Length-Controlled Margin-Based Preference Optimization without Reference Model

Anonymous ACL submission

Abstract

Direct Preference Optimization (DPO) is a widely adopted offline algorithm for preference-based reinforcement learning from human feedback (RLHF), designed to improve training simplicity and stability by redefining reward functions. However, DPO is hindered by several limitations, including length bias, memory inefficiency, and probability degradation. To address these challenges, we propose Length-Controlled Margin-Based Preference Optimization (LMPO), a more efficient and robust alternative. LMPO introduces a uniform reference model as an upper bound for the DPO loss, enabling a more accurate approximation of the original optimization objective. Additionally, an average log-probability optimization strategy is employed to minimize discrepancies between training and inference phases. A key innovation of LMPO lies in its Length-Controlled Margin-Based loss function, integrated within the Bradley-Terry framework. This loss function regulates response length while simultaneously widening the margin between preferred and rejected outputs. By doing so, it mitigates probability degradation for both accepted and discarded responses, addressing a significant limitation of existing methods. We evaluate LMPO against state-of-the-art preference optimization techniques on two open-ended large language models, Mistral and LLaMA3, across six conditional benchmarks. Our experimental results demonstrate that LMPO effectively controls response length, reduces probability degradation, and outperforms existing approaches.

1 Introduction

Human feedback is essential for aligning large language models (LLMs) with human values and objectives (Jiang et al., 2024; Chang et al., 2024), ensuring that these models act in ways that are helpful, reliable, and safe. A common strategy for achieving this alignment is reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022), which fine-tunes language models using human evaluations. While RLHF has shown substantial success (Schulman et al., 2017), it also introduces notable challenges in optimization due to its multi-step design. This process first involves training a reward model to evaluate outputs based on human preferences, and then optimizing a policy model to maximize the assigned rewards. The complexity of these sequential steps often complicates the implementation and reduces efficiency (Chaudhari et al., 2024).

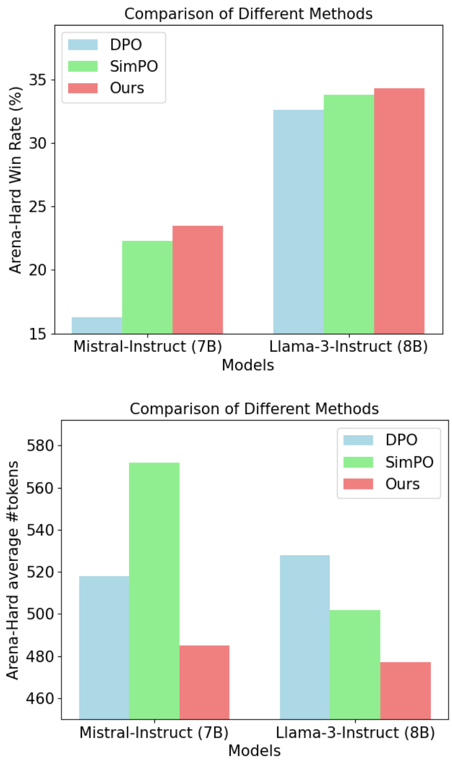

Figure 1: Comparison with DPO and SimPO under the Mistral-Instruct and Llama3-Instruct models in the Arena-Hard benchmark. Our proposed method, LMPO, achieves the highest win rate while utilizing an exceptionally low average token count across both models.

In response to these challenges, researchers have started exploring simpler alternatives that avoid the intricate, multi-stage nature of RLHF. One promising method is Direct Preference Optimization (DPO) (Rafailov et al., 2024), which streamlines the process by reformulating the reward function. This approach enables direct learning of a policy model from preference data, eliminating the need for a separate reward model. As a result, DPO offers greater stability and is more practical to implement.

DPO estimates implicit rewards using the log-probability ratio between a policy model's response and that of a supervised fine-tuned (SFT) model, enabling preference learning without an explicit reward function. However, this implicit reward may misalign with the log-probability metric during inference. Moreover, DPO's reliance on both policy and SFT models significantly increases GPU usage, especially for LLMs. The DPO loss, derived from the Bradley-Terry model, can create training imbalances, as it does not ensure an increase in the probability of positive samples—potentially reducing both positive and negative probability simultaneously. Unlike IPO (Azar et al., 2024), which constrains probability variation but weakens response distinction, DPO also exhibits length bias, favoring longer responses due to preference label distribution inconsistencies (Lu et al., 2024). This issue, common in multi-stage RLHF methods, allows models to exploit verbosity for higher rewards without improving output quality, often generating responses nearly twice as long as labeled data.

To address these challenges, we introduce a novel approach incorporating a length-controlled margin-based loss function to mitigate both length bias and probability reduction. Our method consists of two key components: (1) a reference-free loss function that reduces memory inefficiency and aligns generation metrics via average log probability, and (2) a Length-Controlled Margin-Based term with two kinds of normalization methods, which minimizes probability reduction while alleviating length bias and preserving model performance. In summary, our method offers the following advantages:

- **Memory efficiency**: Our method does not rely on an extra reference model, making it more lightweight and easier to implement compared to DPO and other reference-dependent methods.
- **Reduction of length bias and probability decrement**: By incorporating a specially designed margin-based term, our method effectively reduces both positive and negative probability decrements, similar to traditional NLL loss, while also addressing length bias without impairing model performance.
- **Competitive performance**: Despite being reference-free, our method demonstrates competitive performance when compared to DPO and its variants (Hong et al., 2024a; Ethayarajh et al., 2024). This performance advantage is consistent across a variety of training setups and comprehensive instruction-following benchmarks, including AlpacaEval 2 (Li et al., 2023) and Arena-Hard v0.1 (Li et al., 2024).

