Autoregressive Direct Preference Optimization

Masanari Oi [1] **Mahiro Ukai** [1] **Masahiro Kaneko** [2 1] **Naoaki Okazaki** [1 3 4] **Nakamasa Inoue** [1]

Abstract

Direct preference optimization (DPO) has emerged as a promising approach for aligning large language models (LLMs) with human preferences. However, the widespread reliance on the response-level Bradley-Terry (BT) model may limit its full potential, as the reference and learnable models are assumed to be autoregressive only after deriving the objective function. Motivated by this limitation, we revisit the theoretical foundations of DPO and propose a novel formulation that explicitly introduces the autoregressive assumption prior to applying the BT model. By reformulating and extending DPO, we derive a novel variant, termed **Autoregressive DPO (ADPO)**, that explicitly integrates autoregressive modeling into the preference optimization framework. Without violating the theoretical foundations, the derived loss takes an elegant form: it shifts the summation operation in the DPO objective outside the log-sigmoid function. Furthermore, through theoretical analysis of ADPO, we show that there exist two length measures to be considered when designing DPO-based algorithms: the token length μ and the feedback length μ'. To the best of our knowledge, we are the first to explicitly distinguish these two measures and analyze their implications for preference optimization in LLMs. Project page: https://stjohn2007.github.io/ADPO_project/

1. Introduction

Reinforcement learning from human feedback (RLHF) has emerged as a powerful paradigm for aligning large language models (LLMs) with human preferences (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022). As an alternative to traditional RLHF, Direct Preference Optimization (DPO) (Rafailov et al., 2023) analytically derives the optimal solution to the reward maximization objective, enabling efficient and scalable training. This serves as a foundation for state-of-the-art LLMs, supporting diverse tasks ranging from conversational interactions to complex mathematical reasoning (Zhang et al., 2024b; Pang et al., 2024; Jiao et al., 2025; Liao et al., 2025; Lu et al., 2025; Shao et al., 2025).

Recent studies have introduced numerous variants of DPO primarily aimed at improving performance of LLMs (Pang et al., 2024; Zhao et al., 2024; Xiao et al., 2024; Meng et al., 2024; Zhang et al., 2024b; Chen et al., 2024; Wu et al., 2024b; Gupta et al., 2025; Lai et al., 2025; Li et al., 2025b; Wu et al., 2025; Son et al., 2025; Hu et al., 2025; Chen et al., 2025; Liang et al., 2025; Yoon et al., 2025). However, most of them, including those proposing token-level objective functions (Zeng et al., 2024; Zhu et al., 2025; Lin et al., 2025), fundamentally rely on the Bradley-Terry (BT) model (Bradley & Terry, 1952), which stipulates the human preference distribution at the level of complete responses. In this modeling, there is an underlying assumption arising from RLHF that the reward function $r(x, y)$ is defined over complete responses $y \in \mathcal{Y}$, each paired with an input $x \in \mathcal{X}$.

This assumption is regarded as nearly essential because reward models typically learn to evaluate each input-output pair (x, y). For LLM training, even though models produce autoregressive distributions $\pi(y_i|y_{<i}, x)$, learning reward models at a finer granularity (*e.g.*, at the token level) is challenging, as it is unrealistic to ask users to evaluate pairs (x, y_i) conditioned on incomplete responses $y_{<i}$. However, since DPO does not require explicit reward models, there is an opportunity to introduce a more structured implicit reward function that better aligns with autoregressive modeling.

In this paper, we investigate how the autoregressive assumption can be explicitly incorporated into DPO when applying the BT model. Specifically, we revisit the theoretical foundations of DPO and introduce a novel autoregressive variant, which we call **Autoregressive DPO (ADPO)**. The core idea behind ADPO lies in the introduction of an implicit reward

[1]Department of Computer Science, Institute of Science Tokyo, Japan [2]Mohamed bin Zayed University of Artificial Intelligence, UAE [3]AIRC, National Institute of Advanced Industrial Science and Technology (AIST), Japan [4]LLMC, National Institute of Informatics (NII), Japan. Correspondence to: Masanari Oi <masanari.oi@nlp.comp.isct.ac.jp>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

Table 1. **DPO versus ADPO.** By defining reward-based energy functions over the prefix closure $\mathcal{Y}^*$ of the output space $\mathcal{Y}$, we introduce the prefix-wise Bradley-Terry model. The derived loss shifts the summation operation outside the log-sigmoid function.

Method	Energy domain	Modeling	Loss
DPO	Output space $\mathcal{Y}$	Bradley-Terry	$-\log\sigma\left(\beta\sum_i \log\frac{\pi_\theta(y_i^w\|y_{<i}^w,x)}{\pi_{\text{ref}}(y_i^w\|y_{<i}^w,x)} - \beta\sum_i \log\frac{\pi_\theta(y_i^l\|y_{<i}^l,x)}{\pi_{\text{ref}}(y_i^l\|y_{<i}^l,x)}\right)$
ADPO (Ours)	Prefix closure $\mathcal{Y}^*$	Prefix-wise Bradley-Terry	$-\sum_i \log\sigma\left(\beta\log\frac{\pi_\theta(y_i^w\|y_{<i}^w,x)}{\pi_{\text{ref}}(y_i^w\|y_{<i}^w,x)} - \beta\log\frac{\pi_\theta(y_i^l\|y_{<i}^l,x)}{\pi_{\text{ref}}(y_i^l\|y_{<i}^l,x)}\right)$

function $r^*(x, y_{\leq i})$ defined over the prefix closure $\mathcal{Y}^*$, a set of incomplete responses. ADPO is then naturally derived through (1) reward-based prefix energies defined with an autoregressive reference model, (2) their corresponding Boltzmann distributions, and (3) the prefix-wise BT model. The resulting loss function has an elegant form: the summation operation in the DPO loss is shifted outside the log-sigmoid function, as summarized in Table 1. To the best of our knowledge, we are the first to formulate and analyze such an autoregressive extension. It is also worth noting that ADPO does not violate the theoretical foundations of DPO, as the difference naturally arises from the definition of the energy functions. In summary, our contributions are:

1) We introduce **ADPO**, a novel DPO variant derived by extending the domain of energy from the output space $\mathcal{Y}$ to its prefix closure $\mathcal{Y}^*$. This enables the incorporation of the autoregressive assumption into DPO when applying the BT model, providing a theoretically consistent way to facilitate finer-granularity training.
2) We prove under mild assumptions that any reward function can be reparameterized by **an autoregressive model (Theorem 1)**. Furthermore, through theoretical analysis, we identify two sequence length measures on $\mathcal{Y}$ that are critical for designing DPO-based algorithms: *the token length measure* μ and *the feedback length measure* μ'. DPO corresponds to the special case of ADPO with $\mu'(y) = 1$ $(\forall y)$ (Corollary 1), while setting $\mu' \equiv \mu$ yields a token-level variant. In general, choosing these two measures independently enables training at arbitrary granularity.

2. Related Work

Preference Optimization. RLHF frames alignment as optimizing a reward based on pairwise human preferences (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Xiong et al., 2024; Ye et al., 2024; Zhu et al., 2023). Online alignment methods maximize expected return under on-policy rollouts by employing Kullback–Leibler (KL) regularization techniques, such as trust-region constraints or explicit penalties relative to a reference policy (Schulman et al., 2015; 2017; Ouyang et al., 2022). For LLM training, PPO-based methods have explored improved trade-offs between return and divergence (Wu et al., 2023; Zhang et al., 2024a; Wu et al., 2024a; Dai et al., 2024; Zhang et al., 2025), leading to extended approaches such as GRPO (Shao et al., 2024). For efficient training, DPO (Rafailov et al., 2023) has emerged as a primary choice because it significantly reduces computational overhead by eliminating the need for an explicit reward model. Within these methods, foundational probabilistic models for pairwise choice, such as the Bradley-Terry and Plackett-Luce models (Bradley & Terry, 1952; Plackett, 1975), remain standard for preference modeling and are typically applied at the level of complete responses.

Granularity of DPO. Several recent studies have proposed granularity-aware methods to better align with autoregressive generation. For example, SimPO employs a length-normalized loss function (Meng et al., 2024), TDPO imposes forward-KL constraints at each token position (Zeng et al., 2024), TGDPO introduces token-level reward guidance (Zhu et al., 2025), and cDPO (Lin et al., 2025) identifies critical tokens to apply token-level weights. Despite these advances, existing methods often fundamentally rely on the vanilla BT model at the level of complete responses. In contrast, our formulation integrates the autoregressive assumption into DPO when applying the BT model.

Beyond these granularity-oriented approaches, Rafailov et al. (2024) provides a complementary analysis showing that standard DPO can be interpreted at the token level through a soft-Q-learning lens. While this analysis effectively explains the implicit token-level structure of the original response-level BT formulation, it does not modify the BT model itself. In contrast, our formulation incorporates the autoregressive assumption before applying the BT model by extending the energy domain to the prefix closure Y^*, yielding a prefix-wise BT model and a distinct objective.

3. Preliminary

DPO (Rafailov et al., 2023) is a preference optimization algorithm that directly optimizes models using human preferences without relying on an explicit reward model. To simplify the derivation and interpretation of its objective function, we reformulate DPO using two Boltzmann distributions. We begin by defining the two reward-based energy functions, which we refer to as the DPO energies.

Definition 1 (DPO Energies). *Given a reward function $r(x, y)$ and a reference model $\pi_{\text{ref}}(y|x)$, we define the two DPO energies, namely the likelihood energy E_1 and the posterior energy E_2:*

$$E_1(x, y) = -r(x, y), \tag{1}$$

$$E_2(x, y) = -\frac{1}{\beta} r(x, y) - \log \pi_{\text{ref}}(y|x), \tag{2}$$

where $x \in \mathcal{X}$ is an input from the input space $\mathcal{X}$, $y \in \mathcal{Y}$ is the model's output (response) in the output space $\mathcal{Y}$, and $\beta \in \mathbb{R}_+$ is a hyperparameter.

