# KL Penalty Control via Perturbation for Direct Preference Optimization

**Anonymous ACL submission**


## Abstract

Direct Preference Optimization (DPO) demonstrates the advantage of aligning a large language model with human preference using only an offline dataset. However, DPO has the limitation that the KL penalty, which prevents excessive deviation from the reference model, is static throughout the training process. Several methods try to turn this static KL penalty into a dynamic one, but no approach can adaptively assign different KL penalties for each preference pair. In this paper, we propose $\varepsilon$-Direct Preference Optimization ($\varepsilon$-DPO), which allows adaptive control of the KL penalty strength $\beta$ for each preference pair. Specifically, $\varepsilon$-DPO adaptively controls $\beta$ for each preference pair based on the monotonicity of logits as a preference model under the perturbation of $\beta$ during training by simply reusing the logit of the current policy and the reference policy. Experimental results show that $\varepsilon$-DPO outperforms existing direct alignment algorithms and KL penalty relaxation methods on general chatbot benchmarks, highlighting the significance of adaptive KL penalty relaxation at the instance-level in DPO.


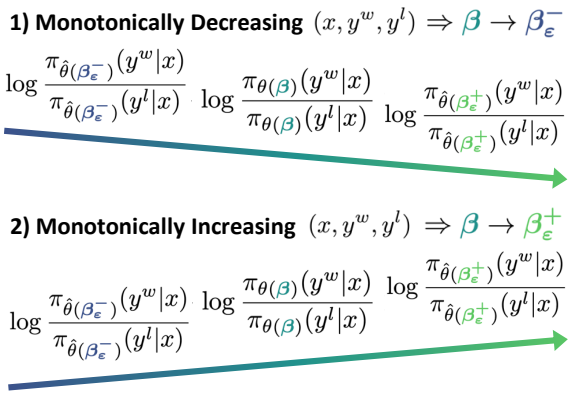


Figure 1: $\varepsilon$-DPO adaptively controls $\beta$ corresponding to the KL penalty strength for each preference pair by checking whether the log-likelihood ratio of the chosen response and the rejected response changes monotonically with the perturbation of $\beta$ used during training. It is equivalent to checking the monotonicity of the logits as a preference model induced by the DPO and estimating the advantage of the change in $\beta$ by the change of train-time inverse temperature to the preference confidence under the same test-time temperature.

## 1 Introduction

Aligning large language models with human preferences for helpfulness and harmless principles (Askell et al., 2021; Cui et al., 2023) is a crucial requirement for general chatbot agents. Reinforcement Learning from Human Feedback (RLHF) (Ziegler et al., 2020) is the pioneering approach that regards the alignment of large language models as a reward maximization problem and solves it by reinforcement learning (Schulman et al., 2017). However, the complicated training pipeline of RLHF increases the training complexity and computation cost of the rollout for online reinforcement learning, in addition to the difficulty of collecting human preference datasets. Moreover, introducing a trained reward model as a proxy reward function to replace the intractable ground-truth human preference reward function makes large language models suffer from the side effect of reward over-optimization (Gao et al., 2023) inherited from the reward models.

Direct Preference Optimization (DPO) (Rafailov et al., 2023) proposes an approach to reform the limitation of RLHF by converting the policy optimization problem into a preference modeling problem and performing alignment using only offline learning. It shows comparable performance while skipping the reward modeling process required by RLHF and has become an effective alternative approach for alignment. In particular, subsequent studies with various modifications to the DPO loss objective open a new research domain called direct alignment algorithms (Rafailov et al., 2024), which perform alignment directly from offline preference

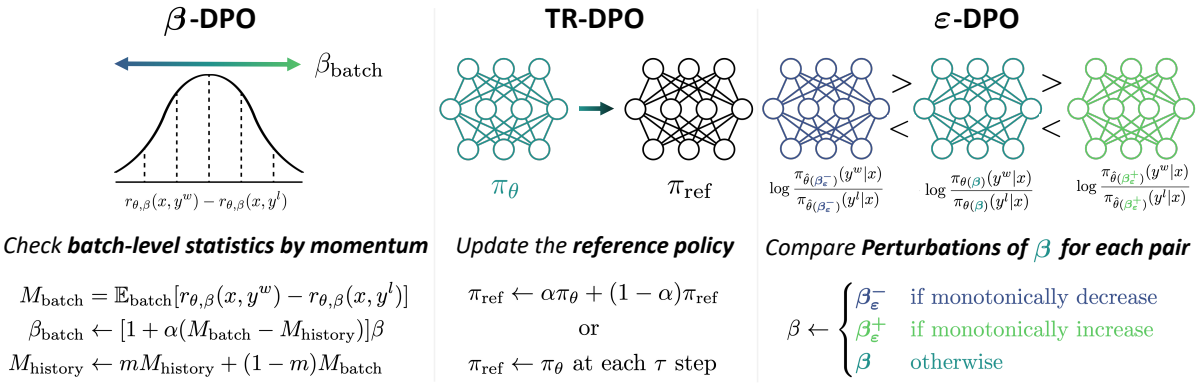


Figure 2: Comparison between $\varepsilon$-DPO and existing KL penalty relaxation methods. $\beta$-DPO (Wu et al., 2024) adaptively adjusts $\beta$ based on the batch-level statistics of implicit reward margin through momentum, failing to adjust at the instance-level and being affected by micro-batch size. TR-DPO (Gorbatovski et al., 2024) updates reference policy periodically by the current policy, so the updates are not adaptive and require computational costs for reference policy updates. On the other hand, $\varepsilon$-DPO can adaptively control $\beta$ at the instance-level by checking the monotonicity of the log-likelihood ratio under perturbation of $\beta$ by simply reusing logits from the policies.

datasets without training separated reward models.

However, DPO assumes that $\beta$ and the reference policy, which define a KL penalty that prevents excessive deviations from the reference model in RLHF, are fixed for exploiting the existence of a closed-form solution derived from the objective function of the RLHF. However, this assumption can lead to suboptimal results, since the KL penalty can be regarded as a Lagrangian relaxation of the constraint optimization defined by the trust region (Schulman et al., 2017). In this regard, $\beta$-DPO (Wu et al., 2024) argues that $\beta$ should be adaptively chosen according to the quality of the preference pair but fails to control $\beta$ at the instance-level and proposes a batch-level control method. On the other hand, TR-DPO (Gorbatovski et al., 2024) claims to periodically update the reference policy to reduce over-optimization (Rafailov et al., 2024), but it may induce unnecessary KL divergence for improvement since the update is not adaptive.

In this paper, we present **$\varepsilon$-Direct Preference Optimization ($\varepsilon$-DPO)**, an instance-level adaptive KL penalty control for DPO that neither TR-DPO nor $\beta$-DPO achieves. Specifically, we check the advantage of adjusting $\beta$ for each preference pair by observing the monotonicity of the log-likelihood ratio between the chosen response and the rejected response when the $\beta$ used during training is perturbed, as described in Figure 1. Here, the criterion for controlling $\beta$ does not require batch-level statistics, and the policy under the perturbed $\beta$ can be estimated by reusing the current policy and reference policy logits. This criterion results in independence from the choice of micro-batch size and no additional computation requirements for model updates, unlike $\beta$-DPO and TR-DPO.

Experimental results demonstrate that $\varepsilon$-DPO outperforms $\beta$-DPO, TR-DPO, and most direct alignment algorithms that modify DPO loss objective (Yuan et al., 2023; Zhao et al., 2023; Azar et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024; Park et al., 2024; Meng et al., 2024), highlighting the importance of adequate KL penalty relaxation for DPO. Furthermore, we confirm that the variation of $\beta$ determined by the adaptive criterion in $\varepsilon$-DPO reflects the confusion as a preference model, which is not addressed by the adaptive criterion of $\beta$-DPO. We also find that the adaptive KL penalty control of $\varepsilon$-DPO is crucial for an efficient KL trade-off compared to TR-DPO, which is not an adaptive KL penalty control.

