+0.7}$	$64.5^{+0.0}$	$58.6^{+3.3}$	64.9 (2)	$38.8^{+5.5}$	$31.7^{+3.5}$	$61.4^{+2.4}$	44.0 (2)
β-SimPO	$71.1^{+0.2}$	$62.8^{-1.7}$	$59.0^{+3.7}$	64.3 (3)	$34.5^{+1.2}$	$29.5^{+1.3}$	$59.1^{+0.1}$	41.0 (3)
γ-SimPO	**72.0**$^{+1.1}$	**65.9**$^{+1.4}$	**62.0**$^{+6.7}$	**66.6 (1)**	**39.0**$^{+5.7}$	**34.3**$^{+6.1}$	**64.8**$^{+5.8}$	**46.0 (1)**

Table 3: The improvements in results from AlpacaEval2 and Arena-Hard across various plug-and-play methods. "LC" stands for Length-Controlled, while "WR" represents Raw Win Rates. The methods rDPO and rSimPO refer to the application of rDPO to the DPO and SimPO methods, respectively. And the method β-SimPO refer to the application of β-DPO to the SimPO method. The superscript indicates enhancements based on the baselines, such as DPO and SimPO, where red represents improvement and blue signifies degradation. Performance metrics displayed in **bold** indicate the best results among the plugins. For each metric, a higher value denotes better performance.

Method	Label Flip Rate 0.1				Label Flip Rate 0.2			
	AlpacaEval 2		Arena-Hard	Average	AlpacaEval 2		Arena-Hard	Average
	LC(%)	WR(%)	WR(%)	Avg. (rank)	LC(%) / ↑	WR(%)	WR(%)	Avg. (rank)
SimPO	$26.2^{+0.0}$	$28.6^{+0.0}$	$20.0^{+0.0}$	24.9 (4)	$24.5^{+0.0}$	$26.1^{+0.0}$	$18.7^{+0.0}$	23.1 (4)
rSimPO	$27.1^{+0.9}$	**28.7**$^{+0.1}$	$21.9^{+1.9}$	25.9 (2)	$25.8^{+1.3}$	$27.0^{+0.9}$	$19.6^{+0.9}$	24.1 (3)
β-SimPO	$26.6^{+0.4}$	$28.4^{-0.2}$	$21.6^{+1.6}$	25.5 (3)	$26.3^{+1.8}$	$27.9^{+1.8}$	$18.9^{+0.2}$	24.4 (2)
γ-SimPO	**28.2**$^{+2.0}$	$28.3^{-0.3}$	**22.4**$^{+2.4}$	**26.3 (1)**	**25.9**$^{+1.4}$	**26.3**$^{+0.2}$	**21.9**$^{+3.2}$	**24.7 (1)**

Table 4: The results from AlpacaEval2 and Arena-Hard illustrate the performance of various plug-and-play methods in a noisy label setting. In this context, "LC" stands for Length-Controlled, and "WR" stands for Raw Win Rates. Metrics that are highlighted in **bold** indicate the best performance among the plugins evaluated. The base model used for this analysis is Mistral-7B-Instruct. For each metric, a higher value indicates better performance.

ing rSimPO, β-SimPO, and our proposed γ-SimPO, all utilizing the Mistral-7B-Instruct as the base model. As shown in Table 4, γ-SimPO outperforms the other methods, achieving the highest average score, which is further illustrated in Figure 4. These results demonstrate that γ-SimPO exhibits

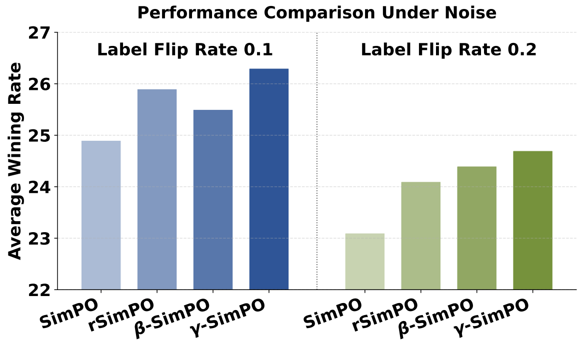

Figure 4: Comparison of average winning rates with randomly flipped labels at different probabilities.

robustness in scenarios involving noise, such as label flipping.

γ-PO requires no additional computing overhead. To evaluate the computational efficiency of γ-PO, we measure the training duration when combining γ-PO with DPO and SimPO across four base models. As shown in Table 5, γ-PO maintains computational efficiency comparable to the baseline methods while also delivering performance gains. Specifically, γ-DPO has only a 0.7% longer training time than DPO, while γ-SimPO demonstrates an even more minor increase of 0.5% over SimPO. These results confirm that our method performs better without significantly increasing GPU resource consumption.