2 Related Work

Alignment with Reinforcement Learning Reinforcement learning with human feedback (RLHF) often utilizes the Bradley-Terry model (Bradley and Terry, 1952) to estimate the probability of success in pairwise comparisons between two independently evaluated instances. Additionally, a reward model is trained to assign scores to these instances. Reinforcement learning algorithms, such as proximal policy optimization (PPO) (Schulman et al., 2017), are used to train models to maximize the reward model's score for the selected response, ultimately enabling LLMs to align with human preferences (Stiennon et al., 2020; Ziegler et al., 2019). A notable example is InstructGPT (Ouyang et al., 2022), which showcased the scalability and adaptability of RLHF in training instruction-following language models. Alternative approaches, such as reinforcement learning with language model feedback (RLAIF (Lee et al., 2023)), may also serve as feasible substitutes for human feedback (Bai et al., 2022; Sun et al., 2023). Nevertheless, RLHF encounters challenges, including the need for extensive hyperparameter tuning due to the instability of PPO (Rafailov et al., 2024) and the sensitivity of the reward models (Wang et al., 2024). Consequently, there is a pressing demand for more stable preference alignment algorithms.

Alignment Without Reward Models Several techniques for preference alignment reduce the reliance on reinforcement learning. Direct Policy Optimization (DPO) (Rafailov et al., 2024) is a method that integrates reward modeling with preference learning. And Identity Preference Optimization

(IPO) (Azar et al., 2024) is introduced to mitigate potential overfitting issues in DPO. In contrast to RLHF and DPO, an alternative approach called Kahneman-Tversky Optimization (KTO) (Ethayarajh et al., 2024) is proposed, which does not require pairwise preference datasets. Additionally, Preference Ranking Optimization (PRO) (Song et al., 2024) introduces the incorporation of the softmax values from the reference response set into the negative log-probability (NLL) loss, allowing for a unified approach to supervised fine-tuning and preference alignment.

Alignment Without Reference Models Due to the reliance of DPO and DPO-like methods on both the policy model and the SFT model during the alignment process, they impose greater demands on GPU resources. Several techniques have been developed to alleviate this GPU requirement by eliminating the need for a reference model. CPO (Xu et al., 2024) demonstrates that the ideal loss function without a reference model can serve as the upper bound of the DPO loss, with the SFT loss acting as a replacement for the KL divergence. ORPO (Hong et al., 2024a) models the optimal reward as a log-odds function, removing the need for an additional fixed reference model. MaPO (Hong et al., 2024b) builds on the ORPO approach by introducing a margin-aware term for aligning diffusion models without a reference model. SimPO (Meng et al., 2024) adopts a similar reference-free preference learning framework as CPO but with improved stability due to its specific length normalization and target reward margin, leading to superior performance in various benchmarks.

3 Method

In this section, we begin by briefly introducing the main concept of DPO. We then propose a uniform, reference-free model based on average log-probability to address the memory and speed inefficiencies of DPO. Next, we incorporate a margin term with two kind of normalization and design a length-controlled margin-based loss function to fully leverage its benefits. Finally, we provide a detailed explanation of the margin term, illustrating how it reduces length bias and mitigates the probability decrement.

3.1 Direct Preference Optimization (DPO)

We derive our method by first examining DPO (Rafailov et al., 2024), which provides a more straightforward optimization goal within the framework of RLHF (Ziegler et al., 2019; Stiennon et al., 2020). DPO operates on a dataset of source sentences, x, paired with both preferred translations, y_w, and less preferred ones, y_l. This dataset, containing comparison examples, is denoted as $\mathcal{D} = \left\{x^{(i)}, y_w^{(i)}, y_l^{(i)}\right\}_{i=1}^{N}$. The loss function for DPO is formulated as a maximum likelihood estimation for a policy model parameterized by π_θ:

$$\mathcal{L}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\Big] \tag{1}$$

where π_{ref} refers to a SFT model, σ represents the sigmoid function, and β is a scaling hyperparameter. The formulation of the DPO loss is based on a reparameterization of the true reward signal and the corresponding optimal policy, borrowing from the PPO framework (Schulman et al., 2017). This loss allows DPO to be trained in a supervised fine-tuning manner, as it makes exclusive use of labeled preference data without requiring any interaction between the agent and its environment which is a shortcoming for PPO.

3.2 Revisiting Bradley-Terry Model

DPO in Section 3.1 uses a statistical model commonly used for sporting events called Bradley-Terry. The Bradley-Terry model stipulates that the human preference distribution p^* can be written as:

$$p^*(y_w \succ y_l \mid x) = \frac{\exp(r^*(x,y_w))}{\exp(r^*(x,y_w))+\exp(r^*(x,y_l))}. \tag{2}$$

The BT model used in DPO is the original form. There are some variants that make some improvements on the BT model. Rao-Kupper model (Rao and Kupper, 1967) considers model human preference with ties: $p^*(y_w = y_l \mid x)$, which means two responses (y_w, y_l) are considered equal with respect to the prompt x.

