Triple Preference Optimization: Achieving Better Alignment with Less Data in a Single Step Optimization

Anonymous ACL submission

Abstract

Large Language Models (LLMs) perform well across diverse tasks, but aligning them with human demonstrations is challenging. Recently, Reinforcement Learning (RL)-free methods like Direct Preference Optimization (DPO) have emerged, offering improved stability and scalability while retaining competitive performance relative to RL-based methods. However, while RL-free methods deliver satisfactory performance, *they require significant data to develop a robust Supervised Fine-Tuned (SFT) model* and *an additional step to fine-tune this model on a preference dataset*, which constrains their utility and scalability. In this paper, we introduce **Triple Preference Optimization (TPO)**, a new preference learning method designed to align an LLM with three preferences without requiring a separate SFT step and using considerably less data. Through a combination of practical experiments and theoretical analysis, we show the efficacy of TPO as a single-step alignment strategy. Specifically, we fine-tuned the Phi-2 (2.7B) and Mistral (7B) models using TPO directly on the UltraFeedback dataset, achieving superior results compared to models aligned through other methods such as SFT, DPO, KTO, IPO, CPO, and ORPO. Moreover, the performance of TPO without the SFT component led to notable improvements in the MT-Bench score, with increases of **+1.27** and **+0.63** over SFT and DPO, respectively. Additionally, TPO showed higher average accuracy, surpassing DPO and SFT by **4.2%** and **4.97%** on the Open LLM Leaderboard benchmarks.

1 Introduction

LLMs are trained across a wide array of tasks, demonstrating their remarkable versatility in solving diverse tasks (Brown et al., 2020; Narayanan et al., 2021; Bubeck et al., 2023). However, their training on data of varying quality can lead to many issues, such as the generation of toxic or harmful text under certain contexts (Perez et al., 2022; Ganguli et al., 2022), and in general, the generation of outputs that are not desired by humans. Hence, it is crucial to align LLMs with human expectations and preferences that prioritize their helpfulness, honesty, and harmlessness (Bai et al., 2022).

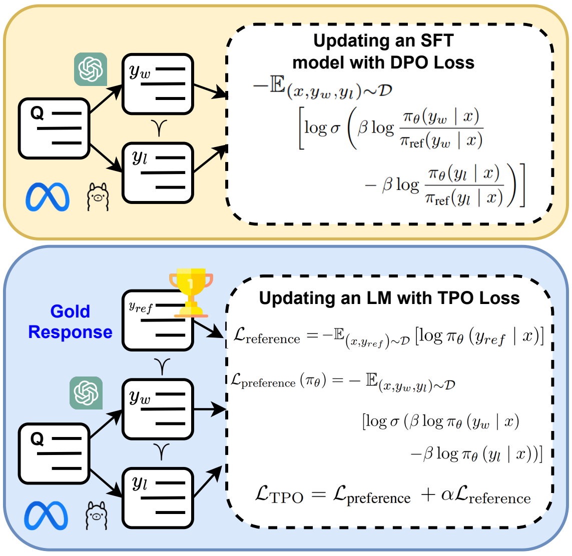

Figure 1: Comparison of the loss functions of TPO and DPO. TPO's loss function incorporates two main objectives. Its first term optimizes the log probability of preferences ($\mathcal{L}_{\text{preference}}(\pi_\theta)$), which demonstrates that optimizing preferences doesn't necessitate a reference model (See Section 3). Through its second term, TPO aims to learn the gold standard response ($\mathcal{L}_{\text{reference}}$). This aspect of the loss function is regulated by a parameter α, which serves as a parameter controlling the extent to which the policy model learns the gold standard response.

Supervised Fine-Tuning (SFT) is a direct alignment method that involves fitting a model to human-written data (Sanh et al., 2022). However, this approach fails to fully impart the human perspective to the model. During training, the model only receives a reference response for each input, thus lacking exposure to incorrect answers and preferences, which ultimately constrains its performance on downstream tasks (Touvron et al., 2023).

A prominent method in AI alignment for LLMs is Reinforcement Learning with Human Feedback (RLHF) (Ouyang et al., 2022). Despite its impressive performance relative to SFT, RLHF faces limitations such as instability and susceptibility to reward hacking (Liu et al., 2024). Consequently, a recent approach called Direct Preference Optimization (DPO) (Rafailov et al., 2023) has emerged. DPO is an RL-free method that directly optimizes human preferences by shifting from RL to simple binary cross-entropy. However, DPO encounters several limitations: 1) high dependency on the SFT part (Tunstall et al., 2023), 2) tendency to overfit beyond a single epoch (Azar et al., 2023), and 3) inefficient learning and memory utilization (Xu et al., 2024).

To address these limitations, various alignment methods have been proposed for dialogue systems (Tunstall et al., 2023), harmful and helpfulness question answering (Wu et al., 2023), summarization (Zhao et al., 2023), and translation (Xu et al., 2024) and all these studies include a separate SFT component. During SFT, models are fine-tuned to generate appropriate responses to the corresponding input prompts. Meanwhile, in DPO, models are fine-tuned to enhance the likelihood of generating preferred responses over less desirable ones and not to stray far away from the SFT model (Rafailov et al., 2023).

In this paper, we introduce the **Triple Preference Optimization (TPO)**, a new preference learning approach. In TPO, we combine the two separate optimization steps (supervised fine-tuning and preference learning) into a single step based on Pareto Front concept (Lotov and Miettinen, 2008), with the training data having both the gold standard response (as in SFT) and the preferences (as in PPO/DPO) in a consolidated format. Thus, our training data will be of the form *(input prompt, gold standard response* (y_{ref})*, preferred response* (y_w)*, less-preferred response* (y_l)*)*. Specifically, we jointly optimize a policy model with $-\mathbb{E}_{(x,y_{ref})\sim\mathcal{D}}\left[\log \pi_\theta\left(y_{ref} \mid x\right)\right]$ and $-\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \pi_\theta\left(y_w \mid x\right) - \beta \log \pi_\theta\left(y_l \mid x\right)\right)\right]$ in one step (See Figure 1).

Our results show that TPO exhibits impressive performance compared to SFT across various benchmarks and outperforms other alignment methods such as DPO. Specifically, Mistral (7B), fine-tuned by TPO and trained with **six times less data** than other alignment techniques, outperforms SFT, DPO, KTO, IPO, CPO, and ORPO across nine benchmarks on the Open LLM Leaderboard. Notably, Mistral aligned with TPO achieved a +0.72 increase in the MT-Bench score over SFT.

Overall, TPO addresses two key shortcomings in alignment tasks. Firstly, by removing π_{ref} justified in Section 3, TPO mitigates the inefficient learning and memory utilization issues observed in DPO, IPO, and KTO, allowing for more computational efficiency with less memory usage. Secondly, TPO enhances performance over SFT and other alignment methods by maximizing the likelihood of gold response, regularized by parameter $\boldsymbol{\alpha}$. and simultaneously optimizing between two preferences (preferred and less-preferred responses). Despite TPO's need for three preferences and its higher cost relative to other methods, our findings reveal that it's possible to considerably lessen the training data required and still achieve superior outcomes (See Table 1).

Our findings suggest that a separate SFT step is not necessary for TPO and, in certain scenarios, having one may even hinder TPO's performance (See Tables 1 and 2).