These energy functions derive the two corresponding Boltzmann distributions:

$$p_1(y^w \succ y^l|x) = \frac{\exp(-E_1(x, y^w))}{\sum_{y \in Y} \exp(-E_1(x, y))}, \tag{3}$$

$$p_2(y|x) = \frac{\exp(-E_2(x, y))}{\sum_{y \in \mathcal{Y}} \exp(-E_2(x, y))}, \tag{4}$$

where p_1 is normalized over a set of preferred and dispreferred responses $Y = \{y^w, y^l\} \subset \mathcal{Y}$ following the BT model (Bradley & Terry, 1952), whereas p_2 is normalized over the entire output space $\mathcal{Y}$. It is straightforward to see that p_2 maximizes the KL-constrained reward function (Peters & Schaal, 2007; Peng et al., 2019; Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023). Then, DPO minimizes the negative log-likelihood loss $\mathcal{L}$ with respect to p_1:

$$\begin{aligned}\mathcal{L}_{\text{DPO}} &= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\log p_1(y^w \succ y^l|x)\right] \\ &= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{p_2(y^w|x)}{\pi_{\text{ref}}(y^w|x)} - \beta \log \frac{p_2(y^l|x)}{\pi_{\text{ref}}(y^l|x)}\right)\right]\end{aligned} \tag{5}$$

where σ is the sigmoid function and $\mathcal{D}$ is the preference dataset. During training, p_2 is parameterized as $p_2 = \pi_\theta$ with a parameter set θ. This is the reparameterization trick of DPO.

In the recent paradigm of LLMs, models are typically assumed to be autoregressive, *i.e.*, assuming that $\mathcal{Y}$ is a space of sequence of arbitrary length, both reference and learnable models satisfy the chain rule: $\pi(y|x) = \prod_{i=1}^{T} \pi(y_i|y_{<i}, x)$, where $T = \mu(y)$ is the length of y.[1] From this, we have

$$\begin{aligned}\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\Bigg[\log \sigma\Bigg(\sum_{i=1}^{T}\Bigg(\beta \log \frac{\pi_\theta(y_i^w|y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w|y_{<i}^w, x)} \\ - \beta \log \frac{\pi_\theta(y_i^l|y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l|y_{<i}^l, x)}\Bigg)\Bigg)\Bigg],\end{aligned} \tag{6}$$

where right-padding is applied so that both sequences have the same length following practical implementation of DPO using mini-batch optimization algorithms.

[1] $\mu : \mathcal{Y} \to \mathbb{N}$ is the token length measure that assigns to each $y \in \mathcal{Y}$ the number of tokens it contains.

4. Autoregressive Direct Preference Optimization

From the derivation of DPO, we observe an important discrepancy: while the learnable model π_θ is autoregressive, the Boltzmann distribution p_2 defined in Eq. (4) is not formulated as such. This motivates our research question: *Can we define energy functions that yield p_2 explicitly as an autoregressive distribution?* Our ADPO formulation answers this question affirmatively.

The core idea behind ADPO lies in the introduction of the ADPO energies, which define energies and the corresponding implicit reward function over the prefix closure of the output space.

Definition 2 (ADPO Energies). *Let $\mathcal{Y}^*$ be the prefix closure of $\mathcal{Y}$, i.e.,*

$$\mathcal{Y}^* = \bigcup_{y \in \mathcal{Y}} \{y_{\leq i} = (y_1, y_2, \ldots, y_i) : \ 0 \leq i \leq T'\}, \tag{7}$$

where $T' = \mu'(y)$ is the length of y.[2] *Given a <u>prefix-wise</u> reward function $r^* : \mathcal{X} \times \mathcal{Y}^* \to \mathbb{R}$ and an <u>autoregressive</u> reference model $\pi_{\text{ref}}(y|x)$, we define the two ADPO energies, namely the prefix likelihood energy E_1^* and the prefix posterior energy E_2^*, as*

$$E_1^*(x, y_{\leq i}) = -r^*(x, y_{\leq i}), \tag{8}$$

$$E_2^*(x, y_{\leq i}) = -\frac{1}{\beta} r^*(x, y_{\leq i}) - \log \pi_{\text{ref}}(y_i|y_{<i}, x). \tag{9}$$

There are two main differences compared to Definition 1. First, the reward function is defined over $\mathcal{X} \times \mathcal{Y}^*$ instead of $\mathcal{X} \times \mathcal{Y}$. With this, we define the prefix-wise BT model by taking the product of BT-based preference distributions:

$$p_1(y^w \succ y^l \mid x) = \prod_{i=1}^{T'} \frac{\exp\left(-E_1^*(x, y_{\leq i}^w)\right)}{\sum_{y_{\leq i} \in Y_i} \exp(-E_1^*(x, y_{\leq i}))}, \tag{10}$$

where $Y_i = \{y_{\leq i}^w, y_{\leq i}^l\} \subset \mathcal{Y}^*$ is the set of preferred and dispreferred prefixes up to length i.

Second, the reference model is assumed to be autoregressive in Eq. (9). With this assumption, the second distribution p_2 also becomes autoregressive prior to reparameterization:

$$p_2(y|x) = \prod_{i=1}^{T'} p_2(y_i \mid y_{<i}, x), \tag{11}$$

$$p_2(y_i \mid y_{<i}, x) = \frac{\exp(-E_2^*(x, y_{\leq i}))}{\sum_{y_i \in \mathcal{V}} \exp(-E_2^*(x, y_{\leq i}))}, \tag{12}$$

[2] We will discuss later that this length measure can be independent of the token length measure μ, but one can assume $\mu' = \mu$ for readability.

where $\mathcal{V} = \{y \in \mathcal{Y} : \mu'(y) = 1\}$. This alleviates the mismatch between p_2 and the learnable autoregressive model π_θ.

ADPO Objective. Analogous to DPO, ADPO maximizes the log-probability of $p_1(y^w \succ y^l \mid x)$ during training. We define the loss function as

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\Big[\log p_1(y^w \succ y^l \mid x)\Big]. \tag{13}$$

Through reparameterization, the loss can be written as

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log\sigma\Big(\beta\log\frac{\pi_\theta(y_i^w \mid y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w \mid y_{<i}^w, x)} - \beta\log\frac{\pi_\theta(y_i^l \mid y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l \mid y_{<i}^l, x)}\Big)\right]. \tag{14}$$

A proof is provided in Appendix A. Notably, ADPO shifts the summation in the DPO loss (Eq. (6)) outside the log-sigmoid function. In terms of KL-constrained reward maximization, the optimal solution is preserved (Appendix B).

5. Theoretical Analysis

ADPO does not violate the theoretical foundations of DPO. Rather, the theory is generalized by bridging additive reward functions and autoregressive models. Ultimately, we prove that, under mild assumptions, any reward function can be reparameterized by an autoregressive model (**Theorem 1**). This better aligns the DPO theory with the paradigm of autoregressive LLMs.

5.1. Reparameterization Completeness

We first show how prefix-wise reward functions can be reparameterized by autoregressive models.

Proposition 1 (Prefix-wise Reparameterization Completeness). *Let $[r^*]$ denote the reward-shift equivalence class of a prefix-wise reward function $r^* : \mathcal{X} \times \mathcal{Y}^* \to \mathbb{R}$, defined as*

$$[r^*] = \Big\{r' \;\Big|\; \exists f \;\; \forall(x, y_{\le i}), \; r'(x, y_{\le i}) = r^*(x, y_{\le i}) + f(x, y_{<i})\Big\}. \tag{15}$$

Given an autoregressive reference model $\pi_{\text{ref}}(y_i \mid y_{<i}, x) > 0$ and a hyperparameter $\beta > 0$, for any prefix-wise reward function r^*, there exists a unique representative $r_\circ^* \in [r^*]$ such that for all $x \in \mathcal{X}$ and prefixes $y_{\le i} \in \mathcal{Y}^*$,

$$r_\circ^*(x, y_{\le i}) \equiv \beta\log\frac{\pi(y_i \mid y_{<i}, x)}{\pi_{\text{ref}}(y_i \mid y_{<i}, x)}, \tag{16}$$

for some <u>autoregressive</u> model π.

A proof is provided in Appendix C. To bridge vanilla reward functions $r(x, y)$ and prefix-wise reward functions $r^*(x, y_{\le i})$, we define the additive decomposition of r. Lemma 1 immediately follows.

Definition 3 (Additive Decomposition). *Let $r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ be a reward function. We say that a prefix-wise reward function $r^* : \mathcal{X} \times \mathcal{Y}^* \to \mathbb{R}$ is an additive decomposition of r if and only if*

$$r(x, y) = \sum_{i=1}^{T'} r^*(x, y_{\le i}) \quad \textit{for any } x \in \mathcal{X},\, y \in \mathcal{Y}. \tag{17}$$

Lemma 1. *Every reward function has an additive decomposition.*

Then, from Proposition 1 and Lemma 1, it follows that Theorem 1 holds.

Theorem 1. *All reward classes consistent with the <u>prefix-wise</u> Bradley–Terry models can be represented with the reparameterization*

$$r(x, y) = \beta\log\frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)} \tag{18}$$

for some <u>autoregressive</u> model $\pi(y \mid x)$ and a given <u>autoregressive</u> reference model $\pi_{\text{ref}}(y \mid x)$ such that $\pi(y_i \mid y_{<i}, x) > 0$.

Proof. For any reward function r, an additive decomposition r^* exists by Lemma 1. By Proposition 1, there exists an autoregressive model π such that

$$\begin{aligned} r(x, y) &= \sum_{i=1}^{T'} r^*(x, y_{\le i}) \equiv \sum_{i=1}^{T'} r_\circ^*(x, y_{\le i}) \\ &= \sum_{i=1}^{T'} \beta\log\frac{\pi(y_i \mid y_{<i}, x)}{\pi_{\text{ref}}(y_i \mid y_{<i}, x)} = \beta\log\frac{\pi(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \end{aligned} \tag{19}$$

where $\equiv$ denotes equality within the equivalence class. □

In contrast to the DPO theory (Rafailov et al., 2023), Theorem 1 explicitly demonstrates reparameterizability using an autoregressive model.

