EARLIER TOKENS CONTRIBUTE MORE: LEARNING DIRECT PREFERENCE OPTIMIZATION FROM TEMPORAL DECAY PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct Preference Optimization (DPO) has gained attention as an efficient alternative to reinforcement learning from human feedback (RLHF) for aligning large language models (LLMs) with human preferences. Despite its advantages, DPO suffers from a length bias, generating responses longer than those from the reference model. Existing solutions like SimPO and SamPO address this issue but uniformly treat the contribution of rewards across sequences, overlooking temporal dynamics. To this end, we propose an enhanced preference optimization method that incorporates a temporal decay factor controlled by a gamma parameter. This dynamic weighting mechanism adjusts the influence of each reward based on its position in the sequence, prioritizing earlier tokens that are more critical for alignment. By adaptively focusing on more relevant feedback, our approach mitigates overfitting to less pertinent data and remains responsive to evolving human preferences. Experimental results on several benchmarks show that our approach consistently outperforms vanilla DPO by 5.9-8.8 points on AlpacaEval 2 and 3.3-9.7 points on Arena-Hard across different model architectures and sizes.

1 INTRODUCTION

Direct Preference Optimization (DPO) (Rafailov et al., 2023) has recently emerged as a highly efficient alternative for aligning large language models (LLMs) with human preferences (Askell et al., 2021; Ouyang et al., 2022). Unlike reinforcement learning from human feedback (RLHF), which involves training a reward model followed by iterative policy updates, DPO reframes the problem as a binary classification task directly over human preference data. Compared to supervised fine-tuning, DPO enables the model not only to learn what is good but also to be aware of what is bad. This formulation allows DPO to optimize preference alignment in a single-stage training process, bypassing the complexities of reinforcement learning, such as policy sampling or extensive hyperparameter tuning. By leveraging an analytical mapping between reward functions and optimal policies, DPO fine-tunes LLMs efficiently and stably, offering superior performance in tasks like sentiment control, summarization, and dialogue generation while reducing computational overhead.

Despite its advantages, DPO suffers from a length bias problem, which is caused by the unbalanced length preference due to the non-uniform length distribution of chosen and rejected responses. This leads to generated responses tending to be longer than those of the reference model if the majority of chosen responses are longer than the rejected ones. Several approaches have emerged to address this issue. One such method is SimPO (Meng et al., 2024), which introduces a more streamlined framework by eliminating the need for a reference model. Instead of relying on a pre-trained reference model for comparison, SimPO uses the average log probability of a generated sequence as the implicit reward signal. This innovation reduces computational complexity and memory usage, making SimPO a more efficient alternative to DPO. However, our experiments have revealed that SimPO suffers from unexpected performance issues when applied to data not generated through self-sampling. Similarly, Lu et al. (2024) proposed SamPO to address the length bias inherent in DPO. By constraining the reward computation to the shorter time-series range between the chosen and rejected responses, SamPO mitigates biases arising from sequence length disparities, thereby refining the preference optimization process.

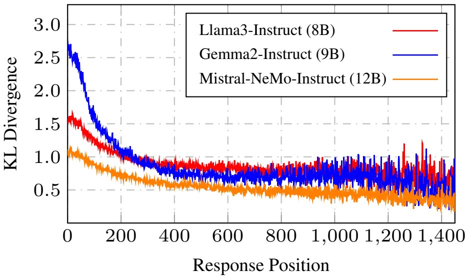


Figure 1: Visualization of KL divergence of instruct models and their DPO variants. The results include three widely used open-source LLMs: Llama3, Gemma2, and Mistral-NeMo. Observation here indicates earlier tokens contribute more during alignment.

Both of these studies, however, treat the contribution of each reward across the entire sequence as uniform. We posit that this uniform treatment does not fully capture the nuances of preference optimization. Specifically, the temporal dynamics within a sequence may influence the importance of certain tokens or segments over others. To validate this conjecture, we plot the KL divergence between the instruct models and their DPO variants on three widely used open-source models, where the results are shown in Figure 1. We notice that the KL divergence remains larger at earlier tokens but gradually decreases along the positions, which indicates earlier tokens' distributions are more likely affected by DPO. This observation aligns with the finds of previous studies (Lin et al., 2024; Yang et al., 2023) that alignment is more critical for earlier tokens. This is also consistent with the nature of next-token prediction, where an accurate prefix allows subsequent tokens to be generated on a more reliable foundation, thereby improving the overall quality and score of the output (Edunov et al., 2018). In other words, the uncertainty of earlier tokens is much lower, and the calibration for more recent tokens is higher than that for earlier ones (Wang et al., 2020).

Building on this observation, we propose an enhanced version of DPO, namely temporal decay based DPO (short for D^{2}PO), that integrates a temporal decay factor, controlled by a gamma parameter, to further refine the influence of preference data during training. Our method introduces a dynamic weighting mechanism that modulates the contribution of each reward based on its temporal relevance, allowing the model to prioritize earlier feedback over more recent tokens. To this end, when the coefficient is slightly less than 1, it gradually reduces the influence of more recent rewards, which are inherently dependent on past rewards.

By incorporating this adaptive temporal decay mechanism, D^{2}PO not only facilitates earlier tokens to contribute more but also maintains the computational efficiency that makes DPO such a compelling approach for preference optimization. Experimental results on several widely used benchmarks, including AlpacaEval2, Arena-Hard and MT-bench, demonstrate the effectiveness on both off-policy and on-policy configurations. For example, in on-policy setups, D^{2}PO outperforms DPO by up to 5.9-8.8 performance gains in terms of win rate on AlpacaEval2 and 3.3-9.7 points on Arena-Hard, respectively. Similarly, in off-policy setups, our method also demonstrates performance improvements. As a bonus of this decay mechanism which helps in reducing the overestimation of rewards caused by length bias in preference pairs, our method could be easily extended to reference-free mode, and it also can beat SimPO (Meng et al., 2024) by a big margin. Specifically, our best reference-free D^{2}PO model can achieve 62.4 LC win rate on AlpacaEval 2 and 63.6 win rate on Arena Hard, which is competitive with the reference-based model. Overall, the proposed D^{2}PO not only deliver significant performance gains, but also retains the simplicity and stability of DPO, which is easy to implement for research or industrial applications.

2 RELATED WORK

2.1 REINFORCEMENT LEARNING FROM HUMAN FEEDBACK(RLHF)

The classical RLHF pipeline (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022) consists of two distinct stages: The reward modeling phase and the RL phase.

Reward modeling phase. The reward modeling is a binary classification task. Given a prompt, the comparison pair (y_1, y_2) is collected by querying the supervised fine-tuning (SFT) model. Then, the preference $y_w \succ y_l$ is labeled by human which is used to train a reward model. Typically, Bradley-Terry model (Bradley & Terry, 1952) which quantifies the likelihood of one action being preferred over another is usually used to modeling the preference relations:

$$p\left(y_1 \succ y_2 \mid x\right)=\frac{\exp \left(r\left(x, y_1\right)\right)}{\exp \left(r\left(x, y_1\right)\right)+\exp \left(r\left(x, y_2\right)\right)}=\sigma\left(r\left(x, y_1\right)-r\left(x, y_2\right)\right) \tag{1}$$

Reinforcement learning phase. With the reward model in place, the second phase involves optimizing a policy through reinforcement learning, such as proximal policy optimization (PPO) (Schulman et al., 2017), aiming to maximize the learned reward while ensuring the policy remains close to a predefined reference policy (Korbak et al., 2022). This optimization is crucial for preventing model drift and maintaining alignment with human judgments, which is typically formulated as:

$$\max_{\theta} \mathbb{E}_{x \sim D, y \sim \pi_\theta(\cdot|x)}\left[r_\phi(x, y)\right]-\beta \mathbb{E}_{x \sim D}\left[\mathrm{KL}\left(\pi_\theta(\cdot \mid x) \| \pi_{\mathrm{ref}}(\cdot \mid x)\right)\right] \tag{2}$$

2.2 DIRECT ALIGNMENT ALGORITHMS (DAAS)

RLHF has become a cornerstone in the training of LLMs, facilitating their alignment with human preferences. However, the classical RLHF framework (Ouyang et al., 2022) is characterized by a two-stage training process, which includes reward modeling, and reinforcement learning. This complexity introduces several challenges and limitations, including reward over-optimization (Gao et al., 2022; Dubois et al., 2023) and training instability (Wu et al., 2023). Nowadays, DAAs have emerged as a promising alternative. The standard DAAs can be divided into two major categories based on whether to consider a reference model.

Reference-based methods. The reference-based methods in DAAs utilize a pre-existing model, often a supervised fine-tuned (SFT) model, as a reference point during the optimization process. This reference model serves as a baseline to which the updated model is compared, ensuring that updates do not deviate excessively from the initial, presumably safe and aligned, model configuration. DPO (Rafailov et al., 2023) is the most popular reference-based alignment algorithm and after its appearance, more researchers attempt to modify objective function for better performance. KTO (Ethayarajh et al., 2024) distinguishes itself by its capability to train from non-paired preference data, providing a unique angle on optimization. IPO (Azar et al., 2023) learns directly from preferences without relying on the Bradley-Terry model assumption that assumes that pairwise preferences can be substituted with pointwise rewards. R-DPO (Park et al., 2024) is an enhanced derivative of DPO, fortified with an additional regularization term designed to mitigate the tendency to exploit length biases, thus ensuring more balanced and diverse response generation.