So in order to better distinguish the two responses, we define the loss response as a home-filed team in the BT model. And we may incorporate a home-court advantage by including an intercept term h:

$$p^*(y_w \succ y_l \mid x) = \frac{\exp\left(r^*(x, y_w)\right)}{\exp\left(r^*(x, y_w)\right) + h \exp\left(r^*(x, y_l)\right)} = \frac{1}{1 + h \exp\left(-d(x, y_w, y_l)\right)}. \tag{3}$$

For DPO, $d(x, y_w, y_l)$ means the term in function σ, which is outlined in Section 3.1. DPO mitigates several issues inherent in conventional RLHF techniques and has found widespread application in modern models, including Meta's recently released Llama 3.1 model (Dubey et al., 2024). Despite these advantages, DPO presents notable drawbacks when compared to standard supervised fine-tuning. One major limitation is its inefficiency in memory usage, as it requires doubling the memory to accommodate both the trained policy and the reference policy concurrently. Additionally, DPO suffers from reduced computational efficiency, as the model must be executed separately for each policy, effectively doubling the processing time. So it is of vital importance to investigate a reference model-free RLHF method.

A recent method called CPO(Xu et al., 2024) has proved that when π_{ref} is defined as π_w, an ideal policy that precisely aligns with the true data distribution of preferred data, the DPO loss $\mathcal{L}(\pi_\theta; \pi_w) + C$ is upper bounded by $\mathcal{L}(\pi_\theta; U)$, where C is a constant. So following this proof, we use a uniform reference model to approximate $d(x, y_w, y_l)$:

$$d(x, y_w, y_l) = \log \pi_\theta(y_w|x) - \log \pi_\theta(y_l|x). \tag{4}$$

Next, in DPO, the implicit reward is formulated using the log ratio of the probability of a response between the current policy model and the SFT model. However, this reward formulation is not directly aligned with the metric used to guide generation, which is approximately the average log probability of a response generated by the policy model. So there is an assumption that this discrepancy between training and inference phases may lead to bad performance. In order to eliminate this discrepancy, we replace the log probability with the average log probability in Eq. 4:

$$d(x, y_w, y_l) = \frac{\beta}{|y_w|} \log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|} \log \pi_\theta(y_l|x). \tag{5}$$

3.3 Length-Controlled Margin-Based Loss

To ensure a more pronounced separation in reward scores for responses with greater quality differences, we incorporate a margin term into the Bradley-Terry framework. The modified objective is as follows:

$$d(x, y_w, y_l) = r^*(x, y_w) - r^*(x, y_l) - \lambda m(y_w, y_l, x). \tag{6}$$

Here, $m(y_w, y_l, x)$ represents a margin that quantifies the preference strength between the winning response y_w and the losing response y_l for a given input x, while λ is a scaling factor. The function $r^*(x, y)$ provides the reward score for response y conditioned on input prompt x. By including this margin, the model is better able to differentiate reward scores, especially when the quality gap between responses is substantial.

Recent approaches have adopted this formulation to enhance model performance. For example, the reward models in Llama-2-Chat (Touvron et al., 2023) and UltraRM (Cui et al., 2023) use discrete preference scores as margin terms. SimPO (Meng et al., 2024) employs a fixed margin to guarantee that the reward for the preferred response always exceeds that of the less favored one. Despite these advances, issues such as length bias persist.

In response to this issue, we introduce the Length-Controlled Margin-Based Loss, which is designed to address several key limitations. First, it explicitly controls the length of generated responses, thereby mitigating the bias towards longer outputs often seen in LLMs. Additionally, the loss function regulates the probability decrease for both selected and rejected responses, further ensuring that the model can more clearly distinguish between correct and incorrect responses. Importantly, this framework also aims to increase the margin between the probabilities of chosen and rejected responses, thus amplifying the model's capacity to discriminate between high- and low-quality responses. The full formulation of the Length-Controlled Margin-Based Loss is presented below.

$$m(x, y_w, y_l) = \left(1 - p_\theta(y_w|x)\right) \cdot \left(1 - \left(p_\theta(y_w|x) - p_\theta(y_l|x)\right)^5\right). \tag{7}$$

Normalization: To enhance training stability and regulate the length of model outputs, we employ two distinct normalization techniques: average length normalization and Z-score normalization (Patro, 2015).

(1) average length normalization: To mitigate length bias in LLM-generated outputs, we intro-

duce a dynamic scaling factor, defined as $\frac{|y_w|+|y_l|}{2*|y|}$ to adjust the rewards for both chosen and rejected outputs. This factor is incorporated into Eq. 7, modifying the probability formulation as follows:

$$p_\theta(y|x) = \exp\left(\frac{1}{|y|}\log \pi_\theta(y|x) * \frac{|y_w| + |y_l|}{2 * |y|}\right) \tag{8}$$

(2) Z-score normalization: To stabilize training and prevent the loss from being dominated by scale variations in $m(y_w, y_l, x)$, we apply Z-score normalization to m, yielding:

$$\overline{m}(x, y_w, y_l) = \frac{m(x, y_w, y_l) - a_m}{b_m}, \tag{9}$$

where a_m and b_m denote the mean and standard deviation of m computed over the entire training process.