4.3 Visualization of Dynamic Target Margin

Figure 5 depicts the relationship between the target margin γ_i and the reward gap for various LLMs (Llama, Mistral, Gemma, and Qwen, which stand for Llama3-Instruct (8B), Mistral-Instruct (7B), Gemma2-Instruct (9B), and Qwen2.5-Instruct (7B), respectively). The target margin exhibits a sigmoid-like increase with the reward gap, with the rapid upward slope closely corresponding to the range of the dynamic target margin. This behavior suggests that the target margin is increased when the reward gap exceeds the threshold, prioritizing learning from high-gap pairs. In contrast, when the reward gap is smaller than the threshold, corresponding to low-gap pairs that may contain noise, the target margin is reduced to prevent overfitting to these pairs. Furthermore, we observed that the target margin does not exhibit extreme values despite large or small reward gaps; it varies within a controlled, reasonable range. This visualization confirms the effectiveness of our approach in dynamically adjusting the target margin based on the reward gap, illustrating how it can be tailored to enhance learning from high-gap pairs while minimizing the influence of unreliable pairs.

4.4 Hyperparameter

We thoroughly investigate how the hyperparameter τ in γ-PO affects the performance of AlpacaEval2 and Arena-Hard. Figure 6 illustrates the performance metrics across different τ values for the Gemma base model. We plot three key metrics: Length-Controlled Winning Rate (AE2:LC) and Raw Winning Rate (AE2:WR) from AlpacaEval2, and Raw Winning Rate from Arena Hard (AH:WR). The results demonstrate a clear upward trend for all three metrics as τ increases when $\tau < 20$, followed by a decline when $\tau > 20$. This suggests that the optimal value of τ is almost the same across all metrics. The range for hyperparameter search is detailed in Table 6, while the final optimal hyperparameters are presented in Table 7.

5 Related Work

5.1 Reinforcement Learning from Human Feedback

Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Azar et al., 2024b) is a crucial post-training step following Supervised Fine-Tuning (SFT). RLHF helps align LLMs outputs with human values such as helpfulness, harmlessness, and honesty (Ouyang et al., 2022b; Bai et al., 2022). It has been widely applied in the revolutionary closed-source LLMs, such as GPT-4 (OpenAI, 2023) and Gemini (Anil et al., 2023), as well as powerful open-source models like LLaMA-3 (Dubey et al., 2024), Gemma-2 (Rivière et al., 2024), and Qwen-2.5 (Yang et al., 2024). The RLHF process generally consists of three main steps. First, a policy π_{SFT} is trained on a large dataset through SFT (Zhou et al., 2023; Ding et al., 2023; Köpf et al., 2023). Next, π_{SFT} generates multiple responses, which are then labeled by humans into preference pairs. These pairs are used to train a reward model, r_θ, that scores the model's responses based on their alignment with human preferences (Chen et al., 2024; Gao et al., 2024; Lightman et al., 2024; Havrilla et al., 2024; Luo et al., 2023; Lambert et al., 2024; Dong et al., 2024). Lastly, the reward model updates the policy model via PPO algorithm (Schulman et al., 2017) in RL (Anthony et al., 2017).

Base Model	DPO	γ-DPO	SimPO	γ-SimPO
Llama	4A100*4h54min	4A100*4h55min^{+1min}	4A100*3h30min	4A100*3h33min^{+3min}
Mistral	4A100*4h45min	4A100*4h53min^{+8min}	4A100*3h41min	4A100*3h42min^{+1min}
Gemma	8A100*4h04min	8A100*4h03min^{-1min}	8A100*3h13min	8A100*3h12min^{-1min}
Qwen	4A100*5h26min	4A100*5h29min^{+3min}	4A100*3h33min	4A100*3h36min^{+3min}
Avg. GPU time	23.22h·A100	23.38h·A100$^{+0.7\%}$	17.17h·A100	17.25h·A100$^{+0.5\%}$

Table 5: A100 GPU time consumption for training DPO, γ-DPO, SimPO, and γ-SimPO across all base models.

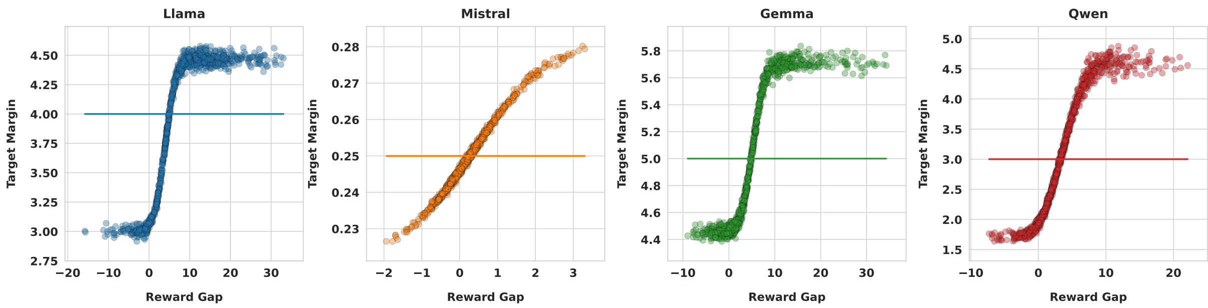

Figure 5: Visualization of dynamic target margin (γ_i) with reward gaps. The horizontal line indicates the initial value of target margin (γ_0).