We summarize our primary contributions as follows:

1. We propose a new preference learning method called Triple Preferences Optimization (TPO) that simplifies the alignment process and reduces two stages to one stage.
2. Theoretically, we derive the TPO objective and show that combining the human expectation data and preference dataset achieves better performance.
3. Comprehensive experiments reveal that the TPO method, applied to two distinct baseline models—Mistral (7 B) and Phi-2 (2.7 B)—outperforms SFT, KTO, IPO, DPO, CPO, and ORPO in terms of performance across ten different benchmarks (refer to Tables 1, 2, and 3).
4. Integrating the SFT step with the preference alignment step and moderating it with a regularization parameter (α) enhances the model's performance while reducing the data required for training (See Figure 3).

2 Related Works

The performance of Large Language Models (LLMs) on a variety of tasks are remarkable (Anil et al., 2023). Nonetheless, effectively aligning LLMs remains a significant challenge. Current studies have fine-tuned LLMs using datasets of human preferences, leading to improvements in translation (Kreutzer et al., 2018), summarization (Stiennon et al., 2022), story-telling (Ziegler et al., 2019), instruction-following (Ramamurthy et al., 2023), and dialogue systems.

RLHF (Christiano et al., 2023) aims to optimize for maximizing the expected reward by interacting with a reward model trained using the Bradley-Terry (BT) model (Bong and Rinaldo, 2022), typically through RL-algorithms like Proximal Policy Optimization (Schulman et al., 2017). While RLHF enhances model performance, it faces challenges such as instability, reward hacking, and scalability inherent in RL-settings.

Recent works (Zhao et al., 2023; Yuan et al., 2023) have presented techniques to overcome these challenges by optimizing relative preferences without relying on reinforcement learning. In particular, DPO (Rafailov et al., 2023) offers a method to directly fit an SFT model to human preferences using the Bradley-Terry (BT) model, providing theoretical insights into the alignment process. However, IPO (Azar et al., 2023) has mathematically revealed the limitations of the DPO approach concerning overfitting and generalization. It proposes a comprehensive objective for learning from human preferences. Zephyr (Tunstall et al., 2023) has improved DPO by utilizing the distillation method.

KTO (Ethayarajh et al., 2023), drawing inspiration from Kahneman and Tversky's influential work on prospect theory (Tversky and Kahneman, 1992), seeks to maximize the utility of LLM outputs directly rather than optimizing the log-likelihood of preferences. By prioritizing the determination of whether a preference is desirable or undesirable, this method eliminates the requirement for two preferences for the same input.

Recently, CPO (Xu et al., 2024) introduced an efficient method for learning preferences by combining maximum-likelihood loss with the DPO loss function, aiming to improve memory usage and learning efficiency. Additionally, ORPO (Hong et al., 2024) proposed a novel approach by incorporating a penalty term to prevent the learning of unpreferred responses while enhancing the likelihood of learning preferred responses.

We observe two primary challenges in the alignment process addressed in mentioned studies. *Firstly*, alignment methods like DPO require an SFT component or perform better with one. *Secondly*, there are concerns regarding inefficient learning and memory usage. Although the CPO approach has shown effectiveness in learning, conflicts between its objectives may limit the policy model's performance. In this research, we explore these limitations and propose a new algorithm to address them.

3 Triple Preference Optimization

In this section, we introduce **Triple Preference Optimization (TPO)**, a new approach to preference learning. This method optimizes a policy model (π_θ) by maximizing the likelihood of the gold response and optimizing for the preferences simultaneously.

Typically, in NLP tasks, we utilize a dataset $D_{reference} = \{x^i, y^i_{ref}\}_{i=1}^N$, where x is the input and y_{ref} is the gold standard response, crafted by humans or large models like GPT-4 and validated by humans. Additionally, for applying preference optimization methods, a dataset $D_{preference} = \{x^i, y^i_w, y^i_l\}_{i=1}^N$ is needed, where y_w and y_l are the preferred and unpreferred responses respectively, generated by smaller models such as LLaMA-3. The aim of TPO is to optimize three preferences concurrently. To achieve this, we merge the $reference$ and $preference$ datasets into one dataset $D_{TPO} = \{x^i, y^i_{ref}, y^i_w, y^i_l\}_{i=1}^N$, establishing a response hierarchy of $y_{ref} \succ y_w \succ y_l$. Further details on the TPO objective will be discussed in the following subsection.

3.1 Deriving the TPO objective

Motivated by the goal of simplifying the alignment process to a single step and enhancing the learning mechanisms of the DPO, we derive the TPO objective. We start with a simple RL objective for aligning an LLM parameterized with θ, represented as π_θ with preferences. The RL objective is just maximizing the expected reward (Ziegler et al., 2019) as shown in Equation 1:

$$\max_{\pi_\theta} \left[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)}[r_\phi(x, y)] \right] \quad (1)$$

where r_ϕ represents the expected reward that the model receives for a given input x and output y. However, maximizing the reward without constraints can lead to distribution collapse in an LLM. Drawing inspiration from the Maximum Entropy Reinforcement Learning (MERL) framework (Hejna et al., 2023), we have modified the

RLHF objective, as detailed in Equation 4. The MERL framework aims to maximize causal entropy alongside the expected reward. This objective is formally defined in Equation 2.

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi_\theta(y|x)}[r_\phi(x,y)] + \beta\mathcal{H}_{\pi_\theta}(y|x)\right] \tag{2}$$

By definition of Entropy,

$$\mathcal{H}_{\pi_\theta}(y|x) = -\sum_y \pi_\theta(y|x)log(\pi_\theta(y|x)) \tag{3}$$

The objective becomes,

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y) - \beta\log\pi_\theta(y|x)\right] \tag{4}$$

Based on this, the optimal policy model induced by a reward function $r(x,y)$ could be derived as shown in Equation 5 (See Appendix A.1). It takes the following form:

$$\pi_r(y|x) = \frac{1}{Z(x)}\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{5}$$

where $Z(x) = \sum_y \exp\left(\frac{1}{\beta}r(x,y)\right)$ is the new partition function. Inspired by (Rafailov et al., 2023), we show that the reward function, in terms of the optimal policy that it induces, is calculated as per Equation 6 given below:

$$r(x,y) = \beta\log\pi_r(y|x) + \beta\log Z(x) \tag{6}$$

Subsequently, we can represent the ground-truth reward $r^*(x,y)$ in the form of its corresponding optimal policy π^* that it induces.

Since the Bradley-Terry model is dependent only on the difference between the two reward functions, i.e., $p^*(y_w > y_l|x) = \sigma(r^*(x,y_w) - r^*(x,y_l))$, where, we can reparameterize it as follows in Equation 7:

$$p^*(y_w > y_l \mid x) = \sigma\Big(\beta\log\pi^*(y_w \mid x) - \beta\log\pi^*(y_l \mid x)\Big) \tag{7}$$

Similar to the reward modeling approach, we model the human preferences, which is now in terms of a parameterized policy π_θ. Thus, we formulate maximum-likelihood objective (*preference* objective) for a dataset $D = \{x^i, y_w^i, y_l^i\}_{i=1}^N$ as outlined in Equation 8:

$$\mathcal{L}_{\text{preference}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\log\sigma\Big(\beta\log\pi_\theta(y_w \mid x) - \beta\log\pi_\theta(y_l \mid x)\Big)\Big] \tag{8}$$

Looking at the Equation 8, the objective is fitting an reward which is reparameterized as $r(x,y) = \beta\log\pi(y|x)$. In section 3.2, we theoretically explain that fitting this reward would ultimately recover the optimal policy.

