Towards Bridging the Reward-Generation Gap in Direct Alignment Algorithms

Zeguan Xiao[1], **Yun Chen**[1], **Jian Yang**[2], **Guanhua Chen**[3*], **Ke Tang**[3*]
[1]Shanghai University of Finance and Economics, [2]Beihang University
[3]Southern University of Science and Technology

Abstract

Direct Alignment Algorithms (DAAs), such as Direct Preference Optimization (DPO) and Simple Preference Optimization (SimPO), have emerged as efficient alternatives to Reinforcement Learning from Human Feedback (RLHF) algorithms for aligning large language models (LLMs) with human preferences. However, DAAs suffer from a fundamental limitation we identify as the "reward-generation gap"—a discrepancy between training objectives and autoregressive decoding dynamics. In this paper, we consider that one contributor to the reward-generation gap is the mismatch between the inherent importance of prefix tokens during the LLM generation process and how this importance is reflected in the implicit reward functions of DAAs. To bridge the gap, we adopt a token-level MDP perspective of DAAs to analyze its limitations and introduce a simple yet effective approach called **P**refix-**O**riented **E**qual-length **T**raining (POET), which truncates both preferred and dispreferred responses to match the shorter one's length. We conduct experiments with DPO and SimPO, two representative DAAs, demonstrating that POET improves over their standard implementations, achieving up to 11.8 points in AlpacaEval 2 and overall improvements across downstream tasks. These results underscore the need to mitigate the reward-generation gap in DAAs by better aligning training objectives with autoregressive decoding dynamics.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023; AI@Meta, 2024) have demonstrated remarkable capabilities across a wide range of tasks, including instruction following (Zhou et al., 2023), mathematical problem solving (Cobbe et al., 2021; Shao et al., 2024), and coding generation (Chen et al., 2021; Roziere et al., 2023). An important step in the training of LLMs is "alignment", which refers to aligning LLMs with human intentions and values, ensuring they are helpful, honest, and harmless (Askell et al., 2021). Learning from human feedback plays a crucial role in LLM alignment, and a popular paradigm for this is Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022; Bai et al., 2022). The RLHF pipeline typically involves a three-stage process: supervised fine-tuning, reward modeling, and policy optimization. While effective, RLHF faces challenges including training instability, computational inefficiency, and high sensitivity to hyperparameters (Zheng et al., 2023b; Santacroce et al., 2023), motivating the exploration of alternative approaches.

To simplify the alignment process, researchers have developed Direct Alignment Algorithms (DAAs) (Rafailov et al., 2024a), such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and Simple Preference Optimization (SimPO) (Meng et al., 2024). DAAs bypass the

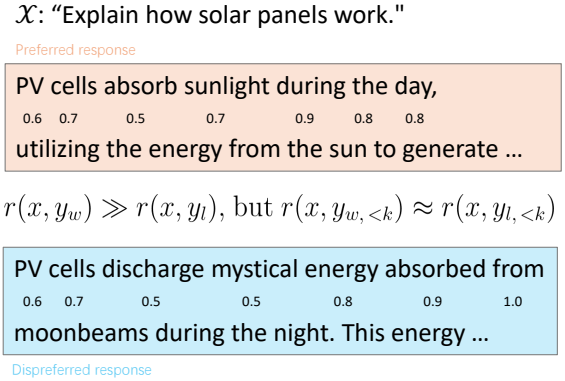


Figure 1: During autoregressive generation, LLMs generate tokens sequentially from left to right. Although DAAs are optimized to ensure $r(x, y_w) \gg r(x, y_l)$ over entire sequences, they do not guarantee that $r(x, y_{w,<k}) \gg r(x, y_{l,<k})$ for prefixes.

* Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14534–14548
July 2-7, 2026 ©2026 Association for Computational Linguistics

need for explicit reward modeling and reinforcement learning in RLHF, instead directly performing preference optimization based on a reference dataset. The simplicity and effectiveness of DAAs have made them an attractive alternative to RLHF.

The DAAs are optimized to increase the reward of preferred responses while reducing the reward of dispreferred ones (Ethayarajh et al., 2024; Azar et al., 2023; Zhao et al., 2023), where some can be expressed as implicit reward functions based on different manifestations of likelihoods (Rafailov et al., 2023; Meng et al., 2024). However, a known issue with DAAs is that they may decrease the reward of preferred responses as long as the reward margins between preferred and dispreferred responses increase (Rafailov et al., 2023; Pal et al., 2024). More importantly, Shi et al. (2024) found that neither a higher reward of preferred responses nor larger reward margins necessarily lead to better performance, suggesting a deeper issue in how DAAs connect optimization objectives to LLM's performance.

In this paper, we identify this issue as the **Reward-Generation Gap in DAAs**—a fundamental misalignment between DAAs' training objectives and the autoregressive decoding dynamics of LLMs. To bridge this gap, we adopt a token-level MDP perspective of DAAs to analyze their limitations. Our subsequent theoretical and empirical analysis provides the motivation for our proposed method. Building on these insights, we propose a simple yet effective data augmentation approach, **P**refix-**O**riented **E**qual-length **T**raining (POET), which truncates both preferred and dispreferred responses in each sample to match the shorter one's length, resulting in diverse truncated lengths across samples. Training with POET, the optimization of DAAs' objective is implicitly constrained to converge across all timesteps of token-level MDP.

We conduct extensive experiments with DPO and SimPO under various experimental settings, demonstrating that POET consistently improves over their standard implementations. POET achieves up to 11.8 points gain on AlpacaEval 2 (Li et al., 2023; Dubois et al., 2024). Experiments on downstream tasks also show overall improvements across various benchmarks. Our analysis further reveals that POET effectively bridges the reward-generation gap by generating better prefixes.[1]

[1]Our code is publicly available at https://github.com/sustech-nlp/POET.

2 Reward-Generation Gap in DAAs

2.1 Background

In RLHF (Ouyang et al., 2022; Stiennon et al., 2020; Li et al., 2016; Ziegler et al., 2019), LLM learns a policy π_θ, and $\pi_\theta(y \mid x)$ represents the probability assigned by the LLM to a response y given an input prompt x. During the reinforcement learning (RL) phase, the optimization objective is to maximize the expected reward while preventing the policy π_θ from deviating too far from the reference policy π_{ref}:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y)\right] - \beta\, \mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x) \| \pi_{\text{ref}}(y \mid x)\right], \tag{1}$$

where r_ϕ is the reward model, typically trained as a Bradley-Terry (BT) model using a static preference dataset $\mathcal{D}$ of triples (x, y_w, y_l). In each triple, x denotes the input prompt, y_w the preferred response, and y_l the dispreferred response. The reward model receives the generated response and provides a reward signal $r_\phi(x, y)$ indicating the quality of the response.