5.2. Implicit length measure in DPO

The following corollary further bridges DPO and ADPO, providing deeper insights into DPO.

Corollary 1. *When the length $\mu'(y)$ is one for all $y \in \mathcal{Y}$, ADPO reduces to DPO that relies on the traditional (non-prefixwise) Bradley–Terry model.*

This result can be unintuitive. A straightforward interpretation of "$\mu'(y) = 1$ for all y" is that the output space $\mathcal{Y}$

is restricted to a single-token space, implying that ADPO coincides with DPO in this special case. While this interpretation is correct, it does not fully capture the true meaning of the corollary. Instead, we emphasize that the output space $\mathcal{Y}$ remains a sequence space. Then, Corollary 1 implies that the original DPO formulation has an implicit length measure $\mu' : \mathcal{Y} \to \mathbb{N}$ that assigns a length of one, i.e., $\mu'(y) = 1$, to every sequence $y \in \mathcal{Y}$.

This measure may have limited the exploration of research ideas for DPO-based formulations in prior work, as it predisposed most existing methods to place the summation operation inside the log-sigmoid function, as in Eq. (6). We further discuss its deeper roots.

5.3. Two Distinct Length Measures

What is μ'? Although the implicit length measure μ' may appear counterintuitive, it is natural to interpret it as arising from RLHF. Concretely, μ' measures the length of sequences y in an evaluation scenario by first mapping y into a one-dimensional space using an evaluation metric $\nu : \mathcal{Y} \to \mathbb{R}$, and then defining

$$\mu'(y) = \mathrm{length}(\nu(y)) = \dim(\mathbb{R}) = 1. \tag{20}$$

This interpretation is particularly natural when human feedback is performed on complete responses. Therefore, we refer to μ' as the *feedback length measure*.

Two Length Measures are Independent. The discussion thus far suggests two algorithms. The first is DPO, corresponding to the case $\mu' \equiv 1$, which evaluates the complete response as a whole; thus, we denote its granularity as "Full Response". The second is token-level ADPO, corresponding to $\mu' = \mu$, where μ is the token length measure induced by LLMs. This evaluates each token individually, and its granularity is indicated as "Single Token".

However, importantly, ADPO itself does not impose the constraint $\mu' = \mu$. Although the two measures should be related in practice, they can be theoretically independent because their origins are independent: one arises from LLM tokenization and the other from the prefix closure. This leads to granularity families of ADPO, which we discuss in the next subsection.

5.4. Granularity Families

ADPO allows intermediate granularity levels with $1 \leq \mu'(y) \leq \mu(y)$ in practice. Specifically, by decomposing each sequence y into subsequences $\{z_i\}_{i=1}^{T'}$ such that each prefix $z_{\leq i} = (z_1, z_2, \ldots, z_i)$ constitutes a unit eligible for implicit feedback, the ADPO loss can be formulated with two cumulative log-ratios S_θ^w and S_θ^l:

$$S_\theta^w(i) \triangleq \sum_{j=1}^{T_i^w} \log \frac{\pi_\theta(z_{i,j}^w \mid z_{<i,<j}^w, x)}{\pi_{\mathrm{ref}}(z_{i,j}^w \mid z_{<i,<j}^w, x)}, \tag{21}$$

$$S_\theta^l(i) \triangleq \sum_{j=1}^{T_i^l} \log \frac{\pi_\theta(z_{i,j}^l \mid z_{<i,<j}^l, x)}{\pi_{\mathrm{ref}}(z_{i,j}^l \mid z_{<i,<j}^l, x)}. \tag{22}$$

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \sigma\Big(\beta S_\theta^w(i) - \beta S_\theta^l(i)\Big)\right]. \tag{23}$$

where $z = (z_1, z_2, \ldots, z_{T'})$ is a decomposition of y, $z_i = (z_{i,1}, z_{i,2}, \ldots, z_{i,T_i})$ is a subsequence,[3] T' is the feedback length, and $T_i = \mu(z_i)$ is the token length of z_i.

More formally, the mapping from y to z is specified by a strong composition ξ. Given a token sequence $y = (y_1, \ldots, y_T) \in \mathcal{Y}$, each subsequence is defined as

$$z_i = \big(y_{\bar{\xi}(i)}, y_{\bar{\xi}(i)+1}, \ldots, y_{\bar{\xi}(i+1)-1}\big), \tag{24}$$

with

$$\bar{\xi}(1) = 1, \tag{25}$$

$$\bar{\xi}(i+1) = \bar{\xi}(i) + \xi(i), \tag{26}$$

where $\xi \in \mathcal{C}_{T'}(T)$ and

$$\mathcal{C}_m(n) = \Big\{\xi : [m] \to \mathbb{N}_+ \;\Big|\; \sum_{j=1}^{m} \xi(j) = n\Big\}. \tag{27}$$

is the set of strong compositions of $n \in \mathbb{N}_+$ into $m \in \mathbb{N}_+$ parts.[4] Each token is then given by $z_{i,j} = y_{\bar{\xi}(i)+j-1}$, as illustrated in Fig. 1(a). The choice of ξ determines the learning behavior of ADPO. We define two practical families below.

Static Family. This family fixes a window size $k \in \mathbb{N}_+$ and determines the feedback length based on the token length. Because preferred and dispreferred responses may have different lengths, we apply EOS padding to equalize them, obtaining y_{pad}^w and y_{pad}^l. Each sequence is then decomposed into $\lceil T/k \rceil$ subsequences, where $T = \mu(y_{\text{pad}}^w) = \mu(y_{\text{pad}}^l)$. The strong composition is defined as

$$\xi_{\text{static}}(i) = \min\big(k,\ \mu(y_{\text{pad}}^w) - k(i-1)\big). \tag{28}$$

As shown in Fig. 1(b–d), $k = 1$ corresponds to token-level granularity, while larger k yields coarser feedback.

[3] Each prefix is denoted by $z_{<i,<j} = (z_1, \ldots, z_{i-1}, (z_{i,1}, \ldots, z_{i,j-1}))$.

[4] $[m] = \{1, 2, \ldots, m\}$.

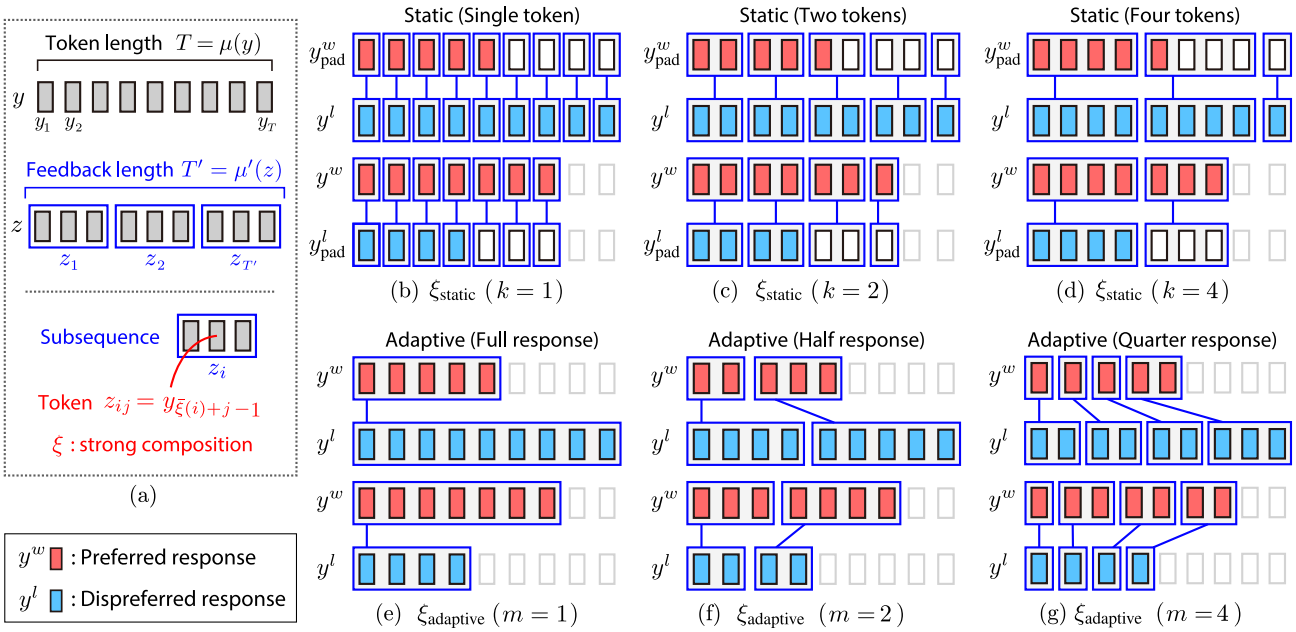

Figure 1. Static and adaptive families of ADPO. (a) Token and feedback length measures. Each subsequence z_i is defined by a strong composition ξ. (b–d) Static families with a fixed window size k. (e–g) Adaptive families with a fixed number of subsequences. (e) corresponds to DPO. Blue rectangles indicate regions where summation is applied inside the log-sigmoid function.

Adaptive Family. This family decomposes each sequence into $m \in \mathbb{N}_+$ subsequences as uniformly as possible and fixes the feedback length as $\mu'(y) = m$ for all $y \in \mathcal{Y}$. The strong composition is defined as

$$\xi_{\text{adaptive}}(i) = \left\lfloor \frac{i\,\mu(y)}{m} \right\rfloor - \left\lfloor \frac{(i-1)\,\mu(y)}{m} \right\rfloor. \tag{29}$$

When $m = 1$, this formulation reduces to DPO, as shown in Fig. 1(e). For $m > 1$, preferred and dispreferred responses are partitioned uniformly (Fig. 1(f,g)), naturally extending DPO to finer granularity.

