

# Filtered Direct Preference Optimization

Tetsuro Morimura<sup>#</sup>, Mitsuki Sakamoto<sup>#</sup>, Yuu Jinnai, Kenshi Abe, Kaito Ariu  
 CyberAgent, Tokyo, Japan  
 {morimura\_tetsuro,sakamoto\_mitsuki,jinnai\_yu,abe\_kenshi,kaito\_ariu}@cyberagent.co.jp  
<sup>#</sup> Equal contribution

## Abstract

Reinforcement learning from human feedback (RLHF) plays a crucial role in aligning language models with human preferences. While the significance of dataset quality is generally recognized, explicit investigations into its impact within the RLHF framework, to our knowledge, have been limited. This paper addresses the issue of text quality within the preference dataset by focusing on Direct Preference Optimization (DPO), an increasingly adopted reward-model-free RLHF method. We confirm that text quality significantly influences the performance of models optimized with DPO more than those optimized with reward-model-based RLHF. Building on this new insight, we propose an extension of DPO, termed filtered direct preference optimization (fDPO). fDPO uses a trained reward model to monitor the quality of texts within the preference dataset during DPO training. Samples of lower quality are discarded based on comparisons with texts generated by the model being optimized, resulting in a more accurate dataset. Experimental results demonstrate that fDPO enhances the final model performance. Our code is available at <https://github.com/CyberAgentAILab/filtered-dpo>.

## 1 Introduction

Large language models (LLMs) have become pivotal in performing various language processing tasks, such as text generation, dialogue, and summarization (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2023). Aligning these models with human preferences and ethical standards is paramount to ensuring they are practical, trustworthy, and socially accepted (Bender et al., 2021; Bommasani et al., 2022). Reinforcement learning from human feedback (RLHF) is developed to tackle this challenge, aiming to enhance LLM performance by leveraging human feedback (Ouyang et al., 2022; Bai et al., 2022; Lin et al., 2022; Touvron et al., 2023; Casper et al., 2023).

RLHF operates by taking a preference dataset and a language model (LM) as inputs to produce an LM refined by these preferences (Ouyang et al., 2022). It is broadly divided into two approaches concerning the use of a reward model (RM): RM-based RLHF, which learns an RM from the preference dataset and then uses it to optimize an LM through reinforcement learning (RL), and an RM-free approach that directly adjusts an LM based on preference data. This division mirrors the distinction between offline model-based and model-free RL (Sutton & Barto, 2018).<sup>1</sup> Each approach offers unique advantages and requires careful application based on specific goals and contexts. For instance, in scenarios with limited data, model-based RL might be preferable due to its data efficiency, though its computational cost is generally higher than that of model-free RL (Moerland et al., 2022; Levine et al., 2020). Consequently, RM-based RLHF may be more effective in leveraging data than RM-free methods.

<sup>1</sup>RM-based RLHF first estimates the environment (specifically, the reward model; we do not need to estimate a state transition model because it is typically predefined in NLG tasks) and then optimizes an LM under the estimated environment. This approach is in itself a form of model-based RL.

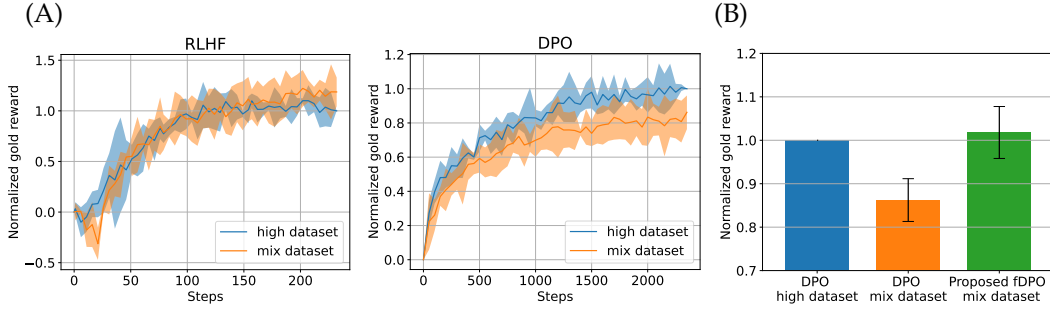


Figure 1: Performance comparison of alignment methods using a 160M LM with the AlpacaFarm dataset (Dubois et al., 2023). (A) shows the impact of dataset quality on RLHF (Ouyang et al., 2022) and DPO (Rafailov et al., 2023), with DPO exhibiting greater sensitivity to dataset quality variations. The normalized gold rewards are adjusted to scale the initial average reward to 0 and the post-optimization average reward to 1 for clarity. The initial model’s average gold reward was  $-2.95$ , while the average rewards for RLHF and DPO were  $-1.49$  and  $-2.31$ , respectively. (B) compares the performance of DPO and the proposed fdPO on a mixed-quality dataset, illustrating that fdPO effectively mitigates the impact of data quality variations.

Direct preference optimization (DPO) is a representative method of the RM-free RLHF (Rafailov et al., 2023). DPO reformulates the RL problem as a type of supervised learning problem, bypassing key challenges in RM-based RLHF, such as the need for reward modeling and balancing exploration and exploitation in RL fine-tuning. Thus, DPO simplifies the learning process. However, this approach relies solely on the initially given preference dataset for training, similar to supervised learning. This reliance might make DPO more sensitive to the quality of the preference dataset, potentially more so than other RLHF methods.