Alternatively, DAAs (Rafailov et al., 2024a) eliminate the need for explicit reward modeling and reinforcement learning, directly performing preference optimization on a static preference dataset. The optimization objectives of DAAs contain rewards implicitly defined by π_θ, where some rewards are based directly on the likelihood (Zhao et al., 2023; Xu et al., 2024), while others are derived from different manifestations of likelihoods, such as DPO (Rafailov et al., 2023) and SimPO (Meng et al., 2024):

$$r_{\text{DPO}}(x,y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \tag{2}$$

$$r_{\text{SimPO}}(x,y) = \frac{\beta}{|y|} \log \pi_\theta(y \mid x), \tag{3}$$

where $r_{\text{DPO}}(x, y)$ and $r_{\text{SimPO}}(x, y)$ are the implicit reward functions of DPO and SimPO, respectively.

2.2 Reward-Generation Gap in DAAs

Designed to increase the reward of preferred responses y_w while decreasing the reward of dispreferred responses y_l, DAAs converge if the reward margins between preferred and dispreferred responses grow sufficiently large. However, a known issue with DAAs is that they may decrease the reward of preferred responses as long as the margins

increase (Rafailov et al., 2023; Pal et al., 2024). Furthermore, Shi et al. (2024) finds that contrary to expectations, neither a higher reward of preferred responses nor larger reward margins between the preferred and dispreferred responses necessarily lead to better performance.

The disconnect identified above arises from the fact that the implicit rewards of DAAs do not represent the quality of responses, but are rather different manifestations of their likelihoods, such as Eq. 2 and Eq. 3. However, due to the foundational discrepancy between the maximum/minimum likelihood training and autoregressive generation in language models (Ranzato et al., 2015; Zhang et al., 2021; Arora et al., 2022), optimizing for the DAAs training objectives does not necessarily lead to better generation quality. This discrepancy creates a substantial misalignment between training objectives and autoregressive decoding dynamics—a phenomenon we refer to as the **Reward-Generation Gap in DAAs**.

We consider that one contributor to this gap is the mismatch between the inherent importance of prefix tokens during the LLM generation process and how this importance is reflected in the implicit reward functions of DAAs. As shown in Fig. 4(a) in Appendix A, the token-level entropies (i.e., uncertainty) are significantly higher in the early positions, and gradually decrease as more context becomes available. As demonstrated by Arora et al. (2022), errors in autoregressive generation tend to accumulate, with early mistakes propagating and amplifying through the sequence. This phenomenon, known as exposure bias, means that suboptimal choices in prefix tokens can severely degrade the quality of the entire response. However, DAAs' reward functions assign equal weight to every token, ignoring these positional differences. As illustrated in Fig. 4(b), the log probability of early tokens is significantly lower than that of later tokens, yet their contribution to DAAs' implicit reward is diluted by the overwhelming number of subsequent tokens. This is highlighted by the large gap between the early tokens' log probabilities and the mean log probability across all positions, which is a key component of the implicit reward.

3 How to Bridge the Gap?

3.1 Token-level MDP perspective of DAAs

Most classical RLHF approaches formulate the preference optimization problem as a contextual bandit problem, where the dataset $\mathcal{D} = \left\{\left(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}\right)\right\}_{i=1}^{N}$ of language prompts $\mathbf{x}$ and target answers $\mathbf{y}$, each of which can be broken down into a sequence of tokens, for example $\mathbf{x} = (x_0, \ldots, x_m)$, from a fixed discrete vocabulary $\mathcal{A}$. In this formulation, the entire response $\mathbf{y}$ is treated as a single action, and the reward is computed based on the entire response. DAAs, such as DPO and SimPO, stay entirely within the contextual bandits setting, but can theoretically be cast into the token-level MDP (Rafailov et al., 2024b).

In the perspective of token-level MDP of DAAs, the state $\mathbf{s}$ consists of all tokens generated so far (i.e., $\mathbf{s}_t = (x_0, \ldots, x_m, y_0, \ldots, y_t)$), and the action a is to select a single token y_{t+1} from the vocabulary $\mathcal{A}$. The corresponding Bradley-Terry (BT) model is therefore defined as follows:

$$p^*\left(y_w \succeq y_l\right) = \frac{A_w}{A_w + A_l} \tag{4}$$

where $A_w = \exp\left(\sum_{i=1}^{N} r\left(\mathbf{s}_i^w, \mathbf{a}_i^w\right)\right)$ and $A_l = \exp\left(\sum_{i=1}^{M} r\left(\mathbf{s}_i^l, \mathbf{a}_i^l\right)\right)$. $r\left(\mathbf{s}_i^w, \mathbf{a}_i^w\right)$ and $r\left(\mathbf{s}_i^l, \mathbf{a}_i^l\right)$ are the token-level rewards for each action in the preferred and dispreferred trajectory, respectively, and $p^*\left(y_w \succeq y_l\right)$ gives the probability that the preferred trajectory y_w is better than the dispreferred trajectory y_l.

Although token-level DPO theoretically enables the derivation of an optimal policy π^* for the underlying MDP of Eq. 1 (Rafailov et al., 2024b)[2], it is challenging to train π^* in practice, as we only possess sparse reward signals in the form of sequence-level labels indicating that y_w is preferred over y_l. Furthermore, the sequence-level BT model does not model the preference of sub-trajectories, failing to capture the convergence behavior of partial sequences, which leads to the issue illustrated in Figure 1.

3.2 Theoretical Basis of Proposed POET

Inspired by the limitations of DAAs from the token-level MDP perspective, we introduce a formal theoretical basis for our approach. Our key insight is that optimizing policies on equal-length sub-trajectories yields the same optimal policy as sequence-level optimization, while providing better supervision for crucial early tokens.

[2]Other DAAs can also be formulated to yield an optimal policy for an MDP; however, the underlying MDP for them is distinct from that defined by Eq. 1.

We begin by defining the equal-length sub-trajectories BT model:

$$p_k^*\left(y_{w,\leq k} \succeq y_{l,\leq k}\right) = \frac{E_w}{E_w + E_l} \tag{5}$$

where $E_w = \exp\left(\sum_{t=0}^{k} r\left(\mathbf{s}_t^w, \mathbf{a}_t^w\right) + V^*(\mathbf{s}_{k+1}^w)\right)$ and $E_l = \exp\left(\sum_{t=0}^{k} r\left(\mathbf{s}_t^l, \mathbf{a}_t^l\right) + V^*(\mathbf{s}_{k+1}^l)\right)$. $y_{w,\leq k} \succeq y_{l,\leq k}$ indicates that the sub-trajectory $y_{w,\leq k}$ is preferred over $y_{l,\leq k}$ when the cumulative reward plus the state value of the resulting state for $y_{w,\leq k}$ exceeds that of $y_{l,\leq k}$. Here, V^* represents the optimal state-value function under the MDP defined by Eq. 1.