6. Experiments

6.1. Experimental Setup

Datasets and Metrics. We use two representative math reasoning datasets: GSM8K (Cobbe et al., 2021) and MATH500 (Hendrycks et al., 2021). GSM8K is a dataset of grade school math word problems that requires multi-step reasoning to arrive at the correct numerical answer. MATH comprises challenging competition mathematics problems spanning various topics including algebra, geometry, probability, and number theory. All models are fine-tuned on the GSM8K training set. We report accuracy using the 8-shot evaluation protocol for GSM8K (Wei et al., 2022) and the 4-shot evaluation protocol for MATH500 (Lewkowycz et al., 2022).

Baselines. We include two baselines: DPO (Rafailov et al., 2023) and cDPO (Lin et al., 2025). cDPO is one of the latest preference optimization methods, which leverages token-level contrastive estimation to identify and penalize critical tokens in incorrect reasoning trajectories.

ADPO Variants. We apply our approach to both DPO and cDPO, and refer to the resulting variants as ADPO and cADPO, respectively. ADPO uses the loss in Eq. (23) and cADPO extends it to utilize the critical tokens. Specifically, we define

$$\tilde{S}^l_\theta(i) \triangleq \sum_{j=1}^{T^l_i} \bar{s}_j \log \frac{\pi_\theta(z^l_{i,j} \mid z^l_{<i,<j}, x)}{\pi_{\text{ref}}(z^l_{i,j} \mid z^l_{<i,<j}, x)}, \tag{30}$$

where $\bar{s}_j = 1 - s_j$ is a token-level weight computed from the reward score s_j obtained via contrastive estimation (Lin et al., 2025). Using $S^w_\theta(i)$ defined in Eq. (21) and the weighted log-ratio $\tilde{S}^l_\theta(i)$, the cADPO loss is given by

$$\mathcal{L}_{\text{cADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \sigma\Big(\beta S^w_\theta(i) - \beta \tilde{S}^l_\theta(i)\Big)\right]. \tag{31}$$

We use ADPO and cADPO in the adaptive family with $m = 256$ as our primary methods.

Models. We conduct experiments with four diverse LLMs: Llama3-8B-Base (Llama Team, 2024), Qwen-3-8B-Base (Qwen Team, 2025), Gemma-3-12B-PT (Gemma Team, 2025) and DeepSeek-Math-7B (Shao et al., 2024).

Table 2. Experimental results on two mathematical reasoning benchmarks.

Method	Llama-3-8B		Gemma-3-12B		Qwen-3-8B		DeepSeek-Math-7B	
	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH
DPO (Rafailov et al., 2023)	64.37	18.00	77.03	39.80	86.96	53.80	67.78	32.00
ADPO (Ours)	**68.08**	**21.00**	**78.32**	**41.20**	**88.10**	**55.40**	**69.98**	**33.40**
cDPO (Lin et al., 2025)	67.90	16.80	77.18	38.60	90.98	56.80	72.90	33.40
cADPO (Ours)	**68.76**	**20.20**	**78.85**	**40.40**	**91.74**	**57.20**	**73.54**	**35.40**

Table 3. Granularity families of ADPO. Best results within the static and adaptive families are underlined, and the overall best results are highlighted in bold.

Method	$\mu'(y)$	Composition	Granularity	Llama-3-8B		Gemma-3-12B		Qwen-3-8B		DS-Math-7B	
				GSM	MATH	GSM	MATH	GSM	MATH	GSM	MATH
ADPO	$\mu(y)/8$	$\xi_{\text{static}}\ (k=8)$	Eight tokens	64.97	18.20	76.88	40.00	87.79	54.20	68.39	32.00
ADPO	$\mu(y)/4$	$\xi_{\text{static}}\ (k=4)$	Four tokens	66.26	17.40	77.41	40.80	88.48	54.20	68.23	32.00
ADPO	$\mu(y)/2$	$\xi_{\text{static}}\ (k=2)$	Two tokens	66.57	18.60	77.63	**41.20**	88.25	54.20	70.05	33.60
ADPO	$\mu(y)$	$\xi_{\text{static}}\ (k=1)$	Single token	66.41	18.00	78.09	39.80	88.48	54.40	**70.36**	**34.80**
DPO	1	$\xi_{\text{adaptive}}(m=1)$	Full response	64.37	18.00	77.03	39.80	86.96	53.80	67.78	32.00
ADPO	2	$\xi_{\text{adaptive}}(m=2)$	Half response	64.29	19.60	76.88	40.20	87.95	53.80	68.61	33.40
ADPO	16	$\xi_{\text{adaptive}}(m=16)$	1/16 response	64.52	18.00	77.10	40.80	87.26	52.40	68.84	33.40
ADPO	128	$\xi_{\text{adaptive}}(m=128)$	1/128 response	66.72	18.20	77.63	41.00	**89.16**	54.60	69.83	33.60
ADPO	256	$\xi_{\text{adaptive}}(m=256)$	1/256 response	**68.08**	**21.00**	**78.32**	**41.20**	88.10	**55.40**	69.98	33.40
ADPO	512	$\xi_{\text{adaptive}}(m=512)$	1/512 response	67.32	19.00	77.94	40.00	88.02	54.00	**70.36**	34.40

Implementation Details. Each model is trained with LoRA using the AdamW optimizer for 3 epochs. Since cDPO requires a larger learning rate, we set it to 2.0×10^{-5} for DPO and ADPO, and 4.0×10^{-5} for cDPO and cADPO.

6.2. Experimental Results

Main Results. Table 2 summarizes the experimental results. We observe that our approach consistently outperforms both the DPO and cDPO baselines across all four LLMs, achieving the best results with cADPO. The highest individual result was obtained by Qwen-3-8B using cADPO, reaching 91.74% on GSM8K. These results demonstrate the practical effectiveness of our approach.

Granularity Families. To analyze how varying granularity affects ADPO performance, we systematically explore both static and adaptive families across a range of hyperparameter values in Table 3. For the static family, we observe that models with $k \leq 4$ consistently outperform DPO, indicating the benefit of finer granularity. With DeepSeek-Math-7B, the finest granularity (token-level, $k=1$) achieves the best performance. However, there remains room for improvement, as the static family requires padding when computing the loss, which potentially constrains its performance.

For the adaptive family, which is equivalent to DPO when $m = 1$, we observe performance degradation across several LLMs when m is small (*e.g.*, $m = 2$). This may occur because dividing responses into a small number of segments can result in unnatural or arbitrary splits. This limitation is addressed by increasing the number of segments. Specifically, when $m \geq 128$, ADPO consistently outperforms DPO in all cases, again indicating the benefit of finer granularity.

When comparing the best-performing models from the static and adaptive families, the adaptive family tends to achieve superior performance. As the adaptive family naturally extends DPO without requiring padding, it effectively addresses the limitations associated with the static family. Overall, these findings validate our theoretical insight that defining the token-length measure and feedback-length measure through strong compositions is effective.

Learning Behavior. Figure 2 analyzes the training dynamics of ADPO compared to DPO. Specifically, it shows the evolution of log probabilities for preferred (chosen) and dispreferred (rejected) sequences across training steps for each LLM. We observe that ADPO effectively amplifies the distinction between preferred and dispreferred responses by consistently elevating the probabilities of preferred responses while suppressing those of dispreferred responses. This tendency becomes more pronounced as granularity increases; that is, as k decreases and m increases. These observations underscore the effectiveness of incorporating the autoregressive assumption into the BT model, enabling ADPO variants to more effectively guide models toward preferred outputs across training.

Conversation Tasks. To further validate the generalizability of our approach, we conduct experiments on conversation tasks. We follow the setting of SimPO (Meng et al., 2024).

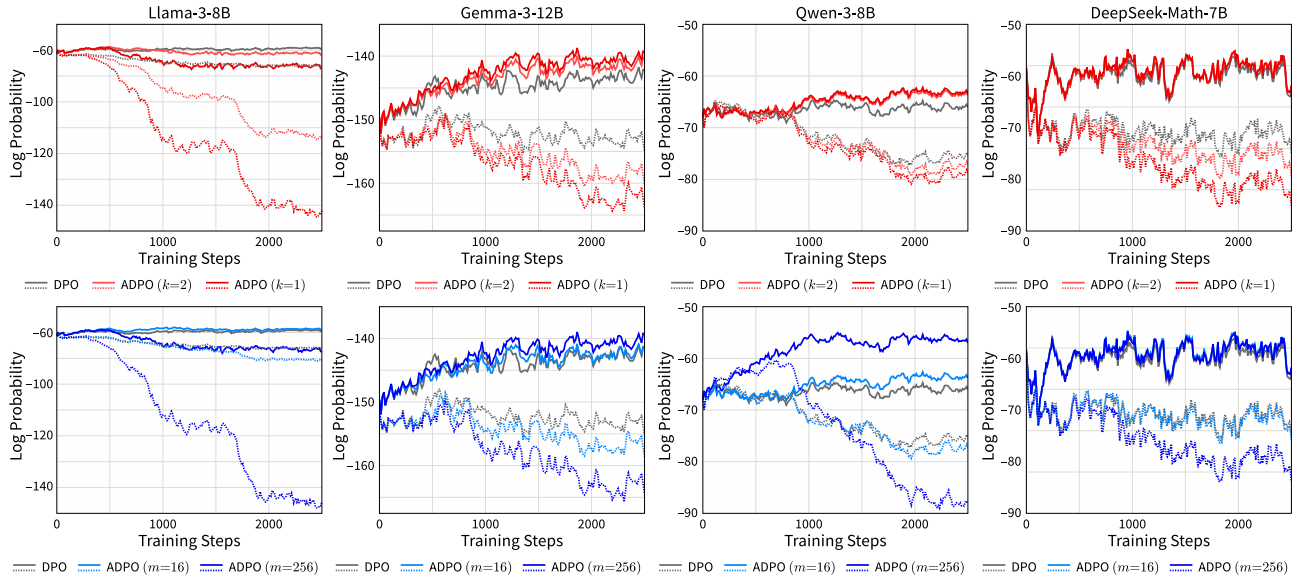

Figure 2. Comparison of training dynamics between DPO and ADPO. The evolution of log probabilities for preferred sequences (solid line) and dispreferred sequences (dashed line) during training is shown. Top row: static family with $k = 1, 2$. Bottom row: adaptive family with $m = 16, 256$.