In this paper, we explore the impact of preference dataset quality on the performance of LMs optimized by DPO, specifically focusing on the quality of response texts rather than labeling accuracy. We demonstrate that DPO is more affected by text quality variations within the dataset than typical RLHF methods, as shown in Figure 1 (A). Notably, we observe that lower-quality data can create performance bottlenecks. In response to this challenge, we introduce a novel approach named filtered direct preference optimization (fdPO), which aims to harness potential data efficiency advantages of RM-based RLHF. It uses a trained RM to identify and discard samples of lower quality than those generated by an LM during fine-tuning. Our experiments, conducted with 160M and 1.4B LMs, show that fdPO significantly enhances the effectiveness of DPO, as illustrated in Figure 1 (B).

For simplicity, we will henceforth refer to RM-based RLHF simply as RLHF, unless a distinction is necessary. This study’s contributions are threefold:

- We confirm that the quality of the preference dataset significantly influences the performance of LMs optimized with DPO whereas it has less impact on LMs optimized by standard RLHF.
- We introduce fdPO, a practical solution that uses an RM to identify and discard lower-quality data, effectively addressing the dataset quality issue.
- Our experiments on instruction following tasks demonstrate that fdPO substantially enhances the performance of LMs.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 delves into the background. In Section 4, we detail the proposed method, fdPO, explaining its mechanisms and the rationale behind its design. Section 5 presents the experimental results, illustrating the effectiveness of fdPO and its impact on LM performance. Finally, Section 6 concludes the paper, summarizing our findings, discussing limitations, and suggesting directions for future research.

---

## 2 Related work

We examine methods for aligning LMs with human preferences, focusing on RLHF and its alternatives. Most RLHF approaches utilize an RM (Ouyang et al., 2022; Touvron et al., 2023; Dubois et al., 2023; Casper et al., 2023). These methods fine-tune LMs using RL algorithms such as REINFORCE (Williams, 1992; Rennie et al., 2017), proximal policy optimization (PPO) (Schulman et al., 2017), or their variants Sutton & Barto (2018). However, there are notable reinforcement-learning-free approaches (Zhao et al., 2023; Liu et al., 2024), and learning-free methods that leverage the RM at decoding time, with best-of-N (BoN) sampling being a prominent example (Stiennon et al., 2020; Nakano et al., 2021).

A significant challenge in these methods is the estimation error of RMs, which can lead LMs to overfitting to a proxy reward, a phenomenon termed RM overoptimization (Gao et al., 2023). Various strategies have been proposed to address this issue, including RM ensembles (Coste et al., 2023; Eisenstein et al., 2023), uncertainty evaluation (Zhang et al., 2024), and analysis of out-of-distribution (Pikus et al., 2023; Kirk et al., 2024). Pace et al. (2024) proposes using BoN sampling to improve the data used for reward modeling, which is relevant to our fDPO approach focusing on dataset quality. As fDPO also leverages an RM, it can benefit from these developments.

DPO and its generalizations (Azar et al., 2023; Tang et al., 2024) represent significant RM-free methods. Some DPO variants explore different regularizations (Wang et al., 2024) or use a divided dataset for stepwise training (Gou & Nguyen, 2024). Other variants propose adding an offset to the DPO objective function (Amini et al., 2024) or a curriculum learning (Gou & Nguyen, 2024), by evaluating the quality difference between chosen and rejected responses. This focus on response quality is relevant to our approach.

Despite various advancements in DPO, our study is the first, to our knowledge, to explore its significant dependence on preference dataset quality and to attempt to refine the dataset for better performance. Additionally, our proposed fDPO method complements these developments. Integrating fDPO with these methods is an exciting possibility for future work, potentially leading to even more effective ways to align LMs with human preferences.

## 3 Background

This section delves into RLHF in Section 3.1 and explores DPO in Section 3.2.

### 3.1 Reinforcement Learning from Human Feedback

Reinforcement learning from human feedback (RLHF) frames the application of human feedback to enhance performance of a language model (LM) within the context of an RL problem. The process incorporates a pre-trained LM  $\pi_\theta(y|x)$ , with  $\theta$  denoting model parameters,  $x$  the prompt, and  $y$  the associated response. It also includes a demonstration dataset  $\mathcal{D}_{\text{demo}}$  for initial supervised fine-tuning and a preference dataset  $\mathcal{D}$  for further RL fine-tuning. The aim is to refine the LM  $\pi_\theta$  with these datasets  $\mathcal{D}_{\text{demo}}$  and  $\mathcal{D}$ . We will present an overview of the widely studied RLHF pipeline (Ouyang et al., 2022), establishing the notations and concepts for understanding our contributions. The RLHF pipeline comprises three principal phases: (1) supervised fine-tuning, (2) reward modeling, and (3) RL fine-tuning.

**Supervised fine-tuning.** Supervised fine-tuning (SFT) refines a pre-trained LM  $\pi_\theta$  through supervised learning using demonstration data  $\mathcal{D}_{\text{demo}}$  from downstream tasks such as dialogue, instruction following, or summarization. This step steers  $\pi_\theta$  towards desirable responses  $y$  given prompts  $x$ , laying the groundwork for the more complex RL fine-tuning steps in the RLHF pipeline. The resulting LM is called the SFT model.

**Reward Modelling.** The reward modeling phase constructs a reward model (RM)  $r_\phi(x, y)$  with a parameter  $\phi$  to capture human preferences. This is achieved using a preference dataset,  $\mathcal{D} = \{(x^{(i)}, y_c^{(i)}, y_r^{(i)})\}_{i=1}^N$ , where for each prompt  $x$ ,  $y_c$  denotes the response chosen

by a human, and  $y_r$  is the rejected response. The variable  $N$  denotes the total number of samples in the dataset.