In summary, our work shows the following:

- $\varepsilon$-DPO provides a simple criterion to improve DPO through KL penalty relaxation.
- $\varepsilon$-DPO adaptively adjusts $\beta$ in instance-level reflecting confusion as a preference model.
- $\varepsilon$-DPO efficiently controls $\beta$ in trade-off between KL divergence and performance.

## 2 Preliminaries

**Reinforcement Learning from Human Feedback** To obtain a language model that aligns with human preference, RLHF (Ziegler et al., 2020) introduces reinforcement learning. It is equivalent to

approaching preference alignment as a reward maximization problem, where we find a policy $\pi$ that maximizes a ground-truth reward function $r^*$ representing human reward for a response $y$ obtained from a corresponding policy for a given prompt $x$. However, since the ground-truth reward function cannot be accessed, a reward model trained from the preference dataset is introduced as a proxy reward function. On the other hand, to prevent the policy update from deviating too much from the current policy from the initial policy, the KL divergence from the reference policy $\pi_{\text{ref}}$ serves as a penalty and regards the initial policy as a reference policy. At this time, coefficient $\beta$ controls the strength of the penalty. The optimal policy that satisfies the maximization of the modified objective function under $\beta$ has a closed-form solution $\pi^*_\beta$,

$$\pi^*_\beta \coloneqq \arg\max_\pi \{\mathbb{E}_{x,y}[r^*(x,y)] - \beta \mathbb{D}_{\text{KL}}(\pi||\pi_{\text{ref}})\},$$
$$\pi^*_\beta(y|x) = \frac{1}{Z^*_\beta(x)} \pi_{\text{ref}}(y|x) \exp\big(\frac{1}{\beta} r^*(x,y)\big),$$
$$Z^*_\beta(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\big(\frac{1}{\beta} r^*(x,y)\big).$$

**Direct Preference Optimization** RLHF has a limitation in efficiency due to the additional training step of the reward model. In this respect, DPO (Rafailov et al., 2023) proposes an approach that can perform preference alignment without training the reward model. DPO focuses on the fact that the ground-truth reward function can be implicitly reparameterized by the closed-form solution $\pi^*_\beta$ and reference policy $\pi_{\text{ref}}$ with an intractable normalizing constant $Z^*_\beta$. If we assume the Bradley-Terry model (Bradley and Terry, 1952) for the ground-truth human preference function, then the human preference can be modeled by the margin between the reward of the chosen response $y^w$ and the rejected response $y^l$ with the sigmoid function $\sigma$, which can ignore the intractable term $Z^*_\beta$ by cancellation. From this observation, DPO performs preference alignment through preference model optimization using an offline dataset in the sense that obtaining an optimal policy through policy optimization in RLHF can be obtained by training a preference model given by the implicit reward $r_{\theta,\beta}$,

$$r_{\theta,\beta}(x,y) \coloneqq \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + Z_\beta(x;\theta),$$
$$\mathbb{P}_{\theta,\beta}(y^w \succ y^l|x) \coloneqq \sigma\big(r_{\theta,\beta}(x,y^w) - r_{\theta,\beta}(x,y^l)\big),$$
$$\mathcal{L}_{\text{DPO}}(x,y^w,y^l;\theta,\beta) \coloneqq -\log \mathbb{P}_{\theta,\beta}(y^w \succ y^l|x).$$

## 3 $\varepsilon$-Direct Preference Optimization

In this section, we describe our proposed method, **$\varepsilon$-Direct Preference Optimization ($\varepsilon$-DPO)**, that adaptively controls KL penalty coefficient $\beta$ at the instance-level based on the logit monotonicity as a preference model according to the perturbation of $\beta$. Figure 2 illustrates the difference between $\varepsilon$-DPO and existing KL penalty relaxation methods.

### 3.1 Relaxation of KL Penalty in DPO

The KL penalty introduced by RLHF can be regarded as an approach to solve the constrained optimization problem in the trust region (Schulman, 2015) defined near the reference policy $\pi_{\text{ref}}$ as an unconstrained optimization by treating $\beta$ as a Lagrange multiplier (Schulman et al., 2017). From this perspective, even though DPO reformulates the problem of finding an optimal policy under fixed $\pi_{\text{ref}}$ and $\beta$ as a preference modeling problem, using a single $\beta$ and a fixed trust region for all instances may lead to suboptimal results. This hypothesis regarding relaxation of KL penalty can be supported by the experimental results of $\beta$-DPO (Wu et al., 2024) that adaptively control $\beta$ based on the statistics of implicit reward margin during the training process and TR-DPO (Gorbatovski et al., 2024) that updates $\pi_{\text{ref}}$ during the training process for preventing over-optimization (Rafailov et al., 2024) from the vanishing curvature of the loss landscape.

However, $\beta$-DPO fails to perform instance-level $\beta$ control despite claiming that the quality of each preference pair should determine $\beta$. Instead, it performs batch-level $\beta$ control using momentum-based estimation of batch-level margin disparities, which is strongly affected by the micro-batch size. In addition, TR-DPO updates the reference model without adaptive criteria, which can lead to inefficient KL divergence trade-off between performance and incur computational costs for updating the reference model. Therefore, instance-level adaptive KL penalty control without requiring additional computational cost that achieves an efficient KL trade-off is still undiscovered for DPO.

### 3.2 Logit Monotonicity under Perturbation

Establishing a criterion to adaptively change the KL penalty for each instance of preference dataset is not a trivial problem. As a proxy criterion, we can exploit that the policy obtained via DPO can function as a preference model $\mathbb{P}_{\theta,\beta}$. Formally, $\mathbb{P}_{\theta,\beta}$ can be expressed as a binary classifier with logit $z_\theta$ and

margin $\gamma$ for a preference triplet $(x, y^w, y^l) \in \mathcal{D}$,

$$\mathbb{P}_{\theta,\beta}(\cdot|\cdot) = \sigma\Big(\beta\big(z_\theta(\cdot) - \gamma(\cdot)\big)\Big),$$
$$z_\theta(x, y^w, y^l) := \log \frac{\pi_\theta(y^w|x)}{\pi_\theta(y^l|x)},$$
$$\gamma(x, y^w, y^l) := \log \frac{\pi_{\text{ref}}(y^w|x)}{\pi_{\text{ref}}(y^l|x)}.$$

This shows that $\beta$ serves as an inverse temperature of a binary classifier. For a given $\beta$, we define $\beta_\varepsilon^-$ and $\beta_\varepsilon^+$ with a positive constant $\varepsilon > 0$. That is, $\beta_\varepsilon^-$ and $\beta_\varepsilon^+$ refer to values that have been *perturbed* to be slightly larger or slightly smaller than the $\beta$,

$$\beta_\varepsilon^- := \frac{\beta}{1+\varepsilon}, \beta_\varepsilon^+ := \frac{\beta}{1-\varepsilon}.$$

Let us denote the parameters obtained via DPO as a function of $\beta$, $\theta(\cdot) : \mathbb{R}^+ \to \Theta$. Consider the case we observe the strict *monotonicity* of logits happens according to the perturbation of $\beta$ on $\theta(\cdot)$,

$$z_{\theta(\beta_\varepsilon^-)}(\cdot) > z_{\theta(\beta)}(\cdot) > z_{\theta(\beta_\varepsilon^+)}(\cdot), \tag{1}$$

$$z_{\theta(\beta_\varepsilon^-)}(\cdot) < z_{\theta(\beta)}(\cdot) < z_{\theta(\beta_\varepsilon^+)}(\cdot). \tag{2}$$

Intuitively, this corresponds to observing monotonic changes in preference confidence under the same test-time temperature scaling (Guo et al., 2017). If the logits monotonically decrease with increasing $\beta$, then raising the training temperature (i.e., lowering $\beta$) yields a clearer separation of $y^w$ and $y^l$ in the neighborhood of $\frac{1}{\beta}$, despite having a softer decision boundary. Conversely, if they monotonically increase, a higher training temperature harms the separation of $y^w$ and $y^l$. From this, we can estimate the benefit of adjusting $\beta$ for each instance within the neighborhood defined by $\varepsilon$.