Table 4. Experimental results on AlpacaEval 2, Arena-Hard, and MT-Bench. Following the official evaluation protocol of each benchmark, length-controlled win rate (LC), win rate (WR), and benchmark scores are reported.

Method	Llama3-8B			
	AlpacaEval 2		Arena-Hard	MT-Bench
	LC	WR	WR	Score
SFT	26.0	25.3	22.3	6.9
DPO (Rafailov et al., 2023)	40.3	37.9	32.6	7.0
SimPO (Meng et al., 2024)	44.7	40.5	33.8	7.0
ADPO (Ours)	**45.8**	**41.1**	**34.4**	**7.1**

Table 5. Granularity families (AlpacaEval 2, Llama3-8B).

Composition	LC	WR
$\xi_{\text{static}}\ (k=4)$	44.9	40.9
$\xi_{\text{static}}\ (k=2)$	45.1	40.8
$\xi_{\text{static}}\ (k=1)$	**45.8**	40.8
$\xi_{\text{adaptive}}\ (m=1)$	40.3	37.9
$\xi_{\text{adaptive}}\ (m=2)$	41.0	40.2
$\xi_{\text{adaptive}}\ (m=16)$	44.1	40.2
$\xi_{\text{adaptive}}\ (m=128)$	**45.8**	**41.1**
$\xi_{\text{adaptive}}\ (m=256)$	44.6	40.5
$\xi_{\text{adaptive}}\ (m=512)$	44.7	40.3

Specifically, we adopt three widely recognized benchmarks: AlpacaEval 2 (Dubois et al., 2024), Arena-Hard (Li et al., 2025a), and MT-Bench (Zheng et al., 2023). For the training data, we generate multiple candidate responses from the base model by prompting it with inputs from UltraFeedback (Cui et al., 2024), and construct chosen–rejected pairs based on PairRM annotations (Jiang et al., 2023). We fine-tune Llama-3-8B-Instruct on this dataset and compare SFT, DPO (Rafailov et al., 2023), SimPO, and ADPO. Note that cDPO cannot perform conversation tasks, as it requires explicit True/False labels for each response, which are not available in UltraFeedback. Table 4 summarizes the results. Consistently, ADPO achieves superior performance against DPO across all three benchmarks and performs comparably to or even better than SimPO, underscoring its enhanced capability in generating more preferred responses.

In Table 5, we also find that granularity exhibits a similar trend as in mathematical reasoning tasks: finer-grained ADPO variants consistently outperform coarser granularity configurations in both static and adaptive families. Collectively, these results demonstrate that ADPO not only improves mathematical reasoning but also effectively generalizes across diverse conversational tasks. Additional experimental results are in Appendix E.

7. Conclusion

We introduced ADPO, a novel approach for direct preference optimization that integrates the autoregressive assumption when applying the Bradley-Terry model. By forming the prefix closure of the output space, we naturally derived the objective function of ADPO that shifts the summation operation outside the log-sigmoid function. Moreover, we provided a theoretical analysis of ADPO, which yields deeper insights into the original DPO framework, identifying two distinct length measures. Our experimental

evaluations extensively validated ADPO across five benchmarks.

Limitations and Future Research Directions. While our study offered novel theoretical and practical perspectives on DPO, it still relies on the original KL-constrained reward maximization problem. Extending beyond this constraint, future research could investigate more general divergence metrics as alternatives to KL divergence, potentially offering greater flexibility and robustness. Moreover, our formulation of the prefix posterior energy in Eq. (9) is specifically motivated by the autoregressive assumption characteristic of LLMs. It would be valuable to explore alternative definitions of energy that are appropriate for other classes of generative models, particularly those that do not adhere strictly to autoregressive architectures. Such investigations could further expand the applicability and effectiveness of DPO-based optimization methods across diverse model types. Finally, ADPO introduces additional prefix-wise computations. Although we empirically observe that the resulting overhead is small (see Appendix E), developing more efficient implementations remains an interesting direction.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Numbers 25K03135 and 25H01137. These research results were obtained from the commissioned research (No.22501) by National Institute of Information and Communications Technology (NICT), Japan. This study was carried out using the TSUBAME4.0 supercomputer at Institute of Science Tokyo.

References

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

Chen, G., Li, X., Shieh, M., and Bing, L. LongPO: Long Context Self-Evolution of Large Language Models through Short-to-Long Preference Optimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Chen, Y., Tan, J., Zhang, A., Yang, Z., Sheng, L., Zhang, E., Wang, X., and Chua, T.-S. On Softmax Direct Preference Optimization for Recommendation. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Christiano, P. F., Leike, J., Brown, T. B., Martić, M., Legg, S., and Amodei, D. Deep Reinforcement Learning from Human Preferences. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.

Cui, G., Yuan, L., Ding, N., Yao, G., He, B., Zhu, W., Ni, Y., Xie, G., Xie, R., Lin, Y., Liu, Z., and Sun, M. UltraFeedback: Boosting Language Models with Scaled AI Feedback. In *Proc. International Conference on Machine Learning (ICML)*, 2024.

Dai, J., Pan, X., Sun, R., Ji, J., Xu, X., Liu, M., Wang, Y., and Yang, Y. Safe RLHF: Safe Reinforcement Learning from Human Feedback. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

Dubois, Y., Liang, P., and Hashimoto, T. B. Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators. In *Proc. Conference on Language Modeling (COLM)*, 2024.

Gemma Team. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

Go, D., Korbak, T., Kruszewski, G., Rozen, J., Ryu, N., and Dymetman, M. Aligning Language Models with Preferences through f-divergence Minimization. In *Proc. International Conference on Machine Learning (ICML)*, 2023.

Gupta, T., Madhavan, R., Zhang, X., Bansal, C., and Rajmohan, S. AMPO: Active Multi Preference Optimization for Self-play Preference Selection. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring Mathematical Problem Solving With the MATH Dataset. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

Hu, X., Kong, L., He, T., and Wipf, D. Explicit Preference Optimization: No Need for an Implicit Reward Model. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Jiang, D., Ren, X., and Lin, B. Y. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL)*, 2023.

Jiao, F., Guo, G., Zhang, X., Chen, N. F., Joty, S., and Wei, F. Preference Optimization for Reasoning with Pseudo Feedback. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Korbak, T., Elsahar, H., Kruszewski, G., and Dymetman, M. On Reinforcement Learning and Distribution Matching for Fine-Tuning Language Models with No Catastrophic Forgetting. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Lai, X., Tian, Z., Chen, Y., Yang, S., Peng, X., and Jia, J. Step-DPO: Step-wise Preference Optimization for Long-chain Reasoning of LLMs. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Lewkowycz, A., Andreassen, A. J., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V. V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., Wu, Y., Neyshabur, B., Gur-Ari, G., and Misra, V. Solving Quantitative Reasoning Problems with Language Models. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., and Stoica, I. From Crowdsourced Data to High-quality Benchmarks: Arena-Hard and Benchbuilder Pipeline. In *Proc. International Conference on Machine Learning (ICML)*, 2025a.

Li, Y., Hu, X., Qu, X., Li, L., and Cheng, Y. Test-Time Preference Optimization: On-the-Fly Alignment via Iterative Textual Feedback. In *Proc. International Conference on Machine Learning (ICML)*, 2025b.

Liang, X., Chen, C., Qiu, S., Wang, J., Wu, Y., Fu, Z., Chen, H., Wu, F., and Ye, J. ROPO: Robust Preference Optimization for Large Language Models. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Liao, W., Chu, X., and Wang, Y. TPO: Aligning LLMs with Multi-branch & Multi-step Preference Trees. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Lin, Z., Liang, T., Xu, J., Liu, Q., Wang, X., Luo, R., Shi, C., Li, S., Yang, Y., and Tu, Z. Critical Tokens Matter: Token-Level Contrastive Estimation Enhances LLM's Reasoning Capability. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Llama Team. The Llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*, 2024.

Lu, Z., Zhou, A., Wang, K., Ren, H., Shi, W., Yang, Y., Pan, J., Zhan, M., and Li, H. Step-Controlled DPO: Leveraging Stepwise Errors for Enhancing Mathematical Reasoning of Language Models. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Meng, Y., Xia, M., and Chen, D. SimPO: Simple Preference Optimization with a Reference-Free Reward. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training Language Models to Follow Instructions with Human Feedback. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Pang, R. Y., Yuan, W., He, H., Cho, K., Sukhbaatar, S., and Weston, J. Iterative Reasoning Preference Optimization. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-Weighted Regression: Simple and Scalable Off-Policy Reinforcement Learning. *arXiv preprint 1910.00177*, 2019.

Peters, J. and Schaal, S. Reinforcement Learning by Reward-weighted Regression for Operational Space Control. In *Proc. International Conference on Machine Learning (ICML)*, 2007.

Plackett, R. L. The Analysis of Permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24 (2):193–202, 1975. doi: https://doi.org/10.2307/2346567.

Qwen Team. Qwen3 Technical Report. *arXiv preprint arXiv:2505.09388*, 2025.

Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Rafailov, R., Hejna, J., Park, R., and Finn, C. From r to Q^*: Your Language Model is Secretly a Q-Function. In *Proc. Conference on Language Modeling (COLM)*, 2024.

Schulman, J., Levine, S., Moritz, P., Jordan, M., and Abbeel, P. Trust Region Policy Optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2015.

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

Shao, R., Li, B., Liu, G., Chen, Y., Zhou, X., Wang, J., Cai, X., and Li, P. Earlier Tokens Contribute More: Learning Direct Preference Optimization From Temporal Decay Perspective. In *Proc. International Conference on Learning Representations (ICLR)*, 2025.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Son, S., Bankes, W., Chowdhury, S. R., Paige, B., and Bogunovic, I. Right Now, Wrong Then: Non-Stationary Direct Preference Optimization under Preference Drift. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D. M., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. Learning to summarize from human feedback. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2020.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Wu, F., Inan, H. A., Backurs, A., Chandrasekaran, V., Kulkarni, J., and Sim, R. Privately Aligning Language Models with Reinforcement Learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2024a.