The comparison between the loss function in Equation 8 and the DPO loss function indicates that the new function is more efficient because it requires only one model during training. However, even though maximizing the objective under the MERL setting prevents distribution collapse, it trains a pessimistic model, which also limits the model from learning the preferred responses effectively. To counteract this limitation, we maximize the likelihood of the gold response. The adjustment is specified in Equation 9.

$$\mathcal{L}_{\text{reference}} = -\mathbb{E}_{(x,y_{ref})\sim\mathcal{D}}\left[\log\pi_\theta\left(y_{ref} \mid x\right)\right] \tag{9}$$

Based on Equations 8, and 9, the TPO is defined as a multi-objective (bi-objective) optimization problem as supported by Pareto Front concept (Lotov and Miettinen, 2008). The TPO loss function is framed as follows:

$$\mathcal{L}_{\text{TPO}} = \mathcal{L}_{\text{preference}} + \alpha\mathcal{L}_{\text{reference}} \tag{10}$$

where hyper-parameter (α) plays a crucial role in moderating the model's learning of the gold response. The impact of the α on the model's performance is detailed in Section 4.3.

Insights into the TPO update. A deeper mechanistic understanding of TPO can be achieved by analyzing the gradient of the $\mathcal{L}_{\text{TPO}}$ loss function. The expression of this gradient in relation to the parameters θ is as follows:

$$\nabla_\theta \mathcal{L}_{\text{TPO}} = -\mathbb{E}_{(x,y_{ref},y_w,y_l)\sim\mathcal{D}} \big[\underbrace{\alpha \nabla_\theta \log \pi(y_{ref}|x)}_{\text{increase likelihood of } y_{ref}}$$
$$+ \beta\sigma(\underbrace{\beta \log \pi_\theta(y_l|x) - \beta \log \pi_\theta(y_w|x)}_{\text{increase weight when reward estimate is wrong}})$$
$$\times [\underbrace{\nabla_\theta \log \pi(y_w|x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l|x)}_{\text{decrease likelihood of } y_l}]\big] \quad (11)$$

where $r(x,y) = \beta \log \pi_\theta (y \mid x)$ is the reward inherently determined by the policy model π_θ. Intuitively, the gradient of the TPO loss function works to increase the likelihood of the gold completions y_{ref}, simultaneously enhancing the preference aspect by amplifying the likelihood of preferred completions y_w and reducing the likelihood of the less-preferred completions y_l, which are weighed by how incorrectly the implicit reward model orders the preferences. (more details on Appendix A.2). Notably, the hyper-parameters β and α significantly influence the performance of the policy model, as discussed further in Section 4.3.

3.2 Theory behind TPO

In this section, we provide a theoretical foundation for the TPO algorithm, drawing inspiration from (Rafailov et al., 2023). We observe that the preference optimization objective aligns with the principles of a Bradley-Terry model, where the reward parameterization is defined as $r(x,y) = \beta \log \pi_\theta(y|x)$. Consequently, we optimize our parametric model π_θ in a manner similar to reward model optimization, as shown by (Ouyang et al., 2022). We expand on the theory underlying this reparameterization of the reward function, illustrating that it does not constrain the range of reward models that can be modeled and ensures accurate retrieval of the optimal policy. We initiate this discussion by following the insights presented in DPO about the equivalent class of reward models.

Definition 3.1 *Two reward functions $r(x,y)$ and $r'(x,y)$ are equivalent iff $r(x,y) - r'(x,y) = g(x)$ for some function g.*

We can state the following two lemmas as it is apparent that there exists an equivalence relation, dividing the set of reward functions into distinct classes.

Lemma 3.1 *Under the Plackett-Luce, and in particular the Bradley-Terry preference framework, two reward functions from the same class induce the same preference distribution. (Rafailov et al., 2023)*

Lemma 3.2 *Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem. (Rafailov et al., 2023)*

The proofs are shown in Appendix A.3.

Theorem 3.1 *Under mild assumptions, all reward classes consistent with Plackett-Luce models can be represented with the reparameterization $r(x,y) = \beta \log \pi(y|x)$ for some model $\pi(y|x)$. (Rafailov et al., 2023)*

As proposed in DPO, upon imposing certain constraints on the under-constrained Plackett-Luce family of preference models, such that we preserve the class of representable reward model, it possible to explicitly make the optimal policy in Equation 5 analytically tractable for all prompts x. The theorem is elaborated in Appendix A.4. We further elaborate our theoretical basis for defining and optimally addressing the TPO objective within a multi-objective optimization framework.

Definition 3.2 *Let f_i denote i^{th} objective, $\mathcal{S}$ denote the feasible policy space, then in a multi-objective optimization setting, a policy $\pi^* \in \mathcal{S}$ is said to be Pareto optimal if there does not exist another policy $\pi \in \mathcal{S}$ such that $f_i(\pi) \leq f_i(\pi^*)$ for all $i = 1,...,k$ and $f_j(\pi) < f_j(\pi^*)$ for at least one index j.*

Looking at the objectives in Equation 8 and Equation 9, it is obvious that optimizing them together is non-trivial; that is, there does not exist a policy that is optimal with respect to both objectives. It can be seen that the objectives are conflicting with each other, especially when $y_{ref} \sim y_w$, as one objective is maximizing the log probability and the other is minimizing the log probability. This means that the objectives are at least partly conflicting. For a multi-objective problem, (Miettinen, 1999) show that optimizing one objective and converting the other objective/s as a constraint with an upper bound, the solution to this $\epsilon - constrained$ problem is Pareto optimal. This shows that optimizing the TPO objective, which is a bi-objective problem, gives an optimal policy that is Pareto optimal as defined in 3.2.

4 Experiments and Results

In this section, we present a comprehensive empirical analysis of TPO, yielding several key find-

Model	Align	ARC	TruthfulQA	Winogrande	HellaSwag	MMLU	Average
Mistral	SFT	60.41	43.73	74.19	81.69	60.92	64.18
Mistral+SFT	DPO	59.04	46.70	76.63	82.10	60	64.91
Mistral+SFT	IPO	59.30	42.22	76.4	81.02	59.93	63.77
Mistral+SFT	KTO	57.84	49.88	76.47	81.61	59.73	65.1
Mistral+SFT	CPO	57.50	53.22	75.92	80.37	58.41	65.08
Mistral	ORPO	58.61	52.77	77.5	82.04	63.26	66.83
Mistral+SFT	TPO (**our**)	58.02	59.05	76.47	80.6	59.48	66.72
Mistral	TPO (**our** $\alpha = 1 \mid \beta = 0.1$)	**61.34**	**60**	**78.21**	**83.18**	63.18	**69.18**
Mistral	TPO (**our** $\alpha = 0.9 \mid \beta = 0.2$)	60.23	57.34	78.29	83.01	**63.75**	68.52

Table 1: Comparing TPO's performance with other alignment methods reveals that the Mistral+TPO model exhibits comparable performance across different benchmarks and, on average, outperforms other methods. In particular, Mistral+TPO performed remarkably on the TruthfulQA benchmark. It's worth noting that the Mistral+TPO model is directly trained with TPO, which contributes to its superior performance. Additionally, for all benchmarks, accuracy is the metric used to gauge performance. More detail about ORPO in Appendix B.1.

