Reward-Augmented Data Enhances Direct Preference Alignment of LLMs

Shenao Zhang[1] **Zhihan Liu**[1] **Boyi Liu**[2] **Yufeng Zhang**[2]
Yingxiang Yang[2] **Yongfei Liu**[2] **Liyu Chen**[2] **Tao Sun**[2] **Zhaoran Wang**[1]
[1]Northwestern University [2]ByteDance

Abstract

Preference alignment in Large Language Models (LLMs) has significantly improved their ability to adhere to human instructions and intentions. However, existing direct alignment algorithms primarily focus on relative preferences and often overlook the qualitative aspects of responses, despite having access to preference data that includes reward scores from judge models during AI feedback. Striving to maximize the implicit reward gap between the chosen and the slightly inferior rejected responses can cause overfitting and unnecessary unlearning of the high-quality rejected responses. The unawareness of the reward scores also drives the LLM to indiscriminately favor the low-quality chosen responses and fail to generalize to responses with the highest rewards, which are sparse in data. To overcome these shortcomings, our study introduces reward-conditioned LLM policies that discern and learn from the entire spectrum of response quality within the dataset, helping extrapolate to more optimal regions. We propose an effective yet simple data relabeling method that conditions the preference pairs on quality scores to construct a reward-augmented dataset. This dataset is easily integrated with existing direct alignment algorithms and is applicable to any preference dataset. The experimental results across instruction-following benchmarks including AlpacaEval 2.0, MT-Bench, and Arena-Hard-Auto demonstrate that our approach consistently boosts the performance of DPO by a considerable margin across diverse models such as Zephyr, Mistral, Qwen2, Llama3.1, Gemma2, and SPPO. Additionally, on six academic benchmarks including GSM8K, GPQA, MUSR, TruthfulQA, BBH, and ARC, our method improves their average accuracy. When applying our method to on-policy data, the resulting DPO model outperforms various baselines and achieves state-of-the-art results on AlpacaEval 2.0. Through comprehensive ablation studies, we demonstrate that our method not only maximizes the utility of preference data but also mitigates the issue of unlearning, demonstrating its broad effectiveness beyond mere dataset expansion.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has recently seen remarkable success in aligning Large Language Models (LLMs) to follow instructions with human intentions. In this approach, AI-generated feedback serves as a stand-in for human preferences, assessing and ranking responses to prompts to construct a preference dataset. This dataset is then utilized in preference optimization algorithms to fine-tune LLMs. Among them, direct preference alignment (Rafailov et al., 2024b; Azar et al., 2023; Zhao et al., 2023; Ethayarajh et al., 2024) that bypasses the need for an explicit reward model has garnered interest for their simplicity and cost efficiency. However, these algorithms mainly concern relative preferences and often overlook the quality of responses and their gaps, leading to limitations in their effectiveness.

Specifically, direct alignment algorithms such as DPO (Rafailov et al., 2024b) focus on maximizing the implicit reward difference between accepted and rejected responses. This approach can lead to overfitting, as high-quality but rejected responses are unnecessarily unlearned (Adler et al., 2024). Even worse, since the dataset provides only a sample estimate of true preferences, the rejected responses can actually be more aligned with human preferences than the accepted ones in expectation. Similarly, due to the unawareness of the responses' qualities, direct alignment will also result in the indiscriminate learning of the chosen responses, even when they are of low quality. As a result, the directly aligned LLMs often struggle to differentiate between responses of varying quality and fail to generalize effectively to more optimal or the highest-reward responses that are sparse in the preference data, which is another limitation.

To address these issues, we propose learning reward-conditioned policies as a straightforward fix to the above issues. By optimizing the LLM to generate responses conditioning on their qualities, the model is allowed to discern and leverage patterns within responses of varied quality. As a result, learning from both chosen and rejected responses alleviates the issue of unnecessarily unlearning high-quality rejected responses; distinguishing between varying-quality chosen responses alleviates the issue of indiscriminately accepting low-quality ones. By identifying common patterns in responses of similar quality and distinguishing them from those of differing quality, the LLM becomes more adept at generalizing to more optimal responses that are sparse in data.