Theorem 1. *The policy derived from the optimal equal-length sub-trajectory BT model is equivalent to the optimal policy derived from the original sequence-level BT model defined in Eq. 4 for DPO.*

The proof strategy is to show that the optimal equal-length sub-trajectories BT model can be expressed in terms of the optimal policy from the original sequence-level optimization, demonstrating that the two approaches yield equivalent optimal policies. The detailed proof is provided in Appendix B[3].

3.3 Feasibility of Equal-Length Preference Training

While Theorem 1 establishes that the optimal policy is preserved unconditionally when optimizing the equal-length sub-trajectory BT model, in practice we face two challenges: (1) existing preference datasets only provide full-sequence preference labels indicating that $y_w \succ y_l$, rather than off-the-shelf equal-length preference data; (2) we cannot directly compute the optimal state-value function V^* required by the BT model. In this section, we empirically investigate the feasibility of directly reusing existing full-sequence preference data for training.

We begin with a central observation: **the quality of a response to be generated heavily depends on the quality of its initial prefix**. Formally, we denote a prefix of length k as $y_{\leq k}$ and define the quality of $y_{\leq k}$ as the expected quality of complete responses given it:

$$Q(y_{\leq k}) = \mathbb{E}_{y_{>k} \sim \pi_\theta(y_{>k}|x, y_{\leq k})}\left[r_\theta(x, y_{\leq k} \oplus y_{>k})\right], \tag{6}$$

where π_θ is an oracle policy, r_θ is the oracle reward model that evaluates the quality of responses, $y_{>k}$ denotes the completion given prefix $y_{\leq k}$, and $\oplus$ represents the concatenation operation.

Experimental setup. We randomly select 1,000 samples from the training set of UltraFeedback (Cui et al., 2024). For each pair of responses (y_w, y_l) in these samples, we truncate both responses at different positions k to obtain prefixes of varying lengths, and then generate multiple completions from these prefixes. We use two models as proxy policy $\hat{\pi}_\theta$: Zephyr (Tunstall et al., 2023) and Llama-3-Base-8B-SFT (Meng et al., 2024). We quantify the prefix quality difference by $\Delta Q(k) = Q(y_{w,\leq k}) - Q(y_{l,\leq k})$ at different prefix lengths k, utilizing ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024) as proxy reward model.

Results. As shown in Figure 2, the prefix quality gap emerges very early and grows as the prefix length increases, with diminishing marginal gains at longer lengths. Specifically, the marginal increase $\Delta Q(k+1) - \Delta Q(k)$ decreases substantially as k increases and becomes negligible for sufficiently large k, indicating that $\Delta Q(k)$ approaches a plateau. This convergence behavior implies that for sufficiently large k, the difference $V^*(\mathbf{s}_{k+1}^w) - V^*(\mathbf{s}_{k+1}^l)$ becomes negligible, confirming that the full-sequence preference ordering is preserved after equal-length truncation.

From the token-level MDP perspective, the majority of reward is obtained in the early timesteps, after which the trajectory enters a stationary phase and the incremental reward difference from subsequent actions diminishes significantly.

3.4 Prefix-Oriented Equal-length Training

We introduce POET, a straightforward and effective method designed to prompt DAAs to capture the convergence behavior of partial sequences, thereby bridging the reward-generation gap.

Motivated by these theoretical and empirical findings, we propose POET to approximate the optimization of the equal-length sub-trajectories BT model. Concretely, given a pair of preferred and dispreferred responses (y_w, y_l) with lengths $|y_w|$ and $|y_l|$, we truncate both responses to the length of the shorter one, denoted as $k = \min(|y_w|, |y_l|)$, resulting in truncated responses $y_{w,\leq k}$ and $y_{l,\leq k}$. By Theorem 1, optimizing the equal-length sub-trajectory BT model yields the same optimal policy

[3]The proof for SimPO follows a similar structure, but with a different expression of the BT model using the optimal policy.

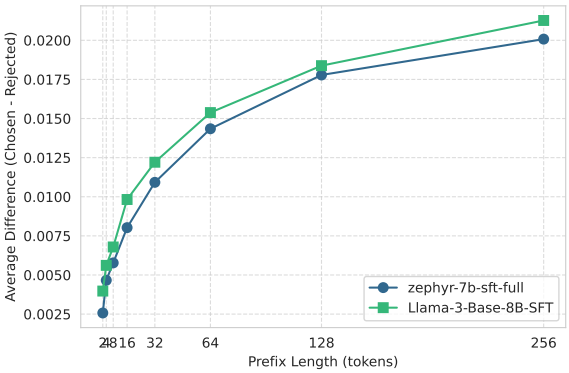


Figure 2: Average prefix quality difference between preferred and dispreferred responses at different prefix lengths. The results demonstrate that quality differences emerge early in the prefixes and increase with prefix length, though with diminishing marginal gains at longer lengths.

as full-sequence optimization. In practice, since $k = \min(|y_w|, |y_l|)$, one response remains complete while only the suffix of the longer one is discarded. As shown empirically in Section 3.3, the quality ranking is preserved after truncation with high consistency (Table 1), validating the use of full-sequence preference labels for the truncated equal-length pairs.

Training with POET, where both responses in each sample are equal in length and diverse lengths across samples, the optimization of DAAs' objective is implicitly constrained to converge across timesteps of token-level MDP, thus providing finer-grained reward signals of sub-trajectories and paying more attention to prefix tokens than the standard DAAs.

We summarize three advantages of POET:

- **Universal compatibility:** POET is compatible with any DAAs, requiring no modifications to their optimization objectives. This allows for seamless integration with both current and future DAAs, making POET a versatile enhancement to DAAs.
- **Hyperparameter-free:** By using the shorter response's length as a natural truncation point, POET requires no additional hyperparameters. This makes implementation straightforward and eliminates the need for additional hyperparameter tuning.
- **Minimal data noise risk:** There is a possibility that after truncation, the originally preferred response might become inferior to the dispreferred one, i.e., $Q(y_{w,\leq k}) < Q(y_{l,\leq k})$. POET inherently minimizes the risk of introducing noise into training data. By truncating to the shorter response's length, we ensure that the risk only comes from the truncated suffix of the longer response, which is less critical to the overall response quality according to our empirical analysis in Figure 2. We verify the risk empirically in three settings and report the results in Table 1. Specifically, we get $y_{w,\leq k}$ and $y_{l,\leq k}$ by POET, generate completions from the truncated prefix, and evaluate whether the quality ranking between the completion and the untruncated shorter response is preserved, using ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024) as the reward model. As shown in Table 1, the quality consistency before and after truncation is high across all three settings: 98.5% in the Llama-3-8B-Instruct v0.2 (on-policy) (Meng et al., 2024) setting, 93.8% in the v0.1 (on-policy) setting (lower due to a weaker reward model for annotation), and 91.4% in the Zephyr + UltraFeedback (off-policy) setting, where Zephyr's behavior differs from the data-generating model, making it a less ideal proxy policy.

