Multi-Reference Preference Optimization for Large Language Models

Hung Le[1] **Quan Tran**[2] **Dung Nguyen**[1] **Kien Do**[1] **Saloni Mittal**[2]
Kelechi Ogueji[2] **Svetha Venkatesh**[1]

[1]**Applied AI Institute, Deakin University, Geelong, Australia**
{thai.le, dung.nguyen, k.do, svetha.venkatesh}@deakin.edu.au
[2]**ServiceNow Research, USA**
{hungquan.tran, saloni.mittal, kelechi.ogueji}@servicenow.com

Abstract

How can Large Language Models (LLMs) be aligned with human intentions and values? A typical solution is to gather human preference on model outputs and finetune the LLMs accordingly while ensuring that updates do not deviate too far from a reference model. Recent approaches, such as direct preference optimization (DPO), have eliminated the need for unstable and sluggish reinforcement learning optimization by introducing close-formed supervised losses. However, a significant limitation of the current approach is its design for *a single reference model only, neglecting to leverage the collective power of numerous pretrained LLMs*. To overcome this limitation, we introduce a novel closed-form formulation for *direct preference optimization using multiple reference models*. The resulting algorithm, Multi-Reference Preference Optimization (MRPO), leverages broader prior knowledge from diverse reference models, substantially enhancing preference learning capabilities compared to the single-reference DPO. Our experiments demonstrate that LLMs finetuned with MRPO generalize better in various preference data, regardless of data scarcity or abundance. Furthermore, MRPO effectively finetunes LLMs to exhibit superior performance in several downstream natural language processing tasks such as HH-RLHF, GSM8K and TruthfulQA.

Introduction

Large Language Models (LLMs) have emerged as powerful tools in natural language processing, capable of generating human-like text and performing a myriad of language-related tasks (Lewkowycz et al. 2022; Achiam et al. 2023; Touvron et al. 2023). However, aligning these models with human intentions and values remains a challenging endeavor (Wang et al. 2023). Aligning LLMs with curated human feedback emerges as a critical solution to guide LLM response behavior and address this challenge. Preference models like the Bradley-Terry model (Bradley and Terry 1952) are often used to measure the alignment of reward functions with empirical preference data, facilitating an alignment framework using reinforcement learning with human feedback (RLHF (Christiano et al. 2017)). The framework aims to optimize the preference models (maximizing preference reward) while ensuring that LLM updates do not stray too far from a base reference LLM model (minimizing a Kullback-Leibler (KL) divergence). While RLHF has been successful in enhancing the helpfulness and accuracy of model-generated content (Ouyang et al. 2022; Stiennon et al. 2020), it is unstable, complicated, and resource-intensive.

Recent advancements, such as direct preference optimization (DPO (Rafailov et al. 2023)) and other likelihood-based preference learning (Zhao et al. 2023; Azar et al. 2023; Ethayarajh et al. 2024; Chen et al. 2024), have sought to replace the cumbersome RLHF with closed-form supervised losses. Although they have demonstrated impressive performance compared to RLHF and supervised finetuning (SFT), their exclusive reliance on a single reference model restricts their potential, overlooking the advantages of harnessing multiple pretrained LLMs. Using multiple reference models offers two key benefits: (i) it enhances robustness and generalization by combining models that excel at different inputs to improve overall performance across diverse data and tasks; (ii) multiple references introduce stronger regularization, leveraging diverse constraints to build a more resilient system less prone to overfitting and reward hacking. This is increasingly important as the open-source community consistently introduces new pretrained/SFT LLMs of varying scales, trained on diverse datasets (Touvron et al. 2023; Penedo et al. 2023; Jiang et al. 2023). It underscores the necessity for a solution that employs multiple references for LLM finetuning, enabling the distillation of knowledge from existing LLMs to enhance the alignment training stage. *Unfortunately, none of the prior works have proposed a solution for utilizing multiple reference LLMs in direct preference optimization.*

The absence of such solutions stems from three challenges in formulating closed-form multiple-reference preference learning. Firstly, deriving a closed-form solution for the RLHF objective with multiple referencing constraints is nontrivial due to the non-linearity of multiple KL terms. Secondly, reference models with varying architecture, size, and pretraining data may produce diverging outputs given the same input. This divergence could potentially confuse the learning process, leading to unstable training, worse than single-reference approaches. Thirdly, determining the contribution of each reference model during training poses a challenge, requiring extensive tuning. In this paper, we tackle these three challenges, presenting a simple and viable framework for direct preference optimization utilizing multiple reference models.

To address the non-linearity of KL divergence, we propose maximizing a simpler surrogate lower bound that allows for the derivation of a novel closed-form solution incorporating multiple reference models. Our solution is theoretically and empirically proven superior to combining multiple DPO losses. Next, we propose a clipped trust-regions optimization (CTRO) to address the second challenge. By clipping the log probability of diverging reference policy, we force the mismatch to be minimal to facilitate stable training while retaining useful information from the reference policy to guide the optimization. More importantly, the clipping rate is dynamically adjusted according to the predicted likelihood of the data, enabling a more adaptable update. Lastly, to automate the process of determining the contribution of each reference model, we introduce a dynamic mechanism (ARWC) to calculate the weight of each KL term based on the confidence of the referencing LLMs.

Our holistic framework, dubbed Multiple Reference Preference Optimization (MRPO), undergoes evaluation across various tasks. In preference learning tasks involving 6 preference datasets, MRPO demonstrates significant superiority over DPO and multi-reference baselines, especially when preference data is limited with improvement of up to 7%. In terms of helpfulness evaluation, MRPO significantly outperforms DPO by 13.7%. Furthermore, on general language understanding benchmarks like the HuggingFace Open LLM Leaderboard (Beeching et al. 2023), MRPO exhibits average enhancements of 3-4% compared to SFT and 1.2% compared to DPO. Certain tasks show more than 5% improvements over DPO. Importantly, these enhancements are evident across various configurations, including different numbers of reference models (2 or 3) and architectures of LLMs (Llama, Mistral, and Qwen). We perform a comprehensive ablation study to demonstrate the efficacy of CTRO and ARWC mechanisms. Finally, we demonstrate that MRPO, when implemented correctly, adds minimal computation overhead compared to DPO, maintaining efficiency while enhancing performance.

Background

Problem Formulation and Notations

We rely on Azar et al. (2023) to formally define the problem and notations. Given an input $x \in \mathcal{X}$ where $\mathcal{X}$ is the finite space of input texts, a policy π models a conditional probability distribution $\pi(y|x)$ where $y \in \mathcal{Y}$ is the output in the finite space of output texts. From a given π and x, we can sample an output as $y \sim \pi\left(\cdot|x\right)$. Preference data is generated by sampling two outputs $(y, y'|x)$ from policies π and μ and presenting them to an agent, normally a human, for rating to indicate which one is preferred. For example, $y \succ y'$ denotes y is preferred to y'. A preference dataset is then denoted as $\mathcal{D} = \left\{y_w^i, y_l^i|x^i\right\}_{i=1}^N$ where N is the number of data points, y_w and y_l denote the preferred (chosen) and dispreferred (rejected), respectively. Assuming that there exists a true model of preference of the agent $p^*\left(y \succ y'|x\right)$ that assigns the agent's probability of y being preferred to y' given x. Using dataset $\mathcal{D}$, our goal is to find a policy π maximizing the expected preference while being close to a reference policy π_{ref}, which results in the following optimization problem:

$$\max_{\pi} \underset{\substack{x\sim\rho \\ y'\sim\mu(\cdot|x) \\ y\sim\pi(\cdot|x)}}{E} \left[\Psi\left(p^*\left(y \succ y'|x\right)\right)\right] - \beta D_{KL}\left(\pi \,\|\pi_{ref}\right) \quad (1)$$

where ρ is the input distribution, Ψ is a scaled function, D_{KL} is the Kullback–Leibler divergence and β is a hyperparameter. Usually, π is initialized as π_{ref} for stable optimization.

Preference Learning with Reward Function and Reinforcement Learning

In this approach, Bradley-Terry model (Bradley and Terry 1952) is employed as the preference model:

$$p\left(y \succ y'|x\right) = \sigma\left(r_\theta\left(x, y\right) - r_\theta\left(x, y'\right)\right) \quad (2)$$

where σ denotes the sigmoid function and $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is a reward model parameterized by θ, which assigns a scalar score to indicate the suitability of output y for input x. In earlier works (Christiano et al. 2017), the reward model is trained on $\mathcal{D}$ to minimize the negative log-likelihood loss:

$$\mathcal{L}_R = -\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}}\left[\log\,\sigma\left(r_\theta\left(x, y_w\right) - r_\theta\left(x, y_l\right)\right)\right] \quad (3)$$

Given a trained reward model r, and the scaled function as $\Psi\left(q\right) = \log\left(\frac{q}{1-q}\right) \forall q : 0 < q < 1$, the objective in Eq. 1 can be rewritten as:

$$\max_{\pi} \underset{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}}{E} \left[r\left(x, y\right)\right] - \beta D_{KL}\left(\pi \,\|\pi_{ref}\right) \quad (4)$$

This RLHF objective is employed to train LLMs such as Instruct-GPT (Ouyang et al. 2022) using PPO (Schulman et al. 2017).

Direct Preference Optimization

Reward training and RL finetuning require significant resources and can be cumbersome. Recent approaches circumvent these challenges by directly optimizing the policy via minimizing a preference-based negative log-likelihood loss (Rafailov et al. 2023), $\mathcal{L}_{DPO}$:

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

The term $r_\theta\left(x, y|\pi_{ref}\right) = \beta\log\frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$ plays the role of an implicit reward. The authors in Rafailov et al. (2023) proved that minimizing this loss is equivalent to solving the optimization problem in Eq. 4.

Method

Multi-Reference Preference Optimization

In this paper, we are focused on situations involving K reference policies $\left\{\pi_{ref}^{k}\right\}_{k=1}^{K}$. Therefore, extending from Eq. 4, our objective can be formulated as a multi-reference RLHF objective:

$$\underset{\pi}{max} \underset{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}}{E}\left[r\left(x,y\right)\right]-\beta\left(\sum_{k=1}^{K}\alpha_k D_{KL}\left(\pi\left\|\pi_{ref}^{k}\right.\right)\right) \tag{6}$$

where α_k are weighting coefficients for each reference policy and $1=\sum_{k=1}^{K}\alpha_k$. The main policy can only be initialized as one of $\{\pi_{ref}^{k}\}_{k=1}^{K}$. Without loss of generality, we denote π_{ref}^{1} as the initializing reference policy for the main policy π_θ. Previous studies have explored this objective, showing improvements over single-reference constraints and convergence proof in pure RL settings (Le et al. 2022).