Model	Align	MT-Bench	BB-causal	BB-sports	BB-formal	OpenBookQA
Mistral	SFT	5.94	51.57	61.76	**51.4**	43.8
Mistral+SFT	CPO	6.2	49.47	70.68	51.07	44.6
Mistral+SFT	DPO	6.64	52.1	71.9	51	46.2
Mistral+SFT	IPO	6.43	51.57	65.01	51.22	44.6
Mistral+SFT	KTO	6.48	53.68	73.42	51.33	45.8
Mistral	ORPO	5.47	54.21	**73.93**	50.4	44.4
Mistral+SFT	TPO (**our**)	**6.66**	54.21	**73.93**	50.84	45.6
Mistral	TPO (**our** $\alpha = 1 \mid \beta = 0.1$)	6.22	55.26	73.63	51.06	**48.2**
Mistral	TPO (**our** $\alpha = 0.9 \mid \beta = 0.2$)	**6.66**	**56.31**	73.32	50.5	47.8

Table 2: In our comparison of TPO with other alignment methods across more benchmarks, Mistral+SFT+TPO and Mistral+TPO emerge as the top performer, surpassing other methods in MT-Bench and BB-causal, BB-sports, OpenBookQA. For BB-causal, BB-sports, BB-formal, and OpenBookQA, performance is evaluated based on accuracy, while MT-Bench uses a scoring system generated by GPT-4. More detail about ORPO in Appendix B.1.

ings: 1) Phi-2+TPO and Mistral+TPO trained on 10K data outperform Phi-2+SFT and Mistral+SFT trained on 200K data by 12.7% and 7.2% on MT-Bench respectively. 2) Phi-2 fine-tuned with TPO surpasses the performance of models aligned with other methods on the MT-Bench. 3) Similarly, Mistral fine-tuned with TPO exceeds the performance of other alignment techniques across the majority of Open LLM Benchmarks. 4) Within the TPO method, the hyper-parameters α and β play a critical role in influencing performance outcomes. 5) An ablation study focusing on batch size adjustments reveals that enlarging the batch size leads to improved performance for models optimized with TPO.

4.1 Experimental Setup

Models. All experiments were conducted using `zephyr-sft-full` and `Mistral-7B-v0.1` as Mistral (7 B), and Phi-2 (2.7 B) (Javaheripi et al., 2023). We utilized the Transformer Reinforcement Learning (TRL) library for fine-tuning (von Werra et al., 2020). It's noted that the notation "+" is used to indicate that a model has been fine-tuned with a specific algorithm, such as "+TPO". Further training details for each method are in Appendix B.

Datasets. In this study, we employ two dialogue datasets: 1) UltraChat (Ding et al., 2023) and 2) UltraFeedback (Cui et al., 2023). UltraChat comprises 200k examples generated by GPT-3.5-TURBO across 30 topics and 20 text material types, offering a high-quality dataset utilized for training the SFT model. Meanwhile, UltraFeedback consists of a 64K set of responses generated by state-of-the-art models such as LLaMA-2 evaluated by a teacher model such as GPT-4. To train TPO, which requires three preferences, we create a custom dataset from the UltraFeedback dataset. Here, the response with the highest score serves as the reference response, the second-highest score as the chosen response, and the lowest score as the rejected response. In light of findings from (Saeidi et al., 2024), which indicate that alignment methods perform better with smaller training sets on one epoch, and due to computational limitations, we restrict our analysis to 12K (10K for training and 2K for evaluation) data points, randomly selected from the custom UltraFeedback dataset (More details in Appendix B).

Evaluation. We evaluate our models in both single-turn and multi-turn scenarios using the MT-Bench benchmark (Ding et al., 2023). MT-Bench is composed of 160 questions covering eight different knowledge domains, designed to be evaluated by GPT-4. To have a comprehensive evaluation we assess all alignment methods using five Open LLM Leaderboard benchmarks including ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), Truthful QA (Lin et al., 2022), and Winogrande (Sakaguchi et al., 2019). We further explore the performance of the models by evaluating them on four benchmarks from Big Bench (bench authors, 2023), including Causal Judgment (causal reasoning), Sports Understanding (commonsense reasoning), Formal Fallacies, and OpenBookQA (Mihaylov et al., 2018).

4.2 Demonstration of TPO Performance

We evaluate the TPO approach against other alignment techniques, such as KTO, IPO, CPO, DPO, and ORPO, using MT-Bench and the Open LLM Leaderboard Benchmarks. Our comparison involves two distinct model configurations: 1) the alignment of an SFT model using TPO and various other alignment methods, and 2) applying TPO directly to fine-tune a pre-trained model. Across all alignment approaches, we utilized Phi-2 (2.7 B) and Mistral (7 B) as the baseline models (More details in Appendix B). Additionally, we compared the ORPO method with a version that excludes the SFT part, the rationale for which is detailed in Appendix B.1.

MT-Bench. The data presented in Table 3 reveals that the Phi-2+TPO method outperforms other alignment techniques, enhancing the MT-Bench score by 12.7% and 7.2% over Phi-2+SFT+DPO and Phi-2+SFT, respectively. Remarkably, Phi-2+TPO achieves this superior performance even when trained on just 10K data, in stark contrast to Phi-2+SFT's training on 200K data (See Table 3). Additionally, the results in Table 2 demonstrate that Mistral+TPO surpasses competing alignment methods in MT-Bench scores. Mistral+TPO trained on 10K data shows a 7.2% improvement over Mistral+SFT, which is trained on 200K data.

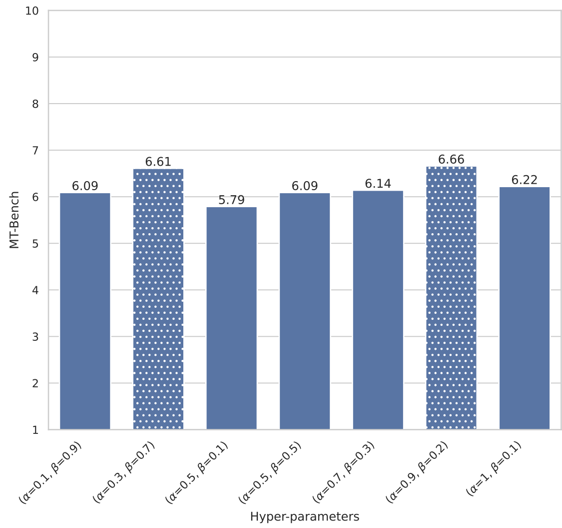

Figure 2: The MT-Bench score for various α and β settings in Mistral+TPO illustrates the influence of α on performance.

The results in Table 2 and Table 6 in the Appendix indicate that TPO exceeds the performance of other alignment methods, inspite of the SFT step being skipped (See Appendix C.1). Furthermore, additional experiments show that TPO achieves greater improvements over DPO, KTO, IPO, and CPO by 13.3%, 13.6%, 2.5%, and 13.3% respectively, on SFT trained on 10K data (See Appendix C.2).

