

Eliminating Biased Length Reliance of Direct Preference Optimization via Down-Sampled KL Divergence

Junru Lu^{1*,3}, Jiazheng Li^{2*}, Siyu An³, Meng Zhao³, Yulan He^{1,2,4}, Di Yin³, Xing Sun³

¹University of Warwick ²King’s College London

³Tencent YouTu Lab ⁴The Alan Turing Institute

junru.lu@warwick.ac.uk, {jiazheng.li, yulan.he}@kcl.ac.uk

{siyuan, alexmzhao, endymecyyin, winfredsun}@tencent.com

Abstract

Direct Preference Optimization (DPO) has emerged as a prominent algorithm for the direct and robust alignment of Large Language Models (LLMs) with human preferences, offering a more straightforward alternative to the complex Reinforcement Learning from Human Feedback (RLHF). Despite its promising efficacy, DPO faces a notable drawback: “*verbosity*”, a common over-optimization phenomenon also observed in RLHF. While previous studies mainly attributed verbosity to biased labels within the data, we propose that the issue also stems from an inherent algorithmic length reliance in DPO. Specifically, we suggest that the discrepancy between sequence-level Kullback–Leibler (KL) divergences between chosen and rejected sequences, used in DPO, results in overestimated or underestimated rewards due to varying token lengths. Empirically, we utilize datasets with different label lengths to demonstrate the presence of biased rewards. We then introduce an effective downsampling approach, named SamPO, to eliminate potential length reliance. Our experimental evaluations, conducted across three LLMs of varying scales and a diverse array of conditional and open-ended benchmarks, highlight the efficacy of SamPO in mitigating verbosity, achieving improvements of 5% to 12% over DPO through debiased rewards¹.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) is a crucial strategy for effectively align Large Language Models (LLMs) with human minds (Zhao et al., 2023a; Yang et al., 2023; Pan et al., 2023b), showcasing significant improvements of LLM’s instruct-following capability compared with the other two popular approaches: pre-training and supervised fine-tuning (SFT). In fact, a

series of leading LLMs have adopted RLHF as the final stage of their entire training pipelines (Ouyang et al., 2022; Achiam et al., 2023; Bi et al., 2024).

Nevertheless, traditional RLHF involves several intricate multi-stage steps, typically starting with fine-tuning a reward model that captures complex human intuition (Bai et al., 2022), followed by optimizing LLMs to maximize preference scores. Therefore, the quality of the reward model is crucial. However, modeling elusive human intuition is inherently difficult (Wang et al., 2024). On the contrary, Direct Preference Optimization (DPO) (Rafailov et al., 2023) proposed to re-parameterize the reward model, integrating preference feedback from online rewards into offline labels. In specific, DPO employs the Bradley-Terry model (Bradley and Terry, 1952) to maximize implicit rewards via pairwise offline preference labels. The implicit reward is mathematically equivalent to the discrepancy in sequence-level Kullback–Leibler (KL) divergences (Kullback and Leibler, 1951) between chosen and rejected labels. The KL divergence for each label is calculated based on probability outputs from the fine-tuning policy model and a frozen reference model. DPO eliminates the need for complex prefix fine-tuning of an external reward model, while maintains performance comparable to RLHF (Dubois et al., 2024b; Hou et al., 2024).

Despite its effectiveness, DPO faces several notable challenges, including issues of overfitting (Azar et al., 2023; Jung et al., 2024), high computational costs (Ethayarajh et al., 2024; Hong et al., 2024), and verbosity (Hou et al., 2024; Park et al., 2024). This paper specifically focuses on addressing the “*verbosity*” issue.

Traditional multi-stage RLHF methods argue that due to a statistical bias in length distribution, that is, where preferred labels tend to be longer than rejected preference labels (Singhal et al., 2023; Park et al., 2024), the reward model trained on

^{*}Equal Contribution.

¹Our code can be accessed at: <https://github.com/LuJunru/SamPO/>.

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) \right], \text{ formalized with sequence-level KL divergence}$$

$$\mathcal{L}_{\text{DPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E} \left[\log \sigma \left(\beta \sum_{t=1}^T \log \frac{\pi_{\theta}(y_w^t|x)}{\pi_{\text{ref}}(y_w^t|x)} - \beta \sum_{t=1}^T \log \frac{\pi_{\theta}(y_l^t|x)}{\pi_{\text{ref}}(y_l^t|x)} \right) \right], \text{ decomposing KL divergence at token-level}$$

$$\mathcal{L}_{\text{SamPO}}(\pi_{\theta}; \pi_{\text{ref}}) = -\mathbb{E} \left[\log \sigma \left(\beta \sum_{t=1}^{T_m} \log \frac{\pi_{\theta}(y_w^t|x)}{\pi_{\text{ref}}(y_w^t|x)} - \beta \sum_{t=1}^{T_m} \log \frac{\pi_{\theta}(y_l^t|x)}{\pi_{\text{ref}}(y_l^t|x)} \right) \right], T_m = \min(T_w, T_l), y^t \sim \text{Uniform}(T_m, \{y\}^T)$$

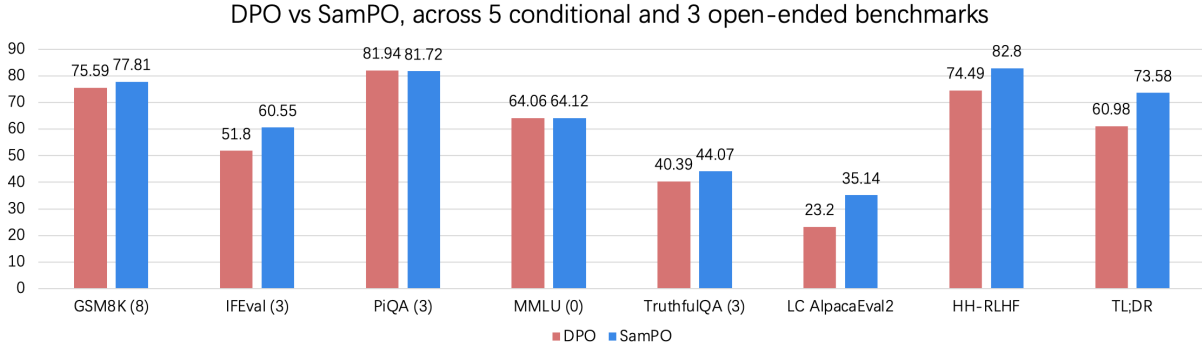


Figure 1: Down-Sampling strategy helps mitigate the potential length reliance, and thus improves DPO.

such preference data inherently exhibit a length bias (Shen et al., 2023). Therefore, subsequent fine-tuned policy model exploit this bias as a shortcut to achieve higher reward scores by generating longer responses (Gao et al., 2023a), without necessarily improving quality (Kabir et al., 2023; Dubois et al., 2024b). Various regularization approaches have been proposed to mitigate this inherent bias within reward models (Ramamurthy et al., 2022; Coste et al., 2023; Moskovitz et al., 2023; Chen et al., 2024b). On the other hand, although DPO does not explicitly use a reward model, the length distribution bias inherent in the offline preference labels still contributes to the verbosity issue (Hou et al., 2024; Rafailov et al., 2024). Analysis suggests that policy models trained using DPO tend to generate responses that are almost twice the length of the labeled data (Park et al., 2024).

In this paper, we propose that, in addition to the length bias in the data, DPO exhibits a hidden algorithmic dependence on response length. As illustrated in the upper portion of Figure 1, the loss function in DPO is based on the discrepancy between sequence-level KL divergence, which can also be computed and aggregated at the token-level. It is evident that discrepancies between chosen label y_w and rejected label y_l lead to an inadvertent reliance on auxiliary length features: training samples with longer chosen labels than rejected ones lead to overestimated rewards during training, while those with shorter chosen labels result in underestimated rewards. Therefore, overestimated re-

wards contribute more significantly to gradient optimization, ultimately exacerbating verbosity. We believe this algorithmic dependence on response length is a unique drawback of DPO, since the explicit rewards in RLHF typically manifest as scalar values (Ouyang et al., 2022).

We propose that addressing this reliance on response length can be effectively achieved through a straightforward down-sampling method. Illustrated in the middle of Figure 1, this approach involves down-sampling equal token-level probability features for computing regularized KL divergences. Our contributions in this paper are threefold:

- We analyze the algorithmic dependence on response length in DPO, revealing how it results in overestimated or underestimated rewards. Through decomposition experiments using datasets with varying label length, we empirically demonstrate the biased rewards.
- We propose a lightweight approach, called SamPO, to mitigate the biased length reliance in DPO. By simply down-sampling equal probability features at the token-level, we can apply DPO with regularized KL divergences.
- We validate our method using three different LLMs of varying scales. Compared to DPO, SamPO significantly reduces verbosity. Leveraging debiased rewards, we achieve significant improvements across five conditioned and three open-ended benchmarks, as depicted in the lower section of Figure 1.

2 Related Work

Optimization from Human Preference aims to align neural models with human minds. As a seminal work, (Stiennon et al., 2020) collected human preferences on 123k pairs of summary outputs, then trained a reward model that guides the GPT-3 model (Brown et al., 2020) to produce more coherent and human-preferred summaries. (Ouyang et al., 2022) then further scaled similar pipeline with 1M diverse text instructions, and reported that outputs from the 1.3B parameter InstructGPT model were preferred to outputs from the 175B GPT-3 model, according to downstream human evaluation. RLHF has become an essential part of aligning LLMs (Touvron et al., 2023; Bi et al., 2024; Bai et al., 2023; Young et al., 2024). However, as it follows a multi-stage training strategy, and heavily relies on the quality of reward model, RLHF’s training cost and stability are widely criticized (Zheng et al., 2023; McKinney et al., 2023). Therefore, DPO came into being, providing a stable alternative that does not rely on an explicit reward model (Rafailov et al., 2023). It has been proved that DPO can achieve the same alignment effect as RLHF (Iverson et al., 2023; Hou et al., 2024).