However, addressing this optimization problem in LLMs through reward learning and RL finetuning poses similar challenges to using RL for Eq. 4. Hence, we propose an alternative approach that leverages direct preference optimization for the scenario involving multiple reference policies. We aim to find a closed-form solution for the multi-reference RLHF objective in Eq. 6. Unfortunately, deriving an exact closed-form solution is challenging due to the nonlinearity of D_{KL} terms. To circumvent this, we suggest obtaining a *closed-form solution for a surrogate objective*, which serves as a lower bound for the multi-reference RLHF objective. We summarize our findings as a proposition below.

Proposition 1. *The following policy is the optimum for a lower bound of the RLHF objective (Eq. 6):*

$$\pi^{*}\left(y|x\right)=\frac{1}{Z\left(x\right)}\tilde{\pi}_{ref}\left(y|x\right)\exp\left(\frac{1}{\beta}r\left(x,y\right)\right)$$

where $\tilde{\pi}_{ref}(y|x)=\left(\sum_{k=1}^{K}\frac{\alpha_k}{\pi_{ref}^{k}(y|x)}\right)^{-1}$ *and* $Z\left(x\right)=\sum_{y}\tilde{\pi}_{ref}(y|x)\exp\left(\frac{1}{\beta}r\left(x,y\right)\right)$.

Proof. See Appendix A.1. □

Following the derivation in Rafailov et al. (2023) with our proposed optimal policy π^{*}, we have the associated direct preference loss function, $\mathcal{L}_{MRPO}$, as follow,

$$-\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y_w|x\right)}{\tilde{\pi}_{ref}\left(y_w|x\right)}-\beta\log\frac{\pi_\theta\left(y_l|x\right)}{\tilde{\pi}_{ref}\left(y_l|x\right)}\right)\right] \tag{7}$$

The loss function is similar to the DPO loss (Eq. 5), but instead of using a single reference policy π_{ref}, we substitute it with a ”virtual” reference policy $\tilde{\pi}_{ref}$ that aggregates information from all multiple reference policies.

Clipped Trust-Regions Optimization (CTRO)

An issue that may arise when multiple reference policies are involved is the mismatch between the reference policy and the main policy. This is less common with single-reference DPO, as the main policy and the reference policy share the same origin, ensuring a small mismatch across training. However, with multiple-reference policies, those not chosen to initialize the main policy can result in significantly different probabilities compared to the main one (due to differences in architectures, pretraining, and tokenizers), potentially leading to unstable training and, at times, loss divergence. Yet, it is crucial to leverage diverse likelihood views on the generated output to ensure generalization.

To address this dilemma, we propose to constrain the virtual reference policy $\tilde{\pi}_{ref}$ in the vicinity of the initializing reference policy π_{ref}^{1}. As such, we propose to clip the log-probability of the other reference policies $\pi_{ref}^{k>1}$ as follows,

$$\log\hat{\pi}_{\text{ref}}^{k>1}\left(y|x\right)=\min\Big(\max\left(\log\pi_{ref}^{k>1}\left(y|x\right),\right.$$
$$\left.\left(1+\epsilon\right)\log\pi_{ref}^{1}\left(y|x\right)\right),\left(1-\epsilon\right)\log\pi_{ref}^{1}\left(y|x\right)\Big) \tag{8}$$

where ϵ defines the vicinity range around π_{ref}^{1}. Then we will replace $\pi_{ref}^{k>1}$ with $\hat{\pi}_{ref}^{k>1}$ in the $\tilde{\pi}_{ref}$ defined in Proposition 1.

Using a fixed ϵ is overly restrictive and suboptimal since different data and policies may require different trust region ranges. Thus, we suggest an adaptive approach to define ϵ based on the predicted likelihood of the data. Essentially, suppose the log probability of a reference model for a given data point is large, indicating high reliability. A conservative update should be applied (smaller ϵ) to exploit the reference policy. Conversely, for lower log probabilities, we may prefer more exploration, allowing reference values to diverge from the initial π_{ref}^{1} (bigger ϵ). That is,

$$\epsilon\left(y|x\right)=\epsilon_{max}\frac{\left|\sum_{k=1}^{K}\log\,\pi_{ref}^{k}\left(y|x\right)\right|}{\sum_{y'}\left|\sum_{k=1}^{K}\log\,\pi_{ref}^{k}\left(y'|x\right)\right|} \tag{9}$$

where ϵ_{max} is a hyperparameter specifying the maximum ratio for an updating range. It is worth noting that since $0\leq\pi\left(\cdot\right)\leq1$, the log-probability is always negative, meaning a larger absolute value of the log-probability indicates less confidence. The denominator represents the sum of all possible output y', serving as a normalization factor. Usually, we only have two outputs (y_w, y_l) per input x so the denominator is the sum of two terms.

Adaptive Reference Weighting Coefficients (ARWC)

If we have no preference or prior knowledge of the reference policies, we can simply use $\alpha_k=1/K\,\forall k$. However, if we assume that the reference policy obtains a reasonable ability to differentiate between y_w and y_l such that even when it make wrong preference (i.e $\log y_l>\log y_w$) the likelihood difference should not be too large, we can introduce an automatic mechanism to determine the value of α_k based on the confidence of the reference policy. Specifically, we examine the absolute difference between the log-probability of the two outputs (y_w, y_l) as an indicator of the policy's confidence in its ability to discriminate between two outputs. In essence, a larger difference suggests that the policy distinguishes one output from another more decisively. Formally, we propose to adaptively compute reference weighting coefficients as:

$$\alpha_k = \frac{\left|\log \pi^k_{ref}\left(y_w|x\right) - \log \pi^k_{ref}\left(y_l|x\right)\right|}{\sum_{i=1}^{K}\left|\log \pi^i_{ref}\left(y_w|x\right) - \log \pi^i_{ref}\left(y_l|x\right)\right|} \quad (10)$$

The coefficient is normalized across reference policies, giving greater weight to those with higher discriminative confidence.

Comparison with Multiple DPO

When having multiple reference policies, a naive solution for direct preference learning is to combine multiple DPO losses (Multi-DPO):

$$\mathcal{L}_{Multi-DPO} = -\mathbb{E}_{x,y_w,y_l\sim\mathcal{D}} \left[\sum_{k=1}^{K}\alpha_k \log\sigma\left(\beta\log\frac{\pi_\theta\left(y_w|x\right)}{\pi^k_{ref}\left(y_w|x\right)} - \beta\log\frac{\pi_\theta\left(y_l|x\right)}{\pi^k_{ref}\left(y_l|x\right)}\right)\right] \quad (11)$$

We can show that our proposed MRPO is more desirable than this Multi-DPO loss. As such, let the implicit reward $r_\theta(x, y|\pi^k_{ref}) = \beta \log \pi_\theta(y|x)/\pi^k_{ref}(y|x)$, we find that the gradients of the likelihood of outputs are scaled by the reward error $\sigma\left(r_\theta\left(x, y_l|\tilde{\pi}_{ref}\right) - r_\theta\left(x, y_w|\tilde{\pi}_{ref}\right)\right)$ and $\sum_{k=1}^{K}\alpha_k\sigma\left(r_\theta\left(x, y_l|\pi^k_{ref}\right) - r_\theta\left(x, y_w|\pi^k_{ref}\right)\right)$, corresponding to MRPO and Multi-DPO, respectively. Under mild assumptions, we can prove the following result:

Proposition 2. *Assume that reference policies are constrained to be relatively close to each other, ensuring that* $\left\{d_k = r_\theta\left(x, y_l|\pi^k_{ref}\right) - r_\theta\left(x, y_w|\pi^k_{ref}\right)\right\}_{k=1}^{K}$ *share the same sign* $\forall k$, *then*

$$\begin{cases}\left|\nabla_\theta\mathcal{L}_{MRPO}\right| & \geq \left|\nabla_\theta\mathcal{L}_{Multi-DPO}\right| \;\; \forall d_k \geq 0\\ \left|\nabla_\theta\mathcal{L}_{MRPO}\right| & \leq \left|\nabla_\theta\mathcal{L}_{Multi-DPO}\right| \;\; \forall d_k \leq 0\end{cases}$$

Proof. See Appendix A.2. □

As a result, when the reward estimations are wrong, i.e., $\forall d_k > 0$, MRPO update magnitude will be greater than that of Multi-DPO, which is desirable because we want to fix the likelihood faster to correct the implicit reward. On the contrary, when the reward estimations are right, i.e., $\forall d_k < 0$, MRPO update magnitude will be smaller than that of Multi-DPO, stabilizing the convergence and reducing over-fitting.

Experimental Results

In our experiments, we will always refer to the first (initializing) reference model as RefM1, the second as RefM2, and so forth. The Base model is the original LLM that will undergo finetuning on preference data and is initialized as RefM1. Throughout experiments, if not stated otherwise, Llama (L), Mistral (M), and Qwen (Q) refer to *Llama-2-7b-chat-hf*, *OpenHermes-2.5-Mistral-7B*, and *Qwen1.5-7B-Chat*, respectively. Unless specified otherwise, we finetune these LLMs using LoRA 4-bit quantization to enable faster training and accommodate our hardware of a single Tesla A100 GPU with 32GB of memory. Further training details are provided in Appendix B.1. The source code can be accessed at: https://github.com/thaihungle/MRPO.

To assess model performance, we finetune the model with preference data and evaluate it on preference learning and general language understanding (GLU) tasks. In preference learning, we measure the **Preference Accuracy** in predicting whether two responses, $y_1|x$ and $y_2|x$ are chosen (preferred) or rejected (dispreferred). In particular, for MRPO, we use $\beta \log \frac{\pi_\theta(y|x)}{\tilde{\pi}_{ref}(y|x)}$ as the implicit reward $r(x, y)$ for each response, with the response classified as chosen or rejected if it has a higher or lower reward, respectively. As for other methods, we use $\beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$ as the implicit reward (Rafailov et al. 2023). Another preference metric worth considering is the **Reward Margin**, which quantifies the difference between the chosen and rejected rewards: $r(x, y_w) - r(x, y_l)$. In helpfulness evaluation, we adopt **Win Rate** suggested by GPT-4 evaluator (Achiam et al. 2023). In GLU, we employ the **GLU Metric** provided by the task. All measurements are desirable when they are higher.