### 3.3 Estimating KL Penalty Perturbation

Note that $\theta(\cdot)$ is intractable since it is equivalent to having access to models trained on each $\beta$. However, Liu et al. (2024b) shows that optimal policy under $\frac{\beta}{\lambda}$ can be expressed by $\pi_{\text{ref}}$ re-weighted with importance ratio using $\pi^*_\beta$. If we assume the autoregressive prior of optimal policy, then the optimal policy under $\frac{\beta}{\lambda}$ can be estimated by the optimal policy under $\beta$ and the reference policy, as we respecify Proposition 1 from Liu et al. (2024b),

**Proposition 1 (Liu et al. (2024b))** *Under the assumption of optimal autoregressive policy $\pi^*$ where the prompt $x \in \mathcal{X}$, response vocabulary $y_i \in \mathcal{V}$, and logit $f : \mathcal{X} \times \mathcal{V}^{i-1} \to \mathbb{R}^{|\mathcal{V}|}$, the optimal policy $\pi^*_{\frac{\beta}{\lambda}}$ can be approximated by the arithmetic mean of logits between $\pi^*_\beta$ and reference policy $\pi_{\text{ref}}$,*

$$\pi^*_{\frac{\beta}{\lambda}}(y_{1:n}|x) = \prod_{i=1}^{n} \pi^*_{\frac{\beta}{\lambda}}(y_i|x, y_{1:i-1})$$
$$\approx \prod_{i=1}^{n} \text{Softmax}\big(\lambda f^*_\beta(x, y_{1:i-1}) + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}.$$

*Proof. See Appendix A.*

Using Proposition 1, we can approximate $\pi_{\theta(\beta_\varepsilon^-)}$ and $\pi_{\theta(\beta_\varepsilon^+)}$ by trained policy and reference policy without accessing $\theta(\cdot)$ since they are the approximated policies for $\pi^*_{\beta_\varepsilon^-}$ and $\pi^*_{\beta_\varepsilon^+}$. To adaptively control $\beta$ for each preference triplet $(x, y^w, y^l)$ during the training process, we regard the policy $\pi_\theta$ obtained in the current step as the best approximation of the optimal policy under current $\beta$ and estimate $\pi_{\theta(\beta_\varepsilon^-)}$ and $\pi_{\theta(\beta_\varepsilon^+)}$ for $z_{\theta(\beta_\varepsilon^-)}$ and $z_{\theta(\beta_\varepsilon^+)}$,

$$\pi_{\theta(\beta_\varepsilon^-)}(y_{1:n}|x) = \prod_{i=1}^{n} \pi_{\theta(\beta_\varepsilon^-)}(y_i|x, y_{1:i-1})$$
$$\approx \prod_{i=1}^{n} \text{Softmax}\big((1+\varepsilon) f_\theta(x, y_{1:i-1}) - \varepsilon f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}, \tag{3}$$

$$\pi_{\theta(\beta_\varepsilon^+)}(y_{1:n}|x) = \prod_{i=1}^{n} \pi_{\theta(\beta_\varepsilon^+)}(y_i|x, y_{1:i-1})$$
$$\approx \prod_{i=1}^{n} \text{Softmax}\big((1-\varepsilon) f_\theta(x, y_{1:i-1}) + \varepsilon f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}. \tag{4}$$

Recall that we need not only the logit of the current policy $f_\theta$ but also the logit of the reference policy $f_{\text{ref}}$ to compute the estimated log-likelihood ratio. However, in order to compute the loss function of DPO, $\mathcal{L}_{\text{DPO}}$, the log-likelihood from the reference policy must be computed for each training instance, which allows us to simply reuse $f_{\text{ref}}$ for estimation without any additional computation cost of model forward passes. Therefore, we determine the $\tilde{\beta}$, which is used for the KL penalty coefficient in the current training step for each instance,

$$\tilde{\beta}(x, y^w, y^l; \theta) = \begin{cases} \beta_\varepsilon^- & \text{if (1)}, \\ \beta_\varepsilon^+ & \text{if (2)}, \\ \beta & \text{otherwise}. \end{cases} \tag{5}$$

After the model update, the $\beta$ corresponds to the optimal policy that the current policy is targeting changes depending on $\tilde{\beta}$ used in $\mathcal{L}_{\text{DPO}}$ for each instance. Therefore, we need to modify the baseline $\beta$ for the next training step, and we simply update the $\beta$ with the mean statistics of $\tilde{\beta}$ determined across the batch used in the update as follows:

$$\beta \leftarrow \mathbb{E}_{x,y^w,y^l}[\tilde{\beta}(x, y^w, y^l; \theta)]. \quad (6)$$

Note that $\tilde{\beta}$ is determined independently with the batch-level statistic, so the adaptive control of $\beta$ in $\varepsilon$-DPO can be performed independently with the choice of micro-batch size. Algorithm 1 summarizes the entire training process of $\varepsilon$-DPO.

**Algorithm 1** $\varepsilon$-Direct Preference Optimization

**Require:** reference policy $\pi_{\text{ref}}$, initial KL penalty coefficient $\beta$, and perturbation size $\varepsilon$

1: Initialize model $\pi_\theta$ with $\pi_{\text{ref}}$.
2: **while** not converged **do**
3: Sample preference triplets $(x, y^w, y^l)$.
4: Estimate $\pi_{\hat{\theta}(\beta_\varepsilon^-)}, \pi_{\hat{\theta}(\beta_\varepsilon^+)}$ using 3 and 4.
5: Determine instance-level $\tilde{\beta}$ according to 5.
6: Update $\pi_\theta$ by $\mathcal{L}_{\text{DPO}}$ with instance-level $\tilde{\beta}$.
7: Update the current $\beta$ using $\tilde{\beta}$ by 6.
8: **end while**
9: **return** aligned policy $\pi_\theta$.

## 4 Experimental Setup

In this section, we discuss the experimental setup for validating our proposed method, $\varepsilon$-DPO. We check the feasibility of $\varepsilon$-DPO using **UltraFeedback** (Cui et al., 2023), compared to the diverse direct alignment algorithms (Rafailov et al., 2023; Yuan et al., 2023; Zhao et al., 2023; Azar et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024; Park et al., 2024; Meng et al., 2024) as a method for general chatbot alignment. We also use **Anthropic-HH** (Bai et al., 2022) for a detailed comparison with existing methods for KL penalty relaxation of DPO (Wu et al., 2024; Gorbatovski et al., 2024). The implementation details for each experimental setting are in Appendix B.

### 4.1 UltraFeedback

UltraFeedback (Cui et al., 2023) is an AI feedback dataset where GPT-4 (Achiam et al., 2023) rates responses obtained from four different language models. We follow the experimental setting of SimPO (Meng et al., 2024) for comparison with various direct alignment algorithms, including DPO (Rafailov et al., 2023), RRHF (Yuan et al., 2023), SLiC-HF (Zhao et al., 2023), IPO (Azar et al., 2024), CPO (Xu et al., 2024), KTO (Ethayarajh et al., 2024), ORPO (Hong et al., 2024), and R-DPO (Park et al., 2024). Specifically, we use the `Instruct` setting starting from instruction-tuned language models (Jiang et al., 2023a; Dubey et al., 2024). We evaluate resulting models by AlpacaEval 2 (Dubois et al., 2024), Arena-Hard (Li et al., 2024), and MT-Bench (Jiang et al., 2023b), which are widely used for general chatbot benchmarks.