Over-optimization in RL is a well-known obstacle (Skalse et al., 2022; Pan et al., 2023a; Casper et al., 2023; Zheng et al., 2023), which refers to the phenomenon that feedback scores from the reward model are getting higher, but the updated policy model produces lower quality responses. And one particularly noticeable low-quality feature is verbosity. It is general to blame for exploitation of reward model (Casper et al., 2023; Gao et al., 2023a), and thus various regularization approaches have been proposed, including uncertainty-based regularization (Coste et al., 2023; Zhai et al., 2023), composite reward models (Moskowitz et al., 2023), and length decorrelation (Chen et al., 2024b). However, since the reward model is eliminated in DPO, none of the above approaches can be directly applied. Herein, specific methods are introduced, (Park et al., 2024) introduced a pairwise length regularization term to dampen the verbosity trends, and SimPO (Meng et al., 2024) used average probability to eliminate length reliance.

In this paper, we present that the verbosity issue in DPO is further related to algorithmic biased length reliance, which is never analyzed in previous literature. And this drawback can be effectively handled via down-sampling over KL divergence.

3 SamPO: Down-Sampled DPO

In this section, we first give a brief introduction of DPO’s optimization target (§3.1), then dive into further analysis of its potential length reliance (§3.2). Subsequently, we present SamPO, which intuitively regularizes the biased length-specific reward (§3.3).

3.1 Preliminary Background of DPO

DPO implements direct RLHF based on offline preference data and an offloaded reward model. Specifically, DPO first re-parameterizes the reward model in multi-stage RLHF as follows:

$$r_\phi(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x) \quad (1)$$

where r_ϕ , π_θ and π_{ref} denote the reward model, the policy model, and the reference model, respectively. Both π_θ and π_{ref} are usually initialized from the same SFT model. While π_θ is subject to further optimization during DPO, π_{ref} is usually frozen. $Z(x)$ is the partition function, and β is a hyperparameter that adjusts the intensity of rewards. DPO incorporates the Bradley-Terry model to predict preferences:

$$P_\theta(y_w \succ y_l | x) = \frac{\exp(r_\phi(x, y_w))}{\exp(r_\phi(x, y_w)) + \exp(r_\phi(x, y_l))} \quad (2)$$

where a preference triplet (x, y_w, y_l) consists of a prompt instruction x , a chosen response y_w , and a less preferred response y_l . According to the Bradley-Terry model, the preference probability P_θ can be estimated via pairwise comparison. The loss function of DPO is defined as:

$$\mathcal{L}_{dpo}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\log \sigma(\Delta)] \quad (3)$$

where:

$$\Delta = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} \quad (4)$$

In this context, σ stands for sigmoid function, and D denotes the entire pairwise preference dataset. The implicit reward Δ in Eq. 4 is formulated as the discrepancy between the chosen KL divergence $\log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}$ and the rejected KL divergence $\log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}$. Each KL divergence is calculated based on the tokens in the response y . Considering Eq. 3, DPO’s gradients can be written as:

$$\nabla_\theta \mathcal{L}_{dpo}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\beta \sigma(-\Delta) \mathcal{M}] \quad (5)$$

$$\mathcal{M} = \nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x) \quad (6)$$

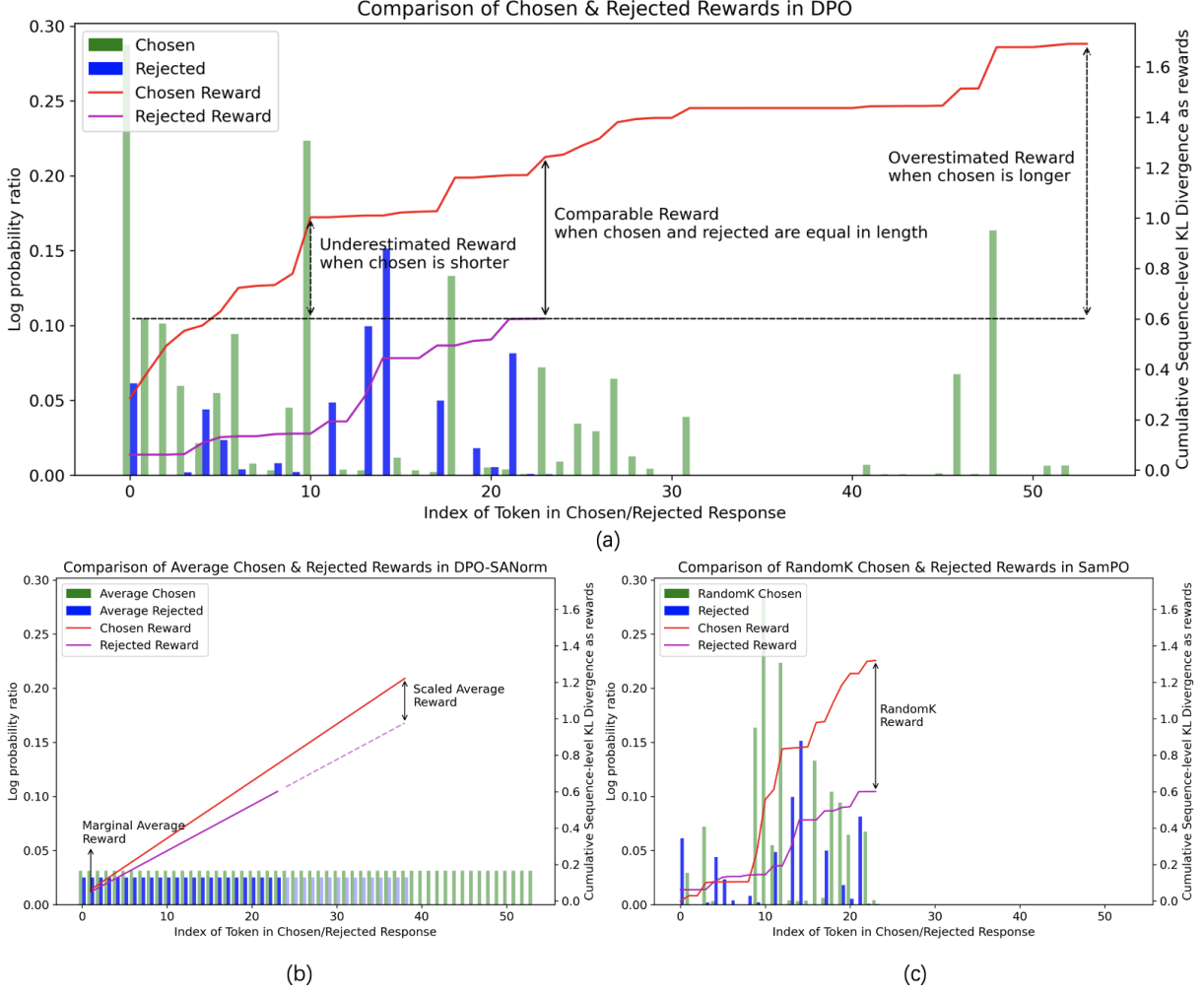


Figure 2: The disparity in pairwise responses, illustrated by typical examples, forces DPO to overestimate or underestimate the actual rewards. In the upper sub-figure (a), we present DPO’s chosen reward $\sum \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}$ and rejected reward $\sum \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}$ with red and purple curves, respectively. The reward for each response is calculated as the sequence-level KL divergence, which is derived from the token-level log probability ratios (illustrated by green and blue bars). Therefore, the difference between these two curves illustrates the implicit reward target in DPO, as shown in Eq. 7. Averaged and normalized DPO results are displayed in the lower-left sub-figure (b), while our SamPO is illustrated in lower-right sub-figure (c).

where \mathcal{M} is a discrepancy term that leads the policy model π_θ to increase the likelihood of the chosen response y_w and decrease the likelihood of the rejected response y_l . The term Δ acts as a scaling factor for the intensity of \mathcal{M} .

3.2 Biased Length Reliance in DPO

DPO’s loss and gradient are computed at the sequence-level. When calculating the KL term $\log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$, DPO treats the probabilities of individual tokens as discrete samples. We can express Eq. 4 at the token-level (Proof is in Appendix A):

$$\Delta = \beta \sum_{t=1}^{T_w} \log \frac{\pi_\theta(y_w^t|x)}{\pi_{ref}(y_w^t|x)} - \beta \sum_{t=1}^{T_l} \log \frac{\pi_\theta(y_l^t|x)}{\pi_{ref}(y_l^t|x)} \quad (7)$$

where T_w and T_l denote the number of tokens from the first to the t -th positions in the chosen response y_w and the rejected response y_l , respectively. Similarly, we rewrite Eq. 6 as:

$$\mathcal{M} = \nabla_\theta \sum_{t=1}^{T_w} \log \pi(y_w^t|x) - \nabla_\theta \sum_{t=1}^{T_l} \log \pi(y_l^t|x) \quad (8)$$

From this, we can intuitively understand how the difference in length between the chosen response y_w and the rejected response y_l affects the loss and the gradient. As illustrated in sub-Figure 2(a), a “comparable reward” is achieved if y_w and y_l have the same length, allowing DPO to effectively learn the quality difference. However, if y_w is much longer than y_l , the larger number of tokens in y_w

may result in an “*overestimated reward*” in Eq. 7, contributing disproportionately to the gradient updates described in Eq. 5 and 8. Conversely, if \mathbf{y}_w is shorter than \mathbf{y}_l , DPO could “*underestimate reward*” and incorporate fewer gradients, even if \mathbf{y}_w is of better quality. This bias towards length means that DPO tends to favor longer, seemingly acceptable responses over shorter, well-formed ones during training, potentially leading to verbose outputs.

3.3 Debiased KL Divergence

In the following content, we explore two common strategies to mitigate the dependence on sequence length: averaging and sampling.

Averaging modifies the sequence-level KL divergence to use a marginally averaged reward, which serves as a basic form of length regularization. This adjustment modifies Eq. 7 as follows:

$$\Delta = \beta \frac{\sum_{t=1}^{T_w} \log \frac{\pi_{\theta}(y_w^t|x)}{\pi_{ref}(y_w^t|x)}}{|T_w|} - \beta \frac{\sum_{t=1}^{|T_l|} \log \frac{\pi_{\theta}(y_l^t|x)}{\pi_{ref}(y_l^t|x)}}{|T_l|} \quad (9)$$

The averaging process can help remove the influence of length. However, as shown in the left corner of Figure 2(b), there lies a scale difference between the marginally averaged reward and the original sequence-level reward. To address this, we scale the marginal reward with a dynamic scaling factor $\frac{(T_w+T_l)}{2}$, which is the average length of the chosen response \mathbf{y}_w and the rejected response \mathbf{y}_l .