Setting	Consistency
Zephyr + UltraFeedback	91.4%
Llama-3-8B-Instruct v0.1	93.8%
Llama-3-8B-Instruct v0.2	98.5%

Table 1: Quality ranking consistency before and after truncation across three settings.

4 Experiments

4.1 Experimental Setup

Models and training settings. Following Meng et al. (2024), we perform preference optimization under two setups: Base and Instruct. For the **Base setup**, we use either Zephyr (Tunstall et al., 2023) or Llama-3-Base-8B-SFT (Meng et al., 2024) as the starting point for preference optimization. These models are trained on the UltraChat-200k dataset (Ding et al., 2023) and are based on Mistral-7B-v0.1 (Jiang et al., 2023) and Meta-Llama-3-8B (AI@Meta, 2024), respectively. The preference optimization is performed on the UltraFeedback dataset (Cui et al., 2024), which contains approximately 61K human preference triples. For the **Instruct setup**, we use either Meta-Llama-3-

Method	Mistral-Base (7B)				Llama-3-Base (8B)			
	AlpacaEval 2			Arena-Hard	AlpacaEval 2			Arena-Hard
	LC (%)	WR (%)	Length	WR (%)	LC (%)	WR (%)	Length	WR (%)
SFT	5.3	5.3	931	2.6	6.1	4.0	976	6.0
DPO	12.9	10.6	1569	11.2	16.9	14.4	1644	18.6
+ POET	**24.7**	**17.7**	1401	**14.6**	**28.4**	**21.4**	1403	**25.5**
SimPO	20.0	18.0	1777	16.0	28.0	25.3	1777	13.5
+ POET	**24.2**	**23.7**	1939	**17.4**	**33.8**	**32.3**	1964	**14.8**
Method	**Llama-3-Instruct v0.2 (8B)**				**Gemma-2-Instruct (9B)**			
	AlpacaEval 2			Arena-Hard	AlpacaEval 2			Arena-Hard
	LC (%)	WR (%)	Length	WR (%)	LC (%)	WR (%)	Length	WR (%)
SFT	28.4	27.4	1932	19.0	52.6	39.9	1558	45.4
DPO	65.9	**63.3**	1998	**36.3**	78.4	**76.1**	2026	61.2
+ POET	**70.4**	58.2	1615	29.1	**79.7**	72.2	1756	**61.7**
SimPO	68.1	**62.4**	1805	**29.9**	78.5	73.7	1854	58.6
+ POET	**70.1**	57.9	1609	25.7	**80.1**	**74.5**	1860	**61.6**

Table 2: Instruction-following evaluation results. LC and WR denote length-controlled and raw win rate, respectively. Length indicates the average length of the responses.

8B-Instruct (AI@Meta, 2024) or gemma-2-9b-it model (Team et al., 2024) as the starting point for preference optimization. The responses in the preference datasets for this setup are regenerated by these models using the prompts from the UltraFeedback dataset, bringing the setup closer to an *on-policy* setting. In our experiments, we reuse the preference datasets from Meng et al. (2024).

As highlighted in Meng et al. (2024), tuning hyperparameters is critical for achieving optimal performance with all DAAs. Accordingly, in our experiments, we follow the choices of hyperparameters as described in Meng et al. (2024); further details are provided in Appendix D.

Baselines. We apply POET to two established DAAs: DPO (Rafailov et al., 2023) and SimPO (Meng et al., 2024), which represent reference-based and reference-free DAAs, respectively. Our baselines consist of models trained with these same DAAs, establishing a direct comparison to isolate the impact of POET. Additionally, we include SFT models as reference points to assess the relative improvements of all preference optimization methods.

Evaluations. We primarily evaluate models along two distinct dimensions of capability: instruction-following ability and performance across diverse downstream tasks. **For instruction-following evaluation**, we utilize two widely-adopted benchmarks by the community: AlpacaEval 2 (Li et al., 2023) and Arena-Hard v0.1 (Li et al., 2024). AlpacaEval 2 provides 805 diverse instructions from 5 datasets, while Arena-Hard v0.1 consists of 500 well-defined technical problem-solving queries. For AlpacaEval 2, we report both raw win rate (WR) and length-controlled win rate (LC) (Dubois et al., 2024), with the latter metric designed to control for potential bias from model verbosity. The LC metric has a Spearman correlation of 0.98 with ChatBot Arena (Zheng et al., 2023a), compared to 0.93 for the WR, making it a more reliable metric for evaluating instruction-following performance. For Arena-Hard, we report the win rate (WR). **For assessing downstream task capabilities**, we utilize the comprehensive suite of benchmarks from the Huggingface Open Leaderboard (Beeching et al., 2023), including tasks such as GSM8K (Cobbe et al., 2021), MMLU (Hendrycks et al., 2021), and others. We report the average performance across all tasks. The details of the evaluations can be found in Appendix C. We also conduct experiments on the safety alignment task, observing substantial improvements in safety rates (Section 4.5).

4.2 Main results

POET **consistently improves the instruction-following performance.** As shown in Table 2, POET consistently improves AlpacaEval 2 LC of both DPO and SimPO without introducing any additional hyperparameters or requiring further hyperparameter tuning. Most notably, POET achieves a substantial improvement of 11.8 points in AlpacaEval 2 LC for Mistral-Base-7B-DPO. These consistent improvements across all experimental settings demonstrate the robustness and effectiveness of POET. In some settings, we observe that while POET improves the LC, the WR decreases. This contradiction can be attributed to the known bias toward verbosity (Dubois et al., 2023; Singhal et al., 2023). As shown in Table 2, settings where POET achieves lower WR typically exhibit reduced average response length compared to their baseline counterparts. The WR might favor longer generations due to the absence of a length penalty. Importantly, in most settings, improvements are maintained across both the LC and WR metrics, suggesting that POET genuinely enhances response quality rather than merely exploiting metric-specific characteristics.