With this motivation, we introduce an effective yet simple data relabeling method to construct reward-augmented datasets. We define a goal-conditioned reward using an indicator function that compares the goal reward with the actual quality score, such as the reward value given by the judge model during AI feedback. This allows us to relabel each preference pair, generating two new pairs conditioned on the reward goals of both the chosen and rejected responses. The resulting augmented dataset, which contains these newly conditioned pairs, can enhance the performance of existing direct alignment algorithms. Our method can be applied to any preference dataset and followed by off-the-shelf direct alignment algorithms to boost their performance.

In experiments, we first apply our method on UltraFeedback (Cui et al., 2023) and perform DPO (Rafailov et al., 2024b) on this reward-augmented preference dataset by fine-tuning on various models, including Zephyr-7B-β (Tunstall et al., 2023b), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023a), Qwen2-7B-Instruct (Yang et al., 2024), Llama-3.1-8B-Instruct (Dubey et al., 2024), Gemma-2-9B-It (Team et al., 2024), and SPPO (Wu et al., 2024). The results show that our method consistently boosts the performance of these models as well as their DPO models by a large margin on instruction-following benchmarks such as AlpacaEval 2.0 (Dubois et al., 2024), MT-Bench (Zheng et al., 2024), and Arena-Hard-Auto (Li et al., 2024b). Our method also improves the average accuracy on a variety of academic benchmarks (GSM8K, GPQA, MUSR, TruthfulQA, BBH, and ARC). Moreover, our findings also demonstrate an improved utility of the preference data: a subsequent round of DPO using the reward-augmented data can still significantly enhance the model fine-tuned with DPO; relabeling the binarized preference dataset with the DPO implicit reward leads to further performance gains. Additional ablation studies also suggest that our method addresses the problem of unlearning and is superior not just due to the increased dataset size. When applied to on-policy data, our method enhances the DPO model, enabling it to surpass various baselines and achieve state-of-the-art performance on AlpacaEval 2.0.

2 Background

Consider a language model $\pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$ that takes the prompt $x \in \mathcal{X}$ as input and outputs the response $y \in \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are spaces of prompts and responses, respectively. Given the prompt $x \in \mathcal{X}$, a discrete probability distribution $\pi(\cdot \mid x) \in \Delta_{\mathcal{Y}}$ is generated, where $\Delta_{\mathcal{Y}}$ is the set of discrete distributions over $\mathcal{Y}$. We define the true human preference distribution as

$$p^*(y_1 \succ y_2 \mid x) := \mathbb{E}_h\big[\mathbb{1}(h \text{ prefers } y_1 \text{ over } y_2 \text{ given } x)\big],$$

where h denotes the human rater and the expectation is over h to account for the randomness of the human raters' choices. After pretraining and Supervised Fine-Tuning (SFT), Reinforcement Learning from Human or AI Feedback (Ouyang et al., 2022; Bai et al., 2022b) is typically employed to enhance the ability of the language model to follow instructions with human preferences.

RL from AI Feedback (RLAIF). The RLAIF framework involves two major steps: preference dataset construction with AI feedback and preference optimization. As a surrogate for human preference, AI feedback, including LLM-as-Judge (Zheng et al., 2024; Cui et al., 2023) and Reward-Model-as-Judge (Adler et al., 2024; Dong et al., 2024), can be used to rank responses and generate preference pairs. Specifically, consider the judge model $r(x, y) : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ that outputs a scalar reward value representing the quality of y under x. For each prompt $x \in \mathcal{X}$, two responses, y_1 and

y_2, are independently sampled—either from the same reference model (Xiong et al., 2024; Wu et al., 2024) or several different models (Zhu et al., 2023; Zhang et al., 2024). Then $r(x, y_1)$ and $r(x, y_2)$ are evaluated to determine the preferred response $y_w = \text{argmax}_{y \in \{y_1, y_2\}} r(x, y)$ and dispreferred response $y_l = \text{argmin}_{y \in \{y_1, y_2\}} r(x, y)$. By sampling responses and ranking them for a set of N prompts, we get a preference dataset: $\mathcal{D}_N = \{x^i, y_w^i, y_l^i, r(x^i, y_w^i), r(x^i, y_l^i)\}_{i=1}^N$.