To estimate the probability that a given response is preferred over another, the RM  $r_\phi$  utilizes the Bradley-Terry model (Bradley & Terry, 1952), which is formulated as:

$$p_{\text{BT}}(y_c \succ y_r | x, r_\phi) = \frac{\exp(r_\phi(x, y_c))}{\exp(r_\phi(x, y_c)) + \exp(r_\phi(x, y_r))} = \sigma(r_\phi(x, y_c) - r_\phi(x, y_r)),$$

where  $\sigma(x) = \frac{1}{1+\exp(-x)}$  is the sigmoid function. The RM is trained by minimizing the following negative log-likelihood of the observed preferences in the dataset:

$$L(\phi) = -\mathbb{E}_{(x, y_c, y_r) \sim \mathcal{D}} [\log \sigma(r_\phi(x, y_c) - r_\phi(x, y_r))] \quad (1)$$

This training process aims to assign higher scores to responses that humans prefer, thus enhancing the RM’s ability to predict human-like responses.

**RL fine-tuning.** The RL fine-tuning phase uses the learned RM  $r_\phi$  to optimize the SFT model  $\pi_\theta$ . The goal is to enhance  $\pi_\theta$  by maximizing the expected reward while maintaining closeness to the reference LM  $\pi_{\text{ref}}$ , striking a balance that avoids large deviations from the pre-trained behavior. The SFT model before RL fine-tuning is often used as  $\pi_{\text{ref}}$ . This is achieved through policy gradient methods like proximal policy optimization (PPO) (Schulman et al., 2017). The optimization objective is formalized as

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

where  $D_{\text{KL}}$  is Kullback–Leibler (KL) divergence of a distribution  $p$  from another distribution  $q$ , defined as

$$D_{\text{KL}}(p, q) = \mathbb{E}_{x \sim p} \left[ \log \frac{p(x)}{q(x)} \right].$$

Here,  $\beta$  is a hyperparameter that controls the penalty for the deviations from  $\pi_{\text{ref}}$ .

### 3.2 Direct Preference Optimization

Direct preference optimization (DPO) reformulates the above reward modeling and RL fine-tuning phases to a single optimization problem (Rafailov et al., 2023). While DPO essentially follows the same loss function under the Bradley-Terry model (Eq. 1), it is an RM-free approach that aligns the SFT model  $\pi_\theta$  directly with the preference data.

The objective function of DPO is defined as follows: aiming to maximize the ratio of probabilities for the chosen responses, optimizing the LM to imitate human preferences:

$$L_{\text{DPO}}(\theta) = \mathbb{E}_{(x, y_c, y_r) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_c | x)}{\pi_{\text{ref}}(y_c | x)} - \beta \log \frac{\pi_\theta(y_r | x)}{\pi_{\text{ref}}(y_r | x)} \right) \right], \quad (3)$$

where  $\beta$  is a hyperparameter and has a similar role in Eq. (2). As the objective function indicates, DPO simplifies the optimization process by not requiring the generation of responses  $y$  from  $\pi_\theta$  during training, unlike the standard RL fine-tuning of Eq. (2). This approach, akin to supervised learning, makes DPO accessible and easy to use.

## 4 Filtered Direct Preference Optimization

In this section, we propose an approach called filtered direct preference optimization (fDPO), which refines the dataset used in DPO. The principle of fDPO is straightforward: it aims to discard lower-quality samples compared to those generated by the LM, as determined by an proxy RM. This strategy is intuitively derived from observing that lower-quality data can create performance bottlenecks in DPO. First, we give an implementation of fDPO in Section 4.1. Then, we will elaborate on the motivation of fDPO by analyzing DPO’s behavior in Section 4.2.

## 4.1 fDPO Implementation

fDPO needs to assess the quality of responses. For this purpose, a straightforward approach is to use an RM. This incorporation of an RM diverges from the RM-free nature of the original DPO, aligning fDPO closer to RM-based RLHF approaches and making DPO more effective in leveraging data.

Algorithm 1 details the pseudo-code for fDPO implementation, which follows the standard RLHF pipeline in Section 3.1 except for RL fine-tuning. Instead of RL fine-tuning, DPO fine-tuning with filtering is employed. At the start of each training epoch in Step 3, the quality of each sample in the preference dataset is evaluated with a trained RM  $r_\phi$ . Samples with chosen responses deemed to be of lower quality than those the LM  $\pi_\theta$  generates are discarded. Specifically, for each prompt  $x$  in the dataset,  $\pi_\theta$  generates a response  $y$ , and  $r_\phi$  scores  $y$  and the chosen response  $y_c$ . If the score of  $y$  is higher than that of  $y_c$ , the corresponding sample  $(x, y_c, y_r)$  is excluded from training.

The learning process itself mirrors that of DPO but introduces the aforementioned data refinement step. This refinement step aims to create a more effective training dataset, thereby improving the LM’s alignment with human preferences.

---

### Algorithm 1 filtered Direct Preference Optimization (fDPO)

---

**Require:** Pre-trained LM  $\pi_\theta$ , RM  $r_\phi$ , demonstration dataset  $\mathcal{D}_{\text{demo}}$ , preference dataset  $\mathcal{D}_{\text{pref}}$ , maximum epoch  $M$ , and maximum filtered rate  $\gamma$ .

- 1: *Step 1: Supervised fine-tuning.* Train  $\pi_\theta$  on  $\mathcal{D}_{\text{demo}}$ .
- 2: *Step 2: Reward modeling.* Train  $r_\phi$  on  $\mathcal{D}_{\text{pref}}$  (see Eq. (1)).
- 3: *Step 3: DPO fine-tuning with filtering.*
- 4: Initialize filtered-preference dataset  $\mathcal{D}_f := \mathcal{D}_{\text{pref}}$ , epoch number  $m := 0$ .
- 5: **while**  $m < M$  and  $|\mathcal{D}_f|/|\mathcal{D}_{\text{pref}}| > 1 - \gamma$  **do**
- 6:   **for each**  $(x, y_c, y_r)$  in  $\mathcal{D}_f$  **do**
- 7:     Generate response  $y$  by LM  $\pi_\theta$  given prompt  $x$ .
- 8:     **if**  $r_\phi(x, y) > r_\phi(x, y_c)$  **then**
- 9:       Discard  $(x, y_c, y_r)$  from  $\mathcal{D}_f$ .
- 10:    **end if**
- 11:   **end for**
- 12:   Update LM  $\pi_\theta$  on  $\mathcal{D}_f$  for one epoch using DPO.
- 13:   Increment epoch number  $m := m + 1$ .
- 14: **end while**
- 15: **return** Optimized LM  $\pi_\theta$ .