Objective. Finally, we obtain the LMPO finall loss function by incorporating the above considerations:

$$\mathcal{L}_{\text{LMPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\left(\frac{1}{1+h\exp(-d(x,y_w,y_l))}\right)\right]. \tag{10}$$

where

$$d(x, y_w, y_l) = \frac{\beta}{|y_w|}\log \pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log \pi_\theta(y_l|x) - \lambda\overline{m}(x, y_w, y_l). \tag{11}$$

In summary, LMPO employs an implicit reward formulation that directly aligns with the generation metric, eliminating the need for a reference model. Next, it introduces a margin term $m(\mathbf{x}, \mathbf{y}^w, \mathbf{y}^l)$ with two kinds of normalization methods to help separate the winning and losing responses, alleviate length bias and wining response probability decrement problems.

4 Experiment

4.1 Experimental Setup

Models and training settings. We perform preference optimization with two families of models, Llama3-8B(AI@Meta, 2024) and Mistral-7B(Jiang et al., 2023) under two setups: Base and Instruct.

For the Base experimental setup, following SimPO, we utilize pre-trained models (alignment-handbook/zephyr-7b-sft-full) (Tunstall et al., 2023) and (princeton-nlp/Llama-3-Base-8B-SFT) as SFT models. These SFT models are then used as the foundation for preference optimization on the UltraFeedback dataset (Cui et al., 2023), which collects feedback (y_w and y_l) from LLMs of different quality levels.

For the Instruct experimental setup, we utilize pre-trained instruction-tuned models (mistralai/Mistral-7B-Instruct-v0.2) and (meta-llama/Meta-Llama-3-8B-Instruct) as SFT models. For a fair comparison, we use the same training data as SimPO: (princeton-nlp/llama3-ultrafeedback) and (https://huggingface.co/datasets/princeton-nlp/mistral-instruct-ultrafeedback) for Llama3-8B and Mistral-7B, respectively.

These configurations embody the latest advancements, securing our models a place among the top contenders on various leaderboards.

Evaluation Benchmarks. We evaluate our models using two widely recognized open-ended instruction-following benchmarks: AlpacaEval 2 (Li et al., 2023) and Arena-Hard v0.1 (Li et al., 2024). These benchmarks evaluate the models' conversational abilities across a wide range of queries and are widely used by the research community (Chang et al., 2024). For AlpacaEval 2, we report both the raw win rate (WR) and the length-controlled win rate (LC) (Dubois et al., 2024), with the LC metric designed to mitigate the effects of model verbosity. For Arena-Hard, we report the win rate (WR) against a baseline model.

Additionally, we evaluate the models on six downstream tasks in the Huggingface Open Leaderboard V1, following SimPO (Meng et al., 2024). These downstream tasks include the AI2 Reasoning Challenge (25-shot) (Clark et al., 2018), HellaSwag (10-shot) (Zellers et al., 2019), MMLU (5-shot) (Hendrycks et al., 2020), TruthfulQA (0-shot) (Lin et al., 2021), Winogrande (5-shot) (Sakaguchi et al., 2021), and GSM8K (5-shot) (Cobbe et al., 2021). We report the match accuracy for these conditional benchmarks. Additional details are provided in Appendix A.

Baselines We perform a comparative analysis of our method against several state-of-the-art offline preference optimization techniques, including DPO (Rafailov et al., 2024), IPO (Azar et al., 2024), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024a), R-DPO (Park et al., 2024), and SimPO (Meng et al., 2024). For SimPO, we use the model provided for the Llama3-8B family and replicate the SimPO methodology for the Mistral-7B family in our environment. For

Table 1: AlpacaEval 2 and Arena-Hard results under the four settings. LC and WR denote length-controlled and raw win rate, respectively. Length denotes the length of the generated prompt. We train SFT models for Base settings on the UltraChat dataset. For Instruct settings, we follow the training process of SimPO.

Method	**Mistral-Base (7B)**					**Mistral-Instruct (7B)**				
	AlpacaEval 2			**Arena-Hard**		**AlpacaEval 2**			**Arena-Hard**	
	LC (%)	**WR (%)**	**Length**	**WR (%)**	**Length**	**LC (%)**	**WR (%)**	**Length**	**WR (%)**	**Length**
SFT	6.2	4.6	1082	3.3	437	17.1	14.7	1676	12.6	486
DPO	15.1	12.5	1477	10.4	628	26.8	24.9	1808	16.3	518
IPO	11.8	9.4	1380	7.5	674	20.3	20.3	2024	16.2	740
CPO	9.8	8.9	1827	5.8	823	23.8	28.8	3245	22.6	812
KTO	13.1	9.1	1144	5.6	475	24.5	23.6	1901	17.9	496
ORPO	14.7	12.2	1475	7.0	764	24.5	24.9	2022	20.8	527
R-DPO	17.4	12.8	1335	9.9	528	27.3	24.5	1784	16.1	495
SimPO	17.7	16.5	1803	14.3	709	29.7	31.7	2350	22.3	572
LMPO	20.9	14.9	1351	13.8	458	29.8	28.0	1881	23.5	485
Method	**Llama-3-Base (8B)**					**Llama-3-Instruct (8B)**				
	AlpacaEval 2			**Arena-Hard**		**AlpacaEval 2**			**Arena-Hard**	
	LC (%)	**WR (%)**	**Length**	**WR (%)**	**Length**	**LC (%)**	**WR (%)**	**Length**	**WR (%)**	**Length**
SFT	8.4	6.2	914	1.3	521	26.0	25.3	1920	22.3	596
DPO	18.2	15.5	1585	15.9	563	40.3	37.9	1883	32.6	528
IPO	14.4	14.2	1856	17.8	608	35.6	35.6	1983	30.5	554
CPO	12.3	13.7	2495	11.6	800	28.9	32.2	2166	28.8	624
KTO	14.2	12.4	1646	12.5	519	33.1	31.8	1909	26.4	536
ORPO	12.2	10.6	1628	10.8	639	28.5	27.4	1888	25.8	535
R-DPO	17.6	14.4	1529	17.2	527	41.1	37.8	1854	33.1	522
SimPO	21.6	20.0	1818	26.9	877	43.9	39.0	1788	33.8	502
LMPO	21.3	17.7	1601	30.1	1114	43.7	39.0	1791	34.3	477

the other methods, we report the results provided by SimPO. We also tune the hyperparameters for SimPO and report the best performance achieved.