Direct Alignment from Preference. The objective for the LLM $\pi \in \Delta_{\mathcal{Y}}^{\mathcal{X}}$ is to maximize the KL-regularized expected reward. Recent works (Azar et al., 2023; Zhao et al., 2023; Tunstall et al., 2023b; Ethayarajh et al., 2024) proposed to align the LLM directly with the preference data by deriving the preference loss as a function of the LLM by the change of variables. Among them, the Direct Preference Optimization (DPO) (Rafailov et al., 2024b) loss has the following form:

$$\mathcal{L}_{\text{DPO}}(\pi; \mathcal{D}_N) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right],$$

where β is a hyperparameter corresponding to the KL divergence regularization, $\sigma(\cdot)$ is the logistic function, and π_{ref} is some reference LLM policy, such as the SFT model.

3 Reward-Conditioning Addresses Limitations of Direct Preference Alignment

3.1 Limitations of Direct Alignment from Preference

We will first demonstrate the limitations of vanilla direct alignment over the preference data.

High-Quality Rejected Responses are Unnecessarily Unlearned. The dataset $\mathcal{D}_N$ often contains preference pairs where the rejected response y_l is only marginally worse than the chosen one y_w. Direct alignment algorithms, however, primarily focus on relative preferences and are unaware of the responses' quality values and gaps. Striving to maximize the reparameterized reward gap between the chosen and rejected responses will risk overfitting and the unnecessary unlearning of high-quality responses, potentially diminishing the model's performance by discarding valuable alternatives. Furthermore, in such a finite data regime where only a sample estimate of the true preference is accessible, it can be very possible that $p^*(y_l \succ y_w \mid x) > 0.5$, i.e., y_l is in fact more preferred than y_w in expectation. This issue becomes even more pronounced when the preference data generated with the imperfect judge model is noisy.

We illustrate this limitation with the example in Table 1. For $\mathcal{D}_{N=1}$ that contains a single preference pair[1] with reward $r(x, y_1) = 9$ and $r(x, y_2) = 8$, the optimal policy learned from $\mathcal{D}_{N=1}$ is $\pi^*(y_1 \mid x) = 1$. This causes the model to avoid generating y_2, a response of nearly equivalent quality.

Low-Quality Chosen Responses are Indiscriminately Learned. For a similar reason, direct alignment algorithms also indiscriminately reinforce the chosen responses. As illustrated in Table 2, when $\mathcal{D}_{N=2}$ contains two preference pairs, where one of the chosen responses, y_2, is of low quality, π^* still indiscriminately generates y_2 with an arbitrary probability $0 \leq a \leq 1$, i.e., $\pi^*(y_2 \mid x) = a$.

Reward Sparsity. Preference data often contains responses that, despite being preferred in pairwise comparisons, exhibit substantial variation in quality. As a result, the optimal responses—those associated with the highest reward values—are sparse in the dataset. Since direct alignment algorithms do not account for these reward values, the trained model struggles to differentiate between responses of varying quality and fails to generalize effectively to the sparse optimal responses.

3.2 Reward-Conditioned Policies Resolve the Limitations

A straightforward way to address the limitations of direct alignment algorithms—specifically, their inability to account for the quality of responses—is to optimize a reward-conditioned policy. In this approach, the LLM policy is trained to generate responses corresponding to different reward values,

[1] For simplicity, we write $(x, y_w, y_l, r(x, y_w), r(x, y_l)) \in \mathcal{D}_N$ as $y_w > y_l$.

response	y_1	y_2
$r(x, y)$	9	8
$\mathcal{D}_{N=1}$	$\{y_1 > y_2\}$	
$\pi^*(y \mid x)$	1	0
$\pi^*(y \mid x; r = 9)$	1	0
$\pi^*(y \mid x; r = 8)$	0	1

Table 1: High-quality rejected responses such as y_2 can be unnecessarily unlearned: $\pi^*(\cdot \mid x)$ deterministically generates y_1. Reward-conditioned policies learn both responses and are easier to generalize to $r = 10$ with the extracted features from $r = 8$ and $r = 9$.

response	y_1	y_2	y_3
$r(x, y)$	9	1	0
$\mathcal{D}_{N=2}$	$\{y_1 > y_3\ , \ y_2 > y_3\}$		
$\pi^*(y \mid x)$	$1 - a$	a	0
$\pi^*(y \mid x; r = 9)$	1	0	0
$\pi^*(y \mid x; r = 1)$	0	1	0
$\pi^*(y \mid x; r = 0)$	0	0	1