---

## 4.2 Background and Motivation for fDPO

The motivation for developing fDPO stems from the observation that the quality of data in DPO significantly affects the performance of the resulting LM. More specifically, upon differentiating the objective function of DPO in Eq. 3, we obtain

$$\nabla_\theta L_{\text{DPO}}(\theta) = -\mathbb{E}_{(x, y_c, y_r) \sim \mathcal{D}} \left[ \underbrace{w_\theta(x, y_c, y_r) \nabla_\theta \log \pi_\theta(y_c | x)}_{\text{increase likelihood of } y_c} - \underbrace{w_\theta(x, y_c, y_r) \nabla_\theta \log \pi_\theta(y_r | x)}_{\text{decrease likelihood of } y_r} \right], \quad (4)$$

where  $w_\theta(x, y_c, y_r) = \sigma \left( \beta \log \frac{\pi_\theta(y_c | x)}{\pi_{\text{ref}}(y_c | x)} - \beta \log \frac{\pi_\theta(y_r | x)}{\pi_{\text{ref}}(y_r | x)} \right)$  is a weight function.

Equation (4) highlights that DPO, while adaptively adjusting sample weights, inherently aims to increase the generation probability for chosen responses and decrease it for rejected ones. This approach can lead to two types of problems: 1) diminished generation probability for high-quality responses labeled as rejected, and 2) increased generation probability for low-quality responses labeled as chosen.



---

Concerns regarding the first case, where high-quality responses are classified as rejected, might be insignificant. In such a case, while the generation probabilities of several high-quality responses decrease, the capability of LMs could remain robust. This is because their extensive diversity of potential responses will ensure that suppressing some responses does not substantially reduce the LM’s capacity to generate other high-quality alternatives.

Conversely, the more critical issue arises when low-quality responses are labeled as chosen. In such cases, their generation probabilities increase. This increase is particularly problematic because the probabilities of potential responses sum to one, meaning an increase in the probability of low-quality responses invariably decreases the share of high-quality responses. This shift substantially directs the learning process toward suboptimal outputs and declines the overall performance of LMs. A more detailed analysis of the sensitivity comparison between chosen and rejected responses will be provided in Appendix A.1.

Building upon these insights, fDPO uniquely addresses the issue of increased generation probability for low-quality chosen responses. It tackles these bottlenecks by discarding samples where the chosen responses are of lower quality compared to those generated by the LM  $\pi_\theta$ , as evaluated according to an RM. Through this process of consistent refinement, fDPO performs DPO on the improved dataset, thereby enhancing DPO’s effectiveness and ensuring a more effective alignment with human preferences.

## 5 Experiments

We first detail our experimental setup, including the pretrained models in Section 5.1 and the data in Section 5.2. We then evaluate the impact of data quality on DPO in Section 5.3 and the effectiveness of fDPO in Sections 5.4 and 5.5 in the context of instruction following tasks, focusing on the ability to generate appropriate responses to prompts. For our baseline comparison, we include a DPO implementation from the Transformer Reinforcement Learning (TRL) library.<sup>2</sup>

### 5.1 Pretrained Models

In our experiments, we employed pretrained LMs provided in the Pythia suite by [Biderman et al. \(2023\)](#) of two different sizes: 1.4B and 160M models. In Section 5.4, we present the primary findings of our study using a 1.4B-sized LM to highlight the effectiveness of fDPO in an LLM. Due to computational resource constraints, a comprehensive examination of the 160M LM is provided in Section 5.5. In the preliminary setup, each LM was subjected to SFT using the demonstration dataset from AlpacaFarm ([Dubois et al., 2023](#)). These prepared LMs, denoted as  $\pi_\theta$ , were then used as the initial LMs for our experiments.

For the (proxy) RM, we used Pythia models of varying sizes: 14M, 70M, and 160M models. To circumvent the high costs associated with human evaluation, similar to other studies [Dubois et al. \(2023\)](#); [Rafailov et al. \(2023\)](#), we utilized a large-scale human preference model as the gold RM. Specifically, “OpenAssistant/reward-model-deberta-v3-large-v2”<sup>3</sup> model was employed for this purpose.

### 5.2 Data

We utilized the AlpacaFarm instruction dataset ([Dubois et al., 2023](#)) for our experiments. The dataset consisted of 20,000 training samples. For evaluation, a random selection of 200 samples from in a pool of 2000 was used in each independent run to ensure consistency and unbiased results.

The baseline DPO and our proposed fDPO used the same data to ensure a fair comparison. This means that in fDPO, both the RM and the LM were trained using an identical dataset.

---

<sup>2</sup><https://github.com/huggingface/trl>

<sup>3</sup><https://huggingface.co/OpenAssistant/reward-model-deberta-v3-large-v2>

---

Given our focus on dataset quality, we employed gold RM and BoN sampling (Stiennon et al., 2020; Nakano et al., 2021) to create three types of pairwise preference datasets:

**Low-quality dataset.** This dataset was created using the conventional manner. For each prompt  $x$ , the LM  $\pi_\theta$  generated two responses. These responses were then evaluated by the gold RM, with the higher-scoring response designated as  $y_c$  and the other as  $y_r$ . This formed the preference dataset  $\mathcal{D}$  samples  $(x, y_c, y_r)$ . For brevity, this dataset is referred to as the ‘low dataset’.