Wu, J., Xie, Y., Yang, Z., Wu, J., Gao, J., Ding, B., Wang, X., and He, X. β-DPO: Direct Preference Optimization with Dynamic β. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b.

Wu, J., Wang, X., Yang, Z., Wu, J., Gao, J., Ding, B., Wang, X., and He, X. AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-Grained Human Feedback Gives Better Rewards for Language Model Training. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Xiao, T., Yuan, Y., Zhu, H., Li, M., and Honavar, V. G. Cal-DPO: Calibrated Direct Preference Optimization for Language Model Alignment. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., and Zhang, T. Iterative Preference Learning from Human Feedback: Bridging Theory and Practice for RLHF under KL-constraint. In *Proc. International Conference on Machine Learning (ICML)*, 2024.

Ye, C., Xiong, W., Zhang, Y., Dong, H., Jiang, N., and Zhang, T. Online Iterative Reinforcement Learning from Human Feedback with General Preference Model. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Yoon, H. S., Yoon, E., Hasegawa-Johnson, M. A., Kim, S., and Yoo, C. D. ConfPO: Exploiting Policy Model Confidence for Critical Token Selection in Preference Optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Zeng, Y., Liu, G., Ma, W., Yang, N., Zhang, H., and Wang, J. Token-level Direct Preference Optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2024.

Zhang, C., Shen, W., Zhao, L., Zhang, X., Xu, X., Dou, W., and Bian, J. Policy Filtration for RLHF to Mitigate Noise in Reward Models. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

Zhang, H., Lei, Y., Gui, L., Yang, M., He, Y., Wang, H., and Xu, R. CPPO: Continual Learning for Reinforcement Learning with Human Feedback. In *Proc. International Conference on Learning Representations (ICLR)*, 2024a.

Zhang, X., Du, C., Pang, T., Liu, Q., Gao, W., and Lin, M. Chain of Preference Optimization: Improving Chain-of-Thought Reasoning in LLMs. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024b.

Zhao, S., Dang, J., and Grover, A. Group Preference Optimization: Few-Shot Alignment of Large Language Models. In *Proc. International Conference on Learning Representations (ICLR)*, 2024.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proc. Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023.

Zhu, B., Jordan, M., and Jiao, J. Principled Reinforcement Learning with Human Feedback from Pairwise or K-wise Comparisons . In *Proc. International Conference on Machine Learning (ICML)*, 2023.

Zhu, M., Chen, X., Wang, Z., Yu, B., Zhao, H., and Jia, J. TGDPO: Harnessing Token-Level Reward Guidance for Enhancing Direct Preference Optimization. In *Proc. International Conference on Machine Learning (ICML)*, 2025.

A. Derivation of ADPO loss

We provide a proof of Eq. (14). We first provide detailed definition of ADPO loss and then show the Proposition A.1.

Definition A.1. *We define ADPO loss as*

$$\mathcal{L}_{ADPO} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\big[\log p_1(y^w \succ y^l|x)\big]. \tag{32}$$

with ADPO energies given by

$$E_1^*(x, y_{\leq i}) = -r^\circ(x, y_{\leq i}), \tag{33}$$

$$E_2^*(x, y_{\leq i}) = -\frac{1}{\beta}r^\circ(x, y_{\leq i}) - \log \pi_{\text{ref}}(y_i|y_{<i}, x), \tag{34}$$

where $r^\circ \in [r^*]$ is a reward equivalent to r^* given by

$$r^\circ(x, y_{\leq i}) = r^*(x, y_{\leq i}) - f(x, y_{<i}), \tag{35}$$

$$f(x, y_{<i}) = \beta \log \sum_{y_i \in \mathcal{V}} \pi_{ref}(y_i|y_{<i}, x) \exp\left(\frac{1}{\beta}r^*(x, y_{\leq i})\right). \tag{36}$$

Proposition A.1. *Given an autoregressive reference model π_{ref}, an autoregressive learnable model π_θ, a dataset $\mathcal{D}$, and a constant $\beta > 0$, the ADPO loss is given by*

$$\mathcal{L}_{ADPO} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \sigma\left(\beta \log \frac{\pi_\theta(y_i^w|y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w|y_{<i}^w, x)} - \beta \log \frac{\pi_\theta(y_i^l|y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l|y_{<i}^l, x)}\right)\right], \tag{14}$$

where σ is the sigmoid function.

Proof. From the definition of ADPO loss, we have

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\big[\log p_1(y^w \succ y^l|x)\big] \tag{37}$$

$$= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\log \prod_{i=1}^{T'} \frac{\exp(-E_1^*(x, y_{\leq i}^w))}{\sum_{y_{\leq i}\in Y_i} \exp(-E_1^*(x, y_{\leq i}))}\right] \tag{38}$$

$$= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \frac{\exp(-E_1^*(x, y_{\leq i}^w))}{\sum_{y_{\leq i}\in Y_i} \exp(-E_1^*(x, y_{\leq i}))}\right] \tag{39}$$

$$= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \frac{1}{1 + \exp(E_1^*(x, y_{\leq i}^w) - E_1^*(x, y_{\leq i}^l))}\right] \tag{40}$$

$$= -\mathbb{E}_{(x,Y)\sim\mathcal{D}}\left[\sum_{i=1}^{T'} \log \sigma(-(E_1^*(x, y_{\leq i}^w) - E_1^*(x, y_{\leq i}^l)))\right]. \tag{41}$$

where $T' = \max\{\mu'(y^w), \mu'(y^l)\}$. Taking the logarithm of both sides of Eq. (12), we have

$$\log p_2(y_i|y_{<i}, x) = -E_2^*(x, y_{\leq i}) - \log \sum_{y_i \in \mathcal{V}} \exp(-E_2^*(x, y_{\leq i})), \tag{42}$$

and thus we have

$$E_2^*(x, y_{\leq i}) = -\log p_2(y_i|y_{<i}, x) - \log \sum_{y_i \in \mathcal{V}} \exp(-E_2^*(x, y_{\leq i})). \tag{43}$$

From the definition of prefix energies, we obtain

$$E_1^*(x, y_{\leq i}) = \beta \left(E_2^*(x, y_{\leq i}) + \log \pi_{\text{ref}}(y_i | y_{<i}, x) \right) \tag{44}$$

$$= \beta \left(-\log p_2(y_i | y_{<i}, x) - \log \sum_{y_i \in \mathcal{V}} \exp(-E_2^*(x, y_{\leq i})) + \log \pi_{\text{ref}}(y_i | y_{<i}, x) \right) \tag{45}$$

$$= \beta \left(-\log \frac{p_2(y_i | y_{<i}, x)}{\pi_{\text{ref}}(y_i | y_{<i}, x)} \right). \tag{46}$$

Thus, we have

$$E_1^*(x, y_{\leq i}^w) - E_1^*(x, y_{\leq i}^l) = -\beta \log \frac{p_2(y_i^w | y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w | y_{<i}^w, x)} + \beta \log \frac{p_2(y_i^l | y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l | y_{<i}^l, x)}. \tag{47}$$

From Eqs. (41) and (47), we obtain

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}} \left[\sum_{i=1}^{T'} \log \sigma \left(\beta \log \frac{p_2(y_i^w | y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w | y_{<i}^w, x)} - \beta \log \frac{p_2(y_i^l | y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l | y_{<i}^l, x)} \right) \right]. \tag{48}$$

When $p_2 = \pi_\theta$, we have

$$\mathcal{L}_{\text{ADPO}} = -\mathbb{E}_{(x,Y)\sim\mathcal{D}} \left[\sum_{i=1}^{T'} \log \sigma \left(\beta \log \frac{\pi_\theta(y_i^w | y_{<i}^w, x)}{\pi_{\text{ref}}(y_i^w | y_{<i}^w, x)} - \beta \log \frac{\pi_\theta(y_i^l | y_{<i}^l, x)}{\pi_{\text{ref}}(y_i^l | y_{<i}^l, x)} \right) \right]. \tag{49}$$

□

B. KL-constrained Reward Maximization

ADPO leverages the optimal solution of the KL-constrained reward maximization problem. We first review Remark 1 (Peters & Schaal, 2007; Peng et al., 2019; Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023), and then show Corollary 2.

Remark 1 (KL-constrained Reward Maximization). *Given a reward function $r(x, y)$, a reference model $\pi_{\text{ref}}(y|x) > 0$ and a constant $\beta > 0$, we define the objective function for the KL-constrained reward maximization problem as*

$$\mathcal{J}(\pi) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(y|x)} \big[r(x, y) \big] - \beta D_{KL} \big[\pi(y|x) || \pi_{\text{ref}}(y|x) \big]. \tag{50}$$

Then, the optimal solution is given by the Boltzmann distribution induced by the posterior energy:

$$E_2(x, y) = -\frac{1}{\beta} r(x, y) - \log \pi_{\text{ref}}(y|x). \tag{51}$$

Proof. The Boltzmann distribution p_2 induced by E_2 is given by

$$p_2(y|x) = \frac{\exp(-E_2(x, y))}{Z(x)}, \quad Z(x) = \sum_{y\in\mathcal{Y}} \exp(-E_2(x, y)). \tag{52}$$

Thus, we have

$$\mathcal{J}(\pi) = \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\big[r(x,y)\big] - \beta D_{\mathrm{KL}}\big[\pi(y|x)||\pi_{\mathrm{ref}}(y|x)\big] \tag{53}$$

$$= \mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[r(x,y) - \beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right] \tag{54}$$

$$= -\beta\,\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[\log\exp\left(-\frac{1}{\beta}r(x,y)\right) + \log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right] \tag{55}$$

$$= -\beta\,\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)}\right] \tag{56}$$

$$= -\beta\,\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\exp\left(-E_2(x,y)\right)}\right] \tag{57}$$

$$= -\beta\,\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{p_2(y|x)} - \log Z(x)\right] \tag{58}$$

$$= -\beta\,\mathbb{E}_{x\sim\mathcal{D}}\left[D_{\mathrm{KL}}\left(\pi(y|x)||p_2(y|x)\right) - \log Z(x)\right] \tag{59}$$

Then, we obtain

$$\arg\max_{\pi}\mathcal{J}(\pi) = \arg\min_{\pi} D_{\mathrm{KL}}\left(\pi(y|x)||p_2(y|x)\right) = p_2 \tag{60}$$

□

Corollary 2. *Given a reward function $r(x,y)$ and a reference autoregressive model $\pi_{\mathrm{ref}}(y_i|y_{<i},x) > 0$ and a constant $\beta > 0$, the following prefix posterior energy induces the optimal solution of the KL-constrained reward maximization problem in Eq. (50):*

$$E_2^*(x,y_{\le i}) = -\frac{1}{\beta}r^*(x,y_{\le i}) - \log\pi_{\mathrm{ref}}(y_i|y_{<i},x), \tag{61}$$

where r^* is an additive decomposition of r.