Reference-free methods. In contrast to reference-based methods that depend on a pre-existing model for guidance, reference-free methods forgo the need for such a reference. They directly optimize the model parameters in response to human feedback, which can enhance the flexibility of the optimization process. However, this freedom also presents challenges in controlling the extent of updates. CPO (Xu et al., 2024) leverages sequence likelihood as a reward signal and is trained in conjunction with an SFT objective. ORPO (Hong et al., 2024) is a novel alignment method that integrates an odds ratio-based penalty into the supervised fine-tuning process. SimPO (Meng et al., 2024) uses an average log probability as an implicit reward and introduces a target reward margin to enhance performance without relying on a reference model.

3 METHODOLOGY

3.1 DIRECT PREFERENCE OPTIMIZATION (DPO).

Direct Preference Optimization (DPO) is a pivotal advancement in the field of offline preference-based training for language models. Traditional RLHF involves a complex, multi-stage process that includes training a reward model to align with human preferences and subsequently optimizing a policy model to maximize this reward while staying close to the original model's distribution. DPO simplifies this process by reparameterizing the reward function directly in terms of the policy model, eliminating the need for an explicit reward model:

$$r(x, y)=\beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}+\beta \log Z(x), \tag{3}$$

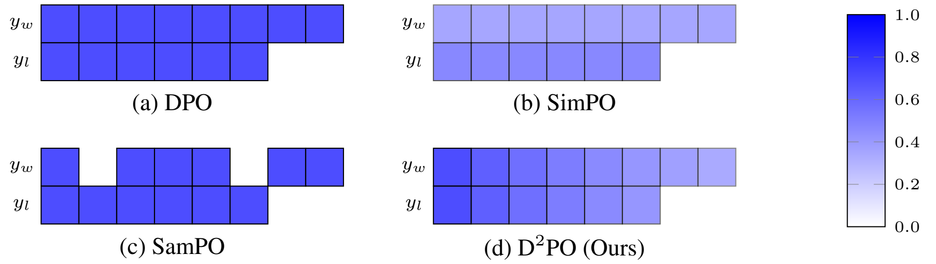


Figure 2: Illustration of coefficients in DPO, SimPO, SamPO, and our D^{2}PO across various positions. Each box represents a coefficient, and the opacity denotes the magnitude, with darker colors indicating higher values. (a) For DPO, the coefficients are uniform across different positions. (b) For SimPO, the coefficients of the chosen y_w and the rejected y_l are normlaized by their lengths $|y_w|$ and $|y_l|$, respectively. (c) In SamPO, the coefficients are selected based on the minimum length of $|y_w|$ and $|y_l|$. (d) Our method introduces a γ factor controlling the coefficients, which decay according to γ^t (e.g., 1, γ, γ^2, ..., γ^T). Here, we use $\gamma = 0.9$ for a clear visualization.

where π_θ, π_{ref} denotes the policy model and reference model, respectively. Z(x) is the partition function, and β is a hyperparameter to control the deviation from the reference model. Substituting this reward into the Bradley-Terry (BT) ranking objective yields the DPO loss:

$$\mathcal{L}_{\text{DPO}}\left(\pi_\theta;\pi_{\text{ref}}\right) = -\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(\mathbf{y}_w\mid\mathbf{x}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_w\mid\mathbf{x}\right)} - \beta\log\frac{\pi_\theta\left(\mathbf{y}_l\mid\mathbf{x}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_l\mid\mathbf{x}\right)}\right)\right] \tag{4}$$

DPO operates by formulating an implicit reward using the log ratio of the likelihood of a response between the current policy model and a supervised fine-tuned (SFT) model. This reward is then incorporated into the Bradley-Terry ranking objective to directly optimize the policy model for preference data. The effectiveness of DPO lies in its ability to simplify the preference optimization process, making it more accessible and efficient for practical applications.

3.2 TEMPORAL DECAY MATTERS IN PREFERENCE LEARNING.

Motivation. Preference learning plays a pivotal role in optimizing large language models (LLMs), especially when leveraging user feedback to align model outputs with human preferences. While methods like Direct Preference Optimization (DPO) and its successors have demonstrated significant potential in this domain, they exhibit a critical oversight: the uniform treatment of tokens across a sequence. As illustrated in Figure 2, DPO, SimPO, and SamPO assign identical coefficients to all tokens within the chosen response y_w and the rejected response y_l. We argue that optimizing each token equally, without considering their positional importance, is suboptimal.

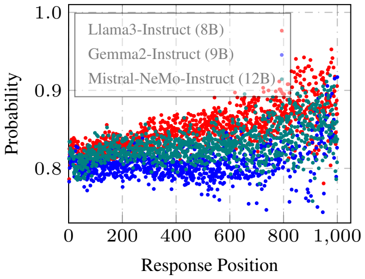


Figure 3: Probability against positions on 1000 samples.

Our observations indicate that earlier tokens receive greater optimization during the preference learning process compared to later ones (see Figure 1). This suggests that the benefits derived from the alignment phase over SFT are predominantly due to the optimization of initial tokens. Additionally, when plotting the prediction probability across different response positions in Figure 3, we find that more recent tokens have higher probabilities than earlier tokens. This indicates that the model becomes increasingly confident in predicting tokens as it progresses through the sequence, likely due to the accumulating contextual information from previous tokens. However, since the accuracy of these later tokens is already high—reaching up to 0.9, further improvements are more likely to come from enhancing the accuracy of the earlier tokens. Therefore, a natural approach is to focus on improving the accuracy of the prefix: *the more accurate the earlier tokens are, the better the overall quality of the sequence will be.*

Temporal Decay Mechanism. To address this issue, we propose introducing a temporal decay mechanism, which has been validated as effective in other domains (Yang et al., 2024), to emphasize the significance of earlier tokens in the sequence. Considering the original DPO loss formulation, the most direct way to prioritize earlier tokens is to incorporate a position-dependent coefficient that decays over time. Multiple decay mechanisms can achieve this, including linear, polynomial, step, and cosine decay functions. After evaluating these options, we chose exponential decay due to its simplicity and effectiveness.

Exponential decay applies a coefficient that decreases exponentially with the token position, represented as γ^t, where γ is the decay rate ($0 < \gamma < 1$) and t is the token's position. This approach provides a smooth and gradual reduction in the influence of later tokens, ensuring that earlier tokens have a more significant impact on the loss calculation. To this end, we adapt this concept to the DPO loss function which defined as:

$$\mathcal{L}_{\mathrm{D^2PO}}\left(\pi_\theta; \pi_{\mathrm{ref}}\right) = -\log\sigma\left(\sum_{t=0}^{T_w} \gamma^t \beta \log \frac{\pi_\theta\left(\mathbf{y}_w^t \mid \mathbf{x}, \mathbf{y}_w^{<t}\right)}{\pi_{\mathrm{ref}}\left(\mathbf{y}_w^t \mid \mathbf{x}, \mathbf{y}_w^{<t}\right)} - \sum_{t=0}^{T_l} \gamma^t \beta \log \frac{\pi_\theta\left(\mathbf{y}_l^t \mid \mathbf{x}, \mathbf{y}_l^{<t}\right)}{\pi_{\mathrm{ref}}\left(\mathbf{y}_l^t \mid \mathbf{x}, \mathbf{y}_l^{<t}\right)}\right) \tag{5}$$

In this formulation, the exponential decay factor γ^t adjusts the contribution of each token based on its position in the sequence. As is shown in Figure 2 (d), coefficients of each token in D^{2}PO gradually decrease along the position (the color from dark to light), placing greater emphasis on earlier tokens. This modification aligns the optimization process with the observed pattern of optimization in preference learning, where initial tokens benefit more from the alignment phase.

3.3 DERIVATION OF D^{2}PO

In the reinforcement learning scenario, two fundamental concepts are the state-value function V and the action-value function Q. The former represents the expected cumulative reward from taking action a in state s, meanwhile the latter represents the reward under state s. We extend the relation between the Q-function and the V-function under a KL divergence constraint, as proposed in Rafailov et al. (2024), by incorporating the temporal decay mechanism:

$$Q^*\left(\mathbf{s}_t, \mathbf{a}_t\right) = \begin{cases} r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \beta \log \pi_{\mathrm{ref}}\left(\mathbf{a}_t \mid \mathbf{s}_t\right) + \gamma V^*\left(\mathbf{s}_{t+1}\right), & \text{if } \mathbf{s}_{t+1} \text{ is not terminal} \\ r\left(\mathbf{s}_t, \mathbf{a}_t\right) + \beta \log \pi_{\mathrm{ref}}\left(\mathbf{a}_t \mid \mathbf{s}_t\right), & \text{if } \mathbf{s}_{t+1} \text{ is terminal} \end{cases} \tag{6}$$

where $\gamma \in (0, 1]$ represents the discount factor. In the assumption of DPO and its subsequent variants, γ is typically set to 1 which indicates long-term returns do not need to decay. However, in auto-regressive scenarios such as language models, the longer the context provided, the lower the uncertainty of the model's predictions (see Figure 3). Therefore, the tokens at the beginning of the response make greater contributions to the total return. Based on this assumption, we can get the formulation of the reward over a trajectory $\tau = \{s_1, a_1, ..., a_{T-1}, s_T\}$:

$$\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) = \sum_{t=0}^{T-1} \gamma^t Q^*(s_t, a_t) - \gamma^t \beta \log \pi_{ref}(a_t, s_t) - \gamma^{t+1} V^*(s_{t+1}) \tag{7}$$