Table 2: Low-quality chosen responses such as y_2 can be learned: π^* indiscriminately generates y_1 and y_2. Reward-conditioned policies distinguish the differences and learn the behaviors corresponding to different reward scores.

enabling it to become aware of and adapt to these reward distinctions. By doing so, the LLM not only learns the patterns associated with the preferred responses but also retains the valuable information from the rejected ones, preventing the unlearning of high-quality rejected responses. For example, in Table 1, reward-conditioned policies learn to generate both y_1 and y_2, instead of unlearning y_2. This reward-based conditioning also enhances the model's ability to differentiate between responses of varying quality, even if both are preferred over a rejected alternative, as illustrated in Table 2. Besides, by extracting common patterns across responses with different quality levels, the LLM becomes more generalizable and is capable of generating the highest-quality responses (e.g., $r = 10$), which are often sparse in the training preference data.

4 Method

With the above motivation, we propose a data relabeling method that constructs a reward-augmented dataset by conditioning the preference pairs on the reward values given by the judge model. Specifically, we define the goal-conditioned reward $r_g(x, y) = \mathbb{1}\big(g = r(x, y)\big)$ as a function of the quality reward r. The objective of the reward-conditioned policy $\pi(y \mid x; g)$ is thus to minimize the absolute difference between the goal reward g and the response reward $r(x, y)$, which is equivalent to maximizing the goal-conditioned reward $r_g(x, y)$, i.e.,

$$\min_{\pi} \mathbb{E}_{g, x \sim \mathcal{D}_N, y \sim \pi(\cdot \mid x; g)} \big[|r(x, y) - g| \big] = \max_{\pi} \mathbb{E}_{g, x \sim \mathcal{D}, y \sim \pi(\cdot \mid x; g)} \big[r_g(x, y) \big]. \tag{4.1}$$

To optimize the RHS of Equation (4.1), we first observe that under the new goal-conditioned reward metric r_g, for each preference pair $x^i, y_w^i, r_w^i, y_l^i, r_l^i$ in $\mathcal{D}_N$, we have

$$r_{g=r_w^i}(x, y_w^i) = 1, r_{g=r_w^i}(x, y_l^i) = 0, r_{g=r_l^i}(x, y_l^i) = 1, r_{g=r_l^i}(x, y_w^i) = 1.$$

Thus, each pair can be relabeled to create two new preference pairs based on two distinct goals: when $g = r_w^i$, $y_w^i \succ y_l^i$; when $g = r_l^i$, $y_l^i \succ y_w^i$. Then any direct alignment algorithm can be applied to this new goal-conditioned preference dataset. Compared to fine-tuning on the original dataset $\mathcal{D}_N$, the model learns to capture not only desirable behaviors but also undesirable ones from the reward-augmented dataset. This approach helps identify patterns across high- and low-quality responses, enabling the LLMs to discern and learn from the entire spectrum of response quality and extrapolate to more optimal responses at inference time, by conditioning on higher reward goals.

We illustrate our method in Figure 1. For each preference pair with index i in $\mathcal{D}_N$, two goals are defined, corresponding to the reward values of the chosen response y_w^i and the rejected response y_l^i. Specifically, under the first goal $g = y_w^i$, the relabeled rewards are $r_g(x, y_w^i) = 1$ and $r_g(x, y_l^i) = 0$. The original ranking of responses remains the same, except that the LLM is preference optimized conditioned on $g = y_w^i$. Similarly, under the second goal $g = y_l^i$, the relabeled rewards are $r_g(x, y_l^i) = 1$ and $r_g(x, y_w^i) = 0$. Thus, the chosen and rejected responses are reversed as y_l^i and y_w^i, respectively. By generating preference pairs conditioned on the goal reward for both the chosen and rejected responses, we obtain a reward-augmented dataset of size $2N$. Finally, this new dataset can be used with any direct alignment algorithm, such as DPO.