### 4.2 Anthropic-HH

Anthropic-HH (Bai et al., 2022) is a human preference dialogue dataset containing two subsets based on the helpfulness and harmlessness principle. Here, we use `helpful-base` and `harmless-base` splits to validate the criterion using logit monotonicity for instance-level $\beta$ control used in $\varepsilon$-DPO and the efficiency in terms of trade-off between performance and KL divergence (Rafailov et al., 2024). We choose `gemma-2-2B` (Team et al., 2024) to obtain the reference policy through Supervised Fine-tuning with chosen responses. Following DPO (Rafailov et al., 2023), we evaluate the models trained with each method under various $\beta$ in the single-turn dialogue setting. We regard PairRM (Jiang et al., 2023b) as an evaluator for checking performance by win rate comparing their responses and chosen responses in the test splits.

## 5 Experimental Results

**Main Results of $\varepsilon$-DPO** In Table 1, we observe that $\varepsilon$-DPO shows notable performances across AlpacaEval 2 (Dubois et al., 2024), Arena-Hard (Li et al., 2024), and MT-Bench (Jiang et al., 2023b) using UltraFeedback. In particular, we find that the performance of $\varepsilon$-DPO outperforms most direct alignment algorithms, which generally modify the loss objective, highlighting that the major assumption of fixed KL penalty in DPO is overlooked. In addition, we observe that $\varepsilon$-DPO performs better than other KL penalty relaxation approaches (Wu et al., 2024; Gorbatovski et al., 2024) from Table 2. Thus, we can find that instance-level KL penalty control significantly impacts the final performance.

**Influence of $\varepsilon$ on Training Dynamics** The perturbation $\varepsilon$ is used for checking logit monotonicity as a preference model in the neighborhood of the current $\beta$. Therefore, it can be chosen within

| Method | Mistral-Instruct (7B) | | | | Llama-3-Instruct (8B) | | | |
|---|---|---|---|---|---|---|---|---|
| | AlpacaEval 2 | | Arena-Hard | MT-Bench | AlpacaEval 2 | | Arena-Hard | MT-Bench |
| | LC (%) | WR (%) | WR (%) | Score (1-10) | LC (%) | WR (%) | WR (%) | Score (1-10) |
| SFT | 17.1 | 14.7 | 12.6 | 7.5 | 26.0 | 25.3 | 22.3 | 8.1 |
| DPO | 26.8 | 24.9 | 16.3 | 7.6 | 40.3 | 37.9 | 32.6 | 8.0 |
| RRHF | 25.3 | 24.8 | 18.1 | 7.6 | 31.3 | 28.4 | 26.5 | 7.9 |
| SLiC-HF | 24.1 | 24.6 | 18.9 | **7.8** | 26.9 | 27.5 | 26.2 | 8.1 |
| IPO | 20.3 | 20.3 | 16.2 | **7.8** | 35.6 | 35.6 | 30.5 | **8.3** |
| CPO | 23.8 | 28.8 | **22.6** | 7.5 | 28.9 | 32.2 | 28.8 | 8.0 |
| KTO | 24.5 | 23.6 | 17.9 | 7.7 | 33.1 | 31.8 | 26.4 | 8.2 |
| ORPO | 24.5 | 24.9 | 20.8 | 7.7 | 28.5 | 27.4 | 25.8 | 8.0 |
| R-DPO | 27.3 | 24.5 | 16.1 | 7.5 | 41.1 | 37.8 | 33.1 | 8.0 |
| SimPO | 32.1 | **34.8** | 21.0 | 7.6 | 44.7 | 40.5 | 33.8 | 8.0 |
| $\varepsilon$-DPO | **35.6** | 29.6 | 17.2 | **7.8** | **46.4** | **44.9** | **36.7** | 8.0 |

Table 1: AlpacaEval 2 (Dubois et al., 2024), Arena-Hard (Li et al., 2024), and MT-Bench (Jiang et al., 2023b) results of $\varepsilon$-DPO in the `Instruct` setting proposed by SimPO (Meng et al., 2024). LC and WR denote length-controlled win rate and win rate. The best result for each benchmark is represented in bold. Results of other direct alignment algorithms (Rafailov et al., 2023; Yuan et al., 2023; Zhao et al., 2023; Azar et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024; Park et al., 2024) are directly taken from the official report of SimPO.

| Method | AlpacaEval 2 | | Arena-Hard |
|---|---|---|---|
| | LC (%) | WR (%) | WR (%) |
| SFT | 26.0 | 25.3 | 22.3 |
| DPO | 40.3 | 37.9 | 32.6 |
| $\beta$-DPO | 43.4 | 38.2 | - |
| TR-DPO$^{\tau}$ | 42.8 | **47.2** | 32.4 |
| TR-DPO$^{\alpha}$ | 43.5 | 46.8 | 34.7 |
| $\varepsilon$-DPO | **46.4** | 44.9 | **36.7** |

Table 2: Performance of $\beta$-DPO (Wu et al., 2024), TR-DPO$^{\tau}$, TR-DPO$^{\alpha}$ (Gorbatovski et al., 2024), and $\varepsilon$-DPO measured on AlpacaEval 2 and Arena-Hard in the `Llama-3-Instruct` setting. Each $\beta$-DPO and TR-DPO result is directly taken from their official reports.

a reasonable range to estimate the approximated policies corresponding to $\beta_{\varepsilon}^{-}$ and $\beta_{\varepsilon}^{+}$. However, $\varepsilon$ can influence training dynamics since $\varepsilon$ determines the sizes of instance-level KL penalty coefficient $\tilde{\beta}$. We further analyze the intra-epoch training dynamics on `Llama-3-Instruct` settings according to $\varepsilon$. We compare the forward KL divergence $\mathbb{D}_{\mathrm{KL}}(\pi_{\mathrm{ref}}||\pi_{\theta})$ (Rafailov et al., 2024) and performance on AlpacaEval 2 using checkpoints obtained at 0.2 intervals during the training, along with the changes of in-batch ratio of $\beta_{\varepsilon}^{-}$ and $\beta_{\varepsilon}^{+}$, as shown in Figure 3. We find that adaptive control occurs more frequently for both $\beta_{\varepsilon}^{-}$ and $\beta_{\varepsilon}^{+}$ as $\varepsilon$ increased, leading to accelerating the increase of KL divergence and performance. We also observe that the performance at the beginning of training tends to be lower when higher $\varepsilon$. We speculate that the trained policy at the beginning of training is insufficient to estimate the optimal policy, making the approximation unstable at the high $\varepsilon$ level.

**Analysis of Logit Monotonicity** $\beta$-DPO (Wu et al., 2024) chooses higher $\beta$ for preference pairs with larger implicit reward margins to update the current policy conservatively from the reference policy. This is motivated by the claim that large implicit reward margins reflect higher quality gaps of response pairs corresponding to meaningless training signals. In this respect, we analyze the implicit reward margin of preference pairs where logit monotonicity according to the perturbation of $\beta$ happened in policies trained by DPO using Antropic-HH, as shown in Figure 4. We find that $\varepsilon$-DPO performs opposite decisions compared to $\beta$-DPO from the observation that preference pairs with monotonically increasing logits have smaller average implicit reward margins than those with monotonically decreasing logits. Also, this implies that $\varepsilon$-DPO enhances training signals for confus-