4.2 Main Results

LMPO achieves competitive performance compared to existing preference optimization methods with controlled length. As shown in Table 1, while all preference optimization algorithms improve over the SFT baseline, LMPO achieves competitive performance compared to existing methods specifically on AlpacaEval 2 and Arena-Hard with controlled length.

AlpacaEval 2: The prompt lengths of LMPO are significantly shorter than those of SimPO in three of the evaluated settings. Notably, in the case of Mistral-Base (7B), LMPO outperforms SimPO by 3.2% in the LC metric, despite utilizing markedly shorter prompt lengths. These results suggest that while LMPO may not lead in terms of LC and WR, its capacity to achieve competitive performance with more efficient prompt lengths positions it as a well-rounded model. It strikes a favorable balance between performance and efficiency, making it particularly suitable for practical applications where both speed and quality are crucial.

Arena-Hard: LMPO achieves the highest win rate while maintaining a shorter prompt length compared to many competitors, making it the most efficient in terms of both performance and prompt length. Its ability to excel in competitive tasks while preserving prompt efficiency positions it as a top choice for complex environments. It is worth noting that the prompt length in the Llama-3-Base (8B) setting is unusually longer than that of other methods. This may be due to the updated Llama-3 tokenizer occasionally introducing two BOS tokens, which can influence the evaluation results.

Overall, LMPO offers a best-in-class combination of strong performance and prompt efficiency, particularly in Arena-Hard, while remaining highly competitive in AlpacaEval 2. Its ability to balance concise outputs with high-quality performance makes it one of the most practical and effective models across these benchmarks.

The importance of the design on the loss term.

Table 2: Ablation studies under Llama-3-Base (8B) settings. We report the win rate and 95% confidence interval for Arena-Hard.

Method	Arena-Hard			
	WR (%)	95 CI high (%)	95 CI low (%)	Length
SimPO	26.9	28.7	25.1	877
LMPO	30.1	32.4	27.7	1114
w/o Z-score normalization	22.5	25.0	20.0	630
w/o avg-length normalization	27.9	29.6	26.2	843
log function	27.9	30.1	25.9	770
cube function	29.3	31.7	27.4	903
sigmoid function	25.2	27.3	22.5	649

As the core contribution of LMPO is to propose a novel loss term $m(x, y_w, y_l) = (1 - p_\theta(y_w|x)) \cdot (1 - (p_\theta(y_w|x) - p_\theta(y_l|x))^5)$, we also evaluate other variants of the reference model. Specifically, we compare LMPO with three variants:

- log function: $m(x, y_w, y_l) = (1 - p_\theta(y_w|x)) \cdot \left(\frac{1}{\alpha} log(\frac{1-(p_\theta(y_w|x)-p_\theta(y_l|x))}{1+(p_\theta(y_w|x)-p_\theta(y_l|x))}) + 0.5\right)$
- cube function: $m(x, y_w, y_l) = (1 - p_\theta(y_w|x)) \cdot (1 - (p_\theta(y_w|x) - p_\theta(y_l|x))^3)$
- sigmoid function: $m(x, y_w, y_l) = (1 - p_\theta(y_w|x)) \cdot \left(\frac{1}{1+\exp(\frac{p_\theta(y_w|x)-p_\theta(y_l|x)}{\beta})}\right)$

where α is a hyperparamater for log function and β is a hyperparamater for sigmoid function.

As shown in Table 2, most of the variants outperform SimPO, highlighting the significance of the loss term. Furthermore, our proposed reference model consistently exceeds the performance of other variants, demonstrating the effectiveness of the proposed design. However, the prompt length of our loss term is the longest among the options, which may affect performance. The log function achieves better performance with a shorter length compared to SimPO. Therefore, exploring improved loss functions will be a key direction for future experiments in LMPO.

All key designs in LMPO are crucial. To further assess the impact of various components in LMPO, we conduct ablation studies by removing key elements. As shown in Table 2, removing Z-score normalization and average-length normalization leads to significant performance drops, underscoring the importance of these components in LMPO. However, removing these two terms reduces the prompt length, suggesting a need to balance model performance with prompt length. Additionally, due to resource limitations, certain aspects of LMPO, such as the home-court advantage, were not removed, which presents an opportunity for future research.

Figure 2: The curves of the chosen (top) and rejected (bottom) log-probabilities during the training process in the Llama-3-Base (8B) setting. The red and green curves represent LMPO and SimPO, respectively.

5 Discussion

5.1 Reduction of probability decrement

First we introduce the loss function SimPO, the loss function for SimPO is formulated as a maximum likelihood estimation for a policy model parameterized by π_θ:

$$\mathcal{L}_{\textbf{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \gamma\right)\right]. \tag{12}$$

where γ is a hyperparameter call target reward margin, which is a constant with no gradient.