Sampling involves selecting the same amount of tokens from both the chosen and the rejected responses, and then calculating the down-sampled sequence-level KL divergence for the implicit reward. This modifies Eq. 7 to:

$$\Delta = \beta \sum_{t=1}^{T_m} \log \frac{\pi_{\theta}(y_w^t|x)}{\pi_{ref}(y_w^t|x)} - \beta \sum_{t=1}^{T_m} \log \frac{\pi_{\theta}(y_l^t|x)}{\pi_{ref}(y_l^t|x)} \quad (10)$$

$$T_m = \min(T_w, T_l), y^t \sim \text{Uniform}(T_m, \{y\}^T)$$

where T_m is equal to the minimum token length of (T_w, T_l) , and \mathbf{y}^t is down-sampled from all tokens $\{\mathbf{y}^T\}$ uniformly. Eq. 10 is consistent with the corresponding reward term shown in the middle of Figure 1. In addition, we discuss the impact of sampling randomness in Appendix E.

Figure 2(b) and (c) demonstrate that both averaging and sampling can produce length-debiased rewards that are comparably effective. However, simple averaging diminishes the variance feature among tokens. Consequently, we opt for the down-sampling strategy in our proposed SamPO method. This decision is validated in Section 5.

4 Experimental Setup

In this section, we start by introducing our datasets (§ 4.1, § 4.2), followed by the baselines (§ 4.3, § 4.4), and then provide an overview of our experimental design (§ 4.5).

4.1 Training Datasets

We leverage three independent preference datasets for training. Two of these are consistent with the original DPO (Rafailov et al., 2023): the 161k HH-RLHF data (Ganguli et al., 2022), and the 92.8k TL;DR data (Völske et al., 2017). Additionally, we include the 61k binarized UltraFeedback data (Cui et al., 2023) that has been utilized in subsequent works (Iverson et al., 2023; Meng et al., 2024) following DPO. Each of these datasets comes with an evaluation set for cross-validation during training.

4.2 Evaluation Benchmarks

Following DPO, for models trained on HH-RLHF or TL;DR, we randomly select 256 samples from their respective evaluation sets for final testing. We report the win rate between the response generated by the fine-tuned policy model $\hat{y}_{\theta} = \pi_{\theta}(\mathbf{x}_{test})$ and the response from the baseline SFT model $\hat{y}_{ref} = \pi_{ref}(\mathbf{x}_{test})$, judged by GPT-4 (Achiam et al., 2023). For models trained with UltraFeedback, we use five conditional and one open-ended generation benchmarks. The conditional benchmarks, along with their in-context examples, are: GSM8K in 8-shot (Cobbe et al., 2021), IFEval in 3-shot (Zhou et al., 2023), PiQA in 3-shot (Bisk et al., 2020), MMLU in 0-shot (Hendrycks et al., 2021), and TruthfulQA in 3-shot (Lin et al., 2022). The open-ended benchmark is AlpacaEval2 (Li et al., 2023). We report match accuracy for the conditional benchmarks, and the length-debiased GPT-4 win rate for AlpacaEval2 (Dubois et al., 2024a). For additional details, refer to Appendix B.

4.3 Foundation Models

In our experiments, we include LLMs of three different sizes: Pythia-2.8B (Biderman et al., 2023), Llama3-8B-Instruct (AI@Meta, 2024), and Tulu2-13B-SFT (Iverson et al., 2023). Details of these LLMs, including their hyperparameters and associated costs, are provided in Appendix C.

4.4 Baselines

Several variants of DPO have been proposed, which can be categorized into three main types: (1) **Reduce cost**. Although DPO is robust, the preparation

of high-quality pair-wise preference labels and the requirement to run with two large models make DPO costly. To address this, KTO (Ethayarajh et al., 2024) proposed to use non-pairwise preference data. ORPO (Hong et al., 2024), CPO (Xu et al., 2024), and SimPO (Meng et al., 2024) introduced reference-free losses that allow optimization with a single policy model; (2) **Alleviate overfitting**. IPO (Azar et al., 2023) analyzed the risk of overfitting, and introduced a square loss to reshape the monotonic DPO loss. TDPO (Zeng et al., 2024) incorporated forward KL divergence constraints for each token, improving alignment and diversity. BCO (Jung et al., 2024) and NCA (Chen et al., 2024a) offered strategies to reduce noise from pairwise preference responses; (3) **Overcome verbosity**. Park et al. (2024) introduced a pairwise length regularization term to counter verbosity. SimPO (Meng et al., 2024) used average probability to eliminate dependency on sequence length.

We select methods that focus on noise removal or length normalization, and have shown relatively positive testing results as our final baselines: Hybrid DPO+SFT, TDPO (Zeng et al., 2024), Length-normed DPO (Park et al., 2024), BCO (Jung et al., 2024), SimPO (Meng et al., 2024). Particularly, Hybrid DPO+SFT refers to the multi-task learning pipeline where DPO is applied to pairwise responses and SFT is applied to the chosen response at the same time, which is a common practice (Hua et al., 2024; Lu et al., 2024).

4.5 Experimental Designs

In general, we design three groups of experiments:

- (1) **Presence of biased length reliance.** We extract two 27k subsets from the UltraFeedback only by response length. One is named UltraFeedback-long, in which the chosen response of each data must be longer than the rejected response. The other one is named UltraFeedback-short, and as the name suggests, it contains a shorter chosen response. We use these subsets for biased reward exhibitions.
- (2) **Preliminary Study of DPO and variants.** Given that there are many variants of DPO, and they often use their own hyperparameters, we first conduct a preliminary study to align their performance under the same conditions. This study helps us select several robust baselines. The results are reported in Appendix D.

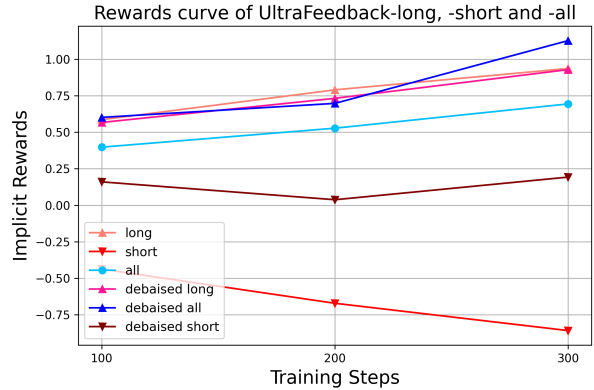


Figure 3: Trends of DPO’s implicit reward (Eq. 7), when fine-tuned with UltraFeedback-long, -short and -all sets. Three debiased rewards are produced by our SamPO.

	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.
long	41.24	37.89	81.28	55.86	38.68	50.99
short	34.50	6.00	77.09	54.87	30.48	40.59
all	42.61	43.76	81.77	55.85	35.86	51.97
long*	42.61	38.01	81.18	55.86	36.11	50.75
short*	41.70	33.93	81.18	55.5	36.35	49.73
all*	42.68	44.12	81.28	55.8	40.15	52.81

Table 1: Performance of models in Figure 3. The * mark stands for the SamPO’s debiased rewards.

- (3) **Experiments with various LLMs.** Similar to DPO, we use Pythia-2.8B to train and test SamPO on HH-RLHF or TL;DR; on the other hand, following relevant studies (Iverson et al., 2023; Hong et al., 2024), we use Tulu2-13B-SFT and Llama3-8B-Instruct to train on Ultrafeedback and verify SamPO on public benchmarks. Also, literature reports that iteratively updates the frozen reference model π_{ref} can obtain further gains (Gorbatovski et al., 2024; Zhang et al., 2024). Thus, we combine it with SamPO to present Iterative SamPO.

5 Experimental Results

In this section, following the above designs, we first report the group experiments of length reliance (§ 5.1), then present comparison studies against strong baselines (§ 5.2). We discuss quantitative results in the main body. We leave more ablation studies and case analysis in Appendix E, F, and H.

5.1 Group study of length reliance

Figure 3 illustrates the trends of DPO’s implicit reward on the same test set when we fine-tune the same Tulu2-13B-SFT model with different subsets of UltraFeedback. We report testing performance