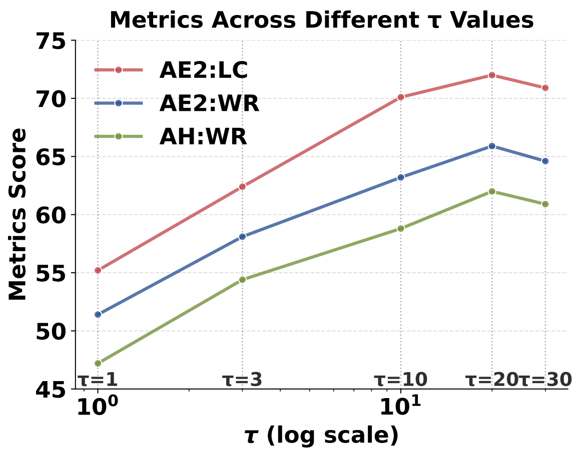

Figure 6: Performance metrics across different τ settings for the Gemma model, showing Length-Controlled Winning Rate (AE2:LC) and Raw Winning Rate (AE2:WR) from AlpacaEval2, and Raw Winning Rate (AH:WR) from Arena Hard.

5.2 Direct Preference Optimization

Recent studies introduce several enhancements to DPO, including IPO (Azar et al., 2024a), KTO (Ethayarajh et al., 2024), CPO (Xu et al., 2024), ORPO (Hong et al., 2024), R-DPO (Park et al., 2024), β-DPO (Wu et al., 2024), SimPO (Meng et al., 2024), α-DPO (Wu et al., 2025a), WPO (Zhou et al., 2024), and RainbowPO (Zhao et al., 2024a). In addition, Dr. DPO (Wu et al., 2025b) uses Distributionally Robust Optimization to handle noisy data and align human preferences. Huang et al. (2025) introduce the alignment potential metric M_{AP} to optimize preference data selection for LLM alignment.

6 Conclusion

In this work, we introduce a plug-and-play approach called γ-PO, which is applicable to DPO and its variants. The key idea behind γ-PO is personalizing the target margin based on the reward margin, increasing them for high-confidence pairs to prioritize learning from these instances while reducing them for low-reward margin pairs to mitigate the impact of uncertain preference, which improves the handling of ambiguous preference data and reduces the risk of noisy data distorting the alignment policy. Furthermore, γ-PO offers a simple and efficient solution with minimal impact on training efficiency, making it a valuable tool for improving alignment in practical applications. Extensive experiments demonstrate that the γ-PO enhances the alignment performance, achieving robust results on AlpacaEval2 and Arena-Hard. Our findings highlight the potential of dynamic margin strategies as a promising direction for improving the alignment of LLMs, particularly in the presence of noisy data. There is still substantial opportunity for deeper exploration of this approach to enhance model alignment in real-world settings.

Limitations

Although the γ-PO method demonstrates significant improvements in LLMs alignment, it introduces several limitations that warrant further investigation and refinement in future work.

Additional Hyperparameter. The γ-PO method, while enhancing performance, introduces an additional hyperparameter, τ, which significantly impacts model alignment. Future efforts will focus on eliminating this parameter, aiming to compute the adaptive margin automatically using existing parameters.

Limited Evaluation. In our experiments, we used two evaluation benchmarks: AlpacaEval2 (Li and Zhang, 2023) and Arena-Hard (Li et al., 2024). Although these benchmarks are widely used in the field (Wu et al., 2025a; Meng et al., 2024; Wu et al., 2024), we observed that their results are influenced by the choice of parameters (*e.g.*, the version of 'alpaca_eval'). In the future, incorporating a broader range of LLMs evaluation metrics and developing benchmarks that are more robust to parameter variations could provide a more comprehensive assessment of model performance.

7 Ethical Considerations

While the UltraFeedback dataset (Cui et al., 2024) provides high-quality preference and textual feedback, covering aspects such as instruction following, truthfulness, honesty, and helpfulness, it still faces challenges in addressing the safety of LLMs. This includes mitigating risks such as toxicity, stereotype bias, adversarial and out-of-distribution robustness, privacy, machine ethics, and fairness (Wang et al., 2023). Future alignment efforts will require the integration of larger and more diverse training datasets to comprehensively address these issues (Ji et al., 2023; Zhao et al., 2024b).