The primary optimization objective in Eq. 12 is to maximize the margin between the chosen and rejected probabilities, without directly controlling either of them. This lack of control may result in a reduction in both probabilities during training. Furthermore, a decrease in the chosen probability contradicts the goal of aligning the language model with human preferences.

In LMPO, we introduce a constraint term, $1 - p_\theta(y_w|x)$. By minimizing the loss function, LMPO effectively maximizes the exponentiated

Table 3: AlpacaEval 2 results for Hyperparameter Selection under Mistral-Base (7B) settings. LC and WR denote length-controlled and raw win rate, Length denotes the length of the generated prompt, STD means standard deviation of win rate.

Method	AlpacaEval 2			
	Lc (%)	WR (%)	STD (%)	Length
λ=0.05	16.1	14.6	1.1	1751
λ=0.2	16.6	15.0	1.0	1726
λ=1.0	20.9	14.9	1.1	1351

log-probability, implicitly imposing a constraint on the log-probability. It is worth noting that the constraint term we use is similar to the SFT loss employed in CPO (Xu et al., 2024). However, relying solely on the SFT loss can impose an excessive constraint, which may negatively impact the performance of the method. Therefore, we combine the latent constraint term with a margin term to balance the reduction of probability decrement while maximizing the margin.

As shown in Figure 2, it is evident that LMPO imposes a constraint on the log-probabilities of both chosen and rejected responses, in contrast to SimPO. Despite this constraint, LMPO is still able to maximize the margin between these two probabilities, with the margins being similar to those of SimPO. By reducing the probability decrement and maximizing the margin, LMPO can achieve competitive performance when compared to SimPO.

5.2 Hyperparameter Selection

As shown in Eq. 11, LMPO employs a hyperparameter λ to control the margin loss term. Additionally, since Z-score normalization is applied to compute the overall margin loss during the training process, adjusting λ can significantly affect $\overline{m}(x, y_w, y_l)$, thereby influencing the model's preferences.

We selected three values for the hyperparameter λ: 0.05, 0.2, and 1.0, and applied them to the LMPO algorithm under the Mistral-Base (7B) setting. The results of AlpacaEval 2 are presented in Table 3. It is evident that as λ increases, the WR remains relatively stable, while the LC increases with λ, and the length of the generated prompt decreases. These findings suggest that LMPO has a notable impact on prompt length control and performs well in scenarios requiring length regulation.

To demonstrate the effect of hyperparameter selection on the reduction of probability decrement, we present the training curves for these three training processes. The results are shown in Figure 3.

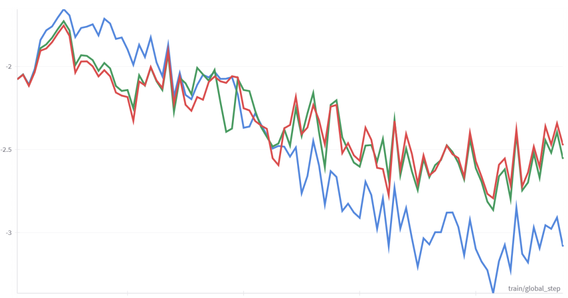

Figure 3: The curves of the chosen log-probabilities during the training process in the Mistral-Base (7B) setting. The red, green and blue curves represent λ=0.05, λ=0.2 and λ=1.0, respectively.

It is clear that as λ increases, the log-probabilities of the selected prompts decrease significantly, and the corresponding curves decline rapidly. These findings indicate that increasing λ may adversely affect the latent constraint mechanism in LMPO, which is undesirable for its intended performance.

Therefore, selecting an appropriate hyperparameter for LMPO is crucial, as it depends on the specific scenario. Choosing an optimal hyperparameter can strike a balance between achieving better performance in a length-controlled setting and minimizing the reduction in probability decrement.

6 Conclusion

In this paper, we introduce LMPO, which uses a length-controlled margin-based loss function to mitigate length bias and probability reduction. It features a reference-free loss for memory efficiency and a margin-based term with two normalization methods to balance probability control and model performance. Without requiring a reference model, it remains lightweight while effectively reducing length bias and probability decrement. Despite its simplicity, the method achieves competitive results compared to DPO and its variants across multiple benchmarks, including two open-ended benchmarks: AlpacaEval 2, Arena-Hard v0.1 and six conditional benchmarks used in Huggingface open leaderboard V1.

Limitations

The constraints of LMPO are outlined as follows:

Settings. The settings we use in our paper are based on those from the early version of SimPO. In later versions, SimPO adopts other configurations, such as Llama-3-Instruct v0.2 and Gemma. For a more in-depth analysis, updating the settings is necessary.

Performance. LMPO does not outperform SimPO in AlpacaEval 2 and struggles with downstream tasks, particularly underperforming in mathematical settings like GSM8K. To improve its performance, further updates are needed, such as selecting a better loss function and employing more effective normalization methods. Additionally, the updated Llama3 tokenizer occasionally introduces two BOS tokens, which can impact evaluation results. For example, this causes an unusually long generated prompt for LMPO in AlpacaEval 2 under the Llama-3-Base setting. Therefore, it may be necessary to use the pre-update Llama3 tokenizer.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card. *Github*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

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.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Shreyas Chaudhari, Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, Ameet Deshpande, and Bruno Castro da Silva. 2024. Rlhf deciphered: A critical analysis of reinforcement learning from human feedback for llms. *arXiv preprint arXiv:2404.08555*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv: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 arXiv:2404.04475*.