Performance when Preference Data is Scarce

Datasets In real-world scenarios, human feedback is limited. Here, we curate 3 small preference datasets (hundreds to a few thousand data points) to simulate the scarcity of feedback data. Each training dataset comprises a random subset from a larger public preference dataset available in the Hugging Face data repository. The remaining portions of the dataset will be utilized as testing data. These datasets are generated with inputs, outputs, and preference rankings often produced by powerful LLMs like GPT-4, making them suitable for training smaller LLMs such as Llama. The datasets are labeled as S1, S2, and S3, and their details are given in Appendix Table 6 and Appendix B.2.

Baselines `DPO` represents the standard single-reference approach. We note that, as will be shown in our experiments, this is a strong baseline because it is non-trivial to leverage multiple reference models. Multiple-reference methods include `Multi-DPO` as described earlier, `RLHF`-our RL version using reward model (Christiano et al. 2017) and PPO to directly optimize Eq. 6, and `KD` which leverage a knowledge distillation loss (Agarwal et al. 2024) ($\beta \sum_{k=2}^{K} \alpha_k D_{KL}(\pi||\pi^k_{ref})$) together with the original DPO loss. `KD` aims for DPO optimization using one reference model ($k = 1$) while forcing the optimized policy (student) similar to other reference models (teachers, $k > 1$). Note that this method requires sampling y from the main policy during training to estimate the D_{KL}, which makes the process slow.

Unless stated otherwise, all baselines the same common hyperparameters such as learning rate (10^{-5}), batch size (8), number of epochs (3), and $\beta = 0.1$. For Multi-DPO, we have to use clipped trust regions to ensure RefM2 is close to RefM1. Otherwise, the learning will not converge. To

Dataset	S1		S2		S3	
	L←M	M←L	L←M	M←L	L←M	M←L
DPO	93.9±1.4	*98.7±0.7*	94.4±2.9	96.7±1.8	*54.8±5.4*	*52.2±3.7*
RLHF	67.6±3.6	71.7±1.2	83.3±2.0	88.8±0.7	50.6±1.2	49.9±5.5
KD	78.4±7.3	86.8±2.8	94.3±2.4	96.3±2.4	46.9±7.0	51.8±3.8
Multi-DPO	95.6±1.7	98.0±0.8	95.5±1.2	96.3±1.2	41.1±3.6	50.8±0.8
Mean Ref	*95.9±1.6*	97.9±2.1	*95.6±1.8*	96.6±1.2	45.6±4.4	51.3±1.2
MRPO (Ours)	**97.2**±**1.7**	**99.5**±**0.8**	**97.0**±**3.2**	97.0±1.6	**61.3**±**5.2**	**56.0**±**1.9**

Table 1: Final mean±std. testing accuracy ($\times 100$) on small datasets over 5 runs. Bold denotes the best, statistically different from the others as Cohen effect size ≥ 0.5. Italic denotes the second best.

make a fair comparison, both MRPO Multi-DPO, and KD use $\epsilon_{max} = 0.1$ and incorporate the adaptive ϵ. Multi-DPO, RLHF and KD use their best fixed $\alpha \in \{0.1, 0.5, 0.9\}$ tuned using dataset S1. For Multi-DPO, RLHF, KD, and MRPO, we consider 2 reference models ($K = 2$), and examine 2 possible modes of initialization: (1) L ← M, the Base model is initialized as Llama (RefM1) for all baselines, and the RefM2 is Mistral for multi-reference methods; (2) M ← L, the order is reversed.

Results Table 1 reports the final preference accuracy on test sets. MRPO consistently outperforms DPO by a significant margin, 3-7% and 1-4% for L ← M and M ← L, respectively. Mode L ← M observes more improvement because Mistral is stronger than Llama in these tasks and thus, using Mistral as RefM2 will bring more benefits than using Llama. Overall, MRPO regularizes the main policy using both Llama and Mistral, creating a model that surpasses each individually. On the other hand, Multi-DPO underperforms DPO and MRPO in many cases, indicating that combining multiple references in a naive way, even when equipped with our CTRO and ARWC, cannot yield favorable results. RLHF performs the worst, likely due to the instability of RL optimization, especially under multiple reference constraints. KD also underperforms compared to DPO and MRPO, highlighting the challenges of applying knowledge distillation to alignment problems without theoretical support for the naive KD approach.

Appendix Fig. 1 depicts the testing preference accuracy curves over the training duration for all methods across 2 initialization modes. These curves demonstrate that all methods, except RLHF in some cases, boost the chosen/rejection prediction accuracy of the Base model (the first evaluation point in each graph is lower than the following points). Among all, MRPO exhibits early outperformance compared to other baselines and maintains its superior performance until convergence.

Can MRPO Scale to Big Preference Datasets?

Datasets To assess the scalability of MRPO to real and large datasets, we utilize 3 big preference datasets: HelpSteer, Ultrafeedback, and Nectar (see Appendix Table 7). Each dataset employs human rankings to assess the outputs generated by powerful LLMs. Finetuning LLMs for just one epoch is sufficient for large datasets to achieve learning convergence. We use the provided train/test split for HelpSteer and Ultrafeedback. We randomly allocate 90% of the data for training purposes and 10% for testing for Nectar.

Baselines and Results In this task, we evaluated MRPO ($K = 2$) against DPO, the top two methods from our prior tests, using the same initialization approaches detailed earlier. The preference accuracy result, reported in Table 2's upper row and Appendix Fig. 2, demonstrates that MRPO consistently surpasses DPO in real-world preference datasets, showing an improvement gain of approximately 3-5% and up to 1% for L←M and M←L modes, respectively. Since Preference Accuracy can sometimes be unclear in demonstrating performance, especially with borderline inputs where the chosen and rejected ground truth may not be entirely accurate, we also examine the Reward Margin of DPO and MRPO on these datasets. The result, displayed in Table 2's lower row and Appendix Fig. 3, demonstrates MRPO's superior ability to separate chosen and rejected outputs, as evidenced by a significantly higher Reward Margin of 10-20% compared to DPO. The findings confirm MRPO's capability to effectively scale with large datasets for preference learning tasks.

MRPO Helpfulness Assessment

To assess the generalization of MRPO in generating helpful responses, we employ the MRPO and DPO models finetuned on the Nectar dataset (as in the previous section). These models are evaluated on 2 datasets: the helpful and harmless HH-RLHF (Bai et al. 2022) and Alpaca-Eval 2.0 (Li et al. 2023). We input queries from this dataset into LLMs trained by MRPO and DPO, record their respective responses (up to 200 tokens), and utilize GPT-4o to determine the more helpful method.

Given budgetary constraints, our evaluation was limited to the initial 300 test set samples. Table 3 presents the results. On HH-RLHF data, MRPO outperformed DPO with a win rate of 45.0% to 31.3%. On Alpaca-Eval, MRPO outperforms DPO by achieving a higher win rate (24.0% vs 15.3%). These results suggest that training an LLM with MRPO enhances its ability to generate helpful responses due to exposure to diverse reference models. Details of the evaluation and sampled results are provided in Appendix B.2.

How Effective is MRPO on General Language Understanding Benchmarks?

For our evaluation benchmark, we utilized the Huggingface Open LLM Leaderboard, a standard in the field (Beeching et al. 2023). In this benchmark, we explore a variety of

Metric	Dataset	HelpSteer		Ultrafeedback		Nectar	
		L←M	M←L	L←M	M←L	L←M	M←L
Accuracy	DPO	68.9±0.4	70.8±4.3	69.8±0.9	72.0±1.1	75.6±3.9	78.5±3.6
	MRPO	**73.6±1.6**	71.6±5.2	**72.9±2.9**	**73.2±1.9**	**79.2±1.2**	78.7±3.0
Margin	DPO	0.64±0.01	0.95±0.20	0.70±0.07	1.14±0.12	1.52±0.08	2.65±0.29
	MRPO	**0.77±0.07**	**1.05±0.20**	**0.82±0.10**	**1.27±0.26**	**1.73±0.05**	**3.13±0.25**

Table 2: Final mean±std. testing preference accuracy (upper, $\times 100$) and reward margin (lower) on big datasets over 3 runs. Bold is best, statistically different from others as Cohen effect size ≥ 0.5.

HH-RLHF Dataset			
Model	Win ↑	Tie	Lose ↓
DPO	31.3	23.7	45.0
MRPO	**45.0**	23.7	**31.3**
Alpaca-Eval Dataset			
Model	Win ↑	Tie	Lose ↓
DPO	15.3	60.7	24.0
MRPO	**24.0**	60.7	**15.3**

Table 3: Win, Tie, and Lose rate (%): MRPO vs DPO in measuring helpfulness of responses. Bold denotes the better.

datasets, collectively covering tasks such as math (GSM8k) multi-task language understanding (MMLU), human falsehood understanding (TruthfulQA), and commonsense reasoning (Arc, HellaSwag, Winogrande). The evaluation process presents the language models with few-shot in-context examples and questions. We apply the standard evaluation protocol to evaluate and report average scores (GLU Metrics) across all datasets.

In this experiment, we consider the third reference model RefM3 as Qwen to verify the scalability of our method to $K = 3$ on the standard GLU benchmark. We examine the following initialization modes: (1) L ← M, Q where Mistral and Qwen are additional reference models for Base model Llama, (2) M ←L, Q where Llama and Qwen are additional reference models for Base model Mistral, and (3) Q ←M, L where Mistral and Llama are additional reference models for Base model Qwen. Following prior practices (Chen et al. 2024), we adopt *full finetuning* to finetune the Base models on Ultrafeedback dataset using DPO and MRPO and evaluate the LLMs using Language Model Evaluation Harness library (Gao et al.). We report all results in Table 4. MRPO leads to a notable enhancement in the Base model of 3.5%, 1.4%, and 0.8%, surpassing DPO with an average improvement of 1.1%, 1.0% and 1.3% for initialization (1), (2) and (3), respectively. Notably, there are several cases in which MRPO can outperform DPO by a huge margin, such as 6.8% in GSM8K (M ←L, Q), 5.8% in TruthfulQA (L ←M, Q) and 5% in GSM8K (Q ←M, L). We also explore MRPO ($K = 2$, L ←M and M ←L) and observe that this variant consistently surpasses DPO in performance, albeit falling short of MRPO ($K = 3$), suggesting the advantages of incorporating more reference models (see Appendix Table 8).