ACKNOWLEDGEMENT

This work was supported by Ant Group Research Fund.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. In *Findings of the 62th Annual Meeting of the Association for Computational Linguistics*, pages 9954–9972, Bangkok, Thailand.

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. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint*, abs/2312.11805.

Thomas Anthony, Zheng Tian, and David Barber. 2017. Thinking fast and slow with deep learning and tree search. In *Advances in Neural Information Processing Systems 30*, pages 5360–5370, Long Beach, CA.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024a. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 238, pages 4447–4455, Valencia, Spain.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Rémi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024b. A general theoretical paradigm to understand learning from human preferences. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 238, pages 4447–4455, Palau de Congressos, Valencia.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Chris Olah, Benjamin Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint*, abs/2204.05862.

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

Lichang Chen, Chen Zhu, Jiuhai Chen, Davit Soselia, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024. ODIN: disentangled reward mitigates hacking in RLHF. In

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust DPO: aligning language models with noisy feedback. In *Proceedings of the Forty-first International Conference on Machine Learning*, Vienna, Austria.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30*, pages 4299–4307, Long Beach, CA.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *arXiv preprint*, abs/2407.21783.

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

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: model alignment as prospect theoretic optimization.

Leo Gao, John Schulman, and Jacob Hilton. 2024. Scaling laws for reward model overoptimization. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria.

Alexander Havrilla, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. 2024. Glore: When, where, and how to improve LLM reasoning via global and local refinements. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, FL.

Kexin Huang, Junkang Wu, Ziqian Chen, Xue Wang, Jinyang Gao, Bolin Ding, Jiancan Wu, Xiangnan He, and Xiang Wang. 2025. Larger or smaller reward margins to select preferences for alignment? In *Proceedings of the 42nd International Conference on Machine Learning*, Vancouver, Canada.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of LLM via a human-preference dataset. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint*, abs/2310.06825.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, CA.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi Rui Tam, Keith Stevens, Abdullah Barhoum, Duc Nguyen, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. 2023. Openassistant conversations - democratizing large language model alignment. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint*, abs/2403.13787.

Tianle Li, Wei-Lin Chiang, Lisa Dunlap Evan Frick, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From live data to high-quality benchmarks: The arena-hard pipeline. *arXiv preprint*, abs/2406.11939.

Xuechen Li and Tianyi Zhang. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Weixin Liang, James Zou, and Zhou Yu. 2020. Beyond user self-reported likert scale ratings: A comparison model for automatic dialog evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1363–1374, Virtual Event.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *Proceedings of the 12th International Conference on Learning Representations*, Vienna, Austria.

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

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems 38*, Vancouver, Canada.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint*, abs/2303.08774.

Long Ouyang, Jeffrey 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. 2022a. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, New Orleans, LA.

Long Ouyang, Jeffrey 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. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, New Orleans, LA.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. In *Findings of the 62th Annual Meeting of the Association for Computational Linguistics*, pages 4998–5017, Bangkok, Thailand.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint*, abs/2408.00118.

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

Hu Su, Yonghao He, Rui Jiang, Jiabin Zhang, Wei Zou, and Bin Fan. 2022. DSLA: dynamic smooth label assignment for efficient anchor-free object detection. *Pattern Recognit*, 131:108868.

Boxin Wang, Weixin Chen, Hengzhi Pei, Chulin Xie, Mintong Kang, Chenhui Zhang, Chejian Xu, Zidi Xiong, Ritik Dutta, Rylan Schaeffer, Sang T. Truong, Simran Arora, Mantas Mazeika, Dan Hendrycks, Zinan Lin, Yu Cheng, Sanmi Koyejo, Dawn Song, and Bo Li. 2023. Decodingtrust: A comprehensive assessment of trustworthiness inGPT models. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Junkang Wu, Xue Wang, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2025a. α-dpo: Adaptive reward margin is what direct preference optimization needs. In *Proceedings of the 13rd International Conference on Learning Representations*, Singapore.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2025b. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. In *Proceedings of the 13th International Conference on Learning Representations*, Singapore.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. β-dpo: Direct preference optimization with dynamic β. In *Advances in Neural Information Processing Systems 37*, Vancouver, Canada.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of LLM performance in machine translation. In *Proceedings of the 41st International Conference on Machine Learning*, Vienna, Austria.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint*, abs/2412.15115.