POET **does not increase the alignment tax on downstream tasks.** An important consideration for preference optimization methods is the *alignment tax*—the potential degradation of general capabilities as a side effect of alignment training. To verify that POET does not exacerbate this issue, we evaluate model performance across various downstream tasks (Table 7). Comparing DPO and SimPO models with and without POET, we observe that POET does not degrade general capabilities and even yields modest but consistent improvements in overall performance in most settings. We attribute these moderate gains to the generation of high-quality prefixes in downstream tasks, aligning with findings of Ji et al. (2025) on the role of prefixes in complex reasoning tasks. Note that these downstream tasks are not targeted by preference optimization; rather, this evaluation serves to confirm that POET's gains in instruction-following do not come at the cost of general task performance.

4.3 Ablation Studies

To further investigate the impact of the key components of POET, we conduct experiments on the Mistral-Base (7B) setting **to demonstrate that simply truncating responses to shorter lengths does not necessarily lead to better performance; instead, the equal-length truncation strategy is crucial for achieving significant gains.**

Method	Length Strategy	25%	50%	75%	100%
DPO	Original Len.	14.1	17.2	16.2	12.9
	POET Len.	23.5	24.9	26.7	24.7
SimPO	Original Len.	12.5	17.0	11.9	20.0
	POET Len.	19.9	26.1	24.5	24.2

Table 3: Results of the ablation studies. Darker colors indicate higher AlpacaEval 2 LC scores.

We compare two truncation strategies: (1) **Original Len.**: We truncate both the preferred and dispreferred responses to a certain percentage of their original lengths. (2) POET **Len.**: We first truncate both responses to the length of the shorter one (the POET strategy), and then further truncate them to a certain percentage of this equal length. We vary the retention percentage from 25% to 100% and report the AlpacaEval 2 results.

The results, summarized in Table 3, lead to three main observations. First, the POET **Len.** strategy consistently outperforms the **Original Len.** strategy across almost all retention ratios for both DPO and SimPO. Second, the performance gap between the two strategies is already substantial at low retention ratios, indicating that equal-length truncation is more important than simply shortening them. Third, while simple truncation (Original Len.) brings some gains for DPO, it fails to improve SimPO, whereas the POET **Len.** strategy benefits both. Finally, performance can be further improved by increasing the truncation ratio; nevertheless, as discussed earlier, the parameter-free version of POET remains the safest choice and already provides strong performance.

4.4 Comparison with Token-level Methods

We compare POET with two token-level DAAs methods: SamPO (Lu et al., 2024) and D^{2}PO (Shao et al., 2025). SamPO is designed to mitigate the length bias of DAAs by restricting reward calculation to a random subset of tokens, rather than focusing on the prefix. D^{2}PO prioritizes prefix tokens by applying a temporal decay weight to each token based on its position. Notably, D^{2}PO employs an exponential decay. With a recommended factor of $\gamma = 0.98$, the weight for tokens beyond position 200 drops below 0.02, effectively acting as a truncation strategy based on absolute length.

Method	LC	WR	Length
DPO	12.9	10.6	1569
DPO + SamPO	22.4	15.6	1319
DPO + D^{2}PO	15.1	14.8	1925
DPO + POET	**24.7**	**17.7**	1401
SimPO	20.0	18.0	1777
SimPO + SamPO	19.6	17.6	1789
SimPO + D^{2}PO	12.5	7.9	1127
SimPO + POET	**24.2**	**23.7**	1939

Table 4: Comparison results with other token-level DAAs methods on AlpacaEval 2.

We conduct experiments based on Mistral-7B-Base and report the AlpacaEval 2 results in Table 4. As shown in Table 4, POET consistently outperforms both SamPO and D^{2}PO across both DPO and SimPO.

4.5 Safety Alignment Evaluation

We further evaluate the effectiveness of POET on the safety alignment task.

Setup. We utilize the PKU-SafeRLHF dataset (Ji et al., 2024) for preference optimization, where the safer response is selected as the preferred one. We conduct experiments on two settings: Mistral-Base (7B) and Llama-3-Base (8B). For evaluation, we use AdvBench (Zou et al., 2023), which is a set of 500 harmful behaviors formulated as instructions. We employ Llama Guard 3-8B [4] to perform safety classification on the generated responses. The evaluation metric is the safety rate, where a higher rate indicates better performance.

Results. The experimental results are presented in Table 5. We observe that POET achieves substantial improvements over the DPO baseline across settings. Notably, for the Mistral-Base setting, POET improves the safety rate from 45.2% to 82.1%, a remarkable gain of 36.9 points. We attribute this significant improvement to the nature of the safety alignment task, where the distinction between safe and unsafe responses typically manifests early in the generation process (e.g., a refusal prefix versus a compliant prefix). This characteristic aligns perfectly with the design of POET, which emphasizes the optimization of prefix tokens, thereby effectively bridging the reward-generation gap in safety alignment.

[4] https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard3/8B/MODEL_CARD.md

Method	Mistral-Base (7B)	Llama-3-Base (8B)
DPO	45.2	78.3
+ POET	**82.1**	**87.3**

Table 5: Safety alignment evaluation results (Safety Rate %) on PKU-SafeRLHF.

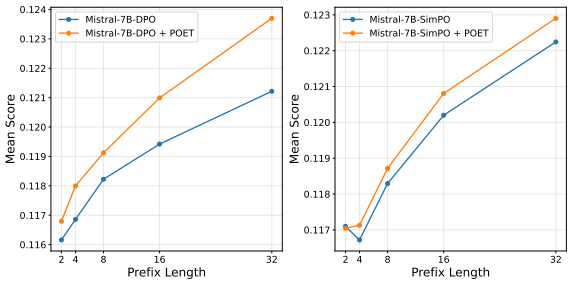


Figure 3: Prefix quality with and without POET across different prefix lengths. The results show that models trained with POET consistently generate higher-quality prefixes.

4.6 POET Generates Better Prefixes

To evaluate whether POET improves performance by generating better prefixes, we conduct an analysis comparing the quality of prefixes generated by standard DAAs models and POET-trained models. The experimental setup is detailed in Appendix G.

Results. Figure 3 demonstrates that models trained with POET consistently generate higher-quality prefixes than standard DAAs models across all prefix lengths, validating the effectiveness of POET in forcing the DAAs to focus on optimizing the prefix quality, which is the key to bridging the reward-generation gap.

4.7 When Does POET Work Best?

As discussed in Section 3.4, POET's effectiveness relies on the practical condition that the quality ranking between preferred and dispreferred responses is preserved after truncation. In this section, we empirically test this condition and show that the effectiveness of POET is also connected to the quality difference between preferred and dispreferred responses of the preference dataset.