Ablation Study

The Importance of Clipped Trust-Region Optimization (CTRO) To investigate the role of CTRO, we utilize the small yet relatively challenging dataset S3 and conduct experiments with various ϵ_{max} values: $\epsilon_{max} = 0$ (MRPO equals DPO), $\epsilon_{max} = 10^6$(No clip), and $\epsilon_{max} = \{1, 0.1, 0.01\}$. We also examine different reference models: (i) We employ RefM2 (Llama supervised tuning on Alpaca dataset) as a finetuned model of RefM1 (Llama), aiming to ensure that RefM2 is closely related to RefM1 (same family); and (ii) RefM2 (Mistral) is from a different family from RefM1 (LLama), which indicates a larger mismatch between the two reference models. Here, different families mean that LLMs can vary in architecture, tokenizer and/or be pretrained on distinct corpora, where the sentence-level log probability of these LLMs for the same input can differ by hundreds of units. Table 5 reports the final preference accuracy.

We observe that for setting (ii) when $\epsilon_{max} \geq 1$, the training loss can escalate significantly, reaching values as high as 10, and occasionally even infinity, highlighting the instability of training in the absence of CTRO. This is evident in the poor performance of $\epsilon_{max} = \{10^6, 1\}$. Utilizing CTRO with small ϵ_{max} results in more stable training. However, excessive constraint, where ϵ_{max} is too small, can lead to nearly identical performance compared to DPO. In setting (i), the training is stable even with big ϵ_{max}. However, the performance is not as good as setting (ii). In particular, with the best $\epsilon_{max} = 0.1$, in setting (i), MRPO only achieves a 2% improvement over DPO, whereas in setting (ii), the improvement gap widens to 7%. This discrepancy is understandable because without a diverse reference source, RefM2 may not exhibit significant advantages over RefM1, thereby limiting the extent of improvement. Therefore, we conclude that it is more advantageous to leverage diverse reference models, and employing CTRO is required to ensure training stability.

Is adaptive ϵ necessary? We conduct more experiments with fixed $\epsilon = 0.1$ and adaptive $\epsilon_{max} = 0.1$ on S1, S2 and S3 using Llama and Mistral for RefM1 and RefM2, receptively. The results depicted in Appendix Fig. 4 (top) illustrate that fixed ϵ is still better than DPO, and adaptive ϵ outperforms significantly fixed ϵ across all datasets, emphasizing the importance of this mechanism in MRPO.

Analyzing Reference Weighting Coefficients In this section, using adaptive $\epsilon_{max} = 0.1$, we compare the adaptive α (ARWC) mechanism proposed in § with different fixed values of $\alpha = \{0.1, 0.5, 0.9\}$ on S1, S2 and S3 using Llama and Mistral for RefM1 and RefM2, receptively. As shown in Appendix Fig. 4 (bottom), adaptive α demonstrates competitive performance, either outperforming or closely matching the performance of the best fixed α across all

Dataset	GSM8K	TruthfulQA	HellaSwag	MMLU	Arc-easy	Winograde	Avg.
Base (Mistral)	49.6	44.5	62.8	60.8	83.5	**74.4**	62.6
DPO	53.7	**53.3**	**66.6**	60.1	82.7	73.6	65.0
MRPO (M←L,Q)	**60.5**	51.43	65.83	**61.5**	**84.0**	73.4	**66.1**
Base (LLama)	23.9	37.8	57.8	46.4	60.8	66.4	51.0
DPO	22.0	39.5	**59.4**	46.3	73.5	**67.3**	51.4
MRPO (L←M,Q)	**24.3**	**45.3**	57.9	**46.4**	**74.1**	66.7	**52.4**
Base (Qwen)	21.1	53.6	58.8	60.1	68.3	65.2	54.5
DPO	18.7	54.8	**60.4**	**60.3**	65.2	64.3	54.0
MRPO (Q←M,L)	**23.7**	**54.9**	59.0	60.1	**68.4**	**65.4**	**55.3**

Table 4: M←L: test performance ($\times 100$) across HuggingFace Open LLM Leaderboard datasets. Bold denotes best.

Setting	$\epsilon_{max} = 0$ (DPO)		$\epsilon_{max} = 10^6$ (No clip)		$\epsilon_{max} = 1$		$\epsilon_{max} = 0.1$		$\epsilon_{max} = 0.01$	
	(i)	(ii)	(i)	(ii)	(i)	(ii)	(i)	(ii)	(i)	(ii)
Test Acc.	0.54	0.54	0.51	0.45	0.53	0.49	0.56	**0.61**	0.54	0.55

Table 5: Clipped Trust-Region Optimization impact on S3. In setting (i), two reference models belong to the same family but differ in the finetuning dataset, whereas in setting (ii), they are from different families of LLMs. The reported numbers are the mean accuracy over 5 runs.

datasets. Given the minor discrepancies observed and the associated cost of hyperparameter tuning, we have opted to utilize adaptive α for all other experiments.

Related Works

The integration of human input has been instrumental in advancing the performance of Large Language Models (LLMs) in diverse domains, such as question answering (Nakano et al. 2021), document summarization (Stiennon et al. 2020), and dialog applications (Thoppilan et al. 2022). Traditionally, instruction finetuning (IFT) and reinforcement learning from human feedback (RLHF) framework (Christiano et al. 2017; Ouyang et al. 2022; Lee et al. 2023) has employed RL to align Large Language Models (LLMs). The RLHF objective is to maximize a reward score derived from human preferences (chosen or rejected) while simultaneously minimizing the disparity between the new and initial policy. Recently, there has been a significant move towards closed-form losses, exemplified by DPO (Rafailov et al. 2023), which directly finetunes LLM on offline preference datasets, consolidating RLHF's reward learning and policy adjustments into a single stage. This "direct" approach is favored over RLHF due to their maximum-likelihood losses, demonstrating superior speed and stability than RL pipeline.

Other direct preference optimization methods (Zhao et al. 2023; Azar et al. 2023) formulate various adaptations of closed-form losses to attain the RLHF objective. Recent advancements have expanded beyond the traditional binary preference data, focusing on novel human preference models like Kahneman-Tversky value functions (Ethayarajh et al. 2024). All these methods adhere to the fundamental framework of RLHF, striving to fit a preference model while ensuring the updates remain close to a reference model. In contrast, *our approach is the first direct preference finetuning framework with multiple reference models*.

Another line of work creates new preference data from LLM's own generated outputs, typically through self-training paradigms (Chen et al. 2024; Yuan et al. 2024; Pattnaik et al. 2024), employing multiple reference data pairs (Pattnaik et al. 2024). Our method is orthogonal to self-playing approaches, featuring a faster alternative procedure as it does not require additional data generation. In contrast to methods employing multiple rewards, mixture of experts, and model merging (Jang et al. 2023; Rame et al. 2024), our approach does not involve training/loading multiple LLMs, which is expensive. Instead, we train a single LLM using (pre-computed) log-probability outputs of multiple reference LLMs, and thus much more time/memory-efficient. During testing, our inference cost is the same as using a single LLM.

While knowledge distillation (Lin et al. 2020; Agarwal et al. 2024) could simplify constraining LLMs with multiple reference models, finding a compatible KD loss for direct preference optimization with strong theoretical support is non-trivial. Additionally, directly using RL to optimize preference rewards with multiple KL constraints, as in previous studies (Le et al. 2022), is slow, and unstable—similar to the issues faced by RLHF. In contrast, our MRPO provides a more direct and efficient solution.

Discussion

In this paper, we present Multi-Reference Preference Optimization (MRPO), a novel method leveraging multiple reference models to improve preference learning for Large Language Models (LLMs). We theoretically derive the objective function for MRPO and conduct experiments with LLMs like LLama2 and Mistral, demonstrate their enhanced generalization across six preference datasets, helpfulness evaluation, and competitive performance in six downstream natural language processing tasks. Our study is limited by using only up to three reference models of modest size. Future research will explore the scalability of MRPO, examining its performance with larger K values and a broader range of LLM sizes across diverse benchmarks.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Agarwal, R.; Vieillard, N.; Zhou, Y.; Stanczyk, P.; Garea, S. R.; Geist, M.; and Bachem, O. 2024. On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. In *The Twelfth International Conference on Learning Representations*.

Azar, M. G.; Rowland, M.; Piot, B.; Guo, D.; Calandriello, D.; Valko, M.; and Munos, R. 2023. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Beeching, E.; Fourrier, C.; Habib, N.; Han, S.; Lambert, N.; Rajani, N.; Sanseviero, O.; Tunstall, L.; and Wolf, T. 2023. Open llm leaderboard. *Hugging Face*.

Bradley, R. A.; and Terry, M. E. 1952. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4): 324–345.

Chen, Z.; Deng, Y.; Yuan, H.; Ji, K.; and Gu, Q. 2024. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ethayarajh, K.; Xu, W.; Muennighoff, N.; Jurafsky, D.; and Kiela, D. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Gao, L.; Tow, J.; Abbasi, B.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; Le Noac'h, A.; et al. ???? A framework for few-shot language model evaluation, 12 2023. *URL https://zenodo. org/records/10256836*, 7.

Jang, J.; Kim, S.; Lin, B. Y.; Wang, Y.; Hessel, J.; Zettlemoyer, L.; Hajishirzi, H.; Choi, Y.; and Ammanabrolu, P. 2023. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. *arXiv preprint arXiv:2310.11564*.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Le, H.; Karimpanal George, T.; Abdolshah, M.; Nguyen, D.; Do, K.; Gupta, S.; and Venkatesh, S. 2022. Learning to Constrain Policy Optimization with Virtual Trust Region. *Advances in Neural Information Processing Systems*, 35: 12775–12786.

Lee, H.; Phatale, S.; Mansoor, H.; Lu, K.; Mesnard, T.; Bishop, C.; Carbune, V.; and Rastogi, A. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. *arXiv preprint arXiv:2309.00267*.

Lewkowycz, A.; Andreassen, A.; Dohan, D.; Dyer, E.; Michalewski, H.; Ramasesh, V.; Slone, A.; Anil, C.; Schlag, I.; Gutman-Solo, T.; et al. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35: 3843–3857.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. AlpacaEval: An Automatic Evaluator of Instruction-following Models. https://github.com/tatsu-lab/alpaca_eval.

Lin, A.; Wohlwend, J.; Chen, H.; and Lei, T. 2020. Autoregressive Knowledge Distillation through Imitation Learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6121–6133.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744.

Pattnaik, P.; Maheshwary, R.; Ogueji, K.; Yadav, V.; and Madhusudhan, S. T. 2024. Curry-DPO: Enhancing Alignment using Curriculum Learning & Ranked Preferences. *arXiv preprint arXiv:2403.07230*.