Noting that, in the general maximum entropy RL setting (Ziebart et al., 2008; Ziebart, 2010), the optimal policy is given by Boltzmann probability distribution as:

$$\pi^*\left(\mathbf{a}_t \mid \mathbf{s}_t\right) = e^{(Q^*(\mathbf{s}_t, \mathbf{a}_t) - V^*(\mathbf{s}_t))/\beta} \tag{8}$$

By taking the logarithm of the Eq. (8), we can simplified Eq. (7):

$$\begin{aligned} \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t) &= Q^*(s_0, a_0) - \beta \log \pi_{ref}(a_0, s_0) + \sum_{t=1}^{T-1} (\gamma^t \beta \log \frac{\pi_\theta(a_t, s_t)}{\pi_{ref}(a_t, s_t)}) & (9) \\ &= V^*(s_0) + \sum_{t=1}^{T-1} (\gamma^t \beta \log \frac{\pi_\theta(a_t, s_t)}{\pi_{ref}(a_t, s_t)}) & (10) \end{aligned}$$

We can then plug Eq. (10) into the Bradley-Terry ranking objective, which yields our final loss formation as discussed in Eq. (5). This is similar to the standard DPO objective, except for an additional temporay decay term γ. We empirically set $\gamma < 1$ to focus more on short term return rather than long term return.

3.4 REFERENCE-FREE IS CONSISTENT WITH ON-POLICY SETUPS.

Reference-based methods often incorporate a KL divergence constraint to prevent the policy model from deviating significantly from its initial state, which adds computational and memory overhead. In the context of DPO, this constraint appears as an adaptive margin term $\log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_w|x)}$ in the pairwise loss function. This term quantifies the reference model's preference difference between less-preferred (y_l) and preferred (y_w) responses. We note that the DPO loss can be viewed as a specific case of contrastive loss, where $\log \pi_\theta(y)$ measures the relevance between the prompt x and the response y. The adaptive margin ensures that loss values remain moderate, contributing to training stability. However, if the reference model assigns similar probabilities to both y_w and y_l (i.e., the margin approaches zero), the impact of the reference model diminishes, suggesting that it can be easily excluded.

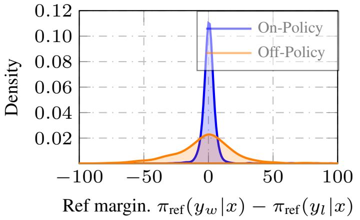


Figure 4: Reference margin of DPO.

To validate this, we analyze the margin distributions in the UltraFeedback dataset under off-policy and on-policy settings. In the off-policy setting, we use original responses, while in the on-policy setting, responses are regenerated by the same model. As illustrated in Figure 4, the on-policy dataset exhibits smaller variance in margins and an average closer to zero compared to the off-policy dataset. This indicates a higher proportion of semi-hard samples in the on-policy data. From this perspective, we can discard the KL divergence constraint under on-policy settings and easily derive the reference-free version loss function:

$$\mathcal{L}_{\text{D}^2\text{PO}}(\pi_\theta) = -\log\sigma\left(\sum_{t=0}^{T_w}\gamma^t\beta\log\pi_\theta\left(\mathbf{y}_w^t \mid \mathbf{x}, \mathbf{y}_\mathbf{w}^{\mathbf{<t}}\right) - \sum_{t=0}^{T_l}\gamma^t\beta\log\pi_\theta\left(\mathbf{y}_l^t \mid \mathbf{x}, \mathbf{y}_\mathbf{l}^{\mathbf{<t}}\right)\right) \tag{11}$$

4 THEORETICAL ANALYSIS

We analyze the influence of the discount factor γ on the performance of our method compared to DPO. Both DPO and our method can be considered as token-level Markov Decision Processes (MDPs) that satisfy the Bellman equation. Here, we define the suboptimality as the performance difference between the optimal policy π^* and a given policy π under specific discount factors, which has been widely discussed in offline RL (Rashidinejad et al., 2021; Jin et al., 2021).

4.1 SUBOPTIMALITY DECOMPOSITION

Definition 1 (Suboptimality). The suboptimality of a policy π with respect to the optimal policy π^*, starting from an initial state s under discount factor γ, is defined as:

$$\text{SubOpt}(\pi, s; \gamma) = V_\gamma^{\pi^*}(s) - V_\gamma^{\pi}(s), \tag{12}$$

where $V_\gamma^\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{H-1}\gamma^t r(s_t, a_t) \mid s_0 = s\right]$ is the expected return of policy π under discount factor γ, and H is the finite horizon. To analyze the influence of the discount factor γ, we consider the suboptimality of our method evaluated with an evaluation discount factor $\gamma_e = 1.0$. We decompose the suboptimality into three terms that separately capture the differences due to the discount factors and the policy discrepancies. In this way, we can rewrite the suboptimality as below:

$$\begin{aligned}\text{SubOpt}(\pi, s; \gamma_e) &= V_{\gamma_e}^{\pi^*}(s) - V_{\gamma_e}^{\pi}(s)\\ &= \underbrace{\left[V_{\gamma_e}^{\pi^*}(s) - V_{\gamma}^{\pi^*}(s)\right]}_{\Delta_1} + \underbrace{\left[V_{\gamma}^{\pi^*}(s) - V_{\gamma}^{\pi}(s)\right]}_{\Delta_2} + \underbrace{\left[V_{\gamma}^{\pi}(s) - V_{\gamma_e}^{\pi}(s)\right]}_{\Delta_3}\end{aligned} \tag{13}$$

This decomposition allows us to separately analyze the impact of the discount factors and the policy differences.

4.2 SUBOPTIMALITY ANALYSIS

Theorem 1(Suboptimality Upper Bound). *Let π^* denote the optimal policy, and π be the policy under a discount factor $\gamma \in [0, 1)$. Assume that rewards are bounded such that $|r(s, a)| \leq R$ for all*

Table 1: We report AlpacaEval 2 (Li et al., 2023) (denoted by AE2), Arena-Hard (Li et al., 2024) (denoted by AH), and MT-Bench (Zheng et al., 2023) (denoted by MB) results under three settings using standard provided samples. Note that LC and WR denote length-controlled and raw win rate, respectively. We used off-the-shelf models as the SFT model. And our judge model is GPT-4-Turbo.

Method	**Llama3-Instruct (8B)**				**Gemma2-Instruct (9B)**				**Mistral-NeMo-Instruct (12B)**			
	AE2		**AH**	**MB**	**AE2**		**AH**	**MB**	**AE2**		**AH**	**MB**
	WR (%)	**LC (%)**	**WR (%)**	**G4-T**	**WR (%)**	**LC (%)**	**WR (%)**	**G4-T**	**WR (%)**	**LC (%)**	**WR (%)**	**G4-T**
SFT	39.1	40.1	27.6	7.5	37.6	47.2	44.1	8.3	44.6	47.7	46.5	8.1
DPO	37.4	40.3	27.7	**7.7**	38.8	48.8	42.5	8.1	44.4	49.3	48.5	8.3
KTO	33.3	38.1	21.0	7.5	39.1	50.0	43.8	8.3	37.4	48.7	35.8	8.2
IPO	42.2	**45.7**	31.9	7.6	41.0	50.0	48.2	8.0	39.8	48.9	39.8	8.2
SamPO	40.7	43.1	26.1	7.5	39.9	50.1	46.9	8.2	43.5	49.5	50.1	8.1
D^{2}PO (ours)	**43.5**	43.0	**37.0**	**7.7**	**45.5**	**51.0**	**50.2**	**8.3**	**51.3**	**54.4**	**51.8**	**8.4**
ORPO	10.6	15.3	6.8	6.3	11.3	21.6	10.2	7.1	9.6	17.0	9.8	6.9
SimPO	0.3*	0.8*	1.4*	1.6*	38.8	50.0	31.6	8.0	46.8	**53.3**	46.6	8.0

Table 2: Following the setting in Meng et al. (2024), we used the on-policy data to obtain the chosen and rejected and applied a stronger reward model. † denotes our reference-free version.