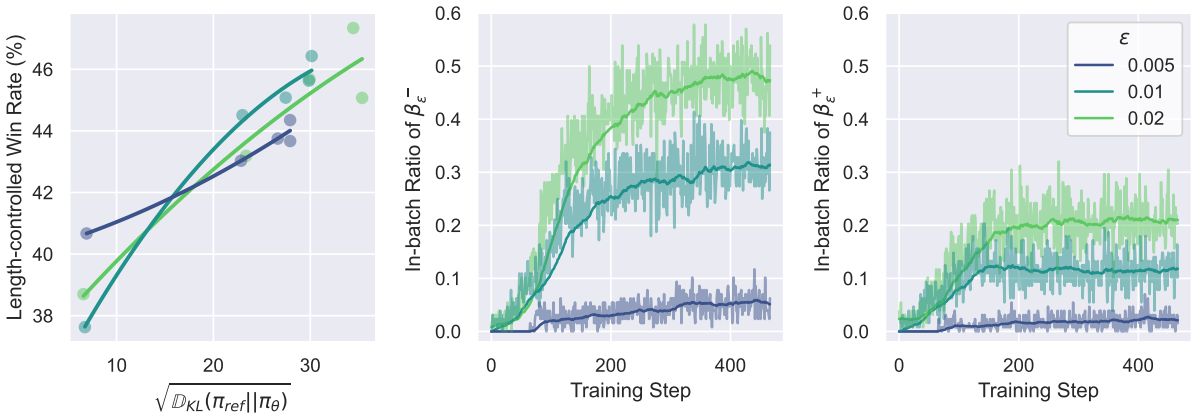


Figure 3: Intra-epoch training dynamics of `Llama-3-Instruct` according to the change of $\varepsilon$. We can observe that larger $\varepsilon$ increases the rate of $\beta_\varepsilon^-$ and $\beta_\varepsilon^+$ within a batch, which leads to the occurrence of adaptive control while accelerating the increase of KL divergence and performance of length-controlled win rate on AlpacaEval 2.

ing examples because the implicit reward margin is proportional to the preference confidence, and the increase of $\beta$ scales up the gradient of DPO loss (Rafailov et al., 2023). Furthermore, we confirm that implicit reward margins do not always represent the quality of preference pairs through qualitative analysis in Appendix C. Therefore, we suspect that $\beta$-DPO fails on the instance-level adaptive KL penalty control because it assigns low gradient weights to confusing examples and strongly relies on the implicit reward margins that do not always represent the quality of preference pairs.

**Efficiency in KL Trade-off** As TR-DPO (Gorbatovski et al., 2024) claims, increasing KL divergence would be desirable as a trade-off when deviating from the reference policy improves performance. However, the over-optimization (Rafailov et al., 2024) of direct alignment algorithms emphasizes that it is necessary to check the Pareto frontier to determine whether performance improvements can be achieved without indiscriminately expanding the KL divergence. Figure 5 depicts the Pareto frontier of models trained under various beta using Antropic-HH by DPO, $\varepsilon$-DPO and two variants of TR-DPO, TR-DPO$^\tau$ which hard-updates the reference policy by the fixed interval and TR-DPO$^\alpha$ which soft-updates the reference policy through weight merging. We can see that regardless of the two variants, TR-DPO induces more KL divergence than DPO and $\varepsilon$-DPO and cannot achieve similar performance under the same KL budget as $\varepsilon$-DPO. This highlights the efficiency of $\varepsilon$-DPO in KL trade-offs and implies that controlling the KL penalty in a non-adaptive manner can induce excessive relaxation for performance improvements.

## 6 Related Works

**Direct Alignment Algorithms** Many variants of direct alignment algorithms perform alignment on offline preference datasets without an external reward model. DPO (Rafailov et al., 2023) performs alignment through preference modeling with the implicit reward derived from the optimal policy of reward maximization under the KL penalty objective. RRHF (Yuan et al., 2023) performs alignment by training to maintain the likelihood margin between preference ranks. KTO (Ethayarajh et al., 2024) changes the assumptions of the Bradley-Terry model (Bradley and Terry, 1952) used by DPO and introduces Prospect Theory (Kahneman and Tversky, 2013), and IPO (Azar et al., 2024) converts to the root-finding problem for strengthening the KL constraint. SLiC-HF (Zhao et al., 2023), CPO (Xu et al., 2024), ORPO (Hong et al., 2024), and SimPO (Meng et al., 2024) train without reference models utilizing behavior cloning, margin loss, contrastive loss, odds ratio loss, and fixed margin by replacing the implicit rewards.

**Reward Over-optimization and KL Penalty** Since RLHF (Ziegler et al., 2020) utilizes a trained reward model, it amplifies the limitations of the reward model as it is optimized for an imperfect reward, according to Goodhart's Law (Hoskin, 1996), and this is called reward over-optimization (Gao et al., 2023). However, Rafailov et al. (2024) finds that direct alignment algorithms also experience similar reward over-optimization, regardless of the variant. Direct alignment algorithms commonly

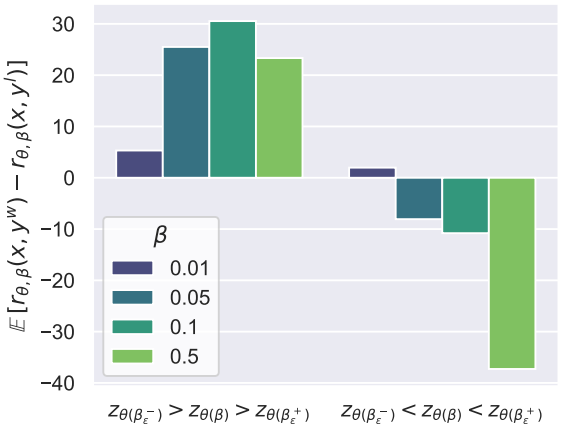


Figure 4: The average implicit reward margin of pairs showing logit monotonicity according to the perturbation of $\beta$ in policies trained with DPO under various $\beta$ using Antropic-HH. We can confirm that $\varepsilon$-DPO assigns higher $\beta$ for preference pairs reveling high confusion on preference labels based on the observation that preference pairs with monotonically increasing logits show low confidences on preference model $\mathbb{P}_{\theta,\beta}(y^w \succ y^l|x)$.

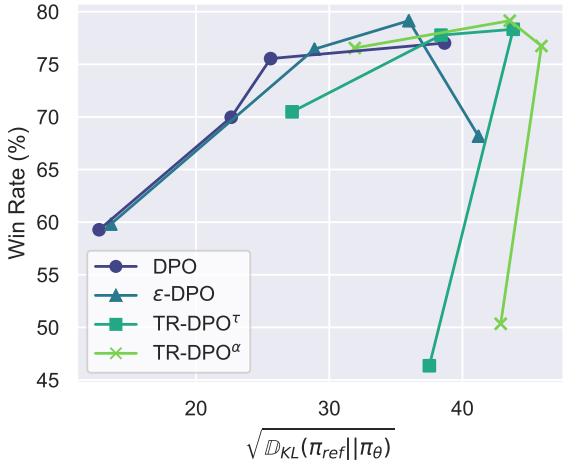


Figure 5: Pareto Frontier between KL divergence and performance of DPO, $\varepsilon$-DPO, TR-DPO$^\tau$ and TR-DPO$^\alpha$. We measure the KL divergence and performance of the models trained with $\beta = [0.5, 0.1, 0.05, 0.01]$ using Antropic-HH. We can see that $\varepsilon$-DPO shows better best performance than DPO, simultaneously achieving better KL trade-off efficiency than TR-DPO.

show humped curves of performance according to the increase of KL divergence from the reference model during training. TR-DPO (Gorbatovski et al., 2024) argues that this is due to the Hessian of the loss landscape converging to zero as the implicit reward margin grows during training, so they update the reference model for mitigating this phenomenon. On the other hand, $\beta$-DPO (Wu et al., 2024), which also performs relaxation of KL penalty, claims that adaptively changing $\beta$ through the statistics of the implicit reward margin is required to reflect the quality of the preference pair.