Hanyang Zhao, Genta Indra Winata, Anirban Das, Shi-Xiong Zhang, David D. Yao, Wenpin Tang, and Sambit Sahu. 2024a. Rainbowpo: A unified framework for combining improvements in preference optimization. *arXiv preprint*, abs/2410.04203.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. 2024b. Wildchat: 1m chatgpt interaction logs in the wild. In *Proceedings of the 12th International Conference on Learning Representations*, Vienna, Austria.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023b. Secrets of RLHF in large language models part I: PPO. *arXiv preprint*, abs/2307.04964.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: less is more for alignment. In *Advances in Neural Information Processing Systems 36*, New Orleans, LA.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. 2024. WPO: enhancing RLHF with weighted preference optimization. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, Miami, FL.

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

A Proof of Theorem 3.1

Theorem 3.1. *Consider the rDPO loss function defined as*

$$\mathcal{L}_{rDPO} = -\varepsilon \log \sigma(r_l - r_w - \gamma_0) \\ -(1-\varepsilon) \log \sigma(r_w - r_l - \gamma_0), \tag{15}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ *is the sigmoid function, and* ε *is a smoothing parameter.*

If we set $\mathcal{L}_{rDPO} = \mathcal{L}_{\gamma\text{-}PO}$ *for simplicity, and denote* $\delta = \gamma_i - \gamma_0$ *and* $m = r_w - r_l$*, then under the condition* $|\delta| \ll |m|$*, the approximate solution for* ε *is given by*

$$\varepsilon \approx \frac{\delta\,[1-\sigma(m-\gamma_0)]}{\log\left(1+e^{m+\gamma_0}\right) - \log\left(1+e^{\gamma_0-m}\right)}. \tag{16}$$

Proof. Starting with the given equation:

$$-\log \sigma(m - \gamma_0 - \delta) \\ = -\varepsilon \log \sigma(-m-\gamma_0) - (1-\varepsilon)\log \sigma(m-\gamma_0), \tag{17}$$

using the definition of the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ and $-\log\frac{1}{x} = \log x$, we rewrite the equation as:

$$\log\left(1+e^{-(m-\gamma_0-\delta)}\right) = \varepsilon \log\left(1+e^{-(-m-\gamma_0)}\right) \\ +(1-\varepsilon)\log\left(1+e^{-(m-\gamma_0)}\right). \tag{18}$$

Assuming that δ is small, we apply a first-order Taylor expansion around $\delta = 0$ to approximate the left-hand side:

$$\log\left(1+e^{-(m-\gamma_0)}\right) + \delta \cdot \frac{e^{-(m-\gamma_0)}}{1+e^{-(m-\gamma_0)}} \\ \approx (1-\varepsilon)\log\left(1+e^{\gamma_0-m}\right) + \varepsilon\log\left(1+e^{m+\gamma_0}\right). \tag{19}$$

Rearranging the terms to solve for δ, we get:

$$\delta \cdot \frac{e^{\gamma_0-m}}{1+e^{\gamma_0-m}} \\ \approx \varepsilon\left[\log\left(1+e^{m+\gamma_0}\right) - \log\left(1+e^{\gamma_0-m}\right)\right]. \tag{20}$$

Recognizing that $\frac{e^{\gamma_0-m}}{1+e^{\gamma_0-m}} = \sigma(\gamma_0 - m) = 1 - \sigma(m-\gamma_0)$, we express δ as:

$$\varepsilon \approx \frac{\delta\,[1-\sigma(m-\gamma_0)]}{\log\left(1+e^{m+\gamma_0}\right) - \log\left(1+e^{\gamma_0-m}\right)}. \tag{21}$$

□

In addition, as training approaches convergence, as shown in Figure 5, most of the values of γ_i lie near γ_0, while δ remains small under the control of τ, thereby satisfying the assumption $|\delta| \ll |m|$.

B Experiment Details

Training hyperparameters. We observed that the alignment performance of LLMs is influenced by the choice of hyperparameters. To ensure a fair comparison, we conducted a hyperparameter search for all methods, with the search range detailed in Table 6. For the remaining hyperparameters, the learning rate was selected from the range [3e-7, 5e-7, 6e-7, 8e-7, 1e-6]. A consistent batch size of 128 was used across all methods. All models were trained for one epoch using a cosine learning rate schedule, which included a 10% warm-up phase. The Adam optimizer was employed (Kingma and Ba, 2015). Additionally, the maximum sequence length was set to 2048.

Hyperparameter in γ-PO. Table 7 shows the hyperparameters of our method.

Decoding hyperparameters. The decoding hyperparameters are consistent with those used in SimPO*. We would like to express our sincere gratitude to the SimPO team for generously sharing their valuable insights.

Computation environment. All training experiments in this paper were conducted using 8×A100 GPUs, following the procedures outlined in the alignment-handbook repository*.