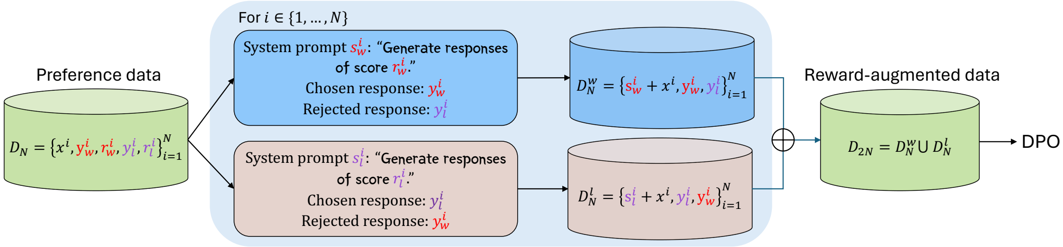

For $i \in \{1, \ldots, N\}$
Preference data
$D_N = \{x^i, y_w^i, r_w^i, y_l^i, r_l^i\}_{i=1}^N$
System prompt s_w^i: "Generate responses of score r_w^i."
Chosen response: y_w^i
Rejected response: y_l^i
System prompt s_l^i: "Generate responses of score r_l^i."
Chosen response: y_l^i
Rejected response: y_w^i
$D_N^w = \{s_w^i + x^i, y_w^i, y_l^i\}_{i=1}^N$
$D_N^l = \{s_l^i + x^i, y_l^i, y_w^i\}_{i=1}^N$
Reward-augmented data
$D_{2N} = D_N^w \cup D_N^l$
DPO

Figure 1: Construction of the reward-augmented preference dataset.

In this work, we implement the reward-conditioned policy $\pi(y \mid x; g)$ as the LLM with a system prompt (or a prefix before the user prompt x if system prompts are not supported by the LLM) such as "generate responses of score g". At inference time, the LLM is conditioned on the highest possible reward value, e.g., $g = 10$, to generate the responses.

5 Related Work

Preference Dataset Construction. In order for the LLMs to follow instructions and better align with human intents, it is common practice to build a preference dataset containing a set of prompts and a pair of responses for each prompt, whose qualities are ranked by humans (Ouyang et al., 2022) or judge models (Bai et al., 2022b). A popular pipeline (Cui et al., 2023; Tunstall et al., 2023b; Wang et al., 2024c; Ivison et al., 2023; Zhu et al., 2023) for constructing offline (i.e., fixed) datasets involves sampling off-policy responses from *various* LLMs for each prompt in the hope to increase the response diversity. The preference data can also be generated online (Guo et al., 2024) or iteratively (Bai et al., 2022a; Xu et al., 2023; Gulcehre et al., 2023; Hoang Tran, 2024; Xiong et al., 2023; Dong et al., 2024; Calandriello et al., 2024; Rosset et al., 2024) by sampling and ranking on-policy responses from the training LLM. Recent works (Zhang et al., 2024; Cen et al., 2024; Xie et al., 2024) have also proposed systematically exploring the responses online and actively eliciting the preference. The proposed method in this paper is orthogonal to the construction ways of the preference data and can be applied to any dataset created either off-policy or on-policy.

Preference Optimization. Preference optimization methods generally follow two approaches. The first involves learning a point-wise reward model, such as the Bradley-Terry model, and using RL algorithms like PPO (Schulman et al., 2017; Zheng et al., 2023; Xu et al., 2024b) or REINFORCE (Williams, 1992; Li et al., 2023; Ahmadian et al., 2024), to maximize the KL-regularized expected reward. The second approach is direct alignment (Rafailov et al., 2024b; Azar et al., 2023; Zhao et al., 2023; Ethayarajh et al., 2024; Liu et al., 2024), which gets rid of a separate reward model that is computationally costly to train. In this work, we mainly focus on the limitations of direct alignment algorithms, particularly their unawareness of the quality aspects of responses. For PPO-style alignment algorithms that fit and maximize an explicit reward, preference data is only used to learn the reward model, and policy training is performed in an online manner, where responses are sampled from the LLM and their reward values directly play a role during the RL optimization. This avoids drawbacks inherent to direct alignment methods, as detailed in Section 3.1.