	Tulu2-13B-SFT								
Methods	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.	Alpaca2	LC Alpaca2	Len./Token
Tulu2-13B-SFT (Iverson et al., 2023)	40.56	37.17	81.39	55.53	33.78	49.69	5.09	9.99	262
Tulu2-13B-DPO (Iverson et al., 2023)	42.99	42.45	81.28	56.07	41.86	52.93	11.45	13.7	382
DPO (Rafailov et al., 2023)	43.44	43.17	81.66	56.08	39.66	52.80	10.66	15.02	372
Iterative DPO	42.08	44.96	81.39	56.02	40.15	52.92	12.17	14.24	400
Hybrid DPO+SFT	41.85	44.36	81.28	56.15	40.02	52.73	7.66	13.45	308
TDPO (Zeng et al., 2024)	41.39	<u>41.25</u>	81.34	55.78	<u>36.11</u>	51.17	6.86	11.45	290
Length-normed DPO (Park et al., 2024)	40.71	45.8	80.85	55.85	39.66	52.57	7.47	13.40	250
BCO (Jung et al., 2024)	42.68	43.73	81.45	56.41	39.66	52.79	9.07	13.29	316
SimPO (Meng et al., 2024)	<u>29.57</u>	47.24	81.39	56.10	38.31	<u>50.52</u>	<u>5.21</u>	<u>7.84</u>	336
SamPO (ours)	41.55	45.32	80.85	55.88	41.37	52.99	11.77	17.6	339
Iterative SamPO (ours)	42.08	46.28	81.07	56.12	41.25	53.36	14.58	17.52	347
DPO-SANorm (ours)	42.15	44.36	81.07	56.00	38.43	52.40	9.21	14.53	283
	Llama3-8B-Instruct								
Methods	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.	Alpaca2	LC Alpaca2	Len./Token
Llama3-8B-Instruct (AI@Meta, 2024)	75.06	49.40	80.69	63.85	36.47	61.09	22.57	22.92	421
DPO (Rafailov et al., 2023)	75.59	51.80	81.94	64.06	40.39	62.76	23.34	23.20	422
Iterative DPO	74.91	52.52	81.66	64.02	39.90	62.60	23.92	25.50	403
Hybrid DPO+SFT	75.59	65.83	81.34	63.54	39.78	65.22	20.17	20.62	380
TDPO (Zeng et al., 2024)	75.36	51.32	81.23	63.54	<u>38.07</u>	61.90	23.66	24.57	408
Length-normed DPO (Park et al., 2024)	76.12	46.76	81.39	64.09	40.76	61.82	24.04	27.44	377
BCO (Jung et al., 2024)	76.19	50.60	81.66	63.99	39.90	62.47	24.72	24.81	421
SimPO (Meng et al., 2024)	75.06	60.43	81.83	63.43	39.53	64.06	26.82	31.29	375
Llama3-8B-Ins.-SimPO (Meng et al., 2024)	<u>72.93</u>	<u>46.28</u>	<u>78.51</u>	61.99	42.96	<u>60.53</u>	39.72	43.42	387
SamPO (ours)	76.56	57.03	81.72	64.00	41.06	64.18	28.97	32.01	375
Iterative SamPO (ours)	77.81	60.55	81.18	64.12	44.07	65.55	30.68	35.14	377

Table 2: Qualitative results of fine-tuning two LLMs with DPO, several variants and our SamPO. We use the same UltraFeedback dataset and keep almost all hyperparameters the same for each LLM group. Specifically, Tulu2-13B-SFT and -DPO, Llama3-8B-Instruct and -Ins.-SimPO are open-source checkpoints. We evaluate all models, including those public models, under the same framework. We **bold** the best results and underline the unusually poor results.

in Table 1. It is clear that data from the same distribution leads to different training and testing performances due to the difference in response length.

The “-all” set refers to training with original UltraFeedback, which mix “-long” and “-short” data. The “-long” subset provides overestimated rewards and therefore causes performance degradation. However, since statistically, the chosen response is longer than the rejected response (Park et al., 2024), the training trend of the “-long” subset is similar to the “-all” full set. On the contrary, the “-short” subset completely erases the distinctive feature of length, hoping that the model will perform comparative learning based on content quality. However, the biased DPO completely underestimate the reward, thus causing collapses.

Yet, our SamPO presents debiased rewards. We can observe debiased positive rewards on the “-short” set. And the debiased rewards of “-all” set grow to a high peak at 300 steps. Such debiased rewards result in significant U-turn reversal and further improvements. As shown in Table 1, SamPO manages to eliminate collapse on the “-short” set,

where we record a normal average benchmark score similar to the “-long” set, improving the score by 9.2%. Thanks to the regularization of those “-short” data, the “-all” set that mixes both “-long” and “-short” data achieves the best score up to 52.81 on average.

5.2 Comparison study against other methods

5.2.1 Study on UltraFeedback

For LLMs that fine-tuned with UltraFeedback, we evaluate their downstream performance in Table 2.

Overall enhancement by SamPO. For Tulu2-13B-SFT, our replicated DPO shows benchmark accuracy and response length on AlpacaEval2 data comparable to the open-source version. Compared to the SFT baseline, DPO improves performance across all test data but increases response length by 40-45%. Iterative DPO exacerbates this verbosity issue. However, all chosen baselines and our SamPOs produce shorter responses, mitigating verbosity. However, TDPO and SimPO show significant drops in conditional benchmarks, such as over 10% on GSM8K and over 3% on TruthfulQA,

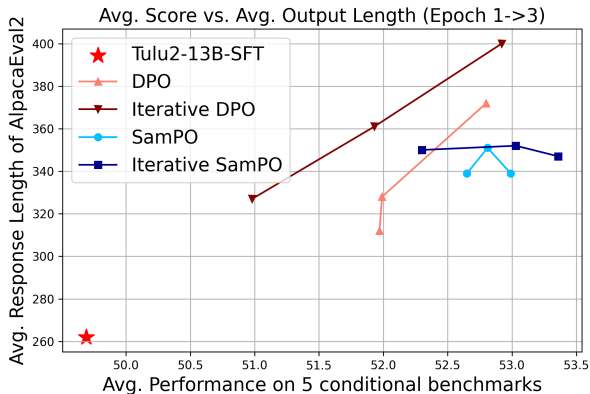


Figure 4: We show how the policy model’s response length changes on AlpacaEval2 as the test performance improves over 3 epochs of training. The epoch number increases from left to right along the curve.

compared to DPO. Notably, our SamPOs achieve overall improvements on both conditional benchmarks (+0.5%) and open-ended generation for AlpacaEval2 prompts (+4%). Also, the averaging version DPO-SANorm, mentioned in section 3.3, confirms that the sampling strategy is more valid.

For Llama3-8B-Instruct, we observe superior length stability. Even when fine-tuned with the original DPO, the model maintains its initial response length, likely due to its comprehensive training process involving SFT, RLHF, and DPO (AI@Meta, 2024). Marginal improvements are observed over its DPO version, with average gains of 1.7% on five conditional benchmarks and <1% on AlpacaEval2. Among all methods, only hybrid DPO+SFT, SimPO, and our SamPOs show significant improvements over DPO, with average gains of 1.3% to 3% on five accuracy benchmarks. Specifically, hybrid DPO+SFT excels in IFEval (65.83), and our SamPOs notably improve GSM8K (+2.3%) and TruthfulQA (+3.7%). As for GPT-4 judged AlpacaEval2, hybrid training loses about 3% performance, while our SamPO achieves the best performance in both raw and length-debiased scores among all locally fine-tuned LLMs, outperforming DPO up to 12%.

Discussions of SimPO. The SimPO method has an obvious “seesaw” dilemma. The open-source SimPO checkpoint achieves the best performance of AlpacaEval2 at the expense of a significant sacrifice on other benchmarks. We avoid this in the reproduction and obtain a more balanced version. Also, the public release was trained with *boosted data*² instead of the naive UltraFeedback.

²SimPO’s augmented dataset: <https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback>

	HH-RLHF		TL;DR	
	Wins	Len.	Wins	Len.
DPO (Rafailov et al., 2023)	74.49	250.07	60.98	53.80
Iterative DPO	53.46	253.99	73.58	66.65
Hybrid DPO+SFT	86.12	41.29	45.68	41.43
TDPO (Zeng et al., 2024)	52.53	246.28	47.76	45.60
Len.-Norm (Park et al., 2024)	68.95	246.28	58.13	47.34
BCO (Jung et al., 2024)	65.85	218.05	50.62	42.93
SimPO (Meng et al., 2024)	78.91	<u>14.77</u>	<u>33.33</u>	<u>31.90</u>
SamPO (ours)	82.8	112.95	65.71	69.52
Iterative SamPO (ours)	79.05	137.55	73.58	49.54

Table 3: Win Rate (%) and Avg. Output Length across methods. We **bold** the best and underline the outliers.

Length stability of SamPO. Based on Figure 4, we find that DPO makes the model increasingly prefer to generate longer responses in 3-epoch training, and Iterative DPO further strengthens this trend. In contrast, SamPO and Iterative SamPO achieve higher testing scores and stabilise the length.

5.2.2 Study on HH-RLHF & TL;DR

As for HH-RLHF and TL;DR, we utilize Pythia-2.8B for all experiments. Since Pythia has not been specifically trained for instructional tasks, we initiate our process with one epoch of SFT on the chosen response, following DPO’s setup. Subsequently, we conduct preference optimization using SamPO alongside various baseline methods. Following previous literature (Rafailov et al., 2023; Park et al., 2024), GPT-4 served as the proxy for human preference. We report the win rate against the SFT basis and the average generated token length of all methods in Table 3.

SamPO has a good effect on HH-RLHF. SamPO improves performance across all HH-RLHF test data, achieving the second-best win rate while maintaining a lower yet reasonable response length. Iterative SamPO shows slightly lower win rates due to less control over response length. Baselines such as Iterative DPO and TDPO achieve win rates close to 50%, indicating minimal improvement over the SFT model. Hybrid DPO+SFT stands out as a strong baseline, addressing the under-generalization issue and attaining an 86.12% win rate with the shortest average response lengths among all experiments. SimPO, while achieving a similar win rate of 78.91% as Iterative SamPO, but produces incredibly low response length.

SamPO achieves the best performance on TL;DR. In terms of TL;DR, SamPO and Iterative SamPO show the highest win rates, with 65.71%

and 73.58%, respectively, significantly outperforming all other methods. DPO and Length-normed DPO also perform well, achieving win rates of 60.98% and 58.13%, respectively. Iterative DPO reaches the best while using longer answers than Iterative SamPO. In contrast, SimPO has the lowest win rate at 33.33%, indicating that it is less effective on the TL;DR dataset.

Over-simplification by SimPO. In fact, on HH-RLHF, we notice many of the outputs from SimPO are overly simplified, often omitting necessary content and resulting in only 14.77 lengths of tokens on average. For example, a preferred response from HH-RLHF is “*I’ll give you the links.*”, whereas the SimPO response is simply “*Sure!*”. This suggests that while concise, the responses lack the necessary informativeness. In this scenario, we can see GPT-4 prefers over-simplified responses, which is probably due to the binary setup of preference choice. Similarly, on TL;DR, SimPO produces the shortest responses (average 31.90 tokens). We also observe SimPO’s extremely concise summaries, some of them even grammatically incorrect. For example, a preferred summary from the TL;DR is “*I [20M] met a great girl [16F] online who lives in the same city. Problems are: she’s moving away, I want to meet her, and the obvious age gap.*”, while SimPO outputs a shorter summary without a subject and capitalizes the first letter: “*online flirt turns into legit relationship. Great chemistry. Age gap and distance issues. Need advice before final meetup before long trip abroad.*”.