C Algorithm for calculating γ

Due to GPU memory constraints, the batch size is relatively small, which may introduce sampling bias, leading to significant variations in reward margins across different batches. To mitigate this issue, we introduce an additional queue to reduce bias. Specifically, we implement a First-In-First-Out (FIFO) reward margin queue, denoted as $\mathcal{H}$. This mechanism maintains an updated record of recent reward margins, and incorporating more samples from $\mathcal{H}$ helps better approximate the true distribution of reward margins. The revised pseudocode is provided in Algorithm 2.

For optimizing γ_i, we apply Mirror Descent and the Multiplicative Weight Update Method to efficiently compute γ. The optimization problem for γ is formulated as follows:

$$p = \arg\min_p(-\mathbb{E}_{\mathcal{D}}[\log \sigma(m-\gamma_i) \\ + \tau\,\mathbb{D}_{\text{KL}}(p||p_0))], \tag{22}$$

where m represents the reward margins, $\gamma_i = m_n p_i \gamma_0$, $\sigma(x)$ is the sigmoid function, p is the

*https://github.com/princeton-nlp/SimPO/tree/main/eval
*https://github.com/huggingface/alignment-handbook

Method	Objective	Hyperparameter
DPO (Rafailov et al., 2023)	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$	$\beta\in[0.01, 0.05, 0.1]$
IPO (Azar et al., 2024a)	$\left(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - \frac{1}{2\tau}\right)^2$	$\tau\in[0.01, 0.1, 0.5, 1.0]$
CPO (Xu et al., 2024)	$-\log\sigma\left(\beta\log\pi_\theta(y_w\|x) - \beta\log\pi_\theta(y_l\|x)\right) - \lambda\log\pi_\theta(y_w\|x)$	$\alpha=1.0,\ \ \beta\in[0.01, 0.05, 0.1]$
KTO (Ethayarajh et al., 2024)	$-\lambda_w\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - z_{\text{ref}}\right) + \lambda_l\sigma\left(z_{\text{ref}} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$, where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta\text{KL}\left(\pi_\theta(y\|x)\|\|\pi_{\text{ref}}(y\|x)\right)\right]$	$\lambda_l=\lambda_w=1.0$ $\beta\in[0.01, 0.05, 0.1]$
ORPO (Hong et al., 2024)	$-\log p_\theta(y_w\|x) - \lambda\log\sigma\left(\log\frac{p_\theta(y_w\|x)}{1-p_\theta(y_w\|x)} - \log\frac{p_\theta(y_l\|x)}{1-p_\theta(y_l\|x)}\right)$, where $p_\theta(y\|x) = \exp\left(\frac{1}{\|y\|}\log\pi_\theta(y\|x)\right)$	$\lambda\in[0.1, 0.5, 1.0, 2.0]$
R-DPO (Park et al., 2024)	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - (\alpha\|y_w\| - \alpha\|y_l\|)\right)$	$\alpha\in[0.05, 0.1, 0.5, 1.0]$ $\beta\in[0.01, 0.05, 0.1]$
rDPO (Chowdhury et al., 2024)	$-(1-\varepsilon)\log\sigma(\text{rg}) - \varepsilon\log\sigma(-\text{rg})$ where $\text{rg} = \beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}$	$\beta\in[0.01, 0.05, 0.1]$ $\varepsilon\in[0.01, 0.05, 0.1]$
SimPO (Meng et al., 2024)	$-\log\sigma\left(\frac{\beta}{\|y_w\|}\log\pi_\theta(y_w\|x) - \frac{\beta}{\|y_l\|}\log\pi_\theta(y_l\|x) - \gamma\right)$	$\beta\in[2.5, 5.0, 10.0]$ $\gamma\in[0.3, 0.4, 0.5, 1.0]$
β-DPO (Wu et al., 2024)	$-\log\sigma\left(\beta_i\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta_i\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$ where $\beta_i = \beta_0 + \alpha(M_i - M_0)\beta_0,\ \ M = \beta_0\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta_0\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}$	$\beta\in[2.5, 10.0],\ \ \gamma\in[0.1, 0.3, 0.5]$ $\alpha\in[1e-2, 5e-2, 0.1, 0.2]$

Table 6: Various preference optimization objectives and hyperparameter search range.

Setting	β	γ	τ	Learning rate
Llama3-Instruct	10.0	0.4	10.0	1e-6
Mistral-Instruct	2.5	0.15	10.0	6e-7
Gemma2-Instruct	10.0	0.5	20.0	8e-7
Qwen2.5-Instruct	10.0	0.3	3.0	1e-6

Table 7: The hyperparameter values in γ-PO used for each training setting.