Open LLM Leaderboard Benchmarks. The primary findings, as detailed in Table 1, highlight that Mistral+SFT+TPO, on average, surpasses other alignment methods. This superior performance is largely attributed to its notable success in the TruthfulQA benchmark despite lagging behind Mistral+SFT+DPO in performance. An intriguing observation from the data is that Mistral+TPO not only excels on average but also leads in performance across all benchmarks, showcasing the effectiveness of the TPO strategy. Specifically, Mistral+TPO achieved average accuracy improvements over Mistral+SFT, Mistral+SFT+DPO, Mistral+SFT+IPO, Mistral+SFT+KTO, Mistral+SFT+CPO, and Mistral+ORPO by 4.97%, 4.27%, 5.37%, 4.07%, 4.07%, and 2.35%, respectively. For additional results, readers are directed to Appendix D.

Model	Alignment Method								
	+SFT	**+SFT+DPO**	**+SFT+IPO**	**+SFT+KTO**	**+SFT+CPO**	**+ORPO**	**+SFT+ORPO**	**+SFT+TPO**	**+TPO**
Phi-2	5.42	6.06	5.91	6.64	6.42	6.06	4.32	6.18	**6.69**

Table 3: The comparison of Phi-2's performance when aligned with various methods on MT-Bench shows that Phi-2+TPO surpasses other alignment techniques. More detail about ORPO in Appendix B.1.

Exploration on More Benchmarks. For a comprehensive evaluation, we assessed the efficacy of the TPO method against various alignment strategies across different benchmarks: BB-causal, BB-sports, BB-formal, and OpenBookQA. As detailed in Table 2, Mistral+SFT+TPO exhibited superior performance on BB-causal and BB-sports benchmarks, while it showed less impressive results on BB-formal and OpenBookQA. Notably, Mistral+TPO not only enhanced the Mistral+SFT+TPO's outcomes on BB-causal and OpenBookQA but also surpassed Mistral+SFT, Mistral+SFT+DPO, Mistral+SFT+IPO, Mistral+SFT+KTO, Mistral+SFT+CPO, and Mistral+ORPO in accuracy by 4.81%, 1.71%, 3.91%, 1.01%, 3.01%, and 1.3%, respectively. Additional results can be found in Appendix D.

4.3 Ablation Studies

In this subsection, we delve into the impact of α and β values, batch size, and learning rate on the performance of the TPO method. Central to our exploration is the TPO method's ability to bypass the SFT stage, thereby assessing its efficacy without this component. Our evaluation focuses on the MT-Bench score and the Open LLM Leaderboard benchmarks to gauge the models' performance.

Impact of α and β. Alpha and Beta serve as crucial hyper-parameters that simultaneously enhance the likelihood of the correct response and refine preference learning. Figure 2 illustrates that the Mistral+TPO model, when set with α=0.9 and β=0.2, outperforms alternatives in terms of performance on the MT-Bench. Additionally, Figure 3 highlights that Mistral+TPO notably excels in the Open LLM Leaderboard benchmarks, boasting an average accuracy performance increase of 5.12% over the SFT method.

Other hyper-parameters. We extend our analysis to examine the influence of various hyperparameters on the TPO's efficacy, including different epochs, learning rates, and batch sizes, specifically with the Mistral+TPO model. We discovered that the learning rate is particularly critical when dealing with smaller datasets; a change by two orders of magnitude prevented the model from converging. Additionally, while different batch sizes do affect performance, there's a threshold beyond which performance plateaus and no longer benefits from increases. Interestingly, we observed that Mistral+TPO, when trained on 10K data, tends to overfit after just one epoch, with additional epochs failing to enhance performance. Nonetheless, we hypothesize that performance improves with larger datasets beyond the initial epoch, as detailed further in Appendix E.

5 Conclusions

In this paper, we begin by addressing the limitations inherent in existing alignment methods. Typically, alignment techniques require an SFT component to achieve notable results. However, incorporating SFT introduces two primary challenges: firstly, fine-tuning a model using SFT demands a substantial dataset (for example, completing a chat task may require fine-tuning with 200K data points). Secondly, generating a preferences dataset by sampling from the SFT model poses additional difficulties, including determining the optimal configuration for producing preferred and less preferred responses. To address these shortcomings, we introduce TPO, a new alignment approach aimed at concurrently optimizing for human preferences and gold responses. Our findings demonstrate the impressive performance of TPO compared to other alignment methods on ten benchmarks. Particularly, Mistral and Phi-2 fine-tuned by TPO achieve increases in the MT-Bench score of +0.72 and +1.27, respectively, compared to SFT, despite being trained on a dataset six times smaller. Another intriguing insight is the significant influence that the values of α and β have on the model's performance.

Limitations and Future Works

While TPO has demonstrated impressive performance compared to other alignment methods across various benchmarks, the requirement to prepare three preferences for each input in a dataset poses challenges. In this section, we outline potential directions for future work. Our evaluation of TPO focused on chat completion tasks, but we are particularly interested in examining its effectiveness in other areas, such as safety and reasoning. Another intriguing aspect for further study is investigating how the quality of reference and preferred responses affects TPO's performance. Notably, our current findings suggest that the reference response is generally better than the preferred response. Investigating whether increasing the preferential difference between these responses enhances performance could yield valuable insights. Additionally, we are interested in exploring TPO's effectiveness in larger models, such as those with 30 B or 70 B, which represents a promising avenue for future work. Drawing inspiration from the new method proposed in (Chatterjee et al., 2024) for fine-tuning diffusion models, we are keen to investigate how these models perform when aligned using the TPO method.

Ethics Statement

We have used AI assistants (Grammarly and ChatGPT) to address the grammatical errors and rephrase the sentences.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences.

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

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Heejong Bong and Alessandro Rinaldo. 2022. Generalized results for the existence and consistency of the mle in the bradley-terry-luce model.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4.

Agneet Chatterjee, Gabriela Ben Melech Stan, Estelle Aflalo, Sayak Paul, Dhruba Ghosh, Tejas Gokhale, Ludwig Schmidt, Hannaneh Hajishirzi, Vasudev Lal, Chitta Baral, et al. 2024. Getting it right: Improving spatial consistency in text-to-image models. *arXiv preprint arXiv:2404.01197*.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences.

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.

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.

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

Kawin Ethayarajh, Winnie Xu, Dan Jurafsky, and Douwe Kiela. 2023. Human-aware loss functions (halos). Technical report, Contextual AI.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, Andy Jones, Sam Bowman, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Nelson Elhage, Sheer El-Showk, Stanislav Fort, Zac Hatfield-Dodds, Tom Henighan, Danny Hernandez, Tristan Hume, Josh Jacobson, Scott Johnston, Shauna Kravec, Catherine Olsson, Sam Ringer, Eli Tran-Johnson, Dario Amodei, Tom Brown, Nicholas Joseph, Sam McCandlish, Chris Olah, Jared Kaplan, and Jack Clark. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned.

Joey Hejna, Rafael Rafailov, Harshit Sikchi, Chelsea Finn, Scott Niekum, W Bradley Knox, and Dorsa Sadigh. 2023. Contrastive prefence learning: Learning from human feedback without rl. *arXiv preprint arXiv:2310.13639*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding.

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

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. 2018. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J. Liu, and Jialu Liu. 2024. Statistical rejection sampling improves preference optimization.

Alexander V. Lotov and Kaisa Miettinen. 2008. *Visualizing the Pareto Frontier*, pages 213–243. Springer Berlin Heidelberg, Berlin, Heidelberg.

Kaisa Miettinen. 1999. *Nonlinear multiobjective optimization*, volume 12. Springer Science & Business Media.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Anand Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on gpu clusters using megatron-lm.

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

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models.

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

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2023. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization.