Method	**Llama3-Instruct (8B)**				**Gemma2-Instruct (9B)**				**Mistral-NeMo-Instruct (12B)**			
	AE2		**AH**	**MB**	**AE2**		**AH**	**MB**	**AE2**		**AH**	**MB**
	WR (%)	**LC (%)**	**WR (%)**	**G4-T**	**WR (%)**	**LC (%)**	**WR (%)**	**G4-T**	**WR (%)**	**LC (%)**	**WR (%)**	**G4-T**
SFT	39.1	40.1	27.6	7.5	37.6	47.2	44.1	8.3	44.6	47.7	46.5	8.1
DPO	46.2	47.6	42.4	7.8	47.0	53.4	56.7	**8.4**	53.5	53.3	59.0	8.4
KTO	42.4	44.8	32.1	7.7	48.3	53.4	57.1	8.3	48.9	51.9	53.2	8.4
IPO	42.9	46.0	34.5	7.9	50.9	50.0	59.7	8.3	53.6	54.4	59.7	8.4
SamPO	44.4	47.2	35.8	**8.0**	45.8	55.2	55.2	8.2	51.1	53.0	58.3	8.3
D^{2}PO (ours)	**47.4**	**53.5**	**47.3**	7.8	**57.2**	**59.7**	**66.4**	8.3	**57.3**	**62.1**	**62.3**	**8.6**
ORPO	37.8	39.3	25.5	7.7	41.9	51.1	45.3	8.2	43.8	47.5	46.0	8.2
SimPO	44.4	50.3	41.9	**7.8**	54.5	58.4	65.0	**8.3**	51.3	55.0	61.9	**8.3**
D^{2}PO† (ours)	**48.0**	**53.9**	**46.1**	7.7	**56.7**	**60.8**	**65.7**	**8.3**	**58.3**	**62.4**	**63.6**	**8.3**

states s and actions a, and consider a finite horizon H. Then, the suboptimality of π compared to π^* when evaluated with an evaluation discount factor $\gamma_e = 1.0$ satisfies the following upper bound:

$$\text{SubOpt}(\pi, s; \gamma_e) \leq 2(H - \frac{1-\gamma^H}{1-\gamma})R + \frac{2(1-\gamma^H)^2}{(1-\gamma)^2} E_{s\sim d^{\pi^*}} \left[\mathbb{TV}(\pi^*(a|s)||\pi(a|s)\right] R \qquad (14)$$

The complete proof is included in Appendix E. This upper bound reveals that the suboptimality depends on both the discount factor γ and the mismatch between π and π^*. Specifically, the first term $H - \frac{1-\gamma^H}{1-\gamma}$ decreases as γ increases, while the second term $\left(\frac{1-\gamma^H}{1-\gamma}\right)^2$ increases, highlighting a trade-off in the choice of γ. As both terms vary monotonically with the discount factor γ but in opposing directions, there exists an optimal value γ^* within the interval $(0, 1)$ that balances these effects to minimize the overall suboptimality.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUPS

Due to page limitations, we provide details regarding the training data, hyperparameters, evaluation benchmarks, and the baselines used for comparison in Appendix A.

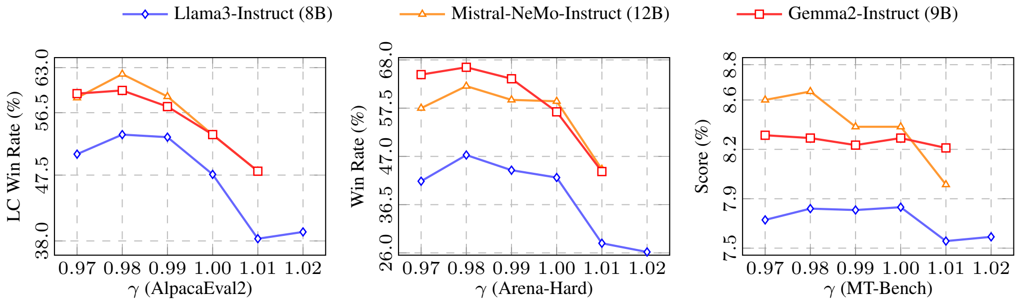


Figure 5: Performance against different γ choices of three open-source models on three benchmarks.

5.2 EXPERIMENTAL RESULTS

In our experiments, we provide a comprehensive comparison of our proposed method against DPO and its variants on both off-policy and on-policy data respectively, along with the baselines introduced in Section 5.1. The baselines are categorized into two broad paradigms: reference-based and reference-free (SimPO and ORPO). Notably, as discussed in Section 3.4, our method can be seamlessly integrated into the reference-free paradigm using on-policy data. We ensure fair comparisons by maintaining consistency in the codebase and experimental settings across all methods evaluated.

Off-policy Setups. Table 1 clearly demonstrates that our method delivers significant improvements in win rates across all configurations. Specifically, when applied to the Llama3, our method outperforms DPO by margins of 6.1% and 2.9% in standard and length-controlled evaluation scenarios, respectively. Similarly, for the Mistral-NeMo model, our method surpasses DPO by margins of 6.9% and 5.1% in standard and length-controlled scenarios, respectively. We observed that reference-free methods exhibited instability when applied to off-policy data, often leading to a degradation in model performance. This issue is particularly evident with SimPO, where previous work observed similar findings (Lu et al., 2024). This phenomenon highlights the challenges associated with reference-free methods in preference optimization on off-policy data.

On-policy Setups. As shown in Table 2, our proposed method, along with all baselines, achieves better results compared to off-policy settings. Notably, our method consistently demonstrates improvements across different setups. Due to the reward model's length preference when selecting on-policy data, models trained on this data are more prone to verbosity. A critical observation in standard evaluations is the inherent bias favoring models that generate longer responses, which tend to achieve higher win rates. However, our method not only achieves superior win rates but also produces significantly shorter responses, showcasing its efficiency in generating concise and relevant outputs. Additionally, when the reference model is omitted, our method outperforms SimPO by 2.4–7.4 in LC win rate and 0.7–4.2 in win rate on AlpacaEval 2 and Arena-Hard, respectively. These findings further underscore the robustness and effectiveness of our approach. This superiority in both reference-based and reference-free contexts emphasizes the versatility and reliability of our method in preference optimization.

6 ANALYSES

γ plays an important role. The temporal decay is one of the main technical contributions of this work, and we would like to show how the γ affects the performance. We conducted ablation studies on three open-source models for robust conclusions. Through results as shown in Figure 5, we see that nearly all variants with γ lower than 1.0 consistently outperform DPO[1]. Also, $\gamma = 0.98$ achieves the highest performance across three benchmarks for these strong open-source models. This indicates that our method is robust to the choice of γ, reducing the need for extensive hyperparameter tuning.

γ larger than 1 is harmful. As highlighted in the previous section, we prioritize earlier feedback over more recent feedback, aligning with the next-token prediction paradigm. We conducted an

---

[1] DPO is a special case of ours where γ equals to 1.0.

Table 3: Results on OpenLLM Benchmark, including reasoning and mathematical testsets. Note that Hella. denotes Hellaswag, Truth. denotes TruthfulQA and Wino. denotes Winogrande.

Method	MMLU	GSM8K	Math	IFEval	ARC-C	Hella.	Truth.	Wino.
	0-shot	0-shot	0-shot	0-shot	25-shot	10-shot	0-shot	5-shot
(a) Llama3-Instruct (8B)								
SFT	61.7	78.5	7.9	68.6	62.0	78.8	51.6	75.5
DPO	56.7	70.5	7.8	65.1	65.1	**79.9**	56.4	74.5
SimPO	55.2	57.5	5.3	60.8	**67.6**	78.8	**63.8**	74.3
D^{2}PO (ours)	**61.4**	**72.0**	**8.5**	**65.6**	65.8	79.0	57.6	**75.1**
(b) Gemma2-Instruct (9B)								
SFT	72.8	87.4	19.4	71.9	71.8	81.7	60.2	77.9
DPO	72.2	88.5	19.4	60.1	69.9	71.5	57.7	72.7
SimPO	72.4	88.2	19.0	**71.5**	68.3	66.5	58.9	73.7
D^{2}PO (ours)	**72.7**	**88.9**	**21.2**	71.2	**71.4**	**81.0**	**61.3**	**76.0**

experiment where the decay factor γ was set to slightly greater than 1.0 to observe the effects. The results could also be observed in Figure 5. When γ exceeds 1.0, rewards linked to later tokens in the sequence receive larger coefficients than those for earlier tokens. However, this adjustment was detrimental to preference optimization, resulting in performance that lagged behind the standard DPO on both the AlpacaEval 2 and Arena-Hard benchmarks. This finding demonstrates the crucial role of earlier tokens in the alignment process and indicates that overemphasizing later tokens can degrade model performance.

Evaluations on OpenLLM Benchmark. To verify whether the improvements of D^{2}PO on the aforementioned RLHF benchmarks, such as Alpaca Eval2, Arena Hard, and MT-bench, come at the expense of general language modeling ability, we conducted a comprehensive evaluation of downstream tasks on the Open LLM leaderboard[2]. Specifically, we employed zero-shot evaluations on MMLU (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), IFEval (Zhou et al., 2023), and TruthfulQA (Lin et al., 2022). Additionally, we performed few-shot evaluations on ARC-C (Clark et al., 2018), Hellaswag (Zellers et al., 2019), and Winogrande (Levesque et al., 2012) according to the official settings in the Open LLM leaderboard. The results are summarized in Table 3, and we observe that:

- In the Llama3-8B configuration, our D^{2}PO method significantly outperforms both DPO and SimPO, particularly on the MMLU and Math benchmarks. Notably, D^{2}PO exhibits less performance degradation on GSM8K compared to SimPO, despite both methods effectively controlling output length. D^{2}PO achieves substantial performance gains on the Math dataset, surpassing the Instruct model by 0.55 points, while the other two methods show a noticeable decline.
- In the Gemma2-9B configuration, we observe a similar pattern, with D²PO demonstrating a significant performance advantage on the Math benchmark. These results suggest that D²PO effectively enhances reasoning and mathematical problem-solving abilities in LLMs across different models. Furthermore, these additional evaluations on specialized datasets confirm that D²PO maintains its effectiveness across various contexts and task types.