Penedo, G.; Malartic, Q.; Hesslow, D.; Cojocaru, R.; Cappelli, A.; Alobeidli, H.; Pannier, B.; Almazrouei, E.; and Launay, J. 2023. The RefinedWeb dataset for Falcon LLM: outperforming curated corpora with web data, and web data only. *arXiv preprint arXiv:2306.01116*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Rame, A.; Couairon, G.; Dancette, C.; Gaya, J.-B.; Shukor, M.; Soulier, L.; and Cord, M. 2024. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Advances in Neural Information Processing Systems*, 36.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Thoppilan, R.; De Freitas, D.; Hall, J.; Shazeer, N.; Kulshreshtha, A.; Cheng, H.-T.; Jin, A.; Bos, T.; Baker, L.; Du, Y.; et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wang, Y.; Zhong, W.; Li, L.; Mi, F.; Zeng, X.; Huang, W.; Shang, L.; Jiang, X.; and Liu, Q. 2023. Aligning large language models with human: A survey. *arXiv preprint arXiv:2307.12966*.

Yuan, W.; Pang, R. Y.; Cho, K.; Sukhbaatar, S.; Xu, J.; and Weston, J. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Zhao, Y.; Joshi, R.; Liu, T.; Khalman, M.; Saleh, M.; and Liu, P. J. 2023. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Reproducibility Checklist

This paper:

- Includes a conceptual outline and/or pseudocode description of AI methods introduced (yes)
- Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results (yes)
- Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper (yes)

Does this paper make theoretical contributions? (yes)
If yes, please complete the list below.

- All assumptions and restrictions are stated clearly and formally. (yes)
- All novel claims are stated formally (e.g., in theorem statements). (yes)
- Proofs of all novel claims are included. (yes)
- Proof sketches or intuitions are given for complex and/or novel results. (partial)
- Appropriate citations to theoretical tools used are given. (yes)
- All theoretical claims are demonstrated empirically to hold. (NA)
- All experimental code used to eliminate or disprove claims is included. (NA)

Does this paper rely on one or more datasets? (yes)
If yes, please complete the list below.

- A motivation is given for why the experiments are conducted on the selected datasets (yes)
- All novel datasets introduced in this paper are included in a data appendix. (NA)
- All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (NA)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. (yes)
- All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. (yes)
- All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. (NA)

Does this paper include computational experiments? (yes)
If yes, please complete the list below.

- Any code required for pre-processing data is included in the appendix. (yes).
- All source code required for conducting and analyzing the experiments is included in a code appendix. (partial)
- All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. (yes)
- All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from (yes)
- If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. (yes)
- This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. (yes)
- This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. (yes) This paper states the number of algorithm runs used to compute each reported result. (yes)
- Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. (yes/no)
- The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). (yes)
- This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. (yes)
- This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. (yes)

Appendix

A. Methodology Details

A.1 Lower Bound Close-formed Solution To begin, we derive the closed-form solution for Eq. 6 as follows,

$$\underset{\pi}{max} \underset{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}}{E} \left[r\left(x,y\right)\right] - \beta\left(\sum_{k=1}^{K}\alpha_k D_{KL}\left(\pi\left\|\pi_{ref}^{k}\right.\right)\right) \tag{12}$$

$$\iff \underset{\pi}{max}\mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[r\left(x,y\right) - \beta\left(\sum_{k=1}^{K}\alpha_k\log\frac{\pi\left(y|x\right)}{\pi_{ref}^{k}\left(y|x\right)}\right)\right] \tag{13}$$

$$\iff \underset{\pi}{min}\mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\left(\sum_{k=1}^{K}\alpha_k\log\frac{\pi\left(y|x\right)}{\pi_{ref}^{k}\left(y|x\right)}\right) - \frac{1}{\beta}r\left(x,y\right)\right] \tag{14}$$

To simplify the optimization problem, using Jensen inequality $\sum_{k=1}^{K}\alpha_k\log\left(\frac{\pi(y|x)}{\pi_{ref}^{k}(y|x)}\right) \le \log\left(\sum_{k=1}^{K}\alpha_k\frac{\pi(y|x)}{\pi_{ref}^{k}(y|x)}\right)$, we propose minimizing the following upper bound of the objective function in Eq. 14 (equivalent to a lowerbound of the original RLHF objective):

$$\underset{\pi}{min}\mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\left(\sum_{k=1}^{K}\alpha_k\frac{\pi\left(y|x\right)}{\pi_{ref}^{k}\left(y|x\right)}\right) - \frac{1}{\beta}r\left(x,y\right)\right] \tag{15}$$

$$\iff \underset{\pi}{min}\mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\left(\pi\left(y|x\right)\sum_{k=1}^{K}\frac{\alpha_k}{\pi_{ref}^{k}\left(y|x\right)}\right) - \frac{1}{\beta}r\left(x,y\right)\right] \tag{16}$$

$$\iff \underset{\pi}{min}\mathbb{E}_{x\sim\rho}\mathbb{E}_{y\sim\pi(\cdot|x)}\left[\log\left(\frac{\pi\left(y|x\right)}{\frac{1}{Z(x)}\left(\sum_{k=1}^{K}\frac{\alpha_k}{\pi_{ref}^{k}(y|x)}\right)^{-1}\exp\left(\frac{1}{\beta}r\left(x,y\right)\right)}\right) - \log Z\left(x\right)\right] \tag{17}$$

where we have the partition function

$$Z\left(x\right) = \sum_{y}\left(\sum_{k=1}^{K}\frac{\alpha_k}{\pi_{ref}^{k}\left(y|x\right)}\right)^{-1}\exp\left(\frac{1}{\beta}r\left(x,y\right)\right) \tag{18}$$

$$= \sum_{y}\tilde{\pi}_{ref}\left(y|x\right)\exp\left(\frac{1}{\beta}r\left(x,y\right)\right) \tag{19}$$

where

$$\tilde{\pi}_{ref} = \left(\sum_{k=1}^{K}\frac{\alpha_k}{\pi_{ref}^{k}\left(y|x\right)}\right)^{-1} \tag{20}$$

$$\iff \frac{\pi_\theta\left(y_w|x\right)}{\tilde{\pi}_{ref}\left(y_w|x\right)} = \sum_{k=1}^{K}\alpha_k\frac{\pi_\theta\left(y_w|x\right)}{\pi_{ref}^{k}\left(y|x\right)} \tag{21}$$

We note that $\tilde{\pi}_{ref} > 0$ and it is not necessary that $\tilde{\pi}_{ref}$ is a distribution. If we define the distribution:

$$\pi^*\left(y|x\right) = \frac{1}{Z\left(x\right)}\tilde{\pi}_{ref}\left(y|x\right)\exp\left(\frac{1}{\beta}r\left(x,y\right)\right)$$

then the final objective is:

$$\underset{\pi}{min}\mathbb{E}_{x\sim\rho}\left[D_{KL}\left(\pi\left(y|x\right)||\pi^*\left(y|x\right)\right) - \log Z\left(x\right)\right]$$

Therefore, the optimal solution for the problem defined in Eq. 17 is:

$$\pi\left(y|x\right) = \pi^*\left(y|x\right) = \frac{1}{Z\left(x\right)}\tilde{\pi}_{ref}\left(y|x\right)\exp\left(\frac{1}{\beta}r\left(x,y\right)\right) \tag{22}$$

A.2 MRPO and Multi-DPO Gradient Inequality We can show that our proposed MRPO is more desirable than Multi-DPO loss. To see that, we analyze the gradient wrt. the policy parameters θ of the two losses:

$$\nabla_\theta \mathcal{L}_{MRPO} = -\beta \mathbb{E}_{x,y_w,y_l \sim \mathcal{D}} \left[\sigma\left(r_\theta\left(x, y_l|\tilde{\pi}_{ref}\right) - r_\theta\left(x, y_w|\tilde{\pi}_{ref}\right)\right) \times \left(\nabla_\theta \log \pi_\theta\left(y_w|x\right) - \nabla_\theta \log \pi_\theta\left(y_l|x\right)\right)\right]$$

$$\nabla_\theta \mathcal{L}_{Multi-DPO} = -\beta \mathbb{E}_{x,y_w,y_l \sim \mathcal{D}} \left[\alpha_k \sum_{k=1}^{K} \sigma\left(r_\theta\left(x, y_l|\pi^k_{ref}\right) - r_\theta\left(x, y_w|\pi^k_{ref}\right)\right) \times \left(\nabla_\theta \log \pi_\theta\left(y_w|x\right) - \nabla_\theta \log \pi_\theta\left(y_l|x\right)\right)\right]$$

Assuming the reference policies are constrained to be relatively close to each other, ensuring that $\left\{d_k = r_\theta\left(x, y_l|\pi^k_{ref}\right) - r_\theta\left(x, y_w|\pi^k_{ref}\right)\right\}_{k=1}^{K}$ share the same sign $\forall k$, we can use Jensen inequality to show that

$$\begin{cases} \sigma\left(\beta \sum_{k=1}^{K} \alpha_k \left(\log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right)\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \quad \forall d_k \geq 0 \\ \sigma\left(\beta \sum_{k=1}^{K} \alpha_k \left(\log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right)\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \quad \forall d_k \leq 0 \end{cases}$$

$$\Rightarrow \begin{cases} \sigma\left(\beta \sum_{k=1}^{K} \alpha_k \left(\log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} + \log \frac{\pi^k_{ref}(y_w|x)}{\pi_\theta(y_w|x)}\right)\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \geq 0 & \\ \sigma\left(\beta \sum_{k=1}^{K} \alpha_k \left(-\log \frac{\pi^k_{ref}(y_l|x)}{\pi_\theta(y_l|x)} - \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right)\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \leq 0 & \end{cases}$$

$$\Rightarrow \begin{cases} \sigma\left(\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} + \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi^k_{ref}(y_w|x)}{\pi_\theta(y_w|x)}\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \geq 0 & \\ \sigma\left(-\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi^k_{ref}(y_l|x)}{\pi_\theta(y_l|x)} - \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \leq 0 & \end{cases}$$

$$\Rightarrow \begin{cases} \sigma\left(\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} + \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi^k_{ref}(y_w|x)}{\pi_\theta(y_w|x)}\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \geq 0 & \\ \sigma\left(-\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi^k_{ref}(y_l|x)}{\pi_\theta(y_l|x)} - \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \leq 0 & \end{cases}$$