Amir Saeidi, Shivanshu Verma, and Chitta Baral. 2024. Insights into alignment: Evaluating dpo and its variants across multiple tasks. *arXiv preprint arXiv:2404.14723*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. 2022. Learning to summarize from human feedback.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Amos Tversky and Daniel Kahneman. 1992. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. `https://github.com/huggingface/trl`.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment.

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*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears.

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

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. Slic-hf: Sequence likelihood calibration with human feedback.

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

Appendix

A Derivation

A.1 Deriving the optimal policy under the Preference Objective

In this section, we derive the optimal policy achieved by optimizing the objective in Equation 4. For a given prompt x, the objective can be analogously written as follows:

$$\max_{\pi} \mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta \log \pi(y|x)\right] s.t. \sum_{y} \pi(y|x) = 1$$

Next, we form a lagrangian for the above objective with λ being the lagrangian multiplier.

$$\mathcal{L} = \sum_{y} \pi(y|x) r(x,y) - \beta \Bigg[\sum_{y} \pi(y|x) \log \pi(y|x)\Bigg] + \lambda \Bigg[1 - \sum_{y} \pi(y|x)\Bigg]$$

Differentiating $\mathcal{L}$ with respect to $\pi(y|x)$ results in,

$$\frac{\partial \mathcal{L}}{\partial_{\pi(y|x)}} = r(x,y) - \beta \Bigg[\log \pi(y|x) + 1\Bigg] - \lambda$$

To obtain the optimal policy, we can set the above equation to zero and solve for $\pi(y|x)$.

$$r(x,y) - \beta \Bigg[\log \pi(y|x) + 1\Bigg] - \lambda = 0$$

$$\log \pi(y|x) = \frac{1}{\beta} r(x,y) - \frac{\lambda}{\beta} - 1$$

$$\pi(y|x) = \exp\left(\frac{1}{\beta} r(x,y)\right). \exp\left(\frac{-\lambda}{\beta} - 1\right)$$

Since $\sum_y \pi(y|x) = 1$, the second exponent is a partition function that does normalization as shown below:

$$\Bigg[\sum_{y} \exp\left(\frac{1}{\beta} r(x,y)\right)\Bigg]. \exp\left(\frac{-\lambda}{\beta} - 1\right) = 1$$

$$\exp\left(\frac{-\lambda}{\beta} - 1\right) = \Bigg[\sum_{y} \exp\left(\frac{1}{\beta} r(x,y)\right)\Bigg]^{-1}$$

Hence, the partition function $Z(x) = \sum_y \exp\left(\frac{1}{\beta} r(x,y)\right)$ and the optimal policy $\pi_r(y|x)$ induced by reward function $r(x,y)$ is therefore given by,

$$\pi_r(y|x) = \frac{1}{Z(x)} \exp\left(\frac{1}{\beta} r(x,y)\right) \quad (1)$$

Now, we can express the reward function in terms of an optimal policy π_r by performing some algebraic transformations on Equation 1 as shown below,

$$\pi_r(y|x).Z(x) = \exp\left(\frac{1}{\beta} r(x,y)\right)$$

Taking logarithm and multiplying by β on both sides,

$$r(x,y) = \beta \log \pi_r(y|x) + \beta \log Z(x) \tag{2}$$

A.2 Deriving the Gradient of the TPO Objective

In this section, we derive the gradient of the TPO objective:

$$\nabla_\theta \mathcal{L}_{\text{TPO}} = -\nabla_\theta \mathbb{E}_{(x, y_{ref}, y_w, y_l) \sim \mathcal{D}} \left[\alpha \log \pi_\theta(y_{ref}|x) + \log \sigma(\beta \log \pi_\theta(y_w|x) - \beta \log \pi_\theta(y_l|x)) \right] \tag{1}$$

We can rewrite the RHS of the Equation 1 as

$$\nabla_\theta \mathcal{L}_{\text{TPO}} = -\mathbb{E}_{(x, y_{ref}, y_w, y_l) \sim \mathcal{D}} \left[\underbrace{\alpha \nabla_\theta \log \pi_\theta(y_{ref}|x)}_{\text{(a)}} + \underbrace{\nabla_\theta \log \sigma(\beta \log \pi_\theta(y_w|x) - \beta \log \pi_\theta(y_l|x))}_{\text{(b)}} \right] \tag{2}$$

In equation 2, the part (b) can be rewritten with

$$u = \beta \log \pi_\theta(y_w|x) - \beta \log \pi_\theta(y_l|x)$$

$$\nabla_\theta \log \sigma(u) = \frac{1}{\sigma(u)} \nabla_\theta \sigma(u)$$

$$\nabla_\theta \log \sigma(u) = \frac{\sigma'(u)}{\sigma(u)} \nabla_\theta(u)$$

Using the properties of sigmoid function function $\sigma'(u) = \sigma(u)(1-\sigma(u)$ and $\sigma(-u) = 1 - \sigma(u)$,

$$\nabla_\theta \log \sigma(u) = \frac{\sigma(u)(1-\sigma(u))}{\sigma(u)} \nabla_\theta(u)$$

$$\nabla_\theta \log \sigma(u) = (1-\sigma(u)) \nabla_\theta(u)$$

$$\nabla_\theta \log \sigma(u) = \sigma(-u) \nabla_\theta(u)$$

$$\nabla_\theta \log \sigma(u) = \beta\sigma(\beta \log \pi_\theta(y_l|x) - \beta \log \pi_\theta(y_w|x)) \left[\nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x)\right] \tag{3}$$

Plugging Equation 3 into Equation 2 we get,

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{TPO}} = &- \mathbb{E}_{(x, y_{ref}, y_w, y_l) \sim \mathcal{D}} \left[\alpha \nabla_\theta \log \pi(y_{ref}|x)\right. \\ &+ \beta\sigma(\beta \log \pi_\theta(y_l|x) - \beta \log \pi_\theta(y_w|x)) \\ &\left. \times \left[\nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x)\right]\right] \end{aligned} \tag{4}$$

A.3 Proof of Lemma

In this section, we will prove the lemmas from Section 3.2.

Lemma 1 Restated. Under the Plackett-Luce preference framework, and in particular the Bradley-Terry framework, two reward functions from the same equivalence class induce the same preference distribution.

Proof. Let's consider two reward functions, $r(x, y)$ and $r'(x, y)$. They are said to be equivalent if they can be related by $r'(x, y) = r(x, y) + g(x)$ for some function g. We analyze this in the context of the general Plackett-Luce model, which includes the Bradley-Terry model (special case when $K = 2$). Here, we denote the probability distribution over rankings generated by a given reward function $r(x, y)$ as p_r. Given any prompt x, responses $y_1, ..., y_K$, and a ranking τ, we can establish the following:

$$
\begin{aligned}
p_{r'}(\tau \mid y_1, \ldots, y_K, x) &= \prod_{k=1}^{K} \frac{\exp(r'(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r'(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}) + g(x))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}) + g(x))} \\
&= \prod_{k=1}^{K} \frac{\exp(g(x)) \exp(r(x, y_{\tau(k)}))}{\exp(g(x)) \sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= p_r(\tau \mid y_1, \ldots, y_K, x),
\end{aligned}
$$

This completes the proof.

Lemma 2 Restated. Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem.