Comparisons of Various Decay Strategies We have proven the importance of temporal decay. Following the classic Markov Decision Process, we use exponential decay as our default decay strategy. Meanwhile, We also consider several variants of decay strategies, including Head decay, Linear decay and Power-Law decay. The detailed decay mechanism are summarized in Table 4. We observe 1-0 decay and Linear Deacy show inferior results to the tenporal decay, and even underperforms with the vanilla DPO. While though Power-Law method also shows promising results, but it cannot properly control the response length competitive results with exponential decay.

Lengthy Debias. Previous studies (Park et al., 2024; Lu et al., 2024; Meng et al., 2024) have demonstrated that DPO is susceptible to length exploitation, as it tends to amplify verbosity biases

---

[2] Open LLM leaderboard is created by huggingface to provide a standardized evaluation setup for LLMs, which includes several popular benchmarks encompassing a wide range of capabilities across multiple domains.

Table 4: Comparison of different decay mechanisms in terms of performance and response length.

Decay Strategy	Rewards	AE2			AH		MB
		WR (%)	**LC (%)**	**Len.**	**WR (%)**	**Len.**	**G4-T**
Exponential	$\sum_{t=0}^{T} \gamma^t \beta \log \frac{p_\theta(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}{p_{ref}(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}$	57.2	59.7	1950	66.4	724	8.3
Head	$\sum_{t=0}^{\gamma T} \beta \log \frac{p_\theta(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}{p_{ref}(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}$	48.6	54.7	1762	57.4	680	8.2
Linear	$\sum_{t=0}^{\gamma T} \left(1 - \frac{t}{\gamma T}\right) \beta \log \frac{p_\theta(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}{p_{ref}(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}$	48.3	54.5	1713	59.4	661	8.3
Power-Law	$\sum_{t=0}^{T} \frac{1}{t^\gamma} \beta \log \frac{p_\theta(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}{p_{ref}(\mathbf{y}_t\|\mathbf{x},\mathbf{y}_{<\mathbf{t}})}$	56.8	57.7	2011	71.2	823	8.5

present in the preference datasets. This propensity can lead to suboptimal outcomes where the model's decisions are disproportionately influenced by the length of the responses rather than their quality or relevance. To investigate the relationship between the length bias of training data and the output length of the model, we visualized the DPO and D^{2}PO loss of 1000 random samples based on the length gap between the chosen and rejected responses. For simplicity, verbosity-biased data refers to pairs in which the chosen response must be longer than the reject response and brevity-biased data refers to the opposite type of data.

From Figure 6, we can see that during the DPO training process, the loss of verbosity-biased data is large, while the loss of brevity-biased data is small. Consequently, DPO prioritizes the optimization of verbosity-biased data, increasing likelihood of longer chosen responses and decreasing likelihood of shorter ones. This kind of imbalance loss can easily cause model verbosity. Meanwhile, D^{2}PO reduce the loss imbalance between verbosity-biased data and brevity-biased data, thereby controlling the output length of the model.

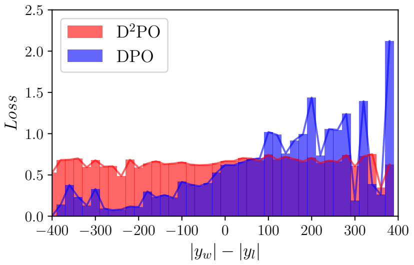


Figure 6: Loss vs. length diff.

Table 5: Human evaluation results on two benchmarks.

Model	**Win**	**Tie**	**Lose**
AlpacaEval 2	116	36	48
Arena-Hard	107	62	31

Human Evaluations To further validate our results, we conducted human evaluations on the AlpacaEval2 and Arena-Hard datasets using the Gemma2-9B model. We enlisted four evaluators, with each person evaluating 50 samples for each benchmark. For each instruction, we randomized the order of the outputs from DPO and D^{2}PO to prevent bias. The evaluators assessed the responses based on three criteria: accuracy, completeness, and relevance, determining which response was better for each sample. If both responses were equally correct or incorrect, the result was considered a tie. As shown in Table 5, our comparison between D²PO and DPO indicates that D²PO achieved a significantly higher win rate than DPO, with an overall win rate of 67% in Arena-Hard and 69% in AlpacaEval 2 (calculated as (win + tie/2) / total).

7 CONCLUSIONS

In this work, we revisited the loss objectives of DPO and its variants, introducing a temporal decay mechanism governed by a parameter γ. Motivated by the observation that earlier tokens contribute more significantly during preference optimization, our dynamic weighting scheme prioritizes these initial tokens, aligning naturally with the next-token prediction paradigm. Extensive experiments demonstrate that our approach consistently outperforms vanilla DPO, achieving notable improvements across diverse benchmarks and model architectures. By enabling DPO to focus more on short-term rewards while retaining its simplicity and stability, our method offers a compelling solution for preference-based fine-tuning of large-scale models. Furthermore, we showed that our method can be extended to a reference-free, on-policy setting, outperforming existing approaches.

REFERENCES

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H. Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan M. Cohen, Sirshak Das, Ayush Dattagupta, Olivier Delalleau, Leon Derczynski, Yi Dong, Daniel Egert, Ellie Evans, Aleksander Ficek, Denys Fridman, Shaona Ghosh, Boris Ginsburg, Igor Gitman, Tomasz Grzegorzek, Robert Hero, Jining Huang, Vibhu Jawa, Joseph Jennings, Aastha Jhunjhunwala, John Kamalu, Sadaf Khan, Oleksii Kuchaiev, Patrick LeGresley, Hui Li, Jiwei Liu, Zihan Liu, Eileen Long, Ameya Sunil Mahabaleshwarkar, Somshubra Majumdar, James Maki, Miguel Martinez, Maer Rodrigues de Melo, Ivan Moshkov, Deepak Narayanan, Sean Narenthiran, Jesus Navarro, Phong Nguyen, Osvald Nitski, Vahid Noroozi, Guruprasad Nutheti, Christopher Parisien, Jupinder Parmar, Mostofa Patwary, Krzysztof Pawelec, Wei Ping, Shrimai Prabhumoye, Rajarshi Roy, Trisha Saar, Vasanth Rao Naik Sabavat, Sanjeev Satheesh, Jane Polak Scowcroft, Jason Sewall, Pavel Shamis, Gerald Shen, Mohammad Shoeybi, Dave Sizer, Misha Smelyanskiy, Felipe Soares, Makesh Narsimhan Sreedhar, Dan Su, Sandeep Subramanian, Shengyang Sun, Shubham Toshniwal, Hao Wang, Zhilin Wang, Jiaxuan You, Jiaqi Zeng, Jimmy Zhang, Jing Zhang, Vivienne Zhang, Yian Zhang, and Chen Zhu. Nemotron-4 340b technical report. *CoRR*, abs/2406.11704, 2024.

AI@Meta. Llama 3 model card. 2024. URL `https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md`.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, John Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom B. Brown, Jack Clark, Sam McCandlish, Christopher Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861, 2021.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *ArXiv*, abs/2310.12036, 2023.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

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

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 489–500. Association for Computational Linguistics, 2018.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. KTO: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Jiwoo Hong, Noah Lee, and James Thorne. ORPO: Monolithic preference optimization without reference model. *ArXiv*, abs/2403.07691, 2024.

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

Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5084–5096. PMLR, 2021.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann (eds.), *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pp. 267–274. Morgan Kaufmann, 2002.

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

Tomasz Korbak, Ethan Perez, and Christopher L. Buckley. Rl with kl penalties is better viewed as bayesian inference. In *Conference on Empirical Methods in Natural Language Processing*, 2022.

Hector Levesque, Ernest Davis, and Leora Morgenstern. The Winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*, 2012.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The Arena-Hard pipeline, April 2024. URL `https://lmsys.org/blog/2024-04-19-arena-hard/`.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. AlpacaEval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023.

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Raghavi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base llms: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *ACL*, pp. 3214–3252, 2022.

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

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

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 E. Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Francis Christiano, Jan Leike, and Ryan J. Lowe. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *ArXiv*, abs/2403.19159, 2024.

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

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function. *CoRR*, abs/2404.12358, 2024.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 11702–11716, 2021.

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

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. Gemma 2: Improving open language models at a practical size, 2024. URL `https://arxiv.org/abs/2408.00118`.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. On the inference calibration of neural machine translation. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pp. 3070–3079. Association for Computational Linguistics, 2020.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *ArXiv*, abs/2310.00212, 2023.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *ArXiv*, abs/2401.08417, 2024.

Shentao Yang, Shujian Zhang, Congying Xia, Yihao Feng, Caiming Xiong, and Mingyuan Zhou. Preference-grounded token-level guidance for language model fine-tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Shentao Yang, Tianqi Chen, and Mingyuan Zhou. A dense reward view on aligning text-to-image diffusion with preference. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *NeurIPS Datasets and Benchmarks Track*, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics.

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

Brian D. Ziebart. *Modeling Purposeful Adaptive Behavior with the Principle of Maximum Causal Entropy*. PhD thesis, Carnegie Mellon University, USA, 2010.

Brian D. Ziebart, Andrew L. Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, pp. 1433–1438. AAAI Press, 2008.