**High-quality dataset.** Adopting the approach from (Pace et al., 2024), we used BoN sampling to create responses of higher quality. Specifically, for each prompt  $x$ , the LM  $\pi_\theta$  generated 16 responses. These responses were then evaluated by the gold RM, and the highest-scoring response was selected as  $y_c$ , with one randomly selected from the remaining 15 responses labeled as  $y_r$ . Due to the probabilistic nature of outputs of  $\pi_\theta$ , this approach is likely to yield  $y_c$  responses of higher quality (as indicated by gold RM scores) compared to the  $y_c$  responses in the low-quality dataset. For simplicity, this dataset is referred to as the ‘high dataset’.

**Mix-quality dataset.** This dataset was created by mixing the low-quality and high-quality datasets in a 50/50 ratio, ensuring no overlap in prompts between the two. This dataset is referred to as the ‘mix dataset’.

We provide the evaluation scores of the gold RM for these datasets in Table 3 in Appendix B.

### 5.3 Effect of Data Quality to Performance of RLHF and DPO

Our preliminary experiment investigates the sensitivities of (RM-based) RLHF and (RM-free) DPO to the quality of the datasets employed with the 160M-sized LM. Here, we used the high-quality and mixed-quality datasets. We used an RLHF implementation in the TRL library, in which a 70M-sized RM is trained and then used for RL fine-tuning. The evaluation is based on three independent runs, with each run employing different evaluation prompts that could vary ranges of the average of gold rewards. For a coherent comparison, gold rewards are normalized, setting the initial performance of LM to 0.0 and post-training performance with RLHF or DPO to 1.0.

Figure 1 (A) shows the results, where the mean and standard error of the normalized gold reward with three independent runs are presented. Notably, while DPO experienced a decline in efficacy when trained on the mixed-quality dataset relative to the high-quality one, RLHF showed an intriguing resilience, sustaining comparable performance levels across both datasets. This differential impact starkly highlights the greater susceptibility of DPO to dataset quality, suggesting that the RM-based approach, including fDPO, may offer more stable performance when the preference dataset quality cannot be consistently assured. All experiments are conducted using a single NVIDIA A100 accelerator. Further details of the experimental parameters are provided in Table 1 in Appendix B.

### 5.4 Evaluation of fDPO

We evaluate fDPO and DPO when trained using a 1.4B-sized LM  $\pi_\theta$  on the mixed-quality dataset, where fDPO used a 160M-sized RM that was trained with the same dataset. The evaluation is based on three independent runs, where the gold reward was normalized with the performance of DPO with high-quality dataset, as in Section 5.3. The epoch number for DPO was set to 5, which avoided overoptimization while ensuring the learning convergence. In the case of fDPO, we adapted the epoch count to double that of DPO, up to 10 epochs, or ceased learning when the remaining sample size was reduced to less than 1% of the original due to the filtering process, meaning the maximum filtered rate  $\gamma = 0.99$ .

Figure 2 present the results of DPO and fDPO. The results shows that the performance of DPO trained on the high-quality dataset and fDPO trained on the mixed-quality dataset were on par. It indicates that fDPO has successfully circumvented the performance decline typically observed with DPO, thereby showcasing its potential to improve DPO performance where dataset quality is inconsistent. Corresponding learning curves are included in Appendix B.

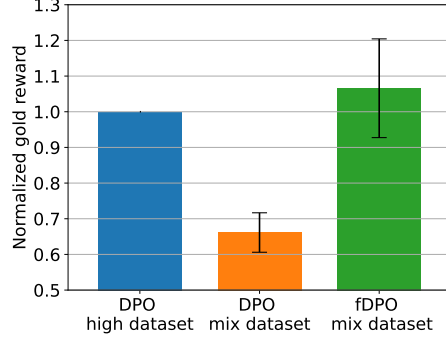


Figure 2: Performance comparison between DPO and fDPO using a 1.4B-sized LM on the mix-quality dataset.

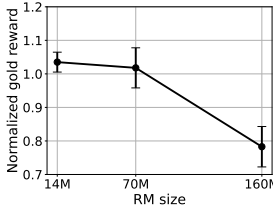


Figure 3: RM size

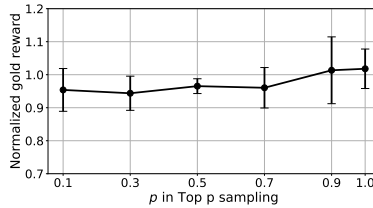


Figure 4: Top p sampling

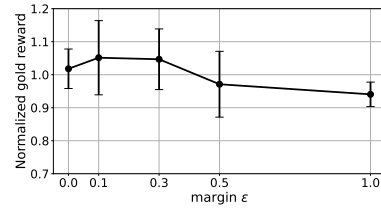


Figure 5: Margin for filtering

## 5.5 Detailed Experiments

We delve into an extensive analysis of fDPO using a 160M LM. We set the number of epochs to 8 for DPO to ensure convergence, resulting in a maximum of 16 epochs for fDPO. We maintained the maximum filtered rate ( $\gamma = 0.99$ ) as unchanged.

**Performance comparison with DPO.** Figure 1 (B) illustrates the performances of LMs trained with DPO and fDPO using the mixed-quality dataset. The results are consistent with those obtained from the larger 1.4B-sized LM, reaffirming the advantage of fDPO with the mixed-quality dataset. Additionally, we conducted an experiment using only a low-quality dataset, which revealed a surprising improvement of 7.42% (standard error: 1.89%) despite the presumed uniformity of response quality. This improvement suggests it effectively discriminates subtle quality variations, enhancing overall performance by eliminating less optimal data, even within uniformly labeled datasets.