probability distribution over the reward gaps, γ_0 is the initial margin value, p_0 is the uniform distribution (used as the target), and τ is a regularization parameter controlling the KL divergence term. For simplicity, We denote $L = -\mathbb{E}_{\mathcal{D}}[\log\sigma(m-\gamma_i) + \tau\,\mathbb{D}_{\text{KL}}(p\|\|p_0)]$. The gradients of the loss with respect to p are computed as:

$$\frac{\partial L}{\partial p_i} = \frac{\gamma_0}{1+e^{m-m_n p_i \gamma_0}} + \tau\left(1+\log(m_n p_i)\right), \tag{23}$$

where m_n is the number of reward margin m. These gradients are then used to iteratively update p using a multiplicative update rule:

$$p_i \leftarrow p_i \cdot \frac{\exp(-\eta\cdot\text{grad}(p_i))}{\sum_i \exp(-\eta\cdot\text{grad}(p_i))}, \tag{24}$$

where η is the learning rate. Furthermore, we derive the second-order gradients with respect to p as:

$$\frac{\partial^2 L}{\partial p_i^2} = \frac{m_n\gamma_0^2 e^{m-m_n p_i\gamma_0}}{(1+e^{m-m_n p_i\gamma_0})^2} + \frac{\tau}{p_i} > 0, \tag{25}$$

Algorithm 2 γ-PO Algorithm

Input: SFT model π_{SFT}, dataset $\mathcal{D}$, total number of iterations T, learning rate α,
Initial policy model $\pi_\theta = \pi_{\text{SFT}}$, empty FIFO queue $\mathcal{H}$,
for $t = 0$ to T **do**
  Sample a batch of tuples (x, y_w, y_l) from $\mathcal{D}$.
  Sample a reward pair (r_w, r_l) from $\mathcal{H}$.
  Calculate the rewards pair (r_w, r_l) via Table 1 and store them in $\mathcal{H}$.
  Compute the adaptive margin γ_i using Equation (9) with reward pairs (r_w, r_l) both from $\mathcal{H}$ current batch.
  Extract the adaptive margin γ_i from the current batch, calculate the loss $\mathcal{L}_{\gamma\text{-PO}}$ using Equation (10),
  Update the policy parameters θ for π_θ via gradient descent: $\theta_{t+1} = \theta_t - \alpha\nabla_{\theta_t}\mathcal{L}_{\gamma\text{-PO}}$.
end for

which indicates that it is a convex optimization problem and that an optimal solution for p exists.

In our implementation, we set $\eta = \min\left(\frac{0.5}{\tau}, 1\right)$ and $m_n = 256$. The optimization process is repeated $T = 20$ times, which allows for sufficient convergence.

Instruction:	Hi, I'm trying to solve a crossword puzzle, but I've never done one of these before. Can you help me out?
DPO output:	I'd love to help you with your crossword puzzle! Here's how we can break it down to make it easier:\n\n**1. Tell me the specifics:**\n\n* **Clues:** Share the **across** and **down** clues you're stuck on. Be as specific as possible about the number of letters each answer should have (e.g., \"7 letters across\").\n* **Theme (if any):** Some crosswords have a theme. Knowing it can give big hints.\n* **Intersection Help:** If you have some letters already filled in, let me know where they intersect with the clues you need help with. This can narrow down possibilities.\n\n**2. I can offer:**\n\n* **Definitions and synonyms:** I can provide definitions or related words to help you decipher clues.\n* **Wordplay hints:** Crosswords often use puns, anagrams, or other wordplay. I can help spot those.\n* **Letter suggestions:** If you have partial answers, I might be able to suggest missing letters.\n\n\nLet's get started! What's your first clue?
γ-SimPO output:	Absolutely! I can definitely help you with your crossword puzzle. To give you the best assistance, I need a bit more information:* **Can you provide the clues, especially the ones you're stuck on?** I'm good at understanding wordplay, synonyms, and different clue types.* **What's the crossword's theme (if any)?** Themes often give hints about interconnected clues.* **Are there any intersecting letters you already have filled in?** Those can be crucial for narrowing down possibilities. Once you provide these, I'll do my best to help you crack the clues and solve the puzzle!
GPT-4-Turbor(1106) output:	Absolutely! I'd be delighted to help you with your crossword puzzle. Just tell me the clues and the number of letters needed for each answer (and any letters you may have already filled in), and I'll do my best to help you find the solutions. If you have any specific questions about how to approach solving crossword puzzles in general, feel free to ask those as well!

Table 8: Output example from Gemma base model.

D Outputs Example

We present sample outputs generated by the Gemma base model, as shown in Table 8.

E The artifacts

Base models. In this manuscript, we use several publicly available resources, including four base models (LLaMA-3-8B-Instruct (Dubey et al., 2024), Mistral-7B-Instruct (Jiang et al., 2023), Gemma-2-9B-Instruct (Rivière et al., 2024), and Qwen-2.5-7B-Instruct (Yang et al., 2024)), which are licensed under their respective terms for research and non-commercial use. Additionally, we utilize the Ultrafeedback Binarized dataset[*], which is publicly available under the CC-BY-4.0 license for academic and research purposes. The reward model, RLHFlow/ArmoRM-Llama3-8B-v0.1 [*], is also publicly available on Hugging Face with terms permitting non-commercial use.