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

A EXPERIMENTAL SETUPS

Model Setting. We conducted preference optimization experiments using three model families: Llama3-8B (AI@Meta, 2024), Gemma2-9B (Team et al., 2024) and Mistral-12B (Jiang et al., 2023). Here, we mainly focused on building our systems upon the instruct models. Thus, we utilized pre-trained instruction-tuned models (e.g., meta-llama/Meta-Llama-3-8B-Instruct, google/gemma-2-9b-it, and nvidia/Mistral-NeMo-12B-Instruct) as the SFT models.[3]

Training Data Our experiments were carried out using the UltraFeedback dataset. Specifically, We categorize the preference data into two types: 1) off-policy data (original response pairs from the UltraFeedback dataset), and 2) on-policy data generated using the SFT models. Similar to SimPO (Meng et al., 2024), for each prompt x, we generated 5 responses using the SFT model with a sampling temperature of 0.8. To validate these responses, we employed RLHFlow/ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024) to assign scores to each response, allowing us to select the highest-scoring response as y_w and the lowest-scoring one as y_l.

Hyperparameters For all models, we set the maximum response length to 2,048 tokens and used a batch size of 128. Optimization was performed using the AdamW optimizer (Kingma & Ba, 2014) with a learning rate of $5e-7$ and a cosine learning rate schedule featuring a 10% warmup period. In preference optimization methods, including DPO and its variants such as our method D^{2}PO and SamPO, we set β to 0.1 to ensure a fair comparison.

Evaluation Benchmarks. We primarily evaluated our models using three widely used open-ended instruction-following benchmarks: MT-Bench (Zheng et al., 2023), AlpacaEval 2 (Li et al., 2023)[4], and Arena-Hard v0.1 (Li et al., 2024). These benchmarks assess the models' versatile conversational capabilities across a diverse set of queries and are widely adopted by the research community. Concretely, AlpacaEval 2 comprises 805 questions from 5 datasets, while MT-Bench encompasses 8 categories with 80 questions. Arena-Hard, an enhanced version of MT-Bench[5], includes 500 rigorously defined technical problem-solving queries. For AlpacaEval 2, we used `alpaca_eval_gpt4_turbo_fn` as the annotator which has a higher human agreement and report both the raw win rate (WR) and the length-controlled win rate (LC) (Dubois et al., 2024), with the LC metric designed to be robust against model verbosity. For Arena-Hard, we reported the WR against the baseline model. For MT-Bench, we report the average MT-Bench score, using `GPT-4-Turbo-2024-04-09` as the judge model[6].

Baselines. We selected several advanced preference optimization baselines, including: IPO (Azar et al., 2023) is a theoretically grounded method that avoids DPO's assumption that pairwise preferences can be replaced with pointwise rewards. KTO (Ethayarajh et al., 2024) learns from non-paired preference data. ORPO (Hong et al., 2024) introduces a reference-model-free odds ratio term to directly contrast winning and losing responses with the policy model, jointly training with the SFT objective. SimPO (Meng et al., 2024) and SamPO (Lu et al., 2024) are both designed to address the issue of model verbosity by applying length normalization. We report details of the experiments in Appendix B. We meticulously tune the hyperparameters for each baseline and report the best performance. We observe that *many DPO variants do not empirically outperform standard DPO*.

AlpacaEval 2 annotator choice AlpacaEval 2 provides various evaluation templates and in the official readme recommends using `weighted_alpaca_eval_gpt4_turbo` as well as `alpaca_eval_gpt4_turbo_fn`. The former is the default annotator in AlpacaEval 2 with a human agreement rate of 65.7% and much cheaper price. In all of our evaluations, we used the latter as the annotator which has a higher agreement rate of 68.1% with human annotation data.

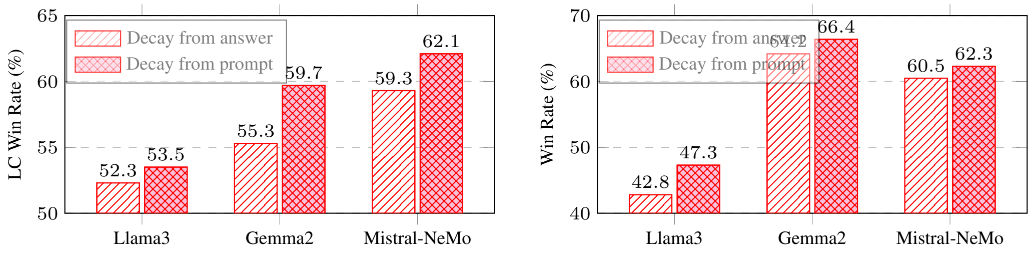
---

[3] The exact nature of the instruction-tuning (whether it includes SFT or the complete RLHF pipeline) of these models is not fully disclosed. For simplicity, we refer to these as SFT models.

[4] `https://tatsu-lab.github.io/alpaca_eval/`

[5] Adler et al. (2024) discussed the existence of incorrect reference answers in MT-Bench, therefore a corrected version of MT-Bench was used.

[6] `GPT-4-Turbo-2024-04-09` provides more accurate reference answers and judgments compared to GPT-4.



(a) AlpacaEval 2 (b) Arena-Hard

Figure 7: Visualization of the LC Win Rate (%) on three models under different decay mechanisms (answer-based and prompt-based decay) in the AlpacaEval 2 and Arena-Hard benchmarks.

B EXPERIMENTAL DETAILS

Considering DPO is a special case of D^{2}PO when $\gamma = 1.0$, to compare the effects of different decay coefficients, we conduct experiments with β fixed at 0.1. In addition, we follow the optimal hyperparameters claimed in SimPO and our code is built on LlamaFactory (Zheng et al., 2024). Across all DAAs run, the models were trained on 32 A100 with a global batch size of 128 (4 gradient accumulation steps). The hyperparameter search range of all methods are displayed in the Table 6.

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

Method	**Objective**	**Hyperparameter**
DPO (Rafailov et al., 2023)	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$	$\beta \in [0.01, 0.1]$
IPO (Azar et al., 2023)	$\left(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - \frac{1}{2\tau}\right)^2$	$\tau \in [0.01, 0.1, 0.5, 1.0]$
KTO (Ethayarajh et al., 2024)	$-\lambda_w\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - z_{\text{ref}}\right) + \lambda_l\sigma\left(z_{\text{ref}} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right),$ where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta\text{KL}\left(\pi_\theta(y\|x)\|\|\pi_{\text{ref}}(y\|x)\right)\right]$	$\lambda_l = \lambda_w = 1.0$ $\beta \in [0.01, 0.05, 0.1]$
SamPO (Lu et al., 2024)	$-\log\sigma\left(\sum_{t=0}^{T_m}\beta\log\frac{\pi_\theta\left(\mathbf{y}_w^t\|\mathbf{x},\mathbf{y}_w^{<t}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_w^t\|\mathbf{x},\mathbf{y}_w^{<t}\right)} - \sum_{t=0}^{T_m}\beta\log\frac{\pi_\theta\left(\mathbf{y}_l^t\|\mathbf{x},\mathbf{y}_l^{<t}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_l^t\|\mathbf{x},\mathbf{y}_l^{<t}\right)}\right)$ where $T_m = min(T_w, T_l), y^t \sim Uniform(T_m, y^T)$	$\beta \in [0.01, 0.1]$
ORPO (Hong et al., 2024)	$-\log p_\theta(y_w\|x) - \lambda\log\sigma\left(\log\frac{p_\theta(y_w\|x)}{1-p_\theta(y_w\|x)} - \log\frac{p_\theta(y_l\|x)}{1-p_\theta(y_l\|x)}\right),$ where $p_\theta(y\|x) = \exp\left(\frac{1}{\|y\|}\log\pi_\theta(y\|x)\right)$	$\lambda \in [0.1, 0.5, 1.0, 2.0]$
SimPO (Meng et al., 2024)	$-\log\sigma\left(\frac{\beta}{\|y_w\|}\log\pi_\theta(y_w\|x) - \frac{\beta}{\|y_l\|}\log\pi_\theta(y_l\|x) - \gamma\right)$	$\beta \in [2.5, 10]$ $\gamma \in [0.3, 0.5, 1.0]$
D^{2}PO	$-\log\sigma\left(\sum_{t=0}^{T_w}\gamma^t\beta\log\frac{\pi_\theta\left(\mathbf{y}_w^t\|\mathbf{x},\mathbf{y}_w^{<t}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_w^t\|\mathbf{x},\mathbf{y}_w^{<t}\right)} - \sum_{t=0}^{T_l}\gamma^t\beta\log\frac{\pi_\theta\left(\mathbf{y}_l^t\|\mathbf{x},\mathbf{y}_l^{<t}\right)}{\pi_{\text{ref}}\left(\mathbf{y}_l^t\|\mathbf{x},\mathbf{y}_l^{<t}\right)}\right)$	$\beta \in [0.1]$ $\gamma \in [0.95, 0.97, 0.98, 0.99]$

C MORE ANALYSES

When to Decay In our default setting, we apply decay from the beginning of the prompt rather than from the first generated tokens. Here, we investigate the implications of these two approaches. Theoretically, during the loss computation, both the chosen token y_w and the rejected token y_l share the same prompt prefix, which results in distinct initial scaling coefficients for the reward at the first generated position. For illustration, if the prompt length is l, then in our default setting, the reward for the first generated token is scaled by y_l while in the alternate setting, the scaling factor would be 1. Through results in Figure 7, we can see that both two settings achieve better results than DPO, and our default setting is much better than the other one. This indicates that proper scaling factor is very important during preference optimization.