Proof. Let's consider two reward functions, $r(x, y)$ and $r'(x, y)$. They are said to be equivalent if they can be related by $r'(x, y) = r(x, y) + g(x)$ for some function g. Let π_r and $\pi_{r'}$ be the optimal policies induced by their corresponding reward functions. By Equation 5, for all x, y we have,

$$
\begin{aligned}
\pi_{r'}(y \mid x) &= \frac{1}{\sum_y \exp\left(\frac{1}{\beta} r'(x, y)\right)} \exp\left(\frac{1}{\beta} r'(x, y)\right) \\
&= \frac{1}{\sum_y \exp\left(\frac{1}{\beta}(r(x, y) + g(x))\right)} \exp\left(\frac{1}{\beta}(r(x, y) + g(x))\right) \\
&= \frac{1}{\exp\left(\frac{1}{\beta} g(x)\right) \sum_y \exp\left(\frac{1}{\beta} r(x, y)\right)} \exp\left(\frac{1}{\beta} r(x, y)\right) \exp\left(\frac{1}{\beta} g(x)\right) \\
&= \frac{1}{\sum_y \exp\left(\frac{1}{\beta} r(x, y)\right)} \exp\left(\frac{1}{\beta} r(x, y)\right) \\
&= \pi_r(y \mid x),
\end{aligned}
$$

This completes the proof.

A.4 Proof of Theorem

Theorem 1 Restated. *For a parameter $\beta > 0$, all reward equivalence classes can be reparameterized as $r(x, y) = \beta \log \pi(y|x)$ for some model $\pi(y|x)$.*

Proof. Consider a reward function $r(x, y)$, which induces an optimal model $\pi_r(y|x)$ under the MERL framework, which takes the form as shown in Eq.5 in Section 3.1. Following, Equation 2 in Section A.1 of Appendix, we have:

$$r(x,y) = \beta \log \pi_r(y|x) + \beta \log Z(x) \tag{1}$$

where $Z(x) = \sum_y \exp\left(\frac{1}{\beta} r(x,y)\right)$ is the partition function of the optimal policy induced by the reward function $r(x,y)$. Let $r^{'}(x,y)$ be a new reward function such that $r^{'}(x,y) = r(x,y) - \beta \log Z(x)$. It is obvious that the new reward function is within the equivalence class of r, and the we have:

$$r^{'}(x,y) = r(x,y) - \beta \log Z(x)$$

From the Equation 1, we get

$$r^{'}(x,y) = \beta \log \pi_r(y|x) + \beta \log Z(x) - \beta \log Z(x)$$

$$r^{'}(x,y) = \beta \log \pi_r(y|x)$$

This completes the proof.

Proposition 1. For a parameter $\beta > 0$, every equivalence class of reward functions has a unique reward function $r(x,y)$, which can be reparameterized as $r(x,y) = \beta \log \pi(y|x)$ for some model $\pi(y|x)$.

$Proof - by - Contradiction.$ Let us assume that we have two reward functions from the same class, such that $r^{'}(x,y) = r(x,y) + g(x)$. Assume that $r^{'}(x,y) = \beta \log \pi^{'}(y|x)$ for some model $\pi^{'}(y|x)$ and $r(x,y) = \beta \log \pi(y|x)$ for some model $\pi(y|x)$, such that $\pi^{'} \neq \pi$. We then have,

$$\begin{aligned} r^{'}(x,y) &= r(x,y) + g(x) \\ &= \beta \log \pi(y|x) + g(x) \\ &= \beta \log \pi(y|x) + \beta \log \exp\left(\frac{1}{\beta} g(x)\right) \\ &= \beta \log \pi(y|x) \exp\left(\frac{1}{\beta} g(x)\right) \\ &= \beta \log \pi^{'}(y|x) \end{aligned}$$

for all prompts x and completions y. Then, we must have $\pi(y|x) \exp\left(\frac{1}{\beta} g(x)\right) = \pi^{'}(y|x)$. Since these are probability distributions, summing over y on both sides,

$$\begin{aligned} \sum_y \left[\pi(y|x) \exp\left(\frac{1}{\beta} g(x)\right)\right] &= \sum_y \pi^{'}(y|x) \\ \exp\left(\frac{1}{\beta} g(x)\right) &= 1 \end{aligned}$$

Since $\beta > 0$, $g(x)$ must be 0 for all x. Therefore, we will have $r(x,y) = r^{'}(x,y)$, which contradicts our initial condition of $\pi^{'} \neq \pi$.

Thus, by contradiction, we have shown that every reward class has a unique reward function that can be represented by the reparameterization in Theorem 3.1.

Datasets	ARC	TruthfulQA	Winogrande	HellaSwag	MMLU	BB-causal	BB-sports	BB-formal	OpenBookQA
# few-shot	25	0	5	10	5	3	3	3	1
Metric	`acc_norm`	`mc2`	`acc`	`acc_norm`	`acc`	`mc`	`mc`	`mc`	`acc_norm`

Table 4: Detailed information of Open LLM Leaderboard and Big Bench benchmarks.

B Training and Evaluation Details

All models were trained using the AdamW optimizer without weight decay. Furthermore, parameter-efficient techniques such as LoRA (Hu et al., 2021) were not employed. The experiments were conducted on 4 A100 GPUs, utilizing bfloat16 precision, and typically required 5-8 hours to complete. All models are trained for one epoch, employing a linear learning rate scheduler with a peak learning rate of 5e-07 and 10% warmup steps. Additionally, the global batch size is set to 16, and $\beta = 0.1$ is used to regulate the deviation from the reference model. For every dataset used in our evaluation, we detail the count of few-shot examples utilized along with the specific metric employed for assessment in Table 4.

The custom UltraFeedback dataset includes y_{ref}, y_w, and y_l for each input x. For a fair comparison, when training alignment methods based on the SFT model, we utilized y_w and y_l under the assumption that the model was trained on y_{ref} during supervised fine-tuning. Conversely, in scenarios where we directly trained a model using alignment methods, we used y_{ref} and y_l.

B.1 Detail Evaluation for ORPO

The central hypothesis of the ORPO method (Xu et al., 2024) suggests that skipping the SFT component can achieve performance comparable to that of SFT and DPO methods. Based on this premise, it is essential to compare a model directly fine-tuned using ORPO against other alignment methods. To test this hypothesis, we designed two experiments: 1) Fine-tuning an SFT model using ORPO, and 2) Fine-tuning a pre-trained model using ORPO.

Model	Align	MT-Bench	ARC	TruthfulQA	Winogrande	HellaSwag	MMLU	BB-causal	BB-sports	BB-formal	OpenBookQA
Mistral	ORPO	5.47	58.61	52.77	77.5	82.04	63.26	54.21	73.93	50.41	44.4
Mistral+SFT	ORPO	4.93	53.92	48.03	75.69	79.69	59.62	50.52	71.19	51.07	43.4
Phi-2	ORPO	6.06	61.17	45.68	74.42	74.69	58.33	55.78	50.7	49.01	52.8
Phi-2+SFT	ORPO	4.32	55.11	49.15	74.74	70.38	55.36	54.21	50.91	49.27	44.8

Table 5: Comparison OPRO method on different scenarios.

The results presented in Table 5 indicate that, consistent with the hypothesis outlined in the paper (Xu et al., 2024), ORPO performs better when the SFT component is omitted. Thus, for our comparisons, we utilized the Mistral+ORPO and Phi-2+ORPO models.