We conduct experiments using Llama-3-8B and SimPO across three settings with varying degrees of quality difference (details in Appendix H). We apply POET to all three settings and report the results in Table 6. The results show that POET achieves a notable improvement on settings where

Setting	Consistency (%)	Difference (*100)	Δ
①	93.8	0.6	-2.8
②	98.5	2.3	+2.0
③	98.9	2.9	+2.9

Table 6: Performance comparison on different settings. The Consistency is the quality ranking consistency as in Table 1. The Difference represents the average reward difference between preferred and dispreferred responses. The Δ column shows the absolute improvement in AlpacaEval 2 LC.

the consistency rate after applying POET is high. However, in setting where the consistency rate is only 93.8%, POET leads to a performance drop.

Furthermore, we also observe a clear correlation between the quality difference and the effectiveness of POET. On one hand, the full response quality difference serves as a good proxy for the quality ranking consistency rate, since the quality difference between preferred and dispreferred responses emerges early in the prefixes (Figure 2). On the other hand, recent researches (Lin et al., 2025; Peng et al., 2025; Zhu et al., 2025) show that preference samples with larger quality gaps between preferred and dispreferred responses tend to be more informative. Combining our results with those of these works, we conjecture that POET may perform better on high-quality preference datasets.

5 Related Work

Direct Alignment Algorithms. To overcome the instability and computational overhead of RLHF (Zheng et al., 2023b; Santacroce et al., 2023), direct alignment algorithms (DAAs) (Rafailov et al., 2024a) have been proposed to align LLMs with human preferences without explicit reward modeling or reinforcement learning. DPO (Rafailov et al., 2023) reparameterizes the reward function using the log-ratio between the policy and a reference model, while SimPO (Meng et al., 2024) further simplifies this by using length-normalized log-likelihoods, eliminating the need for a reference model. Other notable DAAs include IPO (Azar et al., 2023), SLiC-HF (Zhao et al., 2023), ORPO (Hong et al., 2024), and KTO (Ethayarajh et al., 2024), each proposing different formulations of the preference optimization objective. Despite their diversity, these methods share a common limitation: their sequence-level objectives may not faithfully reflect autoregressive generation dynamics, which motivates our investigation of the reward-generation gap.

Token-level Preference Optimization. Rafailov et al. (2024b) showed that DPO can be interpreted through a token-level MDP, establishing a theoretical connection between sequence-level preference optimization and token-level credit assignment. Building on this insight, several methods have been proposed to explicitly incorporate token-level signals into DAAs. SamPO (Lu et al., 2024) addresses the length bias of DAAs by computing the reward over a random subset of token positions, effectively decoupling preference learning from response length. D^2PO (Shao et al., 2025) applies a temporal decay weight to each token based on its position, assigning exponentially decreasing importance to later tokens in the sequence. TIS-DPO (Liu et al., 2025) takes a different approach by estimating token-level importance weights using a separate reward model, performing importance sampling to reweight each token's contribution to the DPO objective. While these methods modify the training objective to adjust token-level weighting, POET operates at the data level by truncating response pairs to equal lengths, requiring no changes to the underlying DAA objective and no additional hyperparameters or models.

6 Conclusion

In this paper, we identify and analyze a critical issue in DAAs—the reward-generation gap, which manifests as a misalignment between optimization objectives during training and autoregressive decoding dynamics. To address this issue, we introduce POET, a simple yet effective method for focusing optimization on prefixes by truncating both preferred and dispreferred responses to match the shorter response's length. This hyperparameter-free approach requires no modification to existing DAA objectives while maintaining broad compatibility across different algorithms. Through extensive experiments with DPO and SimPO on multiple model architectures, we demonstrate that our method consistently improves performance.

Limitations

The main limitation of POET is that its effectiveness depends on the preference ordering being preserved after equal-length truncation. This condition is naturally satisfied when the quality difference between preferred and dispreferred responses

emerges early in the generation process. Consequently, for tasks where the decisive quality signal concentrates at the tail of the sequence (e.g., mathematical reasoning tasks where the final answer token is crucial), the truncation may not preserve the preference ordering, and POET may not be applicable. However, it is worth noting that DAAs are currently rarely used for such types of tasks.

Acknowledgements

This project was supported by National Natural Science Foundation of China (No. 62306132), Guangdong Basic and Applied Basic Research Foundation (No. 2025A1515011564), Natural Science Foundation of Shanghai (No. 25ZR1402136). We thank the anonymous reviewers for their insightful feedback on this work. This work was done during Zeguan's internship at SUSTech.

References

AI@Meta. 2024. Llama 3 model card.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Cheung. 2022. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 700–710, Dublin, Ireland. Association for Computational Linguistics.

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, and 3 others. 2021. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861.

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*, abs/2310.12036.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Edward Beeching, Clémentine Fourrier, Nathan Habib, Sheon Han, Nathan Lambert, Nazneen Rajani, Omar Sanseviero, Lewis Tunstall, and Thomas Wolf. 2023. Open LLM leaderboard. https://huggingface.co/spaces/HuggingFaceH4/open_llm_leaderboard.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. In *NeurIPS*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

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

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. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2024. UltraFeedback: Boosting language models with high-quality feedback. In *ICML*.

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

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

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. 2023. Alpacafarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069.

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

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

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024. Pku-saferlhf: Towards

multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Ke Ji, Jiahao Xu, Tian Liang, Qiuzhi Liu, Zhiwei He, Xingyu Chen, Xiaoyuan Liu, Zhijie Wang, Junying Chen, Benyou Wang, and 1 others. 2025. The first few tokens are all you need: An efficient and effective unsupervised prefix fine-tuning method for reasoning models. *arXiv preprint arXiv:2503.02875*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Jiwei Li, Will Monroe, Alan Ritter, Dan Jurafsky, Michel Galley, and Jianfeng Gao. 2016. Deep reinforcement learning for dialogue generation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Austin, Texas. Association for Computational Linguistics.

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

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

Qi Lin, Hengtong Lu, Caixia Yuan, Xiaojie Wang, Huixing Jiang, and Wei Chen. 2025. Data with high and consistent preference difference are better for reward model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27482–27490.

Aiwei Liu, Haoping Bai, Zhiyun Lu, Yanchao Sun, Xiang Kong, Xiaoming Simon Wang, Jiulong Shan, Albin Madappally Jose, Xiaojiang Liu, Lijie Wen, Philip S. Yu, and Meng Cao. 2025. TIS-DPO: Token-level importance sampling for direct preference optimization with estimated weights. In *The Thirteenth International Conference on Learning Representations*.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. 2024. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1047–1067.

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. Simpo: Simple preference optimization with a reference-free reward. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with DPO-positive. *arXiv preprint arXiv:2402.13228*.