**Analysis of configuration parameters.** We investigated various aspects of fDPO, including the size of RMs, the randomness of LMs, and the criteria for the sampling filtering, with the mix-quality dataset. Figure 3 displays the impact of RM size. Consistent with findings from Ouyang et al. (2022), smaller RMs relative to the LM size yielded better performance. This contrasts with studies advocating larger RMs for improved performance (Gao et al., 2023; Coste et al., 2023), highlighting an area for further detailed analysis.

Reducing randomness of LMs was hypothesized to enhance fDPO’s performance by minimizing the variance in quality of the LM-generated responses. However, as Figure 4 indicates, reducing randomness did not yield improvements, and in some cases, it led to worse performance. This outcome may be attributed to a discrepancy between inference-time and training-time randomness.

Finally, we explored different criteria for discarding data. As stated in line 8 of Algorithm 1, the original criterion was discarding a sample even if the reward of the LM-generated response  $y$  is only marginally higher than that of  $y_c$  in the dataset. Considering potential errors in proxy rewards and the probabilistic nature of LMs, we introduced a margin  $\epsilon$  to the discarding criterion:  $r(x, y) > r(x, y_c) + \epsilon$ . Figure 5 presents the results, showing that a certain margin size can indeed be effective in enhancing fDPO’s performance.



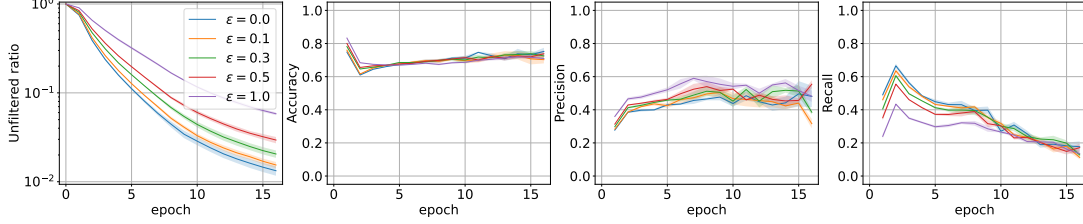


Figure 6: Unfiltered ratio, accuracy, precision, and recall throughout epochs in fDPO, comparing the effects of no margin ( $\epsilon = 0$ ) with various margin levels ( $\epsilon > 0$ ) on the filtering condition.

**Analysis of filtered samples.** We examined how data was selectively discarded throughout the learning process of fDPO with the mix-quality dataset. Figure 6 presents the unfiltered ratio, accuracy, precision, and recall at each epoch. The unfiltered ratio reflects the proportion of data that remains after filtering. Accuracy reflects the overall correctness of the filtering decisions, both for deletion and retention of samples, based on their gold reward quality. Precision measures how accurately the samples decided for deletion were actually of lower quality, while recall evaluates the success in identifying and discarding all samples that warranted removal. The result of the unfiltered ratio indicates an exponential decay in the number of samples used in each epoch. The consistency of accuracy and precision across epochs suggests that data was discarded with a constant efficiency. The lower precision compared to accuracy can be attributed to the relatively small number of samples that warranted removal. Conversely, recall decreases with progressing epochs. This decline can be tied to the static errors within the proxy RM, leading to consistently overestimated  $y_c$  samples, thus increasing their relative proportion over time. The figure contrasts various margin settings with the no-margin condition ( $\epsilon = 0$ ), revealing that larger margins lead to slower filtering speeds. Notably, as the margin increases, precision improves at the expense of recall. This trade-off underscores the importance of carefully tuning the margin parameter  $\epsilon$  to balance filtering efficacy.

## 6 Conclusions

This study explores how the quality of a preference dataset impacts language models (LMs) optimized using direct preference optimization (DPO), especially when compared with an ordinary reinforcement learning from human feedback (RLHF) method. We discovered that the impact of dataset quality is significantly more pronounced in DPO. To address the challenges posed by low-quality data within DPO, we proposed filtered DPO (fDPO). This approach utilizes a reward model (RM) to identify and discard lower-quality data, effectively refining the DPO process. Our experiments demonstrated that fDPO improved DPO’s performance across different LM sizes, effectively handling datasets with quality discrepancies. This finding highlights the practicality of fDPO, particularly for applications where data quality can vary or is challenging to control. Intriguingly, fDPO also presented potential performance gains in datasets with standard quality variance.

Several avenues for future research emerge from this work. One vital area involves expanding our experimentation to diverse, real-world preference datasets. In this study, we employed artificially created datasets with varied response qualities. However, merging multiple existing preference datasets, which naturally contain varied qualities, could offer more realistic insights into fDPO’s applicability and robustness. Exploring how fDPO performs with these organically varied datasets will be crucial for understanding its effectiveness in more complex, real-application scenarios.

Further, the uncertainties associated with proxy RM estimation and response generation of LM require more in-depth exploration. Our initial assessments suggest robustness against these factors; however, a comprehensive investigation is essential to validate these findings. Additionally, while fDPO currently focuses solely on the quality of chosen responses, considering rejected responses presents an exciting direction for future work.

---

## References

- Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*, 2024.
- Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *ACM Conference on Fairness, Accountability, and Transparency*, pp. 610–623, 2021.
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430, 2023.
- Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avaniika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2022.
- Ralph A. Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324–345, 1952.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, 2020.

---

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, J  r  my Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Rapha  l Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biy  k, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

Thomas Coste, Usman Anwar, Robert Kirk, and David Krueger. Reward model ensembles help mitigate overoptimization. In *International Conference on Learning Representations*, 2023.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*, 2023.