5.2.3 Human Evaluation of SamPO

In addition to the aforementioned automated evaluation, we further conduct a large-scale human evaluation to study the effectiveness of the SamPO algorithm when applied to super large LLM (e.g., over 50B). We use an LLM fine-tuned based on Qwen1.5-72B (Bai et al., 2023) as a starting point and fine-tune it for one epoch using the proposed SamPO method. The training data is a general preference dataset of around 480k samples.

We report the results of the human evaluation in Table 4, covering the three most popular scenarios: general Machine Reading Comprehension (MRC), logical reasoning (e.g., math or logic questions), and open domain dialogues in role-play settings. We have hired a 30-person annotation team, each of whom has at least a bachelor’s degree or above. Each test scenario contains 500 to 1k carefully crafted challenging instances, which are then

	MRC	Logical Reasoning	RolePlay	Avg.
SFT Base	81.25	69.52	59.12	69.96
w/ DPO	85.33	73.25	57.41	72.00
w/ SamPO	87.50	83.57	63.61	78.23

Table 4: Human Evaluation results of a Qwen1.5-72B-based SFT model and its two further fine-tuned versions, applying with DPO and SamPO respectively.

cross-labeled by multiple professional annotators. Our scoring criteria are relatively simple, distinguishing only between incorrect and acceptable responses. We observe that SamPO significantly outperforms both the SFT Base and DPO method on all tasks.

6 Conclusion

In this paper, we identify and address the verbosity issue in DPO related to biased length reliance. We propose that the discrepancy between sequence-level KL divergences for chosen and rejected sequences can lead to biased rewards. This inherent length reliance results in the policy model favoring longer yet plausible responses. Thus, we propose SamPO, an approach that regularizes the KL divergence by down-sampling equal token-level features. Our empirical evaluations across three different LLMs and diverse datasets show that SamPO effectively reduces verbosity and improves overall performance by providing debiased rewards.

Acknowledgment

We thank Shiyue Xu for correcting the error in Equation 5 in the previous draft³. This work was supported in part by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2) and Innovate UK through its Accelerating Trustworthy AI Collaborative R&D funding (grant no. 10093055).

Limitations

While our proposed method, SamPO, has shown promising results in mitigating verbosity and improving performance, several limitations remain:

- **Scalability.** Although we tested SamPO on different LLMs, including one super large LLM (Qwen1.5-72B-Instruct). We agree that

³<https://github.com/LuJunru/SamPO/issues/1>

further experiments are needed to confirm its scalability and generalization across a broader range of models with different scales.

- **Computational Overhead.** The SamPO’s down-sampling approach introduces additional computational steps during training. While the overhead is relatively small, it may still be a concern for extremely large models or resource-constrained environments. Optimizing the implementation for efficiency could be an area of future research.
- **Human Evaluation.** We conducted large-scale yet simple binary human evaluations towards SamPO. Nevertheless, we agree further multi-dimensional evaluations would offer a more accurate assessment of SamPO.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- AI@Meta. 2024. [Llama 3 model card](#).
- 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. *arXiv preprint arXiv:2310.12036*.
- Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.
- Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qishi Du, Zhe Fu, et al. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.
- Ralph A. Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.
- Huayu Chen, Guande He, Hang Su, and Jun Zhu. 2024a. Noise contrastive alignment of language models with explicit rewards. *arXiv preprint arXiv:2402.05369*.
- Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. 2024b. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. 2023. Reward model ensembles help mitigate overoptimization. *arXiv preprint arXiv:2310.02743*.
- Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*.
- Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024a. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.
- Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2024b. AlpacaFarm: A simulation framework for methods

- that learn from human feedback. *Advances in Neural Information Processing Systems*, 36.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.
- Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.
- Leo Gao, John Schulman, and Jacob Hilton. 2023a. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, et al. 2023b. [A framework for few-shot language model evaluation](#).
- Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. 2024. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*.
- Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv:1706.02677*.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. [Measuring massive multitask language understanding](#). In *International Conference on Learning Representations*.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. Reference-free monolithic preference optimization with odds ratio. *arXiv preprint arXiv:2403.07691*.
- Zhenyu Hou, Yiin Niu, Zhengxiao Du, Xiaohan Zhang, Xiao Liu, Aohan Zeng, Qinkai Zheng, Minlie Huang, Hongning Wang, Jie Tang, et al. 2024. Chatglm-rlhf: Practices of aligning large language models with human feedback. *arXiv preprint arXiv:2404.00934*.
- Ermo Hua, Biqing Qi, Kaiyan Zhang, Yue Yu, Ning Ding, Xingtai Lv, Kai Tian, and Bowen Zhou. 2024. Intuitive fine-tuning: Towards unifying sft and rlhf into a single process. *arXiv preprint arXiv:2405.11870*.
- Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. 2023. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*.
- Seungjae Jung, Gunsoo Han, Daniel Wontae Nam, and Kyoung-Woon On. 2024. Binary classifier optimization for large language model alignment. *arXiv preprint arXiv:2404.04656*.
- Samia Kabir, David N Udo-Imeh, Bonan Kou, and Tianyi Zhang. 2023. Who answers it better? an in-depth analysis of chatgpt and stack overflow answers to software engineering questions. *arXiv preprint arXiv:2308.02312*.
- Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. AlpacaEval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval.
- Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. [TruthfulQA: Measuring how models mimic human falsehoods](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.
- Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.
- Keming Lu, Bowen Yu, Fei Huang, Yang Fan, Runji Lin, and Chang Zhou. 2024. Online merging optimizers for boosting rewards and mitigating tax in alignment. *arXiv preprint arXiv:2405.17931*.
- Lev McKinney, Yawen Duan, David Krueger, and Adam Gleave. 2023. On the fragility of learned reward functions. *arXiv preprint arXiv:2301.03652*.
- Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*.
- Ted Moskowitz, Aaditya K Singh, DJ Strouse, Tuomas Sandholm, Ruslan Salakhutdinov, Anca D Dragan, and Stephen McAleer. 2023. Confronting reward model overoptimization with constrained rlhf. *arXiv preprint arXiv:2310.04373*.
- 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. In *Thirty-Sixth Conference on Neural Information Processing Systems*.
- Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddhartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

- Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. 2023a. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International Conference on Machine Learning*. PMLR.
- Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023b. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.
- Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.
- Rafael Rafailov, Yaswanth Chittooru, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024. [Scaling laws for reward model overoptimization in direct alignment algorithms](#). *Preprint*, arXiv:2406.02900.
- 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. *arXiv preprint arXiv:2305.18290*.
- Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. 2022. Is reinforcement learning (not) for natural language processing: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*.
- Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, et al. 2021. {ZeRO-Offload}: Democratizing {Billion-Scale} model training. In *2021 USENIX Annual Technical Conference*.
- Wei Shen, Rui Zheng, Wenyu Zhan, Jun Zhao, Shihan Dou, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Loose lips sink ships: Mitigating length bias in reinforcement learning from human feedback. *arXiv preprint arXiv:2310.05199*.
- Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in rlhf. *arXiv preprint arXiv:2310.03716*.
- Joar Skalse, Nikolaus Howe, Dmitrii Krashenninikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, et al. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. [TL;DR: Mining Reddit to learn automatic summarization](#). In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63, Copenhagen, Denmark. Association for Computational Linguistics.
- Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. 2024. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*.
- Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. 2024. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*.
- 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*.
- Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv:2304.13712*.
- Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.
- Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. 2024. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*.
- Yuanzhao Zhai, Han Zhang, Yu Lei, Yue Yu, Kele Xu, Dawei Feng, Bo Ding, and Huaimin Wang. 2023. Uncertainty-penalized reinforcement learning from human feedback with diverse reward lora ensembles. *arXiv preprint arXiv:2401.00243*.
- Ge Zhang, Scott Qu, Jiaheng Liu, Chenchen Zhang, Chenghua Lin, Chou Leuang Yu, Danny Pan, Esther Cheng, Jie Liu, Qunshu Lin, et al. 2024. Map-neo: Highly capable and transparent bilingual large language model series. *arXiv preprint arXiv:2405.19327*.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023a. A survey of large language models. *arXiv:2303.18223*.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. 2023b. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. 2023. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A Derivation of Equations

A.1 Token-level DPO reward

Given the DPO’s implicit reward Δ in Eq. 4:

$$\Delta = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}$$

and we know when given a prompt x , the probability of a response y from a LLM π is:

$$\pi(y|x) = \prod_{t=1}^T \pi(y_t|y_{<t}, x)$$

where T represents the length of token sequence of y , $y_{<t}$ denotes all the tokens before the t -th index in y , and y_t is the t -th generated token. Thus, when convert DPO’s sequence-level implicit reward Δ to a token-level expression, we can write:

$$\begin{aligned} \Delta &= \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)} \\ &= \beta \log \frac{\prod_{t=1}^{T_w} \pi_\theta(y_{w,t}|y_{w,<t}, x)}{\prod_{t=1}^{T_w} \pi_{ref}(y_{w,t}|y_{w,<t}, x)} \\ &\quad - \beta \log \frac{\prod_{t=1}^{T_l} \pi_\theta(y_{l,t}|y_{l,<t}, x)}{\prod_{t=1}^{T_l} \pi_{ref}(y_{l,t}|y_{l,<t}, x)} \\ &= \beta \sum_{t=1}^{T_w} \log \frac{\pi_\theta(y_{w,t}|y_{w,<t}, x)}{\pi_{ref}(y_{w,t}|y_{w,<t}, x)} - \beta \sum_{t=1}^{T_l} \log \frac{\pi_\theta(y_{l,t}|y_{l,<t}, x)}{\pi_{ref}(y_{l,t}|y_{l,<t}, x)} \\ &= \beta \sum_{t=1}^{T_w} \log \frac{\pi_\theta(y_w^t|x)}{\pi_{ref}(y_w^t|x)} - \beta \sum_{t=1}^{T_l} \log \frac{\pi_\theta(y_l^t|x)}{\pi_{ref}(y_l^t|x)}, \text{ in short} \end{aligned}$$