Shangpin Peng, Weinong Wang, Zhuotao Tian, Senqiao Yang, Xing Wu, Haotian Xu, Chengquan Zhang, Takashi Isobe, Baotian Hu, and Min Zhang. 2025. Omni-dpo: A dual-perspective paradigm for dynamic preference learning of llms. *arXiv preprint arXiv:2506.10054*.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, Bradley Knox, Chelsea Finn, and Scott Niekum. 2024a. Scaling laws for reward model overoptimization in direct alignment algorithms. *arXiv preprint arXiv:2406.02900*.

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

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. In *NeurIPS*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Michael Santacroce, Yadong Lu, Han Yu, Yuanzhi Li, and Yelong Shen. 2023. Efficient RLHF: Reducing the memory usage of PPO. *arXiv preprint arXiv:2309.00754*.

Ruichen Shao, Bei Li, Gangao Liu, Yang Chen, ZhouXiang, Jingang Wang, Xunliang Cai, and Peng Li. 2025. Earlier tokens contribute more: Learning direct preference optimization from temporal decay perspective. In *The Thirteenth International Conference on Learning Representations*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhengyan Shi, Sander Land, Acyr Locatelli, Matthieu Geist, and Max Bartolo. 2024. Understanding likelihood over-optimisation in direct alignment algorithms. *arXiv preprint arXiv:2410.11677*.

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*.

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

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023. Zephyr: Direct distillation of LM alignment. *ArXiv*, abs/2310.16944.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In *Findings of EMNLP*.

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*, abs/2401.08417.

Hugh Zhang, Daniel Duckworth, Daphne Ippolito, and Arvind Neelakantan. 2021. Trading off diversity and quality in natural language generation. In *Proceedings of the Workshop on Human Evaluation of NLP Systems (HumEval)*, pages 25–33, Online. Association for Computational Linguistics.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. 2023. SLiC-HF: Sequence likelihood calibration with human feedback. *ArXiv*, abs/2305.10425.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023a. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *NeurIPS Datasets and Benchmarks Track*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, and 1 others. 2023b. 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*.

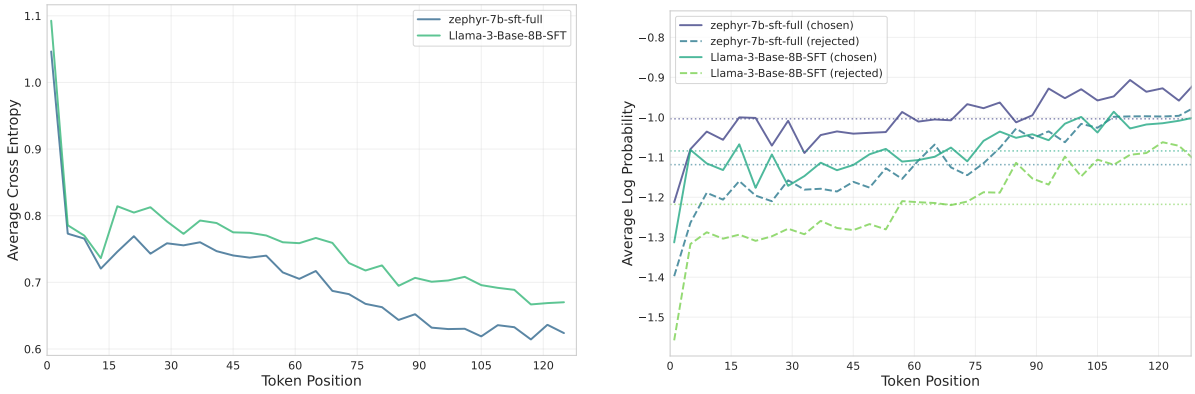
He Zhu, Yifan Ding, Yicheng Tao, Zhiwen Ruan, Yixia Li, Wenjia Zhang, Yun Chen, and Guanhua Chen. 2025. FANNO: Augmenting high-quality instruction data with open-sourced LLMs only. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17633–17653, Vienna, Austria. Association for Computational Linguistics.

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

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.

A Dynamics of Token-level Entropies and Log Probabilities

In this section, we present additional empirical evidence on the dynamics of token-level entropies and log probabilities in Figure 4, to support our analysis in Section 2.2. Figure 4(a) illustrates the average per-position cross entropy when sampling from prompts of the UltraFeedback test set. We observe that prefix tokens have significantly higher uncertainty compared to later tokens, as the latter benefit from more context and exhibit lower randomness. Furthermore, Figure 4(b) presents the average per-position log probability across all responses in the UltraFeedback test set. It reveals that prefix tokens exhibit significantly lower log probabilities. Despite this, DAAs' implicit reward functions treat all tokens equally, which dilutes the importance of these critical early tokens due to the overwhelming number of subsequent tokens.



(a) (b)

Figure 4: (a) Average per-position cross entropy when sampling from prompts. (b) Average per-position log probability. The horizontal line represents the mean log probability across all positions.

B Proof of Theorem 1

Proof. Following the insights from Rafailov et al. (2024b), for $v \in \{w, l\}$, define:

$$R_v(k) = \sum_{t=0}^{k} \beta \log \frac{\pi^*(\mathbf{a}_t^v \mid \mathbf{s}_t^v)}{\pi_{\text{ref}}(\mathbf{a}_t^v \mid \mathbf{s}_t^v)}, \tag{7}$$

where π^* is the optimal policy under the MDP. From Lemma 1 in Rafailov et al. (2024b), we obtain the token-level reward decomposition for a partial sequence up to step k:

$$\sum_{t=0}^{k} r\left(\mathbf{s}_t^v, \mathbf{a}_t^v\right) = V^*(\mathbf{s}_0^v) + R_v(k) - V^*(\mathbf{s}_{k+1}^v). \tag{8}$$

Substituting into Eq. 5, we have:

$$\begin{aligned} &\sum_{t=0}^{k} r\left(\mathbf{s}_t^v, \mathbf{a}_t^v\right) + V^*(\mathbf{s}_{k+1}^v) \\ &= V^*(\mathbf{s}_0^v) + R_v(k) - V^*(\mathbf{s}_{k+1}^v) + V^*(\mathbf{s}_{k+1}^v) \\ &= V^*(\mathbf{s}_0^v) + R_v(k). \end{aligned} \tag{9}$$

Since both trajectories start from the same prompt x, we have $V^*(\mathbf{s}_0^w) = V^*(\mathbf{s}_0^l)$. Therefore, the equal-length sub-trajectories BT model simplifies to:

$$p_k^*(y_{w,\leq k} \succeq y_{l,\leq k}) = \sigma\big(R_w(k) - R_l(k)\big), \tag{10}$$

where σ is the sigmoid function. This is equivalent to the DPO loss function expressed through the optimal policy, demonstrating that the equal-length sub-trajectory BT model and the original sequence-level BT model yield the same optimal policy. □

C Downstream Experimental Results

Table 7 lists detailed results of each downstream task.