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

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*, 10:8–9.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024a. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.

Jiwoo Hong, Sayak Paul, Noah Lee, Kashif Rasul, James Thorne, and Jongheon Jeong. 2024b. Margin-aware preference optimization for aligning diffusion models without reference. *arXiv preprint arXiv:2406.06424*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Ruili Jiang, Kehai Chen, Xuefeng Bai, Zhixuan He, Juntao Li, Muyun Yang, Tiejun Zhao, Liqiang Nie, and Min Zhang. 2024. A survey on human preference learning for large language models. *arXiv preprint arXiv:2406.11191*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. 2024. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*.

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

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

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. *arXiv preprint arXiv:2406.10957*.

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

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

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

S Patro. 2015. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

PV Rao and Lawrence L Kupper. 1967. Ties in paired-comparison experiments: A generalization of the bradley-terry model. *Journal of the American Statistical Association*, 62(317):194–204.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

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

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

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

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

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.

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. *arXiv preprint arXiv:2401.08417*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

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

A Evaluation Details

We outline the specifics of our evaluation framework as follows:

- AI2 Reasoning Challenge: A benchmark for evaluating AI scientific reasoning, consisting of 2,590 multiple-choice questions (Clark et al., 2018). Each question tests science knowledge and reasoning, with highly challenging distractors designed to confuse non-expert models.

- HellaSwag: A benchmark for testing AI commonsense reasoning, consisting of 70,000 multiple-choice questions (Zellers et al., 2019). Each question has a context and four endings, with one correct answer. Adversarial distractors make it highly challenging.

- MMLU: A benchmark for evaluating AI across several diverse tasks, including reasoning, knowledge, and language understanding (Hendrycks et al., 2020). It consists of over 12,000 multiple-choice questions, testing models' performance on tasks ranging from general knowledge to specialized domains.

- TruthfulQA: a benchmark for evaluating AI's ability to generate truthful and factual answers, consisting of 818 multiple-choice questions (Lin et al., 2021). It tests models' capacity to provide accurate information across various domains, with distractors designed to confuse models into providing false answers.

- Winogrande: A benchmark for evaluating AI commonsense reasoning, consisting of 44,000 sentence-pair questions (Sakaguchi et al., 2021). Each question requires selecting the correct word to resolve an ambiguity, with challenging distractors that test subtle reasoning abilities.

- GSM8K: A benchmark for evaluating AI's performance on arithmetic problem solving, consisting of 8,000 high-school-level math word problems (Cobbe et al., 2021). It tests models' ability to reason through multi-step calculations and select the correct solution from multiple choices.

- AlpacaEval2: An open-ended, AI-driven generation benchmark designed to compare model performance (Li et al., 2023). The dataset comprises 805 diverse questions and evaluates model responses against GPT-4, with GPT-4 serving as the judge (Achiam et al., 2023). Additionally, we include a length-debiased win rate to minimize potential biases favoring longer responses (Dubois et al., 2024).

- Arena-Hard v0.1: Arena-Hard is an enhanced version of MT-Bench, consisting of 500 high-quality prompts sourced from real user queries (Li et al., 2024). GPT-4(0613) is used as the baseline model, while GPT-4-Turbo serves as the evaluator. We measure the win rate against the baseline model.

We categorize the first six datasets as conditional benchmarks, and the last two as open-ended benchmarks. Conditional benchmarks require the model to produce answers in a specific format, enabling the calculation of exact match scores or accuracy. Open-ended benchmarks, on the other hand, allow for free-form responses, providing more flexibility in evaluating the model's performance.

For all conditional benchmarks, we employ the well-established evaluation tool lm-evaluation-harness (Gao et al., 2021).And in order to follow Huggingface open leaderboard V1, we use the same version of lm-eval repository. [1]

B Downstream Result Analysis

To demonstrate the effectiveness of our method, we first adhere to established evaluation protocols and report the results of downstream tasks on the Hugging Face Open Leaderboard V1 for all models, as shown in Table 4. Overall, our findings indicate that the impact of our method varies across different tasks.

Minimal degradation in knowledge and reasoning abilities. Compared to the SFT model and other preference optimization methods, our approach largely maintains MMLU performance with only a slight decline. This suggests that our method is effective in preserving both knowledge and reasoning capabilities.

Enhancement of Scientific and Commonsense Reasoning. For ARC and HellaSwag benchmarks, our method generally improves performance compared to the SFT model and demonstrates competitive effectiveness relative to other preference optimization methods. This improvement can be attributed to the preference optimization dataset we used, which contains prompts related to scientific reasoning and commonsense reasoning—domains that closely align with these tasks. Consequently, our method enhances the SFT model's capabilities in these areas.

Enhancement of Truthfulness. For truthfulqa task, we find that our method improves TruthfulQA performance compared to the SFT model and nearly all other preference optimization methods. This improvement can be attributed to the preference optimization dataset, which includes instances that emphasize truthfulness. As a result, the model gains a better understanding of context and generates more truthful responses.

Decline in Mathematical Performance. For the GSM8K task, our method leads to a decline in performance compared to the SFT model and other preference optimization methods. Notably, different preference optimization methods exhibit varying levels of success on this benchmark. We hypothesize that the removal of the reference model in our approach may result in a loss of capability for solving complex arithmetic problems. Given the difficulty of the GSM8K benchmark, several methods have been proposed to address this challenge. For instance, Step-DPO (Lai et al., 2024) treats individual reasoning steps as units for preference optimization rather than evaluating answers holistically, thereby enhancing the long-chain reasoning ability of LLMs.