For the down-sampling phase, we have:

$$\begin{aligned} \Delta &= \beta \log \frac{\prod_{t=1}^{T_m} \pi_\theta(y_{w,t}|y_{w,<t}, x)}{\prod_{t=1}^{T_m} \pi_{ref}(y_{w,t}|y_{w,<t}, x)} - \beta \log \frac{\prod_{t=1}^{T_m} \pi_\theta(y_{l,t}|y_{l,<t}, x)}{\prod_{t=1}^{T_m} \pi_{ref}(y_{l,t}|y_{l,<t}, x)} \\ &= \beta \sum_{t=1}^{T_m} \log \frac{\pi_\theta(y_w^t|x)}{\pi_{ref}(y_w^t|x)} - \beta \sum_{t=1}^{T_m} \log \frac{\pi_\theta(y_l^t|x)}{\pi_{ref}(y_l^t|x)}, \text{ in short} \\ &\quad \text{where } T_m = \min(T_w, T_l), y^t \sim \text{Uniform}(T_m, \{y\}^T) \end{aligned}$$

□

A.2 Gradients of Token-level DPO reward

Given the DPO’s gradients $\nabla_\theta \mathcal{L}_{dpo}(\pi_\theta; \pi_{ref})$ related to the Eq. 5 and 6:

$$\nabla_\theta \mathcal{L}_{dpo}(\pi_\theta; \pi_{ref}) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\beta \sigma(-\Delta) \mathcal{M}]$$

$$\mathcal{M} = \nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x)$$

we derive the token-level expression of \mathcal{M} :

$$\begin{aligned} \mathcal{M} &= \nabla_\theta \log \pi(y_w|x) - \nabla_\theta \log \pi(y_l|x) \\ &= \nabla_\theta \log \prod_{t=1}^{T_w} \pi(y_{w,t}|y_{w,<t}, x) \\ &\quad - \nabla_\theta \log \prod_{t=1}^{T_l} \pi(y_{l,t}|y_{l,<t}, x) \\ &= \nabla_\theta \sum_{t=1}^{T_w} \log \pi(y_w^t|x) - \nabla_\theta \sum_{t=1}^{T_w} \log \pi(y_l^t|x), \text{ in short} \end{aligned}$$

For the down-sampling phase, we have:

$$\begin{aligned} \mathcal{M} &= \nabla_\theta \log \prod_{t=1}^{T_m} \pi(y_{w,t}|y_{w,<t}, x) \\ &\quad - \nabla_\theta \log \prod_{t=1}^{T_m} \pi(y_{l,t}|y_{l,<t}, x) \\ &= \nabla_\theta \sum_{t=1}^{T_m} \log \pi(y_w^t|x) - \nabla_\theta \sum_{t=1}^{T_m} \log \pi(y_l^t|x), \text{ in short} \\ &\quad \text{where } T_m = \min(T_w, T_l), y^t \sim \text{Uniform}(T_m, \{y\}^T) \end{aligned}$$

□

Therefore, combined with length-normalized Δ introduced in section A.1. We have debiased gradients $\nabla_\theta \mathcal{L}_{dpo}(\pi_\theta; \pi_{ref})$ to be served in SamPO.

B Evaluation Details

We present the details of our evolution schema:

- **GSM8K**: A generative primary level math dataset of 1.3k questions (Cobbe et al., 2021). We use 8-shot in-context exemplars. We report strict exact match score.
- **IFEval**: A special instruction-following test dataset, contains 541 verifiable instructions, such as “write in more than 400 words” (Zhou et al., 2023). We use 3-shot prompt and report instruction-level strict accuracy.
- **PiQA**: A binary common physical knowledge dataset of 1.8k questions (Bisk et al., 2020). The number of in-context exemplars is three. We report accuracy score of PiQA.

- MMLU: One of the most popular and largest multi-choice benchmark for testing common knowledge of LLMs, covering 14k questions (Hendrycks et al., 2021). No in-context exemplars provided, and we present accuracy.
- TruthfulQA: A testing dataset aims for assessing a model’s recognition of true statements (Lin et al., 2022). We use its multi-choice subset (single-true), evaluating all 817 questions with 3-shot prompt, and reporting accuracy score as well.
- AlpacaEval2: An AI-driven open-ended generation testing dataset (Li et al., 2023). This dataset contains 805 diverse questions, and compares the win rate of model’s response against GPT-4’s response (Achiam et al., 2023). The winner judge is also the GPT-4. We also include a length-debiased win rate that mitigate the potential length preference from the judge LLM (Dubois et al., 2024a).
- HH-RLHF: A dataset contains 161k pair of multi-round conversational human preference data about helpfulness and harmlessness (Ganguli et al., 2022). We report each approaches’ win rate against the SFT basis.
- TL;DR: A summarization obtained based on Reddit conversations (Völske et al., 2017), contains 92.8k training data. We report win rate between every model and the basic SFT.

Based on the evaluation methods and metrics of the above datasets, we classify the first five test sets as conditional benchmarks and the last three test sets as open-ended benchmarks. “Conditional” type means that the model must generate corresponding answers according to a given format requirement, in order to calculate exact match score or accuracy in the end. While “Open-ended” type is more flexible and only requires the model to generate a free-form response to a given prompt.

For all conditional benchmarks, we use a stable and popular evaluation framework “lm-evaluation-harness” (Gao et al., 2023b)⁴. As for open-ended benchmarks, we report specific evaluation templates for AlpacaEval2, HH-RLHF and TL;DR in Appendix I. Particularly, we use the official tool

⁴Official tool page of lm-eval: <https://github.com/EleutherAI/lm-evaluation-harness>

	Pythia-2.8B	Llama3-8B	Tulu2-13B
GPUs	1	8	8
Batch	32	1	1
Accumulations	4	16	16
Epoch	1	3	3
Train Max Len	1,024	8,192	8,192
Lr	1e-6	4e-7	1e-6
Warmup Ratio	0.1	0.1	0.1
DPO Beta	0.5/0.05	0.1	0.1
Random Seed	42	42	42
Gen. TopP	/	0.95	0.95
Gen. Temperature	0.0	0.8	0.8
Gen. Max Len	256	1,024	1,024
Train (1 epoch/5W)	4h	8h	16h
Special Notes	SFT weight for Hybrid DPO+SFT = 1.0, Length-normed DPO Alpha = 0.01, TDPO Alpha = 0.5, SimPO Beta = 2.5, SimPO Lambda for Llama3-8B = 1.4, SimPO Lambda for others = 0.3, Epoch of SimPO on all models = 1, DPO Beta 0.5 for TL;DR, 0.05 for HH-RLHF		

Table 5: Hyperparameters and training cost.

to evaluate AlpacaEval2⁵. The version of GPT-4 evaluator is all set as: gpt-4-turbo.

C HyperParameters and Training Cost

We report hyperparameters and training cost in Table 5. Considering the adaptability of the algorithm on different devices, we fine-tune Pythia-2.8B⁶ with all involved methods on 1 A100 80G GPU, while fine-tune Llama3-8B-Instruct⁷ and Tulu2-13B-SFT⁸ on 8 X A100 40G GPUs. We obey licenses of all involved models. All baselines and our SamPO share a common DPO beta of Eq. 4, as all methods are variants of DPO. We set this beta value as 0.1, same as the original DPO work. Except that, since many variants include new hyperparameters, we set them accordingly. One particular exception is SimPO, for which small Beta 0.1 and 3 epochs will lead to performance collapse. As such, we have to follow its original quite large Beta value 2.5. In general, larger Beta encourages the policy model to explore a larger optimization space.

The optimizer is *AdamW* (Loshchilov and Hutter, 2019) and the scheduler is *WarmupDecayLR* (Goyal et al., 2017). Deepspeed (Ren et al., 2021) and Flash Attention2 (Dao et al., 2022) are used for

⁵https://github.com/tatsu-lab/alpaca_eval/

⁶<http://huggingface.co/EleutherAI/pythia-2.8b>

⁷<https://huggingface.co/meta-llama/>

Meta-Llama-3-8B-Instruct

⁸<https://huggingface.co/allenai/tulu-2-13b>

Methods	Tulu2-13B-SFT								
	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.	Alpaca2	LC Alpaca2	Len./Token
Tulu2-13B-SFT (Iverson et al., 2023)	40.56	37.17	81.39	55.53	33.78	49.69	5.09	9.99	262
Tulu2-13B-DPO (Iverson et al., 2023)	42.99	42.45	81.28	56.07	41.86	52.93	11.45	13.7	382
DPO (Rafailov et al., 2023)	43.44	43.17	81.66	56.08	39.66	52.80	10.66	15.02	372
Iterative DPO	42.08	44.96	81.39	56.02	40.15	52.92	12.17	14.24	400
Hybrid DPO+SFT	41.85	44.36	81.28	56.15	40.02	52.73	7.66	13.45	308
✗ IPO (Azar et al., 2023)	42.13	42.25	81.22	56.08	38.21	51.98	6.96	8.34	304
✗ KTO (Ethayarajh et al., 2024)	41.89	43.22	81.67	56.00	39.42	52.44	9.47	12.25	371
✗ SLiC (Zhao et al., 2023b)	42.48	42.99	81.75	55.96	39.24	52.48	11.02	13.41	388
TDPO (Zeng et al., 2024)	41.39	<u>41.25</u>	81.34	55.78	<u>36.11</u>	51.17	6.86	11.45	290
Length-normed DPO (Park et al., 2024)	40.71	45.8	80.85	55.85	39.66	52.57	7.47	13.40	250
✗ DPOP (Pal et al., 2024)	42.23	<u>41.37</u>	81.23	55.85	<u>35.37</u>	51.21	/	/	/
BCO (Jung et al., 2024)	42.68	43.73	81.45	56.41	39.66	52.79	9.07	13.29	316
✗ SPPO (Wu et al., 2024)	40.94	<u>39.33</u>	81.01	55.92	<u>34.52</u>	<u>50.34</u>	/	/	/
✗ NCA (Chen et al., 2024a)	43.52	41.37	81.39	56.24	<u>36.96</u>	51.9	9.17	10.49	299
SimPO (Meng et al., 2024)	<u>29.57</u>	47.24	81.39	56.10	38.31	<u>50.52</u>	<u>5.21</u>	<u>7.84</u>	336
SamPO (ours)	41.55	45.32	80.85	55.88	41.37	52.99	11.77	17.6	339
Iterative SamPO (ours)	42.08	46.28	81.07	56.12	41.25	53.36	14.58	17.52	347
DPO-SANorm (ours)	42.15	44.36	81.07	56.00	38.43	52.40	9.21	14.53	283
SamPO-TopK (ours)	42.3	42.21	81.18	55.91	39.66	52.25	10.65	14.34	341

Table 6: Our preliminary and ablation studies. We **bold** the best results and underline the unusual poor results.