Conditional LLM Fine-Tuning. Conditioning LLMs during training has proven effective for aligning responses with specific human objectives. SteerLM (Dong et al., 2023b; Wang et al., 2023b) extends SFT by conditioning the LLM on the multi-dimensional annotated attributes in data, such as humor and toxicity, in order to steer model responses with user customizability. Directional Preference Alignment (DPA) (Wang et al., 2024a) proposed a variant of rejection sampling fine-tuning (Yuan et al., 2023; Dong et al., 2023a) that conditions on the direction of the multi-objective reward, i.e., a user-dependent linear combination of the reward attributes (helpfulness and verbosity in their experiments), that represents diverse preference objectives. These methods aim to train a single LLM that can flexibly adjust to various user preference profiles. On the contrary, our method targets the limitations of direct alignment algorithms by introducing reward-augmented relabeling. This also differs from Conditioned-RLFT (Wang et al., 2023a), which leverages the data source information by learning a class-conditioned policy with RL-free supervised learning. Reward-aware

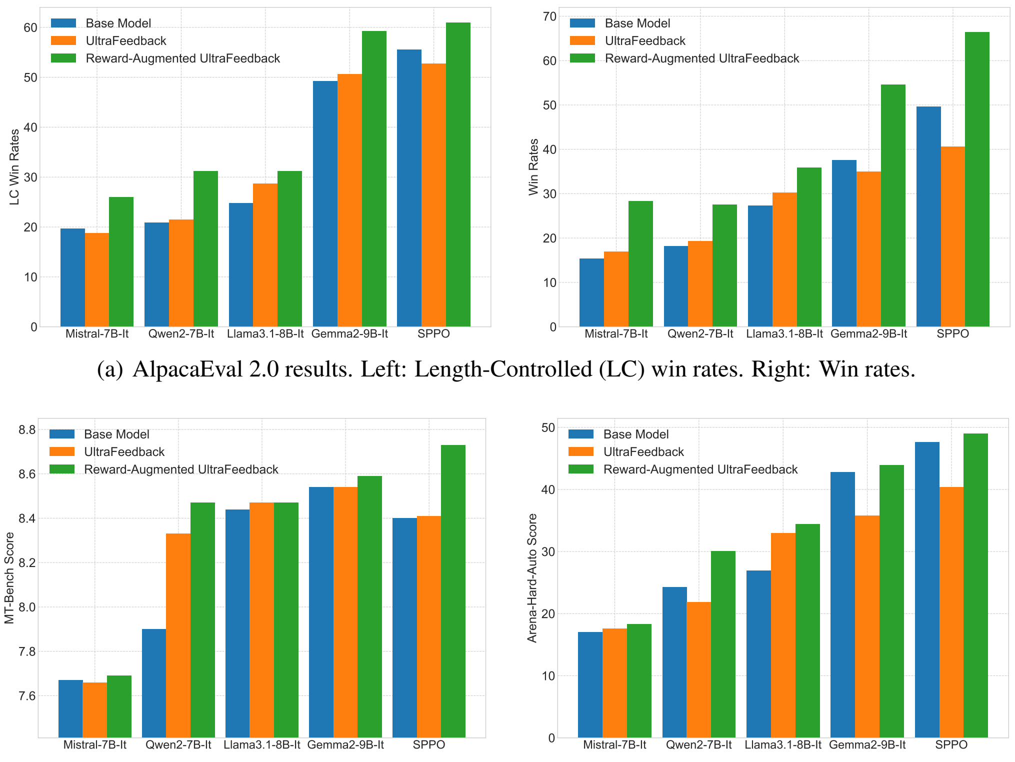
Preference Optimization (RPO), introduced in Nemotron-4 (Adler et al., 2024), attempts to approximate the reward gap using the implicit reward and is motivated to resolve the unlearning issues of DPO, which our work also addresses. However, we show that more limitations beyond unlearning can be simply fixed with reward-conditioned LLMs and propose an easy-to-implement data relabeling method that integrates seamlessly with any direct alignment algorithm. We compare with all the aforementioned methods in our experiments.

6 Experiments

6.1 Reward-Augmented Data Boosts DPO Performance

We begin by conducting experiments to demonstrate that applying the proposed method to fixed offline preference datasets leads to consistent performance improvements in DPO.