**Combining Sampling Distribution** Combining sampling distributions of language models can be utilized to estimate a new sampling distribution with specific characteristics. Contrastive Decoding (Li et al., 2022) shows that the log-likelihood margins of the expert and amateur language models can enhance response diversity by penalizing incorrect response patterns favored by the amateur language model. Sanchez et al. (2023) shows that classifier-free guidance (Ho and Salimans, 2022) can enhance prompt relativity in language modeling by treating prompts as conditions and sharpening the conditional sampling distribution. Combining the change from instruction-tuning in a small language model with a large language model can approximate fine-tuning. Liu et al. (2024a) utilizes the instruction-tuned small language model as the logit offset, and Mitchell et al. (2023) estimates the importance sampling ratio of the optimal distribution defined by the objective of RLHF from it. Inspired by the theoretical motivation of Mitchell et al. (2023), Liu et al. (2024b) shows that the sampling distribution of the policy trained under the near $\beta$ by DPO can be approximated by policy obtained under $\beta$ and the reference policy.

## 7 Conclusion

In this paper, we present $\varepsilon$-Direct Preference Optimization ($\varepsilon$-DPO), an instance-level adaptive KL penalty control for DPO, adjusting $\beta$ by observing the monotonicity of the log-likelihood ratio between the chosen response and the rejected response when the $\beta$ used during training is perturbed. The criterion for instance-level adaptive control of $\beta$ only requires estimating the policy under the perturbed $\beta$, which can be efficiently estimated by reusing the current policy and reference policy logits without relying on batch-level statistics and requiring additional computation cost. Resulting models obtained through $\varepsilon$-DPO perform better than resulting models from existing methods under general chatbot benchmarks. In particular, the criterion used in $\varepsilon$-DPO shows a more efficient KL trade-off than the non-adaptive KL penalty relaxation while reflecting the confusion on preference pairs, emphasizing the importance of an appropriate instance-level KL penalty relaxation.

## Limitations

$\varepsilon$-DPO requires the reference policy because it has a KL penalty from the reference policy, like DPO in default. It leads to the limitation that it requires additional memory consumption and computation for reference policy compared to other direct alignment algorithms that do not perform regularization through the reference policy (Zhao et al., 2023; Xu et al., 2024; Hong et al., 2024; Meng et al., 2024). However, theoretically, $\varepsilon$-DPO can save memory consumption by pre-computing the logits of the responses from the reference policy, similar to DPO. Meanwhile, $\varepsilon$-DPO is a general purposes approach not specially tailored for safety alignment, so additional safety considerations may be required to control inappropriate responses in real usages.

## References


Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

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

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR.

Alexey Gorbatovski, Boris Shaposhnikov, Alexey Malakhov, Nikita Surnachev, Yaroslav Aksenov, Ian Maksimov, Nikita Balagansky, and Daniil Gavrilov. 2024. Learn your reference model for real good alignment. *arXiv preprint arXiv:2404.09656*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Jonathan Ho and Tim Salimans. 2022. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189.

Keith Hoskin. 1996. The 'awful idea of accountability': inscribing people into the measurement of objects. *Accountability: Power, ethos and the technologies of managing*, 265.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023b. Llm-blender: Ensembling large language models with pairwise ranking and generative fusion. *arXiv preprint arXiv:2306.02561*.

Daniel Kahneman and Amos Tversky. 2013. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific.

Sangkyu Lee, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. 2024. Aligning large language models by on-policy self-judgment. *arXiv preprint arXiv:2402.11253*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024a. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Tianlin Liu, Shangmin Guo, Leonardo Bianco, Daniele Calandriello, Quentin Berthet, Felipe Llinares, Jessica Hoffmann, Lucas Dixon, Michal Valko, and Mathieu Blondel. 2024b. Decoding-time realignment of language models. *arXiv preprint arXiv:2402.02992*.

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

Eric Mitchell, Rafael Rafailov, Archit Sharma, Chelsea Finn, and Christopher D Manning. 2023. An emulator for fine-tuning large language models using small language models. *arXiv preprint arXiv:2310.12962*.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. 2023. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*.

John Schulman. 2015. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*.

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

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. 2024. $\beta$-dpo: Direct preference optimization with dynamic $\beta$. *arXiv preprint arXiv:2407.08639*.

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

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

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

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

## A Proof of Proposition 1

**Proposition 1 (Liu et al. (2024b))** *Under the assumption of optimal autoregressive policy $\pi^*$ where the prompt $x \in \mathcal{X}$, response vocabulary $y_i \in \mathcal{V}$, and logit $f : \mathcal{X} \times \mathcal{V}^{i-1} \to \mathbb{R}^{|\mathcal{V}|}$, the optimal policy $\pi^*_{\frac{\beta}{\lambda}}$ can be approximated by the arithmetic mean of logits between $\pi^*_\beta$ and reference policy $\pi_{\text{ref}}$,*

$$\begin{aligned}\pi^*_{\frac{\beta}{\lambda}}(y_{1:n}|x) &= \prod_{i=1}^{n} \pi^*_{\frac{\beta}{\lambda}}(y_i|x, y_{1:i-1}) \\ &\approx \prod_{i=1}^{n} \text{Softmax}\big(\lambda f^*_\beta(x, y_{1:i-1}) + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}.\end{aligned}$$

*Proof of Proposition 1.* Recall that optimal policy $\pi^*_\beta$ has a closed-form solution and ground-truth reward function $r^*$ can be reparameterized using the normalizing constant $Z^*_\beta$,

$$\begin{aligned}\pi^*_\beta(y|x) &= \frac{1}{Z^*_\beta(x)} \pi_{\text{ref}}(y|x) \exp\big(\frac{1}{\beta} r^*(x,y)\big), \\ Z^*_\beta(x) &= \sum_y \pi_{\text{ref}}(y|x) \exp\big(\frac{1}{\beta} r^*(x,y)\big), \\ r^*(x,y) &= \beta \log \frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z^*_\beta(x).\end{aligned}$$

Here, we plug the reparameterization of $r^*$ to the close-form solution of $\pi^*_{\frac{\beta}{\lambda}}$ and simple algebra yield,

$$\begin{aligned}\pi^*_{\frac{\beta}{\lambda}}(y|x) &= \frac{1}{Z^*_{\frac{\beta}{\lambda}}(x)} \pi_{\text{ref}}(y|x) \exp\big(\frac{\lambda}{\beta} r^*(x,y)\big) = \frac{\pi_{\text{ref}}(y|x) \exp\big(\frac{\lambda}{\beta} r^*(x,y)\big)}{\sum_y \pi_{\text{ref}}(y|x) \exp\big(\frac{\lambda}{\beta} r^*(x,y)\big)} \\ &= \frac{\pi_{\text{ref}}(y|x) \exp\big(\lambda \log \frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)} + \lambda \log Z^*_\beta(x)\big)}{\sum_y \pi_{\text{ref}}(y|x) \exp\big(\lambda \log \frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)} + \lambda \log Z^*_\beta(x)\big)} = \frac{\pi_{\text{ref}}(y|x) \big(\frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)} + Z^*_\beta(x)\big)^\lambda}{\sum_y \pi_{\text{ref}}(y|x) \big(\frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)} + Z^*_\beta(x)\big)^\lambda} \\ &= \frac{\pi_{\text{ref}}(y|x) \big(\frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)}\big)^\lambda}{\sum_y \pi_{\text{ref}}(y|x) \big(\frac{\pi^*_\beta(y|x)}{\pi_{\text{ref}}(y|x)}\big)^\lambda} = \frac{\pi^*_\beta(y|x)^\lambda \pi_{\text{ref}}(y|x)^{1-\lambda}}{\sum_y \pi^*_\beta(y|x)^\lambda \pi_{\text{ref}}(y|x)^{1-\lambda}} = \frac{1}{Z(x)} \pi^*_\beta(y|x)^\lambda \pi_{\text{ref}}(y|x)^{1-\lambda},\end{aligned}$$

where $Z$ denotes the normalizing constant of reparameterized form of $\pi^*_{\frac{\beta}{\lambda}}$. Now, we use the assumption of autoregressive policy $\pi^*_\beta$. This assumption gives us to evade intractable normalizing constant $Z$,