In general, our method demonstrates a balanced trade-off in downstream performance. It effectively maintains general knowledge and reasoning abilities while enhancing scientific and commonsense reasoning, as well as truthfulness. However, it comes at the cost of reduced mathematical performance. These results suggest that the choice of preference optimization dataset plays a crucial role in shaping model capabilities. A deeper and more systematic investigation is necessary to fully understand the broader implications of preference optimization.

[1] lm-eval repository of Huggingface open leaderboard V1: https://github.com/EleutherAI/lm-evaluation-harness/tree/b281b0921b636bc36ad05c0b0b0763bd6dd43463

Table 4: Downstream task evaluation results of tasks on the Huggingface open leaderboard V1.

	MMLU (5)	ARC (25)	HellaSwag (10)	TruthfulQA (0)	Winograd (5)	GSM8K (5)	Average
				Mistral-Base			
SFT	60.10	58.28	80.76	40.35	76.40	28.13	57.34
DPO	58.48	61.26	83.59	53.06	76.80	21.76	59.16
IPO	60.23	60.84	83.30	45.44	77.58	27.14	59.09
CPO	59.39	57.00	80.75	47.07	76.48	33.06	58.96
KTO	60.90	62.37	84.88	56.60	77.27	38.51	63.42
ORPO	63.20	61.01	84.09	47.91	78.61	42.15	62.83
R-DPO	59.58	61.35	84.29	46.12	76.56	18.12	57.67
SimPO	59.30	61.86	83.42	46.48	77.19	20.92	58.20
LMPO	58.48	61.43	83.61	50.67	76.87	21.91	58.83
				Mistral-Instruct			
SFT	60.40	63.57	84.79	66.81	76.64	40.49	65.45
DPO	60.53	65.36	85.86	66.71	76.80	40.33	65.93
IPO	60.20	63.31	84.88	67.36	75.85	39.42	65.17
CPO	60.36	63.23	84.47	67.38	76.80	38.74	65.16
KTO	60.52	65.78	85.49	68.45	75.93	38.82	65.83
ORPO	60.43	61.43	84.32	66.33	76.80	36.85	64.36
R-DPO	60.71	66.30	86.01	68.22	76.72	37.00	65.82
SimPO	59.42	65.53	86.07	70.56	76.01	34.87	65.41
LMPO	59.53	65.27	86.12	70.30	76.16	30.63	64.67
				Llama3-Base			
SFT	64.88	60.15	81.37	45.33	75.77	46.32	62.30
DPO	64.31	64.42	83.87	53.48	76.32	38.67	63.51
IPO	64.40	62.88	80.46	54.20	72.22	22.67	59.47
CPO	64.98	61.69	82.03	54.29	76.16	46.93	64.35
KTO	64.42	63.14	83.55	55.76	76.09	38.97	63.65
ORPO	64.44	61.69	82.24	56.11	77.51	50.04	65.34
R-DPO	64.19	64.59	83.90	53.41	75.93	39.27	63.55
SimPO	63.94	65.02	83.09	59.44	77.42	31.54	63.41
LMPO	63.94	64.68	83.03	57.98	77.90	36.01	63.92
				Llama3-Instruct			
SFT	67.06	61.01	78.57	51.66	74.35	68.69	66.89
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	65.72	62.88	78.30	60.74	73.01	50.19	65.14
LMPO	66.08	61.77	76.81	60.06	72.85	43.14	63.45

C Implementation Details

Training hyperparameters. For LMPO, we maintained a consistent batch size of 128 across all four experimental settings. The learning rates were configured as follows: 3e-7 for Mistral-Base (7B), 5e-7 for Mistral-Instruct (7B), 6e-7 for Llama-3-Base (8B), and 1e-6 for Llama-3-Instruct (8B).

Table 5: The hyperparameter values in LMPO used for each training setting.

Setting	β	h	λ	Learning rate
Mistral-Base	2.0	$e^{1.6}$	1.0	3.0e-7
Mistral-Instruct	2.5	$e^{0.25}$	0.2	5.0e-7
Llama-3-Base	2.0	$e^{1.0}$	0.2	6.0e-7
Llama-3-Instruct	2.5	$e^{1.4}$	0.2	1.0e-6

All models were trained for a single epoch using a cosine learning rate schedule with a 10% warmup phase. Optimization was performed using Adam (Kingma, 2014). Furthermore, the maximum sequence length was set to 1024 for Mistral-Base (7B) and 2048 for all other configurations. We use 42 as training random seed.

Hyperparameter in LMPO. Table 5 outlines the hyperparameters used for LMPO across four different settings. For the parameter β, we follow the configuration from SimPO. Among these parameters, h, which represents the home-court advantage, typically requires more careful tuning. For λ, we set it to 1.0 for Mistral-Base and 0.2 for the other settings. As mentioned in the main article, selecting the appropriate value for λ is crucial for LMPO performance.

Evaluation Hyperparameters. The hyperparameters utilized for evaluation in this study align with those adopted in SimPO.[2] We sincerely appreciate the SimPO team for their generous contributions and invaluable insights.

Computational Environment. All training experiments reported in this study were performed on a system equipped with four A100 GPUs, following the procedures outlined in the alignment-handbook repository.[3]

[2] https://github.com/princeton-nlp/SimPO/tree/main/eval

[3] https://github.com/huggingface/alignment-handbook