Table 7: Three benchmarks results with on-policy setups, using gpt-4-1106-preview as the judge model. † denotes our reference-free version.

Method	**Llama3-Instruct (8B)**			**Gemma2-Instruct (9B)**			**Mistral-NeMo-Instruct (12B)**		
	AE2		**AH**	**AE2**		**AH**	**AE2**		**AH**
	WR (%)	**LC (%)**	**WR (%)**	**WR (%)**	**LC (%)**	**WR (%)**	**WR (%)**	**LC (%)**	**WR (%)**
SFT	31.6	31.7	19.7	37.7	48.2	39.9	40.8	44.2	39.7
DPO	41.7	42.9	31.2	46.4	53.1	47.4	53.4	52.6	47.4
KTO	35.7	37.7	25.2	46.5	52.3	49.2	45.9	49.7	45.4
IPO	40.1	43.2	25.2	49.1	48.3	49.5	51.7	52.9	**51.8**
SamPO	39.4	41.9	28.7	46.9	56.7	50.4	49.8	52.5	50.4
D^{2}PO (ours)	**44.5**	**50.1**	**34.1**	**59.3**	**62.3**	**58.4**	**55.0**	**60.6**	50.6
ORPO	31.5	32.5	20.9	39.8	48.2	41.3	40.1	44.5	41.2
SimPO	44.5	49.1	**33.1**	55.1	59.4	56.5	50.7	53.8	**51.0**
D^{2}PO† (ours)	**45.0**	**51.9**	33.0	**58.0**	**61.5**	**56.9**	**56.8**	**59.9**	49.4

Effect of Different Judge Models Here, we mainly evaluated our generated results via `GPT-4-Turbo-0409`, while previous work mainly used `GPT-4-preview-1106` instead. Results in Table 7 show that D^{2}PO delivers consistent performance gains in both two judge models.

Full evaluation results We present the full evaluation of AlpacaEval 2, Arena-Hard and MT-Bench in Table 8 and Table 9. The former is an off-policy setup, while the latter is an on-policy setup. For on-policy setups, we found that DPO can achieve better performance when $beta = 0.01$ and reported this result for fair comparison. Specifically, "-" indicates that the model suffered a collapse during training.

Table 8: Full results on benchmarks under off-policy setups.

Method	**AlpacaEval 2**			**Arena Hard**		**MT-Bench**
	Win Rate (%)	**LC Win Rate (%)**	**Len.**	**Win Rate (%)**	**Len.**	**G4-Turbo**
Llama3-Instruct (8B)						
SFT	39.05	40.13	1971	27.6	581	7.5
DPO	37.38	40.28	1880	27.7	546	**7.7**
KTO	33.29	38.06	1765	21.0	525	7.5
IPO	42.16	45.66	1845	31.9	542	7.6
SamPO	40.68	43.11	1891	26.1	550	7.5
D^2PO (γ=0.95)	45.90	44.78	2113	40.0	636	8.0
D^2PO (γ=0.97)	43.46	43.04	1994	37.0	602	7.7
D^2PO (γ=0.98)	42.73	44.10	1954	35.1	578	7.9
D^2PO (γ=0.99)	41.74	44.03	1912	30.4	560	8.0
D^2PO (γ=1.01)	38.50	40.21	1928	26.1	569	7.5
D^2PO (γ=1.02)	37.69	38.63	1955	26.8	572	7.4
ORPO	10.62	15.32	1386	6.8	764	6.3
SimPO	0.25	0.80	27	1.4	15	1.6
Gemma2-Instruct (9B)						
SFT	37.58	47.23	1566	44.1	608	8.3
DPO	38.81	48.83	1546	42.5	595	8.1
KTO	39.07	50.00	1530	43.8	540	8.3
IPO	41.04	50.03	1630	48.2	608	8.1
SamPO	39.86	50.06	1574	46.9	596	8.2
D^2PO (γ=0.95)	48.07	50.05	1929	53.4	657	8.5
D^2PO (γ=0.97)	45.34	49.70	1824	50.7	636	8.3
D^2PO (γ=0.98)	45.46	50.99	1746	50.2	625	8.3
D^2PO (γ=0.99)	42.10	50.05	1636	50.2	609	8.4
D^2PO (γ=1.01)	38.57	47.45	1577	43.7	612	8.3
D^2PO (γ=1.02)	-	-	-	-	-	-
ORPO	11.30	21.55	1182	10.2	641	7.1
SimPO	38.76	50.00	1508	31.6	475	8.0
Mistral-NeMo-Instruct (12B)						
SFT	44.60	47.71	1879	46.5	575	8.1
DPO	44.41	49.25	1821	48.5	569	8.3
KTO	37.39	48.68	1620	35.8	501	8.2
IPO	39.75	48.85	1634	39.8	506	8.2
SamPO	43.54	49.47	1784	50.1	562	8.1
D^2PO (γ=0.95)	52.17	52.42	2017	54.2	590	8.3
D^2PO (γ=0.97)	51.30	54.43	1879	51.8	562	8.4
D^2PO (γ=0.98)	49.57	55.43	1778	47.8	534	8.3
D^2PO (γ=0.99)	46.96	53.86	1770	45.9	538	8.0
D^2PO (γ=1.01)	43.65	47.60	1829	46.8	564	8.1
D^2PO (γ=1.02)	43.84	47.91	1840	45.6	572	8.0
ORPO	9.64	17.00	1185	9.8	640	6.9
SimPO	46.77	53.28	1704	46.6	500	8.0

Table 9: Full results on benchmarks under on-policy setups. † denotes our reference-free version.

Method	AlpacaEval2			Arena Hard		MT-Bench
	Win Rate (%)	**LC Win Rate (%)**	**Len.**	**Win Rate (%)**	**Len.**	**G4-Turbo**
Llama3-Instruct (8B)						
SFT	39.05	40.13	1971	27.6	581	7.5
DPO (β=0.01)	48.26	49.93	1937	45.2	568	7.8
KTO	42.36	44.77	1901	32.1	545	7.7
IPO	42.92	45.99	1889	34.5	553	7.9
SamPO	44.35	47.17	1890	35.8	536	8.0
D^{2}PO (γ=0.95)	48.13	51.53	1832	42.5	578	7.7
D^{2}PO (γ=0.97)	46.15	50.52	1739	41.6	549	7.7
D^{2}PO (γ=0.98)	47.39	53.54	1705	47.3	518	7.8
D^{2}PO (γ=0.99)	48.01	52.97	1739	44.0	514	7.8
DPO (β=0.1)	46.21	47.60	1971	42.4	627	7.9
D^{2}PO (γ=1.01)	37.25	38.32	1948	28.1	578	7.6
D^{2}PO (γ=1.02)	37.75	39.30	1942	26.2	566	7.6
ORPO	37.75	39.29	1934	25.5	615	7.7
SimPO	44.41	50.34	1704	41.9	477	7.8
D^{2}PO† (γ=0.98)	48.01	53.87	1726	46.1	526	7.7
Gemma2-Instruct (9B)						
SFT	37.58	47.23	1566	44.1	608	8.3
DPO (β=0.01)	54.53	57.05	1948	65.2	768	8.3
KTO	48.26	53.39	1775	57.1	705	8.3
IPO	50.86	50.00	2129	59.7	759	8.3
SamPO	45.78	55.21	1662	55.2	668	8.2
D^{2}PO (γ=0.95)	58.39	59.03	2034	65.5	739	8.5
D^{2}PO (γ=0.97)	56.83	59.25	1949	64.8	715	8.3
D^{2}PO (γ=0.98)	57.20	59.71	1950	66.4	724	8.3
D^{2}PO (γ=0.99)	53.98	57.38	1843	63.9	693	8.2
DPO (β=0.1)	47.02	53.43	1737	56.7	682	8.3
D^{2}PO (γ=1.01)	38.70	48.06	1592	43.7	610	8.2
D^{2}PO (γ=1.02)	-	-	-	-	-	-
ORPO	41.93	51.14	1647	45.3	641	8.2
SimPO	54.47	58.42	1871	65.0	744	8.3
D^{2}PO† (γ=0.98)	56.71	60.76	1894	65.7	687	8.3
Mistral-Nemo-Instruct (12B)						
SFT	44.60	47.71	1879	46.5	575	8.1
DPO (β=0.01)	58.76	57.29	2160	63.6	659	8.3
KTO	48.26	53.39	1775	57.1	705	8.3
IPO	50.86	50.00	2129	59.7	759	8.3
SamPO	45.78	55.21	1662	55.2	668	8.2
D^{2}PO (γ=0.95)	59.25	57.85	2167	60.7	665	8.6
D^{2}PO (γ=0.97)	56.77	58.65	1969	57.5	586	8.6
D^{2}PO (γ=0.98)	57.34	62.07	1853	62.3	546	8.6
D^{2}PO (γ=0.99)	54.29	58.83	1816	59.3	532	8.4
DPO (β=0.1)	53.48	53.32	2081	59.0	624	8.4
D^{2}PO (γ=1.01)	45.34	48.06	1908	44.2	581	8.0
D^{2}PO (γ=1.02)	-	-	-	-	-	-
ORPO	41.93	51.14	1647	45.3	641	8.2
SimPO	54.47	58.42	1871	65.0	744	8.3
D^{2}PO† (γ=0.98)	56.71	60.76	1894	65.7	687	8.3

Stronger Instruct model To verify the robustness properties of our method, we conducted experiments under on-policy setups based on a stronger Instruct model, Gemma2-Instruct (27B). Considering the limited computing resources, we only compared D^{2}PO with DPO and SimPO. Table 10 shows the evaluation results on three benchmarks, demonstrating that our method maintains a certain advantage over stronger models.