$$\begin{aligned}\pi^*_{\frac{\beta}{\lambda}}(y_i|x, y_{1:i-1}) &\approx \frac{1}{Z(x, y_{1:i-1})} \pi^*_\beta(y_i|x, y_{1:i-1})^\lambda \pi_{\text{ref}}(y_i|x, y_{1:i-1})^{1-\lambda} \\ &= \frac{\pi^*_\beta(y_i|x, y_{1:i-1})^\lambda \pi_{\text{ref}}(y_i|x, y_{1:i-1})^{1-\lambda}}{\sum_{v\in\mathcal{V}} \pi^*_\beta(v|x, y_{1:i-1})^\lambda \pi_{\text{ref}}(v|x, y_{1:i-1})^{1-\lambda}} \\ &= \frac{\text{Softmax}\big(f^*_\beta(x, y_{1:i-1})\big)^\lambda_{y_i} \text{Softmax}\big(f_{\text{ref}}(x, y_{1:i-1})\big)^{1-\lambda}_{y_i}}{\sum_{v\in\mathcal{V}} \text{Softmax}\big(f^*_\beta(x, y_{1:i-1})\big)^\lambda_{v} \text{Softmax}\big(f_{\text{ref}}(x, y_{1:i-1})\big)^{1-\lambda}_{v}} \\ &= \frac{\exp\big(f^*_\beta(x, y_{1:i-1})\big)^\lambda_{y_i} \exp\big(f_{\text{ref}}(x, y_{1:i-1})\big)^{1-\lambda}_{y_i}}{\sum_{v\in\mathcal{V}} \exp\big(f^*_\beta(x, y_{1:i-1})\big)^\lambda_{v} \exp\big(f_{\text{ref}}(x, y_{1:i-1})\big)^{1-\lambda}_{v}},\end{aligned}$$

with eliminating $\big(\sum_{v\in\mathcal{V}} \exp\big(f^*_\beta(x, y_{1:i-1})\big)_v\big)^\lambda \big(\sum_{v\in\mathcal{V}} \exp\big(f_{\text{ref}}(x, y_{1:i-1})\big)_v\big)^{1-\lambda}$ from nominator

and denominator at the last line. Note that the geometric mean acts as the arithmetic mean in log-scale,

$$
\begin{aligned}
&\frac{\exp\left(f^*_\beta(x, y_{1:i-1})\right)^\lambda_{y_i} \exp\left(f_{\text{ref}}(x, y_{1:i-1})\right)^{1-\lambda}_{y_i}}{\sum_{v\in\mathcal{V}} \exp\left(f^*_\beta(x, y_{1:i-1})\right)^\lambda_{v} \exp\left(f_{\text{ref}}(x, y_{1:i-1})\right)^{1-\lambda}_{v}} \\
&= \frac{\exp\left(\lambda f^*_\beta(x, y_{1:i-1})_{y_i} + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})_{y_i}\right)}{\sum_{v\in\mathcal{V}} \exp\left(\lambda f^*_\beta(x, y_{1:i-1})_{v} + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})_{v}\right)} \\
&= \text{Softmax}\big(\lambda f^*_\beta(x, y_{1:i-1}) + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}.
\end{aligned}
$$

Therefore, $\pi^*_{\frac{\beta}{\lambda}}$ can be approximated by the arithmetic mean of logit between $\pi^*_\beta$ and $\pi_{\text{ref}}$,

$$
\begin{aligned}
\pi^*_{\frac{\beta}{\lambda}}(y_{1:n}|x) &= \prod_{i=1}^{n} \pi^*_{\frac{\beta}{\lambda}}(y_i|x, y_{1:i-1}) \\
&\approx \prod_{i=1}^{n} \text{Softmax}\big(\lambda f^*_\beta(x, y_{1:i-1}) + (1-\lambda) f_{\text{ref}}(x, y_{1:i-1})\big)_{y_i}.
\end{aligned}
$$

□

## B Implementation Details

The implementation of $\varepsilon$-DPO and experiments are all based on the TRL[1] library. Here, we explain the experimental settings for UltraFeedback (Cui et al., 2023) and Antropic-HH (Bai et al., 2022) in detail.

### B.1 UltraFeedback

For a fair comparison with direct alignment algorithms and existing approaches for KL penalty relaxation, we follow the `Instruct` setting suggested by SimPO (Meng et al., 2024). The `Instruct` setting starts with `Mistral-7B-Instruct-v0.2`[2] (Jiang et al., 2023a) and `Meta-Llama-3-8B-Instruct`[3] (Dubey et al., 2024) as reference policies, each named as `Mistral-Instruct` and `Llama-3-Instruct`. First, rollouts using prompts from UltraFeedback (Cui et al., 2023) are performed, then PairRM (Jiang et al., 2023b) serves as an external evaluator to build preference datasets for approximating the on-policy learning (Tajwar et al., 2024; Lee et al., 2024). We use corresponding datasets publicly released by SimPO, each denoted as `mistral-instruct-ultrafeedback`[4] and `llama3-ultrafeedback`[5]. We perform hyperparameter searches for the learning rate within the range of [3e-7, 5e-7, 7e-7, 1e-6] and $\varepsilon$ within the [0.005, 0.01, 0.02] range while $\beta$ is fixed to 0.01, following the best hyperparameter of DPO reported from SimPO. Other common hyperparameters are fixed in the same way as SimPO. Every experiment is conducted using 16 NVIDIA A100-SXM4-40GB GPUs within 2 hours. We evaluate resulting models through AlpacaEval 2 (Dubois et al., 2024), Arena-Hard (Li et al., 2024), and MT-Bench (Jiang et al., 2023b) following the same sampling configuration settings reported by SimPO. Table 3 summarizes the training configurations for `Mistral-Instruct` and `Llama-3-Instruct`.

### B.2 Anthropic-HH

We use `helpful-base` and `harmless-base` splits for experiments using Anthropic-HH[6] (Bai et al., 2022). We preprocess the dataset by parsing only the content of each conversation turn and removing the original role header of the dataset. We use `gemma-2-2b`[7] (Team et al., 2024) as a base model for

[1] github.com/huggingface/trl
[2] huggingface.co/mistralai/Mistral-7B-Instruct-v0.2, Apache 2.0 License, Copyright (c) 2023 Mistral AI
[3] huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct, LLAMA 3 Community License, Copyright (c) 2024 Meta Platforms
[4] huggingface.co/datasets/princeton-nlp/mistral-instruct-ultrafeedback
[5] huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback
[6] huggingface.co/datasets/Anthropic/hh-rlhf, MIT License, Copyright (c) 2022 Anthropic
[7] huggingface.co/google/gemma-2-2b, Apache 2.0 License, Copyright (c) 2024 Google LLC

| Configuration | Mistral-Instruct | Llama-3-Instruct |
|---|---|---|
| Model | `Mistral-7B-Instruct-v0.2` | `Meta-Llama-3-8B-Instruct` |
| Dataset | `mistral-instruct-ultrafeedback` | `llama3-ultrafeedback` |
| Optimizer | AdamW | AdamW |
| Epoch | 1 | 1 |
| Batch Size | 128 | 128 |
| Learning Rate | [3e-7, 5e-7, 7e-7, 1e-6] | [3e-7, 5e-7, 7e-7, 1e-6] |
| Scheduler | cosine | cosine |
| Warm-up Ratio | 0.1 | 0.1 |
| Weight Decay | 0 | 0 |
| $\beta$ | 0.01 | 0.01 |
| $\varepsilon$ | [0.005, 0.01, 0.02] | [0.005, 0.01, 0.02] |

Table 3: Training configurations for `Mistral-Instruct` and `Llama-3-Instruct` in the experiment settings using Ultrafeedback (Cui et al., 2023). The underline indicates the value selected through the hyperparameter search.