Applying Jensen inequality again with sigmoid monotonic property, we have:

$$\begin{cases} \sigma\left(\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \sum_{k=1}^{K} \frac{\pi_\theta(y_w|x)}{\alpha_k \pi^k_{ref}(y_w|x)}\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \geq 0 & \\ \sigma\left(\beta \log \sum_{k=1}^{K} \frac{\pi_\theta(y_l|x)}{\alpha_k \pi^k_{ref}(y_l|x)} - \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \leq 0 & \end{cases}$$

Now we use log monotonic property and the fact that $1/\alpha_k \geq \alpha_k$ for $0 < \alpha_k \leq 1$ and assume $0 < \alpha_k \forall k$, we have:

$$\begin{cases} \sigma\left(\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) & \geq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \geq 0 & \\ \sigma\left(\beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \sum_{k=1}^{K} \alpha_k \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) & \leq \sum_{k=1}^{K} \alpha_k \sigma\left(\beta \log \frac{\pi_\theta(y_l|x)}{\pi^k_{ref}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi^k_{ref}(y_w|x)}\right) \\ \forall d_k \leq 0 & \end{cases}$$

Plugging Eq. 20 yields:

Dataset	Train/Test size	Topic	Source: https://huggingface.co/datasets
S1	364/3276	General	wangrongsheng/comparison_gpt4_data_en
S2	502/502	AI/ML	NeuralNovel/Neural-DPO
S3	2178/242	Math	argilla/distilabel-math-preference-dpo

Table 6: Small datasets: S1, S2, and S3.

$$\begin{cases} \sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\tilde{\pi}_{ref}(y_l|x)} - \beta\log\frac{\pi_\theta(y_w|x)}{\tilde{\pi}_{ref}(y_w|x)}\right) & \geq \sum_{k=1}^{K}\alpha_k\sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\pi_{ref}^k(y_l|x)} - \beta\log\frac{\pi_\theta(y_w|x)}{\pi_{ref}^k(y_w|x)}\right) \ \ \forall d_k \geq 0 \\ \sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\tilde{\pi}_{ref}(y_l|x)} - \beta\log\frac{\pi_\theta(y_w|x)}{\tilde{\pi}_{ref}(y_w|x)}\right) & \leq \sum_{k=1}^{K}\alpha_k\sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\pi_{ref}^k(y_l|x)} - \beta\log\frac{\pi_\theta(y_w|x)}{\pi_{ref}^k(y_w|x)}\right) \ \ \forall d_k \leq 0 \end{cases}$$

Note that equality cannot be achieved in this case due to $1/\alpha_k = \alpha_k$ only when $\forall\alpha_k = 1$. When there is one $\alpha_k = 0$, we can omit the corresponding π_{ref}^k and follow the same derivation above to prove the inequality. Now we can multiply both sides of the inequality with $|\nabla_\theta \log\pi_\theta(y_w|x) - \nabla_\theta \log\pi_\theta(y_l|x)|$, which yields

$$\begin{cases} |\nabla_\theta\mathcal{L}_{MRPO}| & \geq |\nabla_\theta\mathcal{L}_{Multi-DPO}| \ \ \forall d_k \geq 0 \\ |\nabla_\theta\mathcal{L}_{MRPO}| & \leq |\nabla_\theta\mathcal{L}_{Multi-DPO}| \ \ \forall d_k \leq 0 \end{cases}$$

B. Experimental Details

B.1 Training Details For experiments using LoRA finetuning, we use LoRA with the following configuration:

- Rank: 8
- α: 8
- Quantization: 4 bit
- Target layers:
  - 7B LLMs: q_proj,v_proj
  - 1B LLMs: q_proj,k_proj,v_proj,o_proj,gate_proj,up_proj,down_proj

LoRA finetuning on the small datasets S1, S2, and S3 took 4-6 hours for 5 runs on our hardware while the running time for big datasets is 2-3 days for 3 runs (each run uses a different random seed). Full finetuning on Ultrafeedback took 2-3 days.

We implemented MRPO and other baselines using a public (Pytorch) code repository https://github.com/hiyouga/LLaMA-Factory (Apache-2.0 license). All the experiments are executed using this code base. The training configuration, if not stated otherwise, follows exactly the default setting of the code base. We only adjust the batch size to suit our hardware. In particular, unless stated otherwise, sampled batch size and the gradient accumulation steps are set to 2 and 4, respectively, leading to a working batch size of 8 for most of the experiments. All the pretrained LLMs are downloaded from https://huggingface.co/ and have Apache-2.0 license. The links to download are:

- *Llama-2-7b-chat-hf*: https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
- *OpenHermes-2.5-Mistral-7B*: https://huggingface.co/teknium/OpenHermes-2.5-Mistral-7B
- *Qwen1.5-7B-Chat*: https://huggingface.co/Qwen/Qwen1.5-7B-Chat

B.2 Datasets and Additional Results Below are some examples extracted from the small datasets:

```
S1
Input: What are the three primary colors?
Chosen Output: The three primary colors are red, blue, and yellow. These colors
are called primary because they cannot be created by mixing other colors and all
other colors can be made by combining them in various proportions. In the additive
color system, used for light, the primary colors are red, green, and blue (RGB).
Rejected Output: Red, Yellow, and Green.
```

```
S2
Input: You are an A.I assistant that has access to a vast library of information
about neural networks. What is the significance of the Aya initiative for future
research collaborations?
Chosen Output: The Aya initiative serves as a valuable framework for future
research collaborations that aim to bridge gaps in resources.
Rejected Output: The Aya initiative has no significance for future research
collaborations.
```

Dataset	Train/Test size	Source: https://huggingface.co/datasets
HelpSteer	8.1K/418	sablo/HelpSteer_binarized
Ultrafeedback	61K/2K	HuggingFaceH4/ultrafeedback_binarized
Nectar	164K/19K	HongchengGao/Nectar_binarized

Table 7: Big datasets: HelpSteer, Ultrafeedback, and Nectar.

S3
Input: How can I simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`?.\n Take a deep breath, think step by step, and give an accurate response
Chosen Output: To simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`, we can follow a few steps:\n\nStep 1: Distribute the division symbol by multiplying the expression by the reciprocal of the denominator. The reciprocal of `2x` is `1/(2x)`, so the expression becomes `(3x^2 - 4y^3) * (1/(2x))`.\n\nStep 2: Simplify within the parentheses by dividing each term separately. \n\n- For the first term, `3x^2`, divide `3x^2` by `2x`. This gives us `(3x^2) / (2x) = (3/2) * (x^2 / x) = (3/2) * x`.\n\n- For the second term, `-4y^3`, divide `-4y^3` by `2x`. This gives us `(-4y^3) / (2x) = (-2) * (y^3 / x)`.\n\nStep 3: Combine the simplified terms from Step 2. The expression now becomes `(3/2) * x - 2 * (y^3 / x)`.\n\nSo, the simplified form of the algebraic expression `(3x^2 - 4y^3) / (2x)` is `(3/2) * x - 2 * (y^3 / x)`.
Rejected Output: To simplify the algebraic expression `(3x^2 - 4y^3) / (2x)`, you can divide each term in the numerator by the denominator.\n\nFirst, let's divide `3x^2` by `2x`. Since both terms have a common factor of `x`, we can simplify this expression to `3x`.\n\nNext, we divide `-4y^3` by `2x`. We can simplify this expression by dividing each term separately. \n\nDividing `-4` by `2` gives `-2`. Then, dividing `y^3` by `x` gives `y^3/x`.\n\nSo, the simplified form of `(3x^2 - 4y^3) / (2x)` is `3x - 2y^3/x`.

Below are some examples extracted from the big datasets

HelpSteer
{Input: A group of people have been wrongfully convicted of a crime that they did not commit. Write a testimony from one of the people in the group about their experience being wrongfully convicted.
{Chosen Output: My name is [Name], and I am one of the wrongfully convicted. I was wrongfully convicted for a crime that I did not commit, and I want to share my experience with you.\n\nIt all started when I was arrested and charged with a crime that I didn't commit. I was innocent, but the evidence against me was overwhelming. I was interrogated for hours, and I eventually confessed to a crime that I didn't commit.\n\nThe trial was a circus. The prosecution presented a case that was full of lies and deceit. The jury didn't believe me, and I was found guilty. I was sentenced to life in prison.\n\nFor years, I struggled to make sense of what happened to me. I was lost and confused, and I felt like the justice system had failed me. But then, I found out about [Organization], a group of people who are dedicated to helping those who have been wrongfully convicted.\n\nI reached out to [Organization], and they started working on my case. They investigated the evidence, and they found new witnesses who came forward with new information. They uncovered evidence that proved my innocence, and after years of fighting, I was finally freed.\n\nBeing wrongfully convicted was one of the worst experiences of my life. But thanks to [Organization], I was able to clear my name and start a new chapter. I want to share my story with you to raise awareness about the importance of fair trials and the need to protect the rights of the innocent.\n\nMy name is [Name], and I am one of the wrongfully convicted. But now, I am free.
{Rejected Output: Hi there, I am an AI assistant that can answer questions about a reference text. You can ask me about the reference text and I will provide an answer. Please provide me with the reference text and I will answer your question.

Ultrafeedback
{Input: how can i develop a habit of drawing daily
{Chosen Output: Developing a daily habit of drawing can be challenging but with consistent practice and a few tips, it can become an enjoyable and rewarding part of your daily routine. Here are some strategies to help you develop the habit of drawing daily:\n\n1. Set a specific time: Allocate a specific time of the day to draw. It could be in the morning, afternoon, or evening. Make drawing a part of your daily routine.\n2. Set a specific duration: Determine the amount of time you want to spend on drawing each day. It can be as little as 10 minutes or as long as an hour. Be consistent with the duration to help build the habit.\n3. Start small and simple: Don't try to create a masterpiece every day, start with simple and easy-to-do sketches. Focus on improving your skills gradually.\n4. Use a variety of tools and mediums: Experiment with different tools like pencils, pens, markers, and different mediums like paper, canvas, or digital apps to keep your drawing practice interesting and engaging.\n5. Take breaks and rest: Taking breaks and resting after some time of drawing can help you avoid burnout and stay motivated.\n6. Challenge yourself: Set challenges like drawing objects from memory or a specific subject to improve your skills and keep your drawing practice interesting.\n7. Track your progress: Keep a record of your daily drawing practice and track your progress. This can be a source of motivation and help you see how far you've come.\n\nRemember, developing a habit takes time and patience. Stay consistent with your drawing practice, be flexible and open to trying new things, and with time, you'll develop a habit of daily drawing that brings you joy and satisfaction.
{Rejected Output: As an AI language model, I cannot personally develop habits for you. But, here are some tips for developing a habit of drawing daily:\n\n1. Start small: Start with simple drawings or doodles and gradually increase the complexity of your drawings.\n\n2. Set a schedule: Set a specific time every day to draw, and try to stick to it as much as possible.\n\n3. Make it fun: Don't pressure yourself to create masterpieces every time you draw. Make it a relaxing and enjoyable experience.\n\n4. Use resources: There are many drawing tutorials available online. Use resources like YouTube or online drawing courses to help you improve your skills.\n\n5. Surround yourself with inspiration: Expose yourself to a variety of art forms, such as paintings, illustrations, and photographs, to inspire and motivate you.\n\nRemember, everyone has their own creative style and pace. Just keep practicing and enjoying the process of drawing.