Table 10: Gemma2-Instruct (27B) results under on-policy setups

Method	**AlpacaEval2**			**Arena Hard**		**MT-Bench**
	Win Rate (%)	**LC Win Rate (%)**	**Len.**	**Win Rate (%)**	**Len.**	**G4-Turbo**
SFT	39.68	48.13	1633	60.7	648	8.5
DPO (β=0.1)	48.45	53.66	1786	71.2	715	8.5
DPO (β=0.01)	57.39	59.05	1990	78.6	811	**8.6**
SimPO	54.29	**61.30**	1748	75.3	682	8.5
D^{2}PO (γ=0.98)	**57.52**	60.52	1893	**79.8**	708	**8.6**

D CASE ANALYSIS

We list two concrete cases from AlpacaEval2 to compare D^{2}PO with DPO. Regarding the first case in Figure 8, the DPO model's output overthinks and mistakenly assumes that the stopover time is already included in the total travel time in the second step of its analysis, resulting in the omission of this time in the final calculation. In contrast, the model trained with our method succinctly lists the times and correctly sums them to obtain the total travel time. Figure 9 shows the second case: given the question, "What is the largest ocean in the world?", the DPO model provides a response without explanation. In contrast, the D^{2}PO model produces a well-structured answer with detailed explanations, making the answer more credible.

Figure 8: A case examples of AlpacaEval2, generated by Gemma2-Instruct (9B).

Input	→	`Q: What is the largest ocean in the world?`
Gemma2-9b-it DPO	→	`The largest ocean in the world is the` **`Pacific Ocean`**`.`

Gemma2-9b-it D^{2}PO	→	`The` **`Pacific Ocean`** `is the largest ocean in the world.`
		`Here's why:`
		`*` **`Surface Area:`** `It covers over 30% of the Earth's surface, roughly 63.8 million square miles (165.25 million square kilometers).`
		`*` **`Volume:`** `It holds about half of the Earth's total water.`
		`*` **`Depth:`** `The Pacific Ocean boasts the deepest point on Earth, the Challenger Deep in the Mariana Trench, which plunges over 36,000 feet (10,900 meters) below sea level.`
		`Let me know if you'd like to know more about the Pacific Ocean or other oceans!`

Figure 9: A case examples of AlpacaEval2, generated by Mistral-Nemo-Instruct (12B).

Input →

Q: if it takes 8 hours to fly to Singapore, stop over for an hour and a half and then fly 14 hours to London then what is the total travel time if the plane is delayed in Singapore?

Mistral-Nemo DPO →

To calculate the total travel time including the delay in Singapore, we need to add up all the components of the journey:

1. Flight to Singapore: 8 hours
2. Stopover in Singapore: 1.5 hours (which is already included in the total time as it's part of the journey, not a delay)
3. Delay in Singapore: Let's assume the delay is 'D' hours.
4. Flight from Singapore to London: 14 hours

The total travel time including the delay would be:

Total travel time = Flight to Singapore + Delay in Singapore + Flight to London
Total travel time = 8 hours + D hours + 14 hours
Without knowing the exact duration of the delay (D), we can express the total travel time as:
Total travel time = 22 hours + D

For example, if the delay in Singapore is 2 hours, the total travel time would be:

Total travel time = 22 hours + 2 hours = 24 hours
So, the total travel time including the delay is 22 hours plus the duration of the delay (D).

Mistral-Nemo D^{2}PO →

To calculate the total travel time including the delay in Singapore, let's break down the journey and add the delay:

Total initial travel time without delay:
- Flight to Singapore: 8 hours
- Stopover in Singapore: 1.5 hours
- Flight from Singapore to London: 14 hours

Total without delay = 8 + 1.5 + 14 = 23.5 hours

Now, let's assume the plane is delayed in Singapore by 'D' hours. The delay will only affect the stopover time, so we add 'D' to the stopover duration:

Adjusted stopover time with delay = 1.5 hours + D hours
Now, the total travel time with the delay is:
Total with delay = 8 hours (flight to Singapore) + (1.5 + D) hours (stopover with delay) + 14 hours (flight to London)
Total with delay = 8 + (1.5 + D) + 14
Total with delay = 23.5 + D hours
Without knowing the specific duration of the delay (D), we can express the total travel time with the delay as:
Total travel time with delay = 23.5 hours + Delay in Singapore (D hours)

E THEORETICAL ANALYSIS

In this section, we provide detailed derivation of the upper bound for

$$\begin{aligned}\text{SubOpt}(\pi, s; \gamma_e) &= V_{\gamma_e}^{\pi^*}(s) - V_{\gamma_e}^{\pi}(s) \\ &= \underbrace{\left[V_{\gamma_e}^{\pi^*}(s) - V_{\gamma}^{\pi^*}(s)\right]}_{\Delta_1} + \underbrace{\left[V_{\gamma}^{\pi^*}(s) - V_{\gamma}^{\pi}(s)\right]}_{\Delta_2} + \underbrace{\left[V_{\gamma}^{\pi}(s) - V_{\gamma_e}^{\pi}(s)\right]}_{\Delta_3}\end{aligned}$$

E.1 THE UPPER BOUND OF Δ_1 AND $\Delta 3$

Noting that Δ_1 and Δ_3 both capture the difference in the expected returns of the same policy when evaluated under different γ, we can analyze the upper bound of these two items together.

The term Δ_1 is given by with $\gamma^e = 1.0$:

$$\begin{aligned}\Delta_1 &= V_{\gamma_e}^{\pi^*}(s) - V_{\gamma}^{\pi^*}(s) \\ &= \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{H-1} \gamma_e^t r(s_t, a_t) - \sum_{t=0}^{H-1} \gamma^t r(s_t, a_t)\right] \\ &= \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{H-1} (1 - \gamma^t) r(s_t, a_t)\right].\end{aligned}$$

Assuming the rewards are bounded, i.e., $|r(s, a)| \leq R$, we have:

$$\Delta_1 \leq \sum_{t=0}^{H-1} (1 - \gamma^t) R = (H - \frac{1 - \gamma^H}{1 - \gamma}) R. \tag{15}$$

Similarly, we can obtain the upper bound of Δ_3:

$$\Delta_3 \leq \sum_{t=0}^{H-1} (1 - \gamma^t) R = (H - \frac{1 - \gamma^H}{1 - \gamma}) R. \tag{16}$$

E.2 THE UPPER BOUND OF Δ_2

Lemma 1 Performance Difference Lemma with finite horizon H (Kakade & Langford, 2002)

$$V_{\gamma}^{\pi^*}(s) - V_{\gamma}^{\pi}(s) = \frac{1 - \gamma^H}{1 - \gamma} E_{s \sim d^{\pi^*}} \left[\Sigma_{a \in A} (\pi^*(a \mid s) - \pi(a \mid s)) Q^{\pi}(s, a)\right] \tag{17}$$

where π^* represents optimal policy and π represents policy.

Based on the assumption that the rewards are bounded, i.e., $|r(s, a)| \leq R$, we have:

$$Q^{\pi}(s, a) \leq \sum_{t=0}^{H-1} \gamma^t R = \frac{1 - \gamma^H}{1 - \gamma} R \tag{18}$$

Finally, we can get

$$\begin{aligned}\Delta_2 &= V_{\gamma}^{\pi^*}(s) - V_{\gamma}^{\pi}(s) && (19) \\ &\leq \frac{(1 - \gamma^H)^2}{(1 - \gamma)^2} E_{s \sim d^{\pi^*}} \left[\sum_{a \in A} (\pi^*(a \mid s) - \pi(a \mid s)\right] R && (20) \\ &= \frac{2(1 - \gamma^H)^2}{(1 - \gamma)^2} E_{s \sim d^{\pi^*}} \left[\mathbb{TV}(\pi^*(a|s) || \pi(a|s)\right] R && (21)\end{aligned}$$

E.3 SUBOPTIMAL ANALYSIS

Adding the bounds on Δ_1, Δ_2, and Δ_3, we obtain:

$$\text{SubOpt}(\pi, s; \gamma_e) \leq 2(H - \frac{1-\gamma^H}{1-\gamma})R + \frac{2(1-\gamma^H)^2}{(1-\gamma)^2} E_{s\sim d^{\pi^*}} \left[\mathbb{TV}(\pi^*(a|s)||\pi(a|s)\right] R \tag{22}$$

Since both terms vary monotonically with the temporary decay factor γ but in opposing directions, this implies the existence of an optimal trade-off value, denoted as γ^*, within the interval (0, 1).