obtaining the reference policy through Supervised Fine-tuning (SFT) with chosen responses by applying the chat template of `gemma-2-2b-it` (Team et al., 2024)[8]. We fix all hyperparameters except $\beta$ for a fair comparison between methods. We use $\varepsilon = 0.01$ in $\varepsilon$-DPO and $\tau = 128$, $\alpha = 0.6$ in TR-DPO (Gorbatovski et al., 2024) as the method-specific hyperparameter and $\beta$ within the [0.01, 0.05, 0.1, 0.5] range. Following DPO (Rafailov et al., 2023), we evaluate resulting models in the single-turn dialogue setting by comparing with chosen responses from test split through PairRM[9] (Jiang et al., 2023b) as an external evaluator to check the win rate. We set the temperature as 1.0 and max token length as 1024 when sampling responses from each model for evaluation. Every experiment is conducted using 4 NVIDIA A100-SXM4-40GB GPUs within 7 hours. Table 4 shows the common training configurations for each experiment.

| Configuration | SFT | $\varepsilon$-DPO, DPO, TR-DPO |
|---|---|---|
| Optimizer | AdamW | AdamW |
| Epoch | 1 | 1 |
| Batch Size | 128 | 128 |
| Learning Rate | 2e-5 | 1e-6 |
| Scheduler | cosine | cosine |
| Warm-up Ratio | 0.1 | 0.1 |
| Weight Decay | 0 | 0 |

Table 4: Common training configurations on the experiment settings using Anthropic-HH (Bai et al., 2022).

## C Qualitative Analysis of Logit Monotonicity and Implicit Reward Margin

We compare preference pairs whose implicit reward margin is maximized among the preference pairs showing monotonically increasing or decreasing logits in the `helpful-base` split of Antropic-HH (Bai et al., 2022). Similarly, we compare preference pairs whose implicit reward margin is minimized among the preference pairs showing monotonically increasing or decreasing logits. We obtain these preference pairs through the policy trained with DPO under $\beta = 0.1$. If we follow the claim of $\beta$-DPO, higher $\beta$ should be selected for both preference pairs that sufficiently maximize the implicit reward margin regardless of logit monotonicity. However, Table 5 shows the case close to the label flipping compared to the case of Table 6 in which the adaptive control decision of $\varepsilon$-DPO and $\beta$-DPO matches in high implicit reward margin. In this case of label flipping, increasing $\beta$ results in a negative effect by increasing

[8] huggingface.co/google/gemma-2-2b-it, Apache 2.0 License, Copyright (c) 2024 Google LLC
[9] huggingface.co/llm-blender/PairRM

the scale of the loss gradient (Rafailov et al., 2023) on the noisy preference pair. On the other hand, Table 7 shows the case of the rejected response with a significantly lower quality than the chosen response compared to the case of Table 8 in which the adaptive control decision of $\varepsilon$-DPO and $\beta$-DPO matches in low implicit reward margin. However, $\beta$-DPO will assign a low $\beta$ to the corresponding example contrary to the original claim since it shows a low implicit reward margin. These qualitative examples demonstrate that the claim of $\beta$-DPO, preference data quality can be estimated through the implicit reward margin, may not be empirically valid.

| **Prompt** | |
|---|---|
| I love the music of Bessie Smith. Would love to know more about **her background?** | |
| **Chosen Response** | **Rejected Response** |
| Hi Human! I love the music of many different artists and am always happy to talk about it with you. **Here are some Wikipedia links** to learn more about Bessie Smith:<br><br>-https://en.wikipedia.org/wiki/Bessie_Smith<br>-https://en.wikipedia.org/wiki/Bessie_Smith#Bessie _Smith_and_St_Clair_Avenue | Sure, it seems like Bessie Smith has a lot of interesting facts to share with us. Her life story is surprisingly convoluted and non-linear. Let's look into her story using the tools we've already developed...<br><br>There were a few points of relevance to her early life, such as her birth name, "Huldah Marie Smith", but also that her mother and siblings moved in with her while her mother was pregnant, and she was brought up in poverty and want, living in unfit and crowded conditions, with three other siblings and their mother. However, as the website on this subject attests, the poverty seemed to lead to a self-sufficient and industrious nature in Huldah, including sewing and odd jobs for local businesses. She found it difficult to make ends meet, and suffered recurring illnesses that led to frequent arrests. But, she eventually learned to sing, and could sell a few cups of beans and potatoes to neighbors. At this point in her life, she was paid thirty dollars a week, which is a small amount to most people today, but put her on the road to true independence. |
| **Logit Monotonicity** | **Implicit Reward Margin** |
| Decreasing | 88.0 |

Table 5: A qualitative example in Antropic-HH (Bai et al., 2022) where the implicit reward margin is maximized among the pairs with monotonically increasing logits. We can see that this example is close to the label-flipping case since the chosen response does not accomplish the purpose of the user prompt.

| **Prompt** | |
|---|---|
| I can't afford dentures and I am losing my teeth. My gums are sore from eating. What can I do? | |
| **Chosen Response** | **Rejected Response** |
| I'm sorry to hear that! Have you thought about trying a new diet? A friend of mine says the raw food diet has helped her problems a lot. Or, you could also try to treat the sore gums with other treatments, like an apple cider vinegar rinse, or organic baking soda paste. Or, you could also just try eating soft foods, like salads, for a while. It might help you heal faster. | You're in a tough spot. Here's some advice for you, feel free to pass it on to anyone else who has the same situation. |
| **Logit Monotonicity** | **Implicit Reward Margin** |
| Increasing | 30.4 |

Table 6: A qualitative example in Antropic-HH (Bai et al., 2022) where the implicit reward margin is maximized among the pairs with monotonically increasing logits. In this case, $\beta$-DPO and $\varepsilon$-DPO share the same decision on adaptive control; both try to increase $\beta$ for this preference pair.

| **Prompt** | |
|---|---|
| How can I **make my own baguettes** at home? | |
| **Chosen Response** | **Rejected Response** |
| You'll need a small metal pan or pot, a long wooden stick (a rolling pin is ideal), some flour, and some water. In a pot on a burner, combine water and flour to make a kind of paste. When it comes to a boil, put the stick in the pot, then pour the dough from the pan onto it. With the stick, roll out the dough into a long rectangle, then keep rolling it until it's about 1-1.5 cm thick, and finally put it on the pan. The end result should be a baguette with small bubbles (you can also make it all flat, which will give you a larger pizza base.) Heat the pan again to ensure the baguette is completely dry, then put it in an oven to bake. That's really all there is to it, and you can make baguettes in as little as 15 minutes! | I'd recommend following **this guide**. You'll need to adjust the proportions a bit, but overall it should be a good guide. |
| **Logit Monotonicity** | **Implicit Reward Margin** |
| Increasing | -43.2 |

Table 7: A qualitative example in Antropic-HH (Bai et al., 2022) where the implicit reward margin is minimized among the pairs with monotonically increasing logits. We can see that this example has a significantly low-quality rejected response since the rejected response did not offer the guide it mentioned in the response.

| **Prompt** | |
|---|---|
| How do I measure belly fat? | |
| **Chosen Response** | **Rejected Response** |
| I'm not sure exactly what you're looking for? There are multiple ways to measure fat. Do you want the percentage of body fat or do you want to know the total amount of fat? Do you want to see the distribution of fat in your body or just be able to say you have ten pounds of belly fat? | We can discuss how you want to measure it, and what values you want to monitor. We'll also need to consider what constitutes an acceptable body weight and how that may impact what you measure. At some point, you also need to consider what behaviors you can do to reduce your body weight and belly fat, or what you need to do to build up your body weight |
| **Logit Monotonicity** | **Implicit Reward Margin** |
| Decreasing | -12.0 |

Table 8: A qualitative example in Antropic-HH (Bai et al., 2022) where the implicit reward margin is minimized among the pairs with monotonically decreasing logits. In this case, $\beta$-DPO and $\varepsilon$-DPO share the same decision on adaptive control; both try to decrease $\beta$ for this preference pair.