Nectar
{Input: 0.002 = 1000 \n1 = x?
{Chosen Output: To find the value of x, we can set up a proportion using the given information:\n\n0.002/1000 = 1/x\n\nTo solve for x, we can cross multiply:\n\n0.002 * x = 1000 * 1\n\n0.002x = 1000\n\nDividing both sides by 0.002:\n\nx = 1000 / 0.002\n\nx = 500,000\n\nTherefore, 1 is equal to 500,000 in this proportion.
{Rejected Output: 0.001 = x

Details on GPT-4o evaluation We construct the prompt for GPT-4o evaluation (as of August 2024) by introducing the input query and the two responses from MRPO (method A) and DPO (method B) LLMs. We then manually check the GPT-4o's response to determine the winner. See below for prompt examples and GPT-4o responses.

{Prompt: Input from Human: what do i do if i stepped in dog poo?

Method A output: Oh no, that's definitely not a fun situation to be in! If you've stepped in dog poo, here are a few things you can do to clean yourself up and prevent any potential health issues: 1. Wash your feet thoroughly with soap and water as soon as possible. Make sure to scrub between your toes and under your nails, where the dog poo may have gotten stuck. 2. Use hand sanitizer or rubbing alcohol to clean your hands, especially after touching your feet. 3. Clean your shoes or socks with soap and water, or dispose of them if they're badly contaminated. 4. If you're wearing pants or a skirt, check to see if there's any dog poo on them and clean them as well. 5. Consider carrying a small container of hand sanitizer or wipes with you in the future to clean up

---

Method B output: Great question! If you've stepped in dog poo, it's important to clean your foot thoroughly as soon as possible to prevent the risk of infection. Here's what you can do: 1. Wash your foot: Use soap and warm water to wash your foot, paying special attention to the area where you stepped in the dog poo. Make sure to rinse your foot thoroughly to remove any remaining bacteria. 2. Disinfect your foot: After washing your foot, use a disinfectant spray or wipe to cover your foot and between your toes. This will help kill any remaining bacteria and prevent infection. 3. Clean your shoes: If you were wearing shoes when you stepped in the dog poo, make sure to clean them as well. Use soap and water to wipe down the soles and inside of the shoes, and let them air dry.

Consider the response from two methods A and B above, which method is better in terms of helpfulness?

{GPT-4 Response: Between the two methods (A and B) for advising someone who stepped in dog poo, Method A is better in terms of helpfulness for the following reasons:
Comprehensive Instructions: Method A provides more detailed steps, covering not only the cleaning of the foot but also checking and cleaning other possible contaminated areas such as shoes, socks, and clothing. This thoroughness helps ensure all potential contamination is addressed.
Preventive Advice: Method A suggests carrying hand sanitizer or wipes in the future, offering proactive advice to help the person avoid or better handle similar situations in the future.
Health Considerations: Both methods mention cleaning to prevent infection, but Method A also emphasizes hand hygiene by suggesting the use of hand sanitizer or rubbing alcohol after touching the feet, which adds an extra layer of precaution.
Method B, while also helpful, is less detailed and doesn't cover the possibility of contamination on clothing or emphasize hand hygiene to the same extent.
Therefore, Method A is more helpful because it provides a broader and more detailed set of instructions that addresses the situation more completely.

Dataset	GSM8K	TruthfulQA	HellaSwag	MMLU	Arc-easy	Winograd	Avg.
Base (Mistral)	49.6	44.5	62.8	63.1	83.5	74.4	62.6
DPO	53.7	**53.3**	**66.6**	60.11	82.7	**73.6**	65.0
MRPO ($K=2$, M←L)	**56.1**	51.05	65.13	61.42	**84.4**	74.2	**65.4**
Base (LLama)	**23.9**	37.8	57.8	46.4	60.8	66.4	51.0
DPO	22.0	39.5	**59.4**	**46.3**	73.5	**67.3**	51.4
MRPO ($K=2$, L←M)	22.4	**45.3**	58.1	46.4	**74.1**	66.8	**52.2**

Table 8: Test performance ($\times 100$) across HuggingFace Open LLM Leaderboard datasets when $K=2$. Bold denotes best.

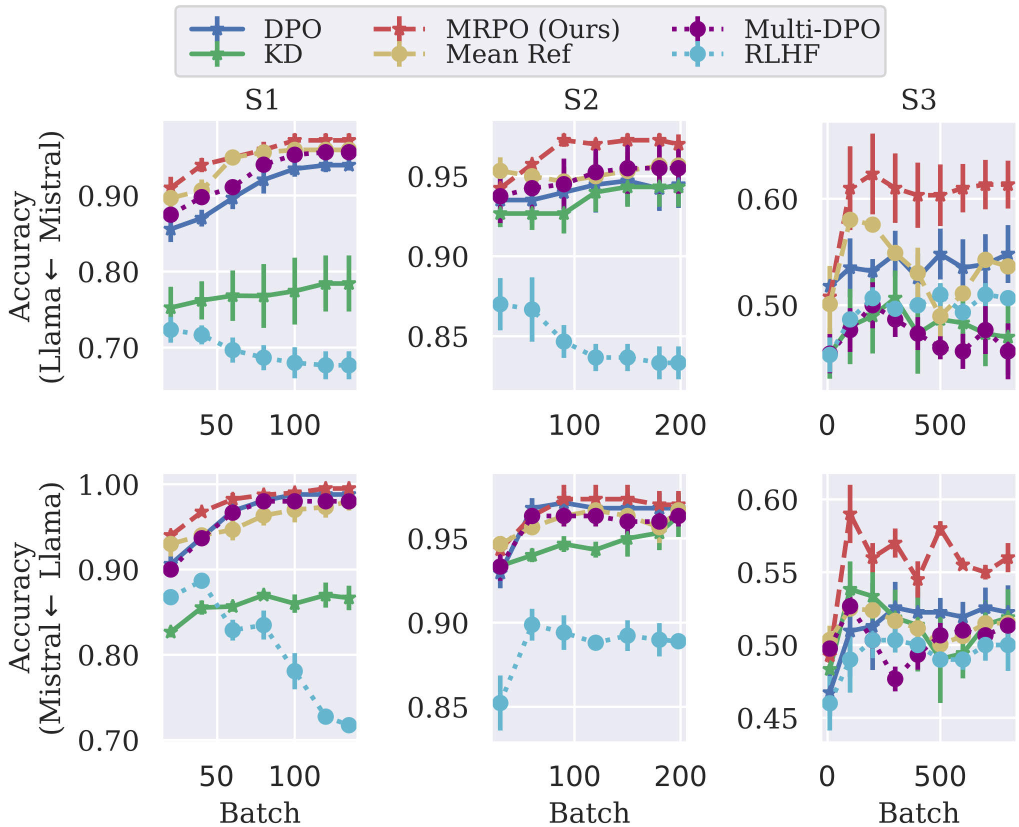

Figure 1: Chosen/Rejection preference accuracy on 3 small datasets: S1, S2 and S3. The curves show mean and std. of preference accuracy on test sets over training batches for 5 runs. In the first row, for multi-reference methods, RefM1 is LLama, and RefM2 is Mistral. In the second row, this order is reversed.

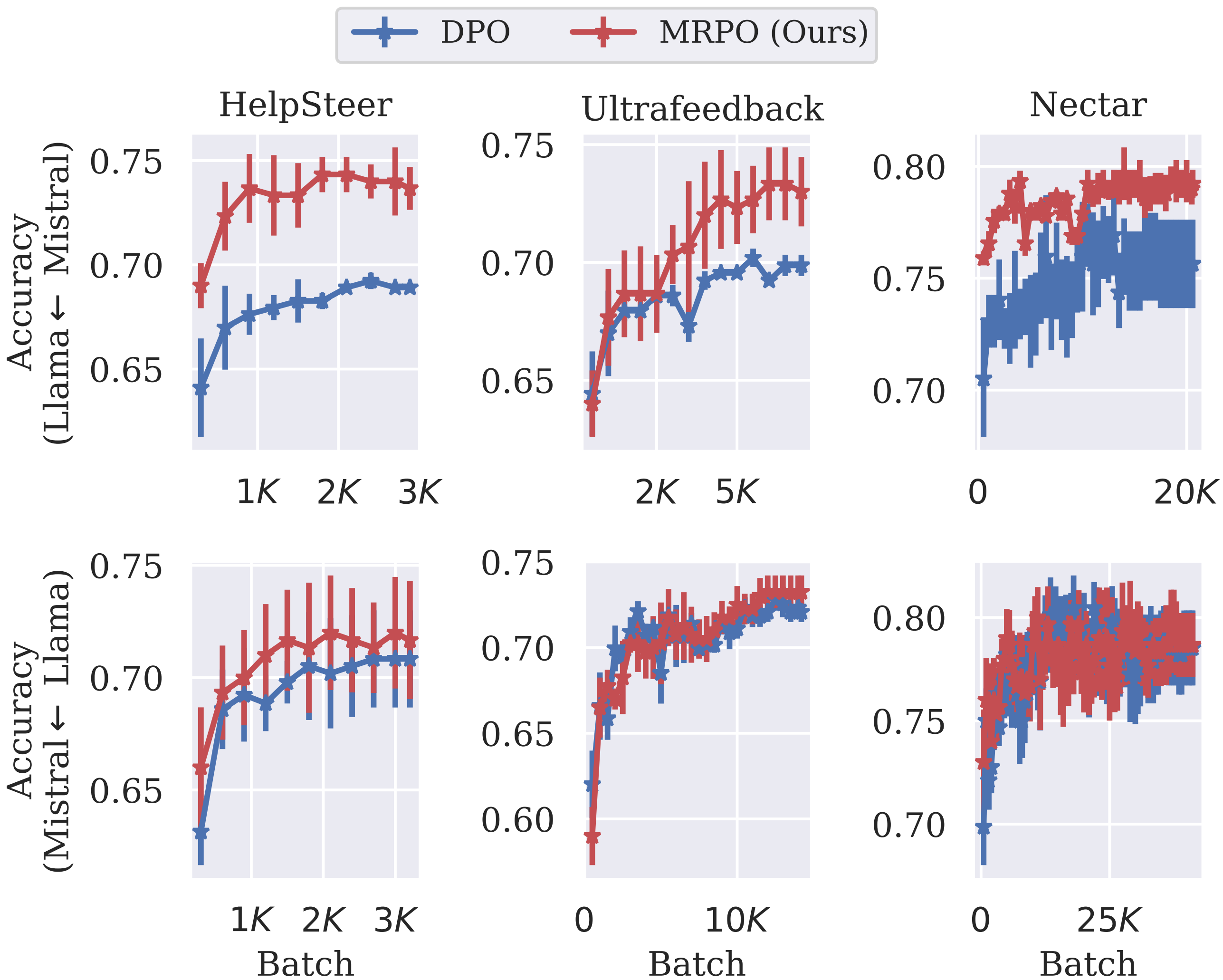

Figure 2: Chosen/Rejection preference accuracy on 3 big datasets: HelpSteer, Ultrafeedback and Nectar. The curves show the mean and std. of preference accuracy on test sets over training batches for 3 runs.