Evaluation. We evaluate our methods using two datasets—AlpacaEval2 (Dubois et al., 2024) and Arena-Hard (Li et al., 2024)—both of which are publicly available and licensed for research purposes. All assets used in this work are provided with clear attribution and comply with the licensing terms of the respective sources. Full license details and terms of use for each artifact can be found in the accompanying repository and documentation.

Dataset. The original UltraFeedback dataset contains 64k prompts, each accompanied by four model completions from various open and proprietary models. GPT-4 was then used to assign scores to each completion based on criteria such as usefulness and honesty. To create UltraFeedback Binarized, we selected the highest overall score as the "chosen" completion and randomly selected one of the remaining three as the "rejected" completion. This defines the preference modeling split

[*]HuggingFaceH4/ultrafeedback_binarized
[*]RLHFlow/ArmoRM-Llama3-8B-v0.1

Name	**Notation**	**Description**
Prompt	x	Prompt is the input sequence passed to the model.
Response	y	Response is the output sequence of the model.
Preferred Response	y_w	The sample has a higher reward score or is preferred by humans.
Non-preferred Response	y_l	The sample has a lower reward score or non-prefered by humans.
Reward of Preferred Response	r_w	Reward score of y_w.
Reward of Less Preferred Response	r_l	Reward score of y_l.
Reward Margin	m	Reward margin between r_w and r_l, *i.e.*, $m = r_w - r_l$.
Margin Batchsize	m_n	The number of preference pairs in one batch.
Initial Target Margin	γ_0	The initial target margin for preference learning.
Adaptive Target Margin	γ_i	The adaptive target margin for preference learning.
Uniform-distribution	p_0	Uniform distribution of γ_0: $p_i = 1/m_n$.
γ-distribution	p_i	The normalized distribution of γ_i: $p_i = \gamma_i^* / \sum_j \gamma_j^*$, where $\gamma_i^* = \max(0, \gamma_i)$.
Policy Model	π_θ	The generative model accepts prompts and outputs response.
Reference Model	π_{ref}	The generative model is employed as a reference to ensure minimal deviation from the policy model.
Preference Dataset	$\mathcal{D}$	Dataset comprising a set of triples (x, y_w, y_l).
Loss Function	$\mathcal{L}$	Loss function.
Historical Reward Gap Queue	$\mathcal{H}$	The First-In-First-Out queue to maintain the recent reward gaps.
Smoothing Parameter	ε	The initial margin for preference learning, also considered as the label flip rate.
Dynamic Smoothing Parameter	ε_i	The smoothing parameter that adaptive to different samples.
Regularization Hyper-parameters	τ	Regularization Hyper-parameters control the deviation from initial margin.

Table 9: Table of Terminology and Notation.

for techniques like reward modeling or DPO. We also created splits for supervised fine-tuning (SFT), using the "chosen" column as the conversation to be modeled, and for generations (such as rejection sampling or PPO). For detailed information on dataset processing, please refer to the accompanying scripts.

F More Evaluation

We also evaluation our methods on MT-Bench (Zheng et al., 2023a), results are shown in Table 10.

Method	**Llama3-Instruct (8B)**	**Mistral-Instruct (7B)**	**Gemma2-Instruct (9B)**	**Qwen2.5-Instruct (7B)**	Avg. (rank)
SFT	8.1	7.5	8.4	8.5	8.13(11)
IPO	9.1	8.4	9.2	8.8	8.88(8)
KTO	9.1	7.9	9.2	8.9	8.78(10)
CPO	8.9	8.8	8.8	8.9	8.85(9)
ORPO	8.8	**9.1**	8.9	8.9	8.93(7)
R-DPO	9.0	8.8	9.2	9.0	9.00(5)
DPO	9.1	8.9	9.2	9.0	9.05(3)
β-DPO	9.2	8.7	**9.3**	9.0	9.05(3)
SimPO	9.0	8.9	9.0	9.0	8.98(6)
γ-DPO	9.2	8.8	**9.3**	**9.1**	9.10(2)
γ-SimPO	**9.3**	**9.1**	9.2	**9.1**	**9.18(1)**
Improve	+1.1%	+0.0%	+0.0%	+1.1%	+1.4%

Table 10: The Win Rate from MT-Bench (Zheng et al., 2023a) across different methods. **Bold** indicates the best performance for each metric, while underlined values represent the best performance excluding our methods, *i.e.,* γ-DPO and γ-SimPO. “Improve” denotes the percentage improvement of the bold value over the underlined one. This metric is considered better when it has a higher value.