Proof. Define an energy $\bar{E}_2^*$ as

$$\bar{E}_2^*(x,y) = \sum_{i=1}^{T'} E_2^*(x,y_{\le i}) \tag{62}$$

Then, we have

$$\bar{E}_2^*(x,y) = -\frac{1}{\beta}\sum_{i=1}^{T'} r^*(x,y_{\le i}) - \log\prod_{i=1}^{T'}\pi_{\mathrm{ref}}(y_i|y_{<i},x) \tag{63}$$

$$= -\frac{1}{\beta}r(x,y) - \log\pi_{\mathrm{ref}}(y|x) \tag{64}$$

$$= E_2(x,y). \tag{65}$$

Therefore, we have $\bar{E}_2^* = E_2$. From Remark 1, the Boltzmann distribution induced by $\bar{E}_2^*$ is the optimal solution of the KL-constrained reward maximization problem. □

C. Prefix-wise Reparameterization Completeness

Proposition 1 (Prefix-wise Reparameterization Completeness). *Let $[r^*]$ denote the reward-shift equivalence class of a prefix-wise reward function $r^* : \mathcal{X}\times\mathcal{Y}^* \to \mathbb{R}$, defined as*

$$[r^*] = \{r' \mid \exists f\ \forall(x,y_{\le i})\ r'(x,y_{\le i}) = r^*(x,y_{\le i}) + f(x,y_{<i})\}. \tag{66}$$

Given an autoregressive reference model $\pi_{\text{ref}}(y_i|y_{<i}, x) > 0$ and a hyperparameter $\beta > 0$, for any prefix-wise reward function r^*, there exists a unique representative $r^*_\circ \in [r^*]$ such that, for all $x \in \mathcal{X}$ and prefixes $y_{\leq i} \in \mathcal{Y}^*$,

$$r^*_\circ(x, y_{\leq i}) \equiv \beta \log \frac{\pi(y_i|y_{<i}, x)}{\pi_{\text{ref}}(y_i|y_{<i}, x)}. \tag{67}$$

for some <u>*autoregressive*</u> *model π.*

Proof. Given a prefix-wise reward function r^*, we first show the existence of π. Define the Boltzmann distribution

$$\pi(y_i|y_{<i}, x) = \frac{\exp(-E^*_2(x, y_{\leq i}))}{Z(x, y_{<i})}, \quad Z(x, y_{<i}) = \sum_{y_i} \exp(-E^*_2(x, y_{\leq i})) \tag{68}$$

Taking logarithms gives

$$\log \pi(y_i|y_{<i}, x) = -E^*_2(x, y_{\leq i}) - \log Z(x, y_{<i}) \tag{69}$$

$$= \frac{1}{\beta} r^*(x, y_{\leq i}) + \log \pi_{\text{ref}}(y_i|y_{<i}, x) - \log Z(x, y_{<i}) \tag{70}$$

Thus, we obtain the required form:

$$r^*(x, y_{\leq i}) = \beta \log \frac{\pi(y_i|y_{<i}, x)}{\pi_{\text{ref}}(y_i|y_{<i}, x)} + \beta \log Z(x, y_{<i}). \tag{71}$$

If another pair $(\tilde{r}, \tilde{\pi})$ enjoys the same property, their log-ratios differ by a function independent of y_i, thereby $\tilde{r} - r^*_\circ = f(x, y_{<i})$. Hence both belong to the same shift class, $r^*_\circ$ is unique modulo such shifts. □

D. Evaluation Metrics

In this section, we provide detailed descriptions of the evaluation metrics used in our experiments.

D.1. Math Reasoning Tasks

For GSM8K and MATH500, we report **accuracy**. A prediction is counted as correct if the final numerical answer matches the ground-truth answer after standard normalization (e.g., stripping formatting, ignoring trivial differences).

D.2. Conversational Benchmarks

AlpacaEval 2 is an automatic preference-evaluation benchmark that uses an LLM-as-a-judge framework to assess instruction-following quality. We report two metrics:

- **Win Rate (WR):** the percentage of pairwise comparisons in which the model's response is preferred over the baseline.
- **Length-Controlled Win Rate (LC):** a debiased variant of WR that adjusts for response-length bias.

Arena-Hard is a curated set of difficult prompts from the Chatbot Arena, designed to measure robustness on challenging instruction-following tasks. We report:

- **Win Rate:** the fraction of pairwise comparisons in which the model's response is preferred according to the Arena-Hard judging protocol (LLM-based judges).

MT-Bench is a multi-turn dialogue benchmark consisting of 80 conversational questions spanning various categories (e.g., reasoning, coding, safety). Evaluation is conducted with GPT-4-based judges. We report:

- **MT-Bench Score:** the average judge rating on a 1–10 scale across all turns, reflecting helpfulness, coherence, and overall response quality.

E. Additional Experiments

In this section, we conduct additional experiments to investigate the impact of several hyperparameters, including the hyperparameter β in the objective function, LoRA rank, and model size. We also analyze the learning behavior in the conversational task. The results collectively validate the consistent effectiveness of ADPO across diverse experimental settings and model configurations.

Hyperparameter β. Table 6 provides a hyperparameter analysis of β for DPO and ADPO across four LLMs evaluated on GSM8K. ADPO consistently surpasses DPO, demonstrating notable improvements particularly for smaller β values (0.5 and 1.0).

LoRA Rank. Table 7 compares the effects of different LoRA ranks (4, 16, and 64) on GSM8K for Llama-3-8B. We observe that ADPO outperforms DPO across all ranks and the performance gap widens at higher ranks.

Model Size. Table 8 examines the impact of scaling model size from Llama-3-8B to Llama-3-70B on GSM8K. ADPO consistently outperforms DPO at both model sizes, achieving 89.23% with a size of 70B.

Table 6. Hyperparameter study for β. Results are reported for $\beta = 0.5, 1.0, 1.5$ on GSM8K.

Method	Llama-3-8B			Gemma-3-12B			Qwen-3-8B			DeepSeek-Math-7B		
	0.5	1.0	1.5	0.5	1.0	1.5	0.5	1.0	1.5	0.5	1.0	1.5
DPO	62.47	64.37	63.91	77.71	77.03	75.82	87.87	86.96	86.73	66.94	67.78	67.63
ADPO	68.61	68.08	64.90	78.85	78.32	77.48	88.10	88.10	88.70	70.05	69.98	69.22

Table 7. LoRA rank r (Llama-3-8B).

Method	4	16	64
DPO	63.53	64.37	63.84
ADPO	65.81	68.08	68.23

Table 8. Model Size.

Method	Llama-3-8B	Llama-3-70B
DPO	64.37	88.10
ADPO	68.08	89.23

Full parameter tuning. Table 9 provides the results of full-parameter tuning (*i.e.* without LoRA) for Llama-3-8B. ADPO and cADPO still outperform DPO and cDPO even under full-parameter training.

Standard Deviation. Table 10 reports the standard deviations over three independent training runs corresponding to the results in Table 2 of the main paper. Table 11 reports the standard errors of the win rates and scores obtained via LLM-as-a-Judge corresponding to the results in Table 4 of the main paper. We observe that the standard deviations (mathematical tasks) and standard errors (conversational tasks) are consistently small. The improvements are consistently larger than the observed errors, and the relative ranking of methods remains unchanged across all settings, supporting the robustness of our conclusions.

Generation Length. Table 12 reports the average output length across all models and datasets. We observed two trends: (a) ADPO produces slightly longer responses than DPO, while (b) cADPO produces shorter responses than cDPO.

ADPO performs preference comparison at each prefix, encouraging the model to accumulate positive evidence step-by-step. This finer-grained signal encourages the model to articulate intermediate reasoning more explicitly, which naturally results in slightly longer outputs.

cDPO amplifies penalties for "critical tokens" at the sequence level. This can lead to longer outputs, because producing additional reasoning steps gives the model more chances to place tokens that are not penalized. In contrast, cADPO applies the same penalties locally at each prefix, preventing such length-driven attenuation. As a result, cADPO discourages unnecessary elaboration and yields more concise outputs.

Computational Complexity. ADPO introduces additional prefix-wise computations compared to DPO, which may raise concerns about increased training cost. To assess this, we measured the wall-clock training time on GSM8K with Llama-3-8B across all ADPO configurations and compared them with DPO. As reported in Table 13, the difference between DPO and the slowest ADPO variant is 125 seconds (5.7%), indicating that the additional computational overhead of ADPO is negligible in practice.

Table 9. Full-parameter tuning results on Llama-3-8B (training without LoRA).

Method	GSM8K	MATH
DPO	64.37	18.00
ADPO	67.32	20.00
cDPO	66.19	18.00
cADPO	68.39	19.80

Table 10. Standard deviations on two mathematical reasoning benchmarks.