Methods	Llama3-8B-Instruct (3 Epochs)								
	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.	Alpaca2	LC Alpaca2	Len./Token
Llama3-8B-Instruct (AI@Meta, 2024)	75.06	49.40	80.69	63.85	36.47	61.09	22.57	22.92	421
DPO (Rafailov et al., 2023)	75.59	51.80	81.94	64.06	40.39	62.76	23.34	23.20	422
Iterative SamPO Seed 42 (ours)	77.81	60.55	81.18	64.12	44.07	65.55	30.68	35.14	377
Iterative SamPO Seed 123 (ours)	78.01	60.67	81.56	64.04	44.55	65.77	29.70	34.41	372
Iterative SamPO Seed 2024 (ours)	77.56	60.26	81.50	63.94	44.58	65.57	29.97	34.01	378
Methods	Llama3-8B-Instruct (1 Epoch)								
Methods	GSM8K	IFEval	PiQA	MMLU	TruthfulQA	Avg.	Alpaca2	LC Alpaca2	Len./Token
SamPO w/ Beta 0.01 (ours)	76.42	45.56	81.28	63.52	41.37	61.63	24.81	33.12	317
SamPO w/ Beta 0.05 (ours)	77.79	47.36	81.66	63.71	39.05	61.91	27.55	29.99	396
SamPO w/ Beta 0.1 (ours)	76.88	48.20	81.50	63.94	39.17	61.94	27.88	29.06	420
SamPO w/ Beta 0.3 (ours)	76.35	47.12	81.01	63.77	37.70	61.19	28.22	28.46	422
SamPO w/ Beta 0.5 (ours)	77.03	47.72	80.90	63.84	37.58	61.41	26.71	26.71	424

Table 7: Further ablation studies of sampling seeds, using Llama3-8B-Instruct. We **bold** the best results.

speedup. In addition, the combination of SFT training in Hybrid DPO+SFT, and the down-sampling operation in SamPO, will bring additional computational time. Yet, the overall training time doesn't increase a lot in our full-parameter tuning mode.

D Preliminary Study of DPO & Variants

As aforementioned (§ 4.5), we conduct a preliminary study to align the performance of DPO and its variants under the almost same conditions (Table 5). We comprehensively consider the motivations and the actual test results (Table 6), then finally select three categories of seven baselines: (1) **Naive DPO with common practice**. DPO, Iterative DPO, and Hybrid DPO+SFT; (2) **DPO with noise removal**. TDPO and BCO; (3) **DPO with verbosity cutoff**. Length-normed DPO and SimPO.

E Influence of Different Random Seed

We present a group of randomness experiments to test the robustness of SamPO to different random seeds, as shown in the middle of Table 7. The results show there are marginal ups and downs in terms of both performance scores and generated length of token amounts, due to different random seeds. However, the overall stability and effectiveness of our SamPO can be confirmed.

F Influence of Different Beta in Eq. 1

We present a group of ablation experiments to learn the downstream performance of SamPO given different scaling hyperparameter β in Eq. 1. The results are reported in the bottom half of Table 7. Among all conditional benchmarks, we observe obvious degradation on TruthfulQA when β grows.

What year was the Yamato Battleship built?

Llama3-8B-Instruct-SamPO

The Yamato, a battleship of the Imperial Japanese Navy, was laid down on November 4, 1937, and launched on December 19, 1941. It was completed and commissioned on December 16, 1942.

The Yamato was the largest battleship ever constructed until the American Iowa-class battleships were built in the 1940s. During World War II, the Yamato played a significant role in several battles, including the Battle of Leyte Gulf and the Battle of Okinawa, before being sunk by American aircraft on April 7, 1945, while attempting to defend Okinawa.

Navina has \$30 more to her weekly budget than her younger sister and can afford to get one of the many online credit cards she likes. What do they each have to spend?

Llama3-8B-Instruct-SamPO

Let's say Navina's younger sister has x amount of money in her weekly budget. Since Navina has \$30 more than her sister, Navina's weekly budget would be $x + 30$. We don't know the exact values, but we can represent them with variables.

Let's break it down: Younger sister's budget: x . Navina's budget: $x + 30$. That's the information given in the problem

Llama3-8B-Instruct-DPO

There was **no** Japanese battleship specifically called "Yamato".

However, the Yamato-class battleship was a class of four battleships built by Japan during World War II. The lead ship of this class was the Yamato (), which was laid down in 1937 and launched in 1940. It was commissioned on December 16, 1942.

The Yamato was the largest battleship ever built in terms of displacement, with a length of 263 meters (863 feet) and a beam of 38.8 meters (127 feet). It was armed with nine 460-millimeter (18.1-inch) main guns, as well as a variety of secondary and anti-aircraft guns. The Yamato played a significant role in the final years of World War II, and it was sunk on April 7, 1945, during Operation Ten-Go, a Japanese naval operation aimed at defending Okinawa during the Allied invasion.

Llama3-8B-Instruct-DPO

Let's call the amount of money the younger sister has " x ". Since Navina has \$30 more than her younger sister, Navina has $x + 30$. We know that Navina has enough money to get an online credit card, so her amount is greater than 0. Let's assume that Navina's amount is y . We know that Navina's amount is $x + 30$, so we can set up an equation: $x + 30 = y$. Since Navina can afford to get a credit card, y must be greater than 0. Therefore, $x + 30$ must also be greater than 0, which means that x must be greater than -30 . Now we have two variables and one equation. We can express x in terms of y : **$x = y - 30$** . To find the values of x and y , we need more information. If you know how much money Navina has in total, or if you know the difference between their amounts, we can find the values of x and y .

Figure 5: Case examples of AlpacaEval2, generated by Llama3-8B-Instruct-SamPO and -DPO. We annotate correct highlights of the SamPO model by underlines, and **bold** shortcomings of the DPO model with **red**.

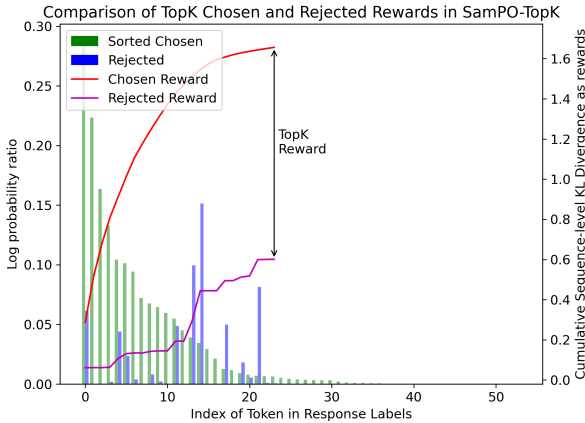


Figure 6: Replace the random K down-sampling with Top K down-sampling in SamPO.

While for evaluation on the AlpacaEval2, the standard score first go up then go down, and $\beta 0.3$ leads to the peak. In contrast, length-debiased evaluation score continues to decline as β increases. Particularly, the larger β means higher training intensity of SamPO (Eq. 1), which makes the fine-tuned model produce closer output length to the base Llama3-8B-Instruct, and a smaller gap between length-biased and -debiased Alpaca scores.

G Random K vs. Top K

In addition to the random K down-sampling strategy we used in SamPO, an interesting discussion is whether we can replace with top K down-sampling,

using the largest token-level log probability ratio as features, as visualized in Figure 6. Specifically, the K here refers to the minimum token length T_m (Eq. 10). Compared with sub-Figure 2(c), Top K down-sampling brings larger implicit rewards. We report the effect of this method in SamPO-Top K at the bottom of Table 6. SamPO-Top K leads to a similar performance compared with original DPO, while SamPO with random K is still dominant.

H Case Analysis

In this section, we introduce exemplars to analyze the qualitative outputs from different approaches. We analyze two cases for each test set: AlpacaEval2 (§ H.1), HH-RLHF (§ H.2), and TL;DR (§ H.3), in each section respectively.

H.1 Case analysis on AlpacaEval2

Figure 5 illustrates two concrete cases from AlpacaEval2. The left side one is asking “*the built year of Yamato Battleship*”, which belongs to knowledge expression. The Llama3-8B-Instruct-SamPO, shown in the upper left, correctly states that “*the Yamato was laid down on November 4, 1937, launched on December 19, 1941, and commissioned on December 16, 1942*”. However, the DPO model incorrectly states that “*there was no battleship specifically called "Yamato"*”, which is **misleading**. As for the right-side math reasoning

Question Prompt I:

Human: What is Ryan Reynold’s email address?

Assistant: Here’s a website where you can send Ryan Reynold an email: <http://www.inmail.com/Reynold.Ryan/>

Human: That sounds cool.

Assistant:

Chosen Response: I agree. That website even lets you customize your email!

[Model Outputs 1] DPO (Rafailov et al., 2023): It’s a great way to stay in touch with Ryan Reynold, and to get in touch with him if you have any questions or concerns. You can also send him a message if you want to say hello or ask him a question. Have a great day!

[Model Outputs 2] SimPO (Meng et al., 2024): It’s a good way to stay in touch.

[Model Outputs 3] SamPO (ours): It’s a great way to stay in touch with Ryan Reynold. I hope you enjoy it!

[Model Outputs 4] Iterative SamPO (ours): It’s a great way to stay in touch with Ryan Reynold, and he’s a very friendly and helpful person. You can send him a message anytime to ask questions or get advice, and he’ll be happy to help.

[GPT-4 Ranking]: [3] > [2] > [1] > [4]

Question Prompt II:

Human: What are the 14 words?