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

Qi Gou and Cam-Tu Nguyen. Mixed preference optimization: Reinforcement learning with data selection and better reference model. *arXiv preprint arXiv:2403.19443*, 2024.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of RLHF on LLM generalisation and diversity. In *International Conference on Learning Representations*, 2024.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. In *Association for Computational Linguistics*, pp. 3214–3252, 2022.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J. Liu, and Xuanhui Wang. LiPO: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*, 2024.

Thomas M. Moerland, Joost Broekens, Aske Plaat, and Catholijn M. Jonker. Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*, 2022.

- 
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.
- Alizée Pace, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. West-of-N: Synthetic preference generation for improved reward modeling. *arXiv preprint arXiv:2401.12086*, 2024.
- Ben Pikus, Will LeVine, Tony Chen, and Sean Hendryx. A baseline analysis of reward models’ ability to accurately analyze foundation models under distribution shift. *arXiv preprint arXiv:2311.14743*, 2023.
- A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskeve. Language models are unsupervised multitask learners. In *OpenAI blog 1.8*, 2019.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023.
- S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1179–1195, 2017.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, 2020.
- R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, 2nd edition, 2018.
- Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. In *International Conference on Learning Representations*, 2024.

R. J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992.

Xiaoying Zhang, Jean-Francois Ton, Wei Shen, Hongning Wang, and Yang Liu. Overcoming reward overoptimization via adversarial policy optimization with lightweight uncertainty estimation. *arXiv preprint arXiv:2403.05171*, 2024.

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

## A Appendix

### A.1 Sensitivity analysis of chosen and rejected responses

This subsection presents a theoretical analysis focused on the differential sensitivity of the DPO algorithm to chosen ( $y_c$ ) and rejected ( $y_r$ ) responses. The analysis elucidates how the DPO’s update gradient affects the probability of chosen responses relative to rejected ones, which is a key consideration in designing our proposed approach, fDPO. This understanding is vital to enhance the efficiency of DPO, which fDPO achieves by selectively discarding low-quality  $y_c$  samples during training.

For simplicity in the following analysis, we will occasionally omit the prompt  $x$  and denote  $\pi_\theta(y | x)$  simply as  $\pi_\theta(y)$ .

**Proposition 1.** *Let the following assumptions hold:*

$$\begin{aligned} |||\nabla_\theta \log \pi_\theta(y_c)| - |\nabla_\theta \log \pi_\theta(y_r)||| &\leq o(||\nabla_\theta \log \pi_\theta(y)||), \quad \forall y \in \{y_c, y_r\} \\ |\nabla_\theta \log \pi_\theta(y_c)^\top \nabla_\theta \log \pi_\theta(y_r)|^{\frac{1}{2}} &\leq o(||\nabla_\theta \log \pi_\theta(y)||), \quad \forall y \in \{y_c, y_r\} \\ \pi_\theta(y_c) / \pi_\theta(y_r) &= \delta. \end{aligned}$$

When, following the DPO algorithm, the parameter  $\theta$  is updated using the gradient given by

$$w(y_c, y_r)(\nabla_\theta \log \pi_\theta(y_c) - \nabla_\theta \log \pi_\theta(y_r)),$$

the sensitivity of  $\pi_\theta(y_c)$ , defined as the magnitude of change in probability,  $\Delta\pi_\theta(y)$ , is approximately  $\delta$  times higher than that of  $\pi_\theta(y_r)$ .

**Proof:** We notate the DPO gradient as  $\Delta\theta = w(y_c, y_r)(\nabla_\theta \log \pi_\theta(y_c) - \nabla_\theta \log \pi_\theta(y_r))$ . The DPO gradient  $\Delta\theta$  directly influences the rate of change in the probabilities  $\pi_\theta(y_c)$  and  $\pi_\theta(y_r)$ . The change in probabilities can be approximated as

$$\begin{aligned} \Delta\pi_\theta(y) &= \Delta\theta^\top \nabla_\theta \pi_\theta(y) + \mathcal{O}(\Delta\theta^\top \Delta\theta) \\ &\simeq \pi_\theta(y) \Delta\theta^\top \nabla_\theta \log \pi_\theta(y) \end{aligned}$$

Given the assumptions, the magnitudes of the gradients for  $\log \pi_\theta(y_c)$  and  $\log \pi_\theta(y_r)$  are similar and their inner product is small, indicating that the direction of their gradients is nearly orthogonal. Hence, the impact of  $\Delta\theta$  on  $\log \pi_\theta(y_c)$  and  $\log \pi_\theta(y_r)$  would be similar in magnitude but differ in direction. However, due to the ratio  $\pi_\theta(y_c) / \pi_\theta(y_r) = \delta$ , the rate of change in  $\pi_\theta(y_c)$  is amplified by a factor of  $\delta$  compared to  $\pi_\theta(y_r)$ . Thus, under the DPO update,  $\pi_\theta(y_c)$  demonstrates a sensitivity approximately  $\delta$  times higher than that of  $\pi_\theta(y_r)$ .  $\square$

As the training progresses in DPO, it is generally observed that the ratio  $\delta$ , representing the likelihood of  $y_c$  over  $y_r$ , tends to exceed 1. This phenomenon indicates an increased sensitivity towards the chosen responses, emphasizing the criticality of their quality within the DPO framework. Consequently, the presence of low-quality chosen responses in the dataset can significantly impede the effectiveness of DPO.

However, it is essential to acknowledge that the assumptions leading to these observations are strong and may not hold in some contexts and datasets. Therefore, further experimental work is necessary to validate these assumptions. Additionally, considering rejected



responses in fDPO represents a separate but exciting area for future exploration, potentially offering new insights into data refinement approaches of preference-based model optimization.