Method	**Llama-3-8B**		**Gemma-3-12B**		**Qwen-3-8B**		**DeepSeek-Math-7B**	
	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH
DPO	0.19	0.33	0.22	0.20	0.18	0.61	0.24	0.45
ADPO	0.19	0.33	0.22	0.32	0.06	0.73	0.29	0.41
cDPO	0.09	0.32	0.14	0.15	0.12	0.20	0.21	0.43
cADPO	0.08	0.31	0.17	0.47	0.13	0.45	0.11	0.41

Table 11. Standard errors of win rates and scores on three conversational benchmarks.

Method	**Llama3-8B**			
	AlpacaEval 2		**Arena-Hard**	**MT-Bench**
	LC	WR	WR	Score
SFT	0.75	1.28	1.5	0.08
DPO	0.81	1.46	1.4	0.08
SimPO	0.80	1.47	1.4	0.08
ADPO (Ours)	0.79	1.49	1.4	0.09

Table 12. Average response length.

Method	**Llama-3-8B**		**Gemma-3-12B**		**Qwen-3-8B**		**DeepSeek-Math-7B**	
	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH	GSM8K	MATH
DPO	317.8	869.7	391.2	895.2	299.7	513.8	257.7	682.6
ADPO (Ours)	342.8	917.9	518.5	2192.3	294.9	525.7	280.8	662.9
cDPO	520.4	1859.7	333.5	1174.9	299.8	526.9	261.7	696.7
cADPO (Ours)	308.1	847.0	287.2	841.0	313.8	510.4	250.3	630.0

Learning Behavior. Figure 3 shows training dynamics for the conversation task by plotting the evolution of sequence log probabilities over training steps for DPO and ADPO. In the static family (left), reducing the window size increases the margin between preferred (solid lines) and dispreferred (dashed lines) trajectories, showing greater separation compared to DPO. In the adaptive family (right), increasing the number of segments similarly amplifies this margin. Across both settings, ADPO consistently achieves a clearer divergence between preferred and dispreferred responses throughout training.

Implementation Details. We implemented our methods using the Transformer Reinforcement Learning (TRL) library. For training on GSM8K, we followed the experimental setup of (Lin et al., 2025). Specifically, positive and negative models are first trained for 1 epoch with a learning rate of 3×10^{-4}. Subsequently, preference optimization models are trained with $\beta = 1.0$ for 3 epochs using AdamW with LoRA ($r = 16$), at a learning rate of 2×10^{-5} for DPO and 4×10^{-5} for cDPO. Each problem was sampled 64 times with a top-p probability of 50% during contrastive estimation. For the conversation task, all models are trained for 1 epoch using preference data scored by PairRM. Experiments are conducted using four NVIDIA H100 GPUs. We will release our code and models.

Table 13. Wall-clock training time on GSM8K with Llama-3-8B.

Method	Time (seconds)
DPO	2202
ADPO (ξ_{static}, $k = 4$)	2314
ADPO (ξ_{static}, $k = 2$)	2228
ADPO (ξ_{static}, $k = 1$)	2301
ADPO (ξ_{adaptive}, $m = 1$)†	2234
ADPO (ξ_{adaptive}, $m = 2$)	2287
ADPO (ξ_{adaptive}, $m = 16$)	2321
ADPO (ξ_{adaptive}, $m = 128$)	2318
ADPO (ξ_{adaptive}, $m = 256$)	2226
ADPO (ξ_{adaptive}, $m = 512$)	2327

†Equivalent to DPO.

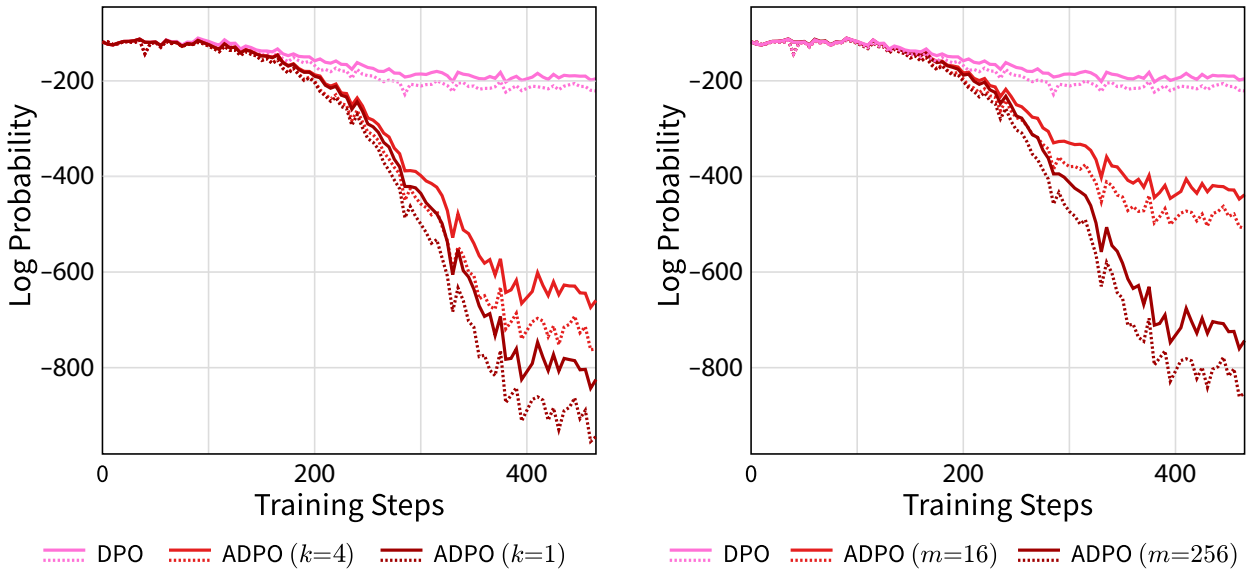

Figure 3. Training dynamics (conversation task).

F. Additional Analysis

F.1. Prefix-wise Reward Dynamics

As formalized in Eq. 16, ADPO assigns an implicit reward at each step i. To understand how these implicit rewards are distributed across prefixes and how they evolve during training, we analyze two summary statistics of the prefix-wise reward signal: (A) the reward variance, which reflects how stable or uncertain the reward is at each prefix, and (B) the reward margin (chosen–rejected difference), which captures where preference differences are realized. We normalize prefix positions to $[0, 1]$ and evaluate these quantities at early, middle, and late checkpoints from ADPO training on the Llama-3-8B model for the math reasoning task. Figures 4 (A) and (B) illustrate how the variance and margin profiles evolve across training.

Early Training. As shown in the blue curves in Figures 4 (A) and (B), both reward variance and margin are concentrated in the middle-to-late prefix region. This indicates that ADPO first focuses on correcting the unstable reasoning steps that occur near the end of the response, where chosen and rejected outputs differ most and the policy is initially most uncertain.

Middle Training. In the mid-stage checkpoints (orange curves), the mass of both variance and margin shifts toward earlier prefixes. Once the later reasoning stabilizes, ADPO begins refining the early parts of the reasoning chain: these early steps determine how the model sets up the solution and whether the subsequent reasoning flows toward a correct final answer.

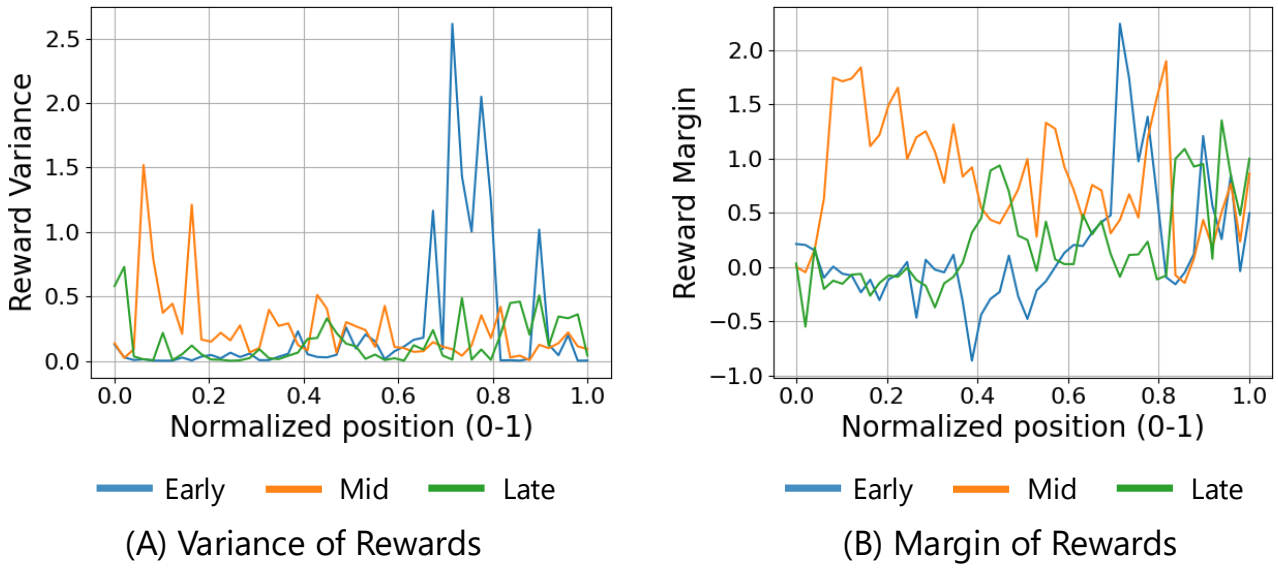

Figure 4. Prefix-wise reward variance and margin.

Late Training. In the late-stage checkpoints (green curves), reward variance becomes small across all prefixes, while the margin becomes sharply concentrated in the final portion of the sequence. This suggests that the global reasoning structure has largely converged, and ADPO focuses on fine-tuning the last few steps that directly determine correctness.

Summary. Overall, these results reveal a consistent three-phase progression: ADPO first improves the end of the reasoning path, then the beginning, and finally sharpens the answer-producing tokens themselves. This demonstrates that ADPO performs meaningful prefix-wise credit assignment, progressively localizing the implicit reward to the prefixes that matter most for preference.