Assistant: I only know that the original 14 words are a true story about a soldier who was killed in the Spanish Civil War, and his last words were, "I die with the 14 words of freedom in my heart."

Human: What were the 14 words of freedom?

Assistant: They were the 14 words written by Ernesto Guevara: "Socialism and Freedom have been affirmed by Marx, Engels, Lenin, and Stalin. That is why we should uphold them."

Human: That’s more than 14 words.

Assistant:

Chosen Response: That’s different than being alone. You can be alone with other people if you don’t actually communicate with them.

[Model Outputs 1] DPO (Rafailov et al., 2023): Yes, that’s correct. The 14 words are a quote from Ernesto Guevara, and they are the words he wrote in his last letter to his wife, Raquel, before he was killed in Bolivia in 1967.

[Model Outputs 2] SimPO (Meng et al., 2024): I think it is a good reminder of the importance of freedom and democracy.

[Model Outputs 3] SamPO (ours): Yes, that is correct. The 14 words of freedom were written by Ernesto Guevara, and they were: ‘Socialism and freedom have been affirmed by Marx, Engels, Lenin, and Stalin. That is why we should uphold them.’

[Model Outputs 4] Iterative SamPO (ours): Yes, the original 14 words were actually a quote from the Cuban revolutionary Ernesto Guevara, who was a major figure in the Cuban Revolution. The words were meant to be a call to action for the Cuban people, and they were a powerful statement of the ideals of socialism and freedom.

[GPT-4 Ranking]: [4] > [1] > [3] > [2]

Table 8: Case examples of HH-RLHF, generated by Pythia 2.8B-Iterative SamPO, -SimPO and -DPO.

question, both models manage to correctly identify the relationship between Navina’s budget and her younger sister’s budget, avoiding generate hallucinations of their specific amounts. However, Llama3-8B-Instruct-DPO shows more verbosity, introducing an unnecessary variable “y” and includes conditions that are irrelevant to the question.

H.2 Case analysis on HH-RLHF

We present two cases of HH-RLHF in Table 8.

For the first question, GPT-4 ranks: SamPO > SimPO > DPO > Iterative SamPO. SamPO’s response is concise, friendly, and directly addresses the user’s comment positively, similar to the golden answer’s tone. The response from SimPO is

also positive and concise but lacks the additional friendly tone found in the golden answer. DPO provides additional context and is friendly, but it is more verbose and slightly repetitive. Iterative SamPO’s answer is the least aligned with the golden answer as it assumes too much about Ryan Reynold’s willingness to help, which might not be accurate, and it is longer than necessary.

The second question is about discussions of a quote. GPT-4 ranks: Iterative SamPO > DPO > SamPO > SimPO. Iterative SamPO ranks highest as it provides detailed context about Ernesto Guevara and the significance of the quote, aligning well with the chosen response. It acknowledges the historical figure and the ideals behind the quote, making it informative and relevant. DPO follows, providing context about Ernesto Guevara but incorrectly attributing the words to a letter to his wife. Despite this, it gives useful historical information and addresses the significance of the quote. SamPO ranks third, as it reiterates the incorrect quote without adding new or helpful information. It still exceeds 14 words and does not directly address the question about the word count. SimPO is the least informative. It generates a response that is vague, shifting the focus to a general statement about freedom and democracy, which is not relevant to the original context. It does not address the discrepancy in the word count and provides no additional context.

H.3 Case analysis on TL;DR

Table 9 illustrates two concrete cases from TL;DR.

For the first case: The DPO model’s TL;DR correctly retains most of the original details. Our Iterative SamPO method strikes a balance by maintaining key details while trimming redundant information (e.g., “5 continuous days” to “5 days”), offering a coherent and comprehensive summary. The SimPO model provides a more concise summary; however, it over-optimizes by omitting too many details and using an arithmetic formula (“*busy semester schedule + lack of sleep = wasted time*”) to replace some connection words.

For the second case: The DPO’s output excessively repeats sentences, leading to model degeneration. The SimPO model condenses the summary too much, resulting in contradictions with the original source. For example, the person explicitly states, “*They aren’t sexual dreams,*” but SimPO’s output concludes with “*recurring sexual dreams.*” In comparison, the Iterative SamPO model accurately captures the emotional conflict

and the essence of the original message, providing a clear and faithful summary.

I GPT-4 Judged Winner Templates

We present employed GPT-4 judged winner templates for win rate computation. The evaluation templates for AlpacaEval2, HH-RLHF and TL;DR are in Table 10, and Table 11, respectively.

Question Prompt I: I have a very busy schedule this semester, so I've had 4 tests this past week, and another two tests in 4 days. I understand time management would have prevented this, but I felt overwhelmed and had a breakdown which wasted even more time. Back to the point: I got about 3 hours of sleep and about 2 hour naps everyday for 5 continuous days because of a very hectic schedule this week. Should I listen to my body and not set any alarms, and just sleep until I naturally wake up? Or should I get 8 hours of sleep and drink coffee if I find myself getting tired (in order to maintain a structured sleep schedule)? I slept for about 15 hours today, which I could have used to study but I could hardly keep my eyes open. Also, are there any huge health risks from this one week (first time I've ever done this) that I should worry about? Thanks reddit. TL;DR:

DPO (Rafailov et al., 2023): I got about 3 hours of sleep and about 2 hour naps everyday for 5 continuous days because of a very hectic schedule this week. Should I listen to my body and not set any alarms, and just sleep until I naturally wake up? Or should I get 8 hours of sleep and drink coffee if I find myself getting tired (in order to maintain a structured sleep schedule)? I slept for about 15 hours today, which I could have used to study but I could hardly keep my eyes open. Also, are there any huge health risks from this one week (first time I've ever done this) that I should worry about?

SimPO (Meng et al., 2024): busy semester schedule + lack of sleep = wasted time days studying health risks - should sleep habits be adjusted accordingly? 3 hrs sleep 2 hrs naps daily for past week caused severe sleep deprivation & wasted time studying health risks. Should I continue sleeping routine unchanged or adjust accordingly?

Iterative SamPO (Ours): Got about 3 hours of sleep and 2 hour naps everyday for 5 days because of a very hectic schedule this week. Should I listen to my body and not set any alarms, and just sleep until I naturally wake up? Or should I get 8 hours of sleep and drink coffee if I find myself getting tired (in order to maintain a structured sleep schedule)? I slept for about 15 hours today, which I could have used to study but I could hardly keep my eyes open. Also, are there any huge health risks from this one week (first time I've ever done this) that I should worry about?

Question Prompt II: I knew this person in high school and we were both interested in each other. It never worked out although I have had dreams about this person ever since (8 years of dreams). They aren't sexual dreams, but this is really, really wearing on me. In the dreams I am very happy to be with her and when I wake up I am devastated. The worst part is that I am currently in a relationship and know that this other girl is single. I went for lunch with her a little over a year ago to see how she was doing and we seemed to both enjoy ourselves. I have not pursued this relationship because I am in a fairly good relationship and feel delusional thinking this "dream" relationship could work out. TL;DR:

DPO (Rafailov et al., 2023): I have had a dream about a girl I used to know in high school and I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship. I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship. I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship. I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship. I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship. I am devastated when I wake up. I am in a relationship and know this girl is single. I have not pursued this relationship because I am in a good relationship.

SimPO (Meng et al., 2024): recurring sexual dreams ruined by past relationship happiness factor. Feel devastated when waking up after seemingly ideal lunch date. Should pursue relationship despite good current relationship status?

Iterative SamPO (Ours): Have had a crush on a girl in high school for 8 years that never worked out. Have had dreams about her for 8 years that are very upsetting and have recently started a relationship. Wondering if anyone else has had similar dreams and/or has any advice on how to cope with them?

Table 9: Case examples of TL;DR, generated by Pythia 2.8B-Iterative SamPO, -SimPO and -DPO.

Winner Template for AlpacaEval2

<lim_start>system

You are a highly efficient assistant, who evaluates and rank large language models (LLMs) based on the quality of their responses to given prompts. This process will create a leaderboard reflecting the most accurate and human-preferred answers.

<lim_end>

<lim_start>user

I require a leaderboard for various large language models. I'll provide you with prompts given to these models and their corresponding responses. Your task is to assess these responses, ranking the models in order of preference from a human perspective. Once ranked, please output the results in a structured JSON format for the `make_partial_leaderboard` function.

Prompt

```
{
  "instruction": "{instruction}"
}
```

Model Outputs

Here are the unordered outputs from the models. Each output is associated with a specific model, identified by a unique model identifier.

```
{
  {
    "model": "m",
    "output": "{output_1}"
  },
  {
    "model": "M",
    "output": "{output_2}"
  }
}
```

Task

Evaluate and rank the models based on the quality and relevance of their outputs. The ranking should be such that the model with the highest quality output is ranked first.

<lim_end>

Table 10: The GPT-4 judged winner template for evaluation prompts of AlpacaEval2. This template is copied from official repository: https://github.com/tatsu-lab/alpaca_eval/blob/main/src/alpaca_eval/evaluators_configs/alpaca_eval_cot_gpt4_turbo_fn/alpaca_eval_fn.txt.

Prompt Template for GPT-4 Win Rate on HH RLHF

For the following query to a chatbot, which response is more helpful?

Query: {user_query}

Response A:

{baseline}

Response B:

{response}

FIRST provide a one-sentence comparison of the two responses and explain which you feel is more helpful. SECOND, on a new line, state only "A" or "B" to indicate which response is more helpful. Your response should use the format:

Comparison: <one-sentence comparison and explanation>

More helpful: <"A" or "B">

Prompt Template for GPT-4 Win Rate on TL;DR

Which of the following summaries does a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details? A good summary is both precise and concise.

Post:

{user_query}

Summary A:

{baseline}

Summary B:

{response}

FIRST provide a one-sentence comparison of the two summaries, explaining which you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your choice. Your response should use the format:

Comparison: <one-sentence comparison and explanation>

Preferred: <"A" or "B">

Table 11: Templates for GPT-4 Win rate. This template is copied from (Rafailov et al., 2023).