C More Experiments

In this section, we assess the performance of alignment methods in two distinct scenarios: 1) skipping the SFT component and 2) aligning an SFT model that has been fine-tuned on a dataset of 10K instances using various alignment techniques.

C.1 Skipping the SFT Component

The primary benefit of using TPO is the ability to skip the SFT component, which often results in better performance for TPO without SFT. In this experiment, we also investigate the effectiveness of other alignment methods without the SFT part. For this purpose, we directly trained a `Mistral-7B-v0.1` model using various alignment techniques like DPO, KTO, IPO, CPO, and ORPO.

The results in Table 6 indicate that without the SFT component, both DPO and IPO fail to match the performance levels of Mistral+SFT. Additionally, the results for KTO and CPO show negligible differences when compared with SFT. Although ORPO recommends bypassing the SFT phase in the alignment process, it seems that a policy model fine-tuned with ORPO underperforms when only one epoch is used. A comparison between the results in Tables 2 and 6 reveals that most of the alignment methods perform better when the SFT part is retained.

Model	Align	MT-Bench
Mistral	SFT	5.94
Mistral	DPO	5.45
Mistral	KTO	6.21
Mistral	IPO	2.06
Mistral	CPO	6.3
Mistral	ORPO	5.47
Mistral	TPO (**our** $\alpha = 0.9 \mid \beta = 0.2$)	6.22
Mistral	TPO (**our** $\alpha = 0.3 \mid \beta = 0.7$)	6.61
Mistral	TPO (**our** $\alpha = 1 \mid \beta = 0.1$)	**6.66**

Table 6: Comparison of the performance of various alignment methods on skipping the SFT part using MT-Bench.

C.2 Aligning an SFT Model with Less Data

In this experiment, we investigate how alignment methods perform when applied to an SFT model trained on significantly less data. TPO utilizes the dataset $D = \{x^i, y^i_{ref}, y^i_w, y^i_l\}_{i=1}^N$. Initially, we fine-tune a `Mistral-7B-v0.1` model on 10K data, which are designated as y_{ref} for TPO. Subsequently, we applied various alignment methods to this fine-tuned model.

Model (training Size)	DPO	CPO	KTO	IPO
+ Mistral+SFT (200K)	6.64	6.2	6.48	6.43
+ Mistral+SFT (10K)	5.33	5.89	5.3	6.41

Table 7: Comparison of the performance of various alignment methods on different SFT models using the MT-Bench. Notably, the score for Mistral+SFT trained on 10K data is 4.2, while the score for Mistral+SFT trained on 200K data is 5.94.

The findings presented in Table 7 suggest that alignment methods yield superior results when applied to an SFT model trained on a larger dataset. It is evident that, when using the same data as for Mistral+TPO, other models perform significantly worse. These results confirm our hypothesis that TPO surpasses other methods with considerably less data.

D More results on Open LLM Leaderboard and Big Bench Benchmarks

Our assessment of Phi-2 through the Open LLM Leaderboard benchmarks, in comparison with various alignment methods, showed that Phi-2+TPO, trained on a dataset of 10K, achieved performance on par with other alignment strategies across the ARC, TruthfulQA, and MMLU benchmarks. Also, The results showed that this model performs better on BB-causal and OpenBookQA.

Model	Align	ARC	TruthfulQA	Winogrande	HellaSwag	MMLU	BB-causal	BB-sports	BB-formal	OpenBookQA
Phi-2	SFT	61	46.01	74.58	74.66	56.48	55.26	51.72	49.54	50.2
Phi-2+SFT	DPO	61.34	51.53	74.82	**75.88**	56.99	**57.36**	**52.63**	49.5	52.2
Phi-2+SFT	IPO	61.43	49.05	**75.05**	75.36	56.83	55.26	51.31	**49.69**	51.2
Phi-2+SFT	KTO	61	52.35	74.98	75.43	57.02	56.31	51.62	49.47	51.4
Phi-2+SFT	CPO	60.49	53.3	**75.05**	74.78	56.94	54.21	50.5	49.48	49.8
Phi-2	ORPO	61.17	45.68	74.42	74.69	58.33	55.78	50.7	49.01	52.8
Phi-2+SFT	TPO (**our**)	61.09	**53.6**	74.82	74.98	56.95	54.21	50.3	49.27	50.6
Phi-2	TPO (**our** $\alpha = 1 \mid \beta = 0.1$)	61.51	45.41	74.34	75.27	**58.38**	55.78	51.44	49.28	53.2
Phi-2	TPO (**our** $\alpha = 0.9 \mid \beta = 0.2$)	**61.6**	46.21	74.66	74.91	58.12	**57.36**	51.31	48.35	**53.4**

Table 8: Comparison between TPO and other alignment methods on Open LLM Leaderboard and Big Bench benchmarks based on Phi-2 model.

E More results on Ablation Studies

This section presents the performance of Mistral+TPO across various learning rate, epoch, and batch size utilizing the MT-Bench score as the benchmark for assessment.

In Figure 3 we compared TPO with SFT on different value of α and β on Open LLM Leaderboard benchmarks.

Model	Align	Learning Rate	Epoch	Batch Size	First Turn (Score)	Second Turn (Score)	Average (Score)
Mistral	TPO (α=1|β=0.1)	5e-07	1	16	6.78	5.66	6.22
Mistral	TPO (α=1|β=0.1)	2e-05	1	16	1	1	1
Mistral	TPO (α=0.9|β=0.2)	5e-07	1	16	7.12	6.2	6.66
Mistral	TPO (α=0.9|β=0.2)	5e-07	1	32	6.98	6.1	6.54
Mistral	TPO (α=0.9|β=0.2)	5e-07	2	16	7.2	6	6.61

Table 9: Performance of the Mistral+TPO on different values of hyper-parameters.

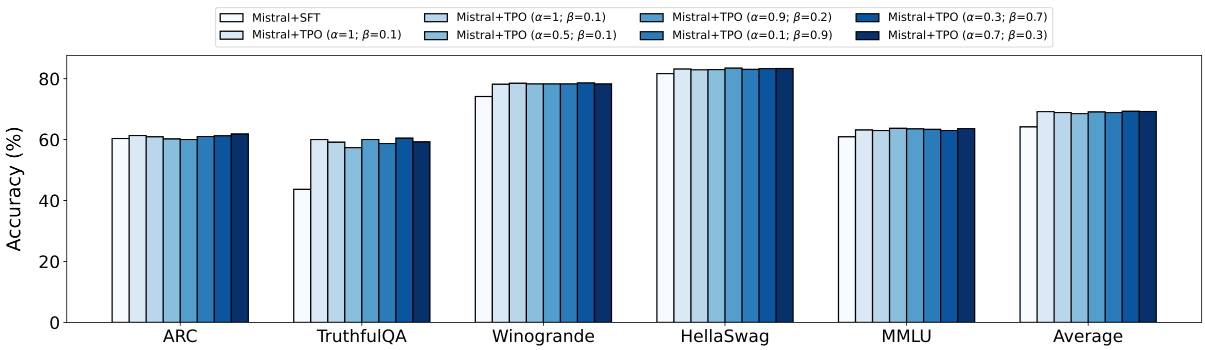

Figure 3: This figure displays the performance of Mistral+TPO across various settings of α and β. In several configurations, Mistral+TPO outperforms SFT on the Open LLM Leaderboard benchmarks. Further discussion is provided in Section 4.3.