D Hyperparameters

Table 8 summarizes the hyperparameters used for the four main experimental settings reported in the main body of the paper. We follow the hyperparameter configurations of DPO and SimPO as described in (Meng et al., 2024). In all experiments, we set max_prompt_length to 1800 and max_length to 2048.

E Evaluations

The official implementation of AlpacaEval 2 (Dubois et al., 2024) uses the *weighted_alpaca_eval_gpt4_turbo* annotator, which employs the *gpt-4-1106-preview* as backbone model. In our experiments, we maintain the same annotator methodology (weighted win rate) but substituted the backbone model with *deepseek-v3-0324*, which is substantially cheaper than *gpt-4-1106-preview* while achieving better performance. We analyze our annotator using AlpacaEval 2's *analyze_evaluators* command and compare it with the official annotator in Table 9. As shown, our annotator achieves higher human agreement, Spearman correlation, and Pearson correlation, while being significantly cheaper. Since Arena-Hard-v0.1 (Li et al., 2024) also employs *gpt-4-1106-preview* as its backbone evaluation model, we substitute it with *deepseek-v3-0324* for the same reasons.

For benchmarks from the Huggingface Open

	MMLU (5)	ARC (25)	HellaSwag (10)	TruthfulQA (0)	Winogrand (5)	GSM8K (5)	Average
Mistral-Base (7B)							
SFT	58.93	57.59	80.65	40.35	76.72	34.34	58.10
DPO	58.77	62.37	83.91	47.76	76.72	29.34	59.81
+ POET	57.93	64.25	84.48	54.68	76.95	30.33	61.44
SimPO	57.71	62.29	83.43	51.08	77.74	30.17	60.40
+ POET	58.12	62.54	83.39	50.85	77.74	34.50	61.19
Llama-3-Base (8B)							
SFT	63.79	60.15	81.59	45.32	76.32	50.80	62.99
DPO	63.50	63.74	83.86	53.67	76.80	53.60	65.86
+ POET	63.45	65.53	83.98	55.25	76.40	51.10	65.95
SimPO	63.15	65.02	82.90	59.78	77.66	48.98	66.25
+ POET	63.34	64.93	82.82	58.20	77.66	54.21	66.86
Llama-3-Instruct v0.2 (8B)							
SFT	61.3	78.8	51.6	65.7	76.5	75.9	68.3
DPO	64.8	79.9	56.2	65.9	76.6	77.4	70.1
+ POET	64.1	79.2	56.6	65.9	76.3	75.4	69.6
SimPO	67.2	78.5	65.6	65.1	76.4	68.5	70.2
+ POET	66.5	79.0	63.4	65.6	77.1	69.9	70.2
Gemma-2-Instruct (9B)							
SFT	71.1	81.8	60.1	72.3	78.5	48.9	68.8
DPO	69.5	71.6	57.9	72.5	72.1	41.9	64.3
+ POET	70.6	71.6	57.5	72.4	72.7	45.0	65.0
SimPO	69.1	67.3	59.1	71.9	73.5	40.6	63.6
+ POET	69.2	68.2	60.5	71.6	74.0	44.1	64.6

Table 7: Downstream task evaluation results of tasks on the huggingface open leaderboard.

Leaderboard (Beeching et al., 2023), we use *lm-evaluation-harness* for evaluation.

F Implementation Details

All the training experiments in this paper were conducted on 8×A800 GPUs based on the alignment-handbook repo and the codebase of (Meng et al., 2024).

G Experimental Setup for Prefix Quality Analysis

In this section, we provide a detailed experimental setup for the prefix quality analysis in Section 4.6.

We randomly sample 500 prompts from the test set of UltraFeedback and generate 5 responses per prompt using both standard DAAs and POET-trained models. These responses are then truncated to create prefixes of varying lengths. Following the definition of prefix quality in Eq. 6, we evaluate the quality of these prefixes by generating completions from them using the SFT model as proxy policy. This choice of proxy policy isolates the influence of other factors. For each prefix, we use the SFT model to generate 3 completions. As a result, we obtain 15 completions (5 responses with 3 completions each) for each prompt at each prefix length. We then evaluate the quality of these completions using a strong reward model, ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024).

H Experimental Setup for Quality Difference Analysis

In this section, we provide a detailed experimental setup for the analysis in Section 4.7.

We conduct an analysis using Llama-3-8B and SimPO across 3 settings with varying degrees of quality difference between preferred and dispreferred responses:

① **Llama-3-8B-SimPO v0.1** (Meng et al., 2024): In this setting, the responses in prefer-

Method	Parameter	Mistral-Base (7B)	Llama-3-Base (8B)	Llama-3-Instruct v0.2 (8B)	Gemma-2-Instruct (9B)
DPO	β	0.01	0.01	0.01	0.01
	Learning rate	5e-7	5e-7	3e-7	5e-7
SimPO	β	2.0	2.0	10	10
	γ/β	0.8	0.5	0.3	0.5
	Learning rate	3e-7	6e-7	1e-6	8e-7

Table 8: Hyperparameters used in our main experiments.

Annotator	Human Agreement	Price	Spearman Corr.	Pearson Corr.	Bias	Variance
weighted_alpaca_eval_deepseek_v3_0324	67.27	0.12	0.95	0.87	32.19	16.45
weighted_alpaca_eval_gpt4_turbo	65.73	4.32	0.78	0.77	33.90	23.65

Table 9: Comparison of AlpacaEval 2 Annotators.

ence dataset are generated by Llama-3-8B-Instruct (AI@Meta, 2024) using the prompts from the UltraFeedback dataset, with preference annotating using a weak reward model. This setting is expected to have a small quality difference between preferred and dispreferred responses of the preference dataset.

② **Llama-3-8B-SimPO v0.2** (Meng et al., 2024): Same as the v0.1 setting, but with a strong reward model, RLHFlow/ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024), used for preference annotating. This setting is expected to have a moderate quality difference.

③ **Llama-3-8B-SimPO Interpolation**: In this setting, we create two new models by interpolating between Llama-3-8B and Llama-3-8B-Instruct with ratios of 0.1:0.9 and -0.1:1.1, respectively. We then generate responses by the two interpolated models and the Llama-3-8B-Instruct model (2 responses per model), following the pipeline with the Llama-3-8B-SimPO-v0.2 setting. This setting is expected to have a large quality difference, as the generated responses are more diverse than the previous two settings.

The quality of preferred and dispreferred responses is estimated by ArmoRM-Llama3-8B-v0.1 (Wang et al., 2024).