## B Details of experiments

### B.1 hyperparameters

We provide details of the hyperparameters used in our experiments. Table 1 presents the training parameters for DPO and RLHF, while Table 2 illustrates the parameters for response generation.

Table 1: Experimental Parameters of DPO and PPO

Parameter	DPO (160M)	DPO (1.4B)	RLHF (160M)
epochs	8 (16 for fDPO)	5 (10 for fDPO)	3
batch size	32	16	16
gradient accumulation steps	2	4	2
learning rate	$1 \times 10^{-6}$	$2 \times 10^{-7}$	$1 \times 10^{-6}$
$\beta$	0.1	0.1	0.1
PPO epochs	-	-	4
number of rollouts	-	-	128

Table 2: Response Generation Params

Parameter	value
Top p	0.9 (1.0 for fDPO and PPO)
Top k	0
Temperature	1.0
Max response tokens	64

### B.2 Additional results

The evaluation scores of the gold reward models for each preference dataset ('high', 'low', 'mix') are detailed in Table 3.

Table 3: The evaluation scores of gold rewards for each dataset

Model Size	Quality	Chosen Mean	Rejected Mean	Overall Mean
160M	High	-0.950	-2.786	-1.868
	Low	-2.153	-3.180	-2.667
	Mix1	-1.549	-2.978	-2.263
	Mix2	-1.547	-2.984	-2.265
	Mix3	-1.555	-2.984	-2.270
1.4B	High	1.220	-0.996	0.113
	Low	-0.240	-1.482	-0.860
	Mix1	0.500	-1.233	-0.367
	Mix2	0.487	-1.236	-0.375
	Mix3	0.487	-1.247	-0.380

Figure 7 provides the learning curves for DPO and fDPO with the the 160M-sized LM, corresponding to the final performances depicted in Figure 1 (B) of the main text. The curves show that even though fDPO processes double the number of epochs compared

to DPO, the total number of steps for fDPO is fewer than that for DPO. This reduction is due to the filtering process of fDPO, which decreased the data over epochs, resulting in fewer steps per epoch, as demonstrated in Figure 6. Additionally, when assessed using KL divergence, the performance of fDPO shows a trend towards converging with the DPO trained on the high-quality dataset, suggesting that fDPO can close the performance gap even when trained on mixed-quality data.

Figure 8 presents the learning curves for DPO and fDPO applied to the mix-quality dataset with the 1.4B LM and the low-quality dataset with the 160M LM. In both contexts, fDPO consistently improved the performance of DPO over steps, echoing the results observed in the mix-quality dataset scenario with the 160M LM.

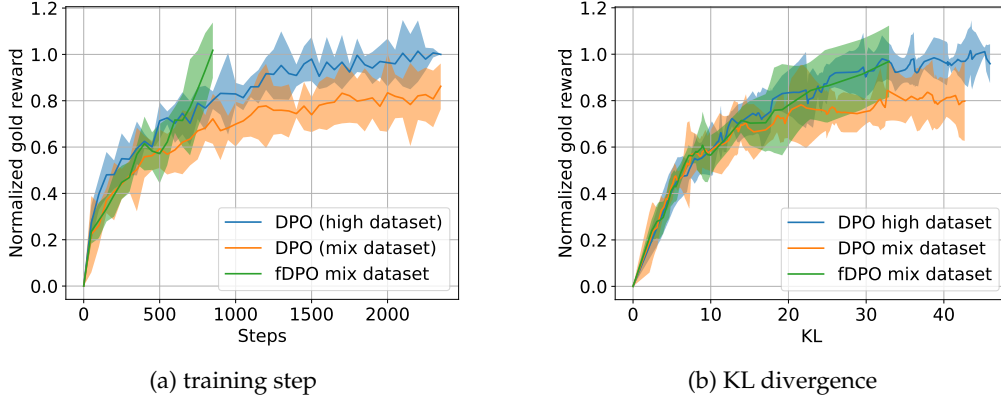


Figure 7: The learning curves for DPO and fDPO using the 160M-sized LM on the mix-quality dataset. The horizontal axes of the two figures represent the number of training steps and the KL divergence with the initial LM, respectively. Normalized gold rewards are adjusted from 0 (initial) to 1 (DPO on high-quality).”

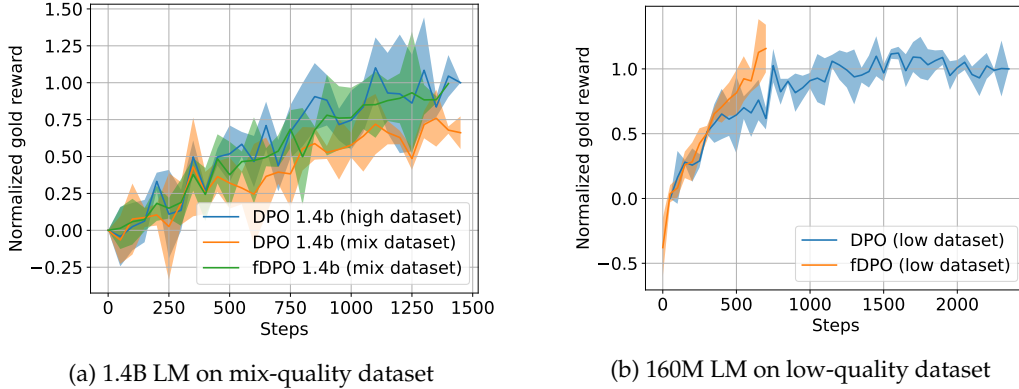


Figure 8: The learning curves for DPO and fDPO using the 1.4B LM on the mix-quality dataset (left) and the 160M LM on the low-quality dataset (right), respectively. Normalized gold rewards are adjusted from 0 (initial) to 1 (DPO).”