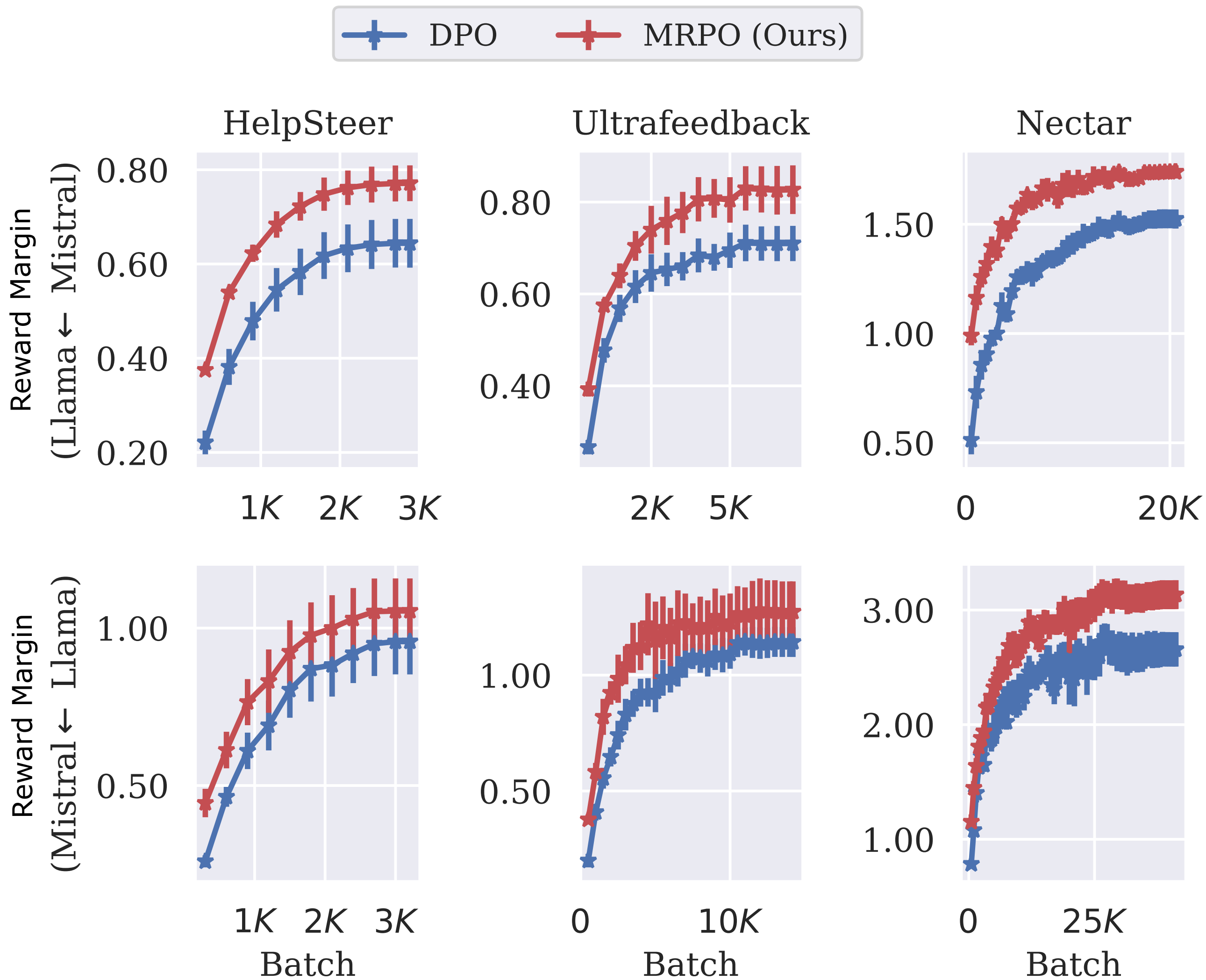

Figure 3: Reward Margin on 3 big datasets: HelpSteer, Ultrafeedback and Nectar. The curves show the mean and std. of reward margin on test sets over training batches for 3 runs.

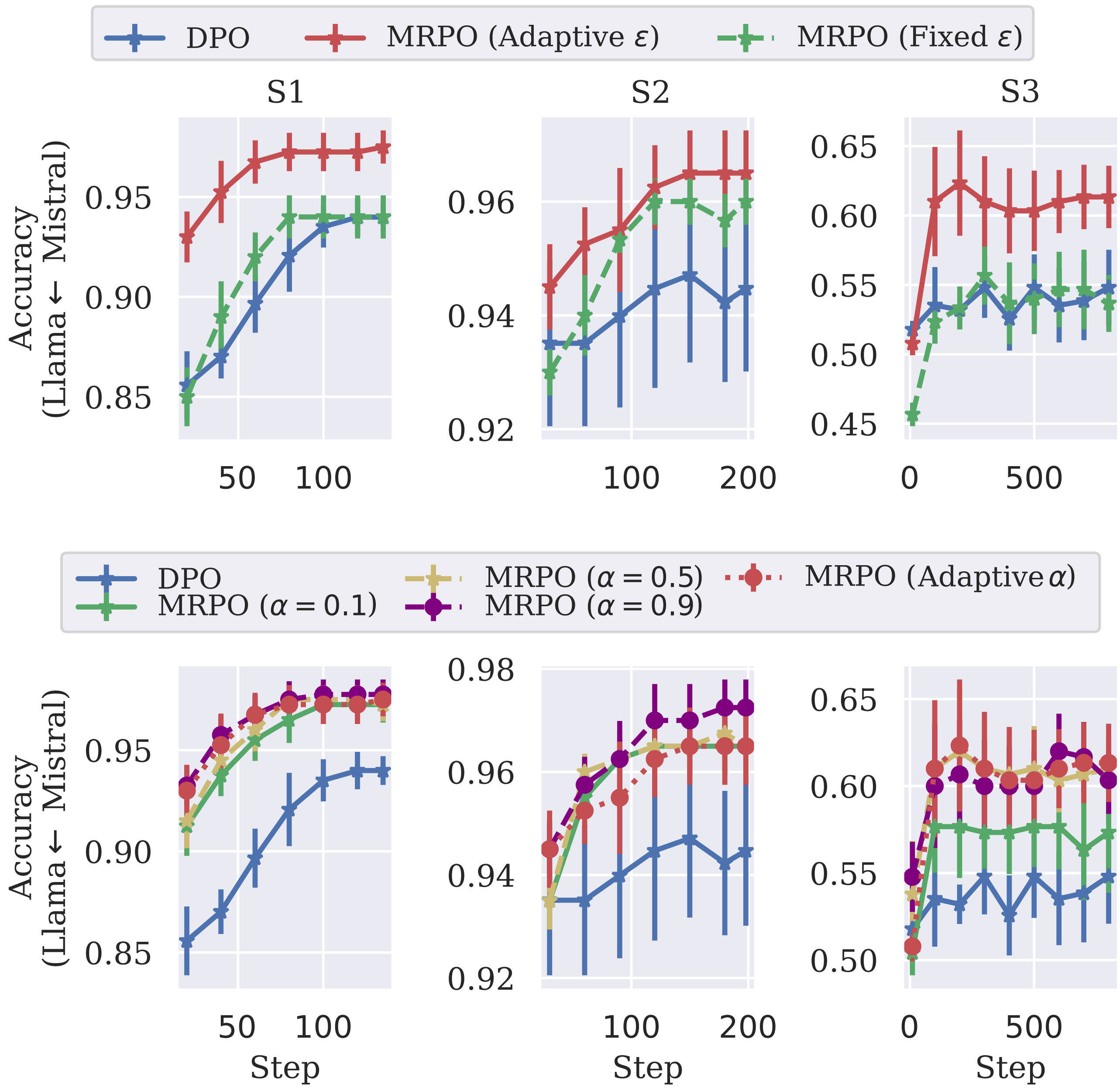

Figure 4: Analysis on ϵ and α using S1, S2 and S3 datasets. The curves show the mean and std. of testing preference accuracy over training batches for 5 runs. In the first row, adaptive $\epsilon_{max} = 0.1$ is compared with fixed $\epsilon = 0.1$. In the second row, adaptive α is compared with different fixed values of α.

Datasets	S1		S2		S3	
	T	S	T	S	T	S
DPO	1.1	5.6	2.3	3.4	6.3	8.2
MRPO	1.2	5.6	2.5	3.4	6.8	8.2

Table 9: Training time (T, hours) and speed (S, iters/s).

Computation Efficiency

In practical implementation, MRPO achieves efficiency by allowing precomputation of log probabilities from reference LLMs on training data, loading one reference LLM into memory at a time. Since log-prob computation is significantly faster than generating full sentences, this process is quick and enables MRPO to optimize the main LLM as fast as DPO. Also, after precomputation, MRPO and DPO have similar memory requirement during training. As shown in Table 9, the total training time (including log-prob precomputation for MRPO) and iteration speed (iters/s) for MRPO closely match those of DPO, demonstrating that MRPO runs just as fast while delivering enhanced performance.