Setup. We adopt the UltraFeedback (Cui et al., 2023) preference dataset containing reward values scored by GPT-4 (LLM-as-Judge) that is ranged between 1 and 10 for each of the preference pairs. Our method constructs reward-augmented data by conditioning on these judge values. We fine-tune on various open-weight LLMs, including Mistral-7B-Instruct-v0.3[2], Qwen2-7B-Instruct[3], Llama-3.1-8B-Instruct[4], Gemma-2-9B-It[5], and SPPO (fine-tuned from Gemma2-9B-It)[6]. We use the DPO



(a) AlpacaEval 2.0 results. Left: Length-Controlled (LC) win rates. Right: Win rates.

(b) MT-Bench average score.

(c) Arena-Hard-Auto score.

Figure 2: Performance of the base models, the models trained with DPO on UltraFeedback, and the models trained with DPO on reward-augmented ultrafeedback on AlpacaEval 2.0, MT-Bench, and Arena-Hard-Auto benchmarks. The complete table is deferred to Appendix A.2.

[2] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3

[3] https://huggingface.co/Qwen/Qwen2-7B-Instruct

[4] https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct

[5] https://huggingface.co/google/gemma-2-9b-it

[6] https://huggingface.co/UCLA-AGI/Gemma-2-9B-It-SPPO-Iter3

implementation in the Huggingface Alignment Handbook (Tunstall et al.). The hyperparameters and prompts that we use are listed in Appendix A.1.

Results. We first report the performance of the trained models on instruction-following benchmarks that use LLM as a judge, including AlpacaEval 2.0 (Dubois et al., 2024), MT-Bench (Zheng et al., 2024), and Arena-Hard-Auto (Li et al., 2024b). The results are shown in Figure 2.

Across all instruction-following benchmarks, we observe that LLMs fine-tuned with DPO on the proposed reward-augmented data consistently outperform both their base models and those fine-tuned using DPO on the original UltraFeedback dataset by a considerable margin. Notably, direct alignment with the original preference data can sometimes degrade the performance of base models on specific benchmarks, such as Arena-Hard-Auto, which involves more complex reasoning tasks. In contrast, alignment using the reward-augmented data consistently yields superior results not only due to the improved style format gained from performing DPO on UltraFeedback.

Besides, we also evaluate the models on academic multi-choice QA benchmarks, including GSM8K (Cobbe et al., 2021), GPQA (Rein et al., 2023), MUSR (Sprague et al., 2023), TruthfulQA (Lin et al., 2021), BBH (Suzgun et al., 2022), and ARC Challenge (Clark et al., 2018). To better reflect the capabilities of LLMs, we adopt various settings for these benchmarks, including zero-shot, few-shot, and few-shot Chain-of-Thought (CoT). The results are shown in Table 3.

Model	GSM8K	GPQA	MUSR	TruthfulQA	BBH	ARC	Average
	(8-s CoT)	(0-s)	(0-s)	(0-s)	(3-s)	(25-s)	
Mistral-7B-Instruct-v0.3	52.39	**30.62**	**47.35**	59.71	46.64	58.53	49.21
+DPO (UltraFeedback)	**53.22**	28.94	47.35	64.74	**47.46**	60.32	**50.34**
+DPO (Reward-Augmented)	51.86	28.02	46.56	**65.90**	46.36	**61.60**	50.05
Qwen2-7B-Instruct	78.24	32.80	44.58	57.31	**55.20**	53.75	53.65
+DPO (UltraFeedback)	78.17	32.80	44.31	**58.91**	54.49	53.75	53.74
+DPO (Reward-Augmented)	**81.05**	**32.97**	**45.77**	57.99	54.94	**54.52**	**54.54**
Llama-3.1-8B-Instruct	76.72	**33.89**	39.95	54.00	50.74	55.38	51.78
+DPO (UltraFeedback)	78.47	33.72	43.39	56.61	51.31	**57.51**	53.50
+DPO (Reward-Augmented)	**78.77**	32.55	**43.52**	**63.32**	**51.57**	56.48	**54.37**
Gemma-2-9B-It	81.35	**36.33**	46.03	60.15	59.42	64.85	58.02
+DPO (UltraFeedback)	83.32	34.14	46.56	65.12	59.78	**66.41**	59.22
+DPO (Reward-Augmented)	**83.62**	35.74	**48.15**	**65.27**	**59.82**	65.87	**59.75**
SPPO	79.83	35.91	44.97	62.56	**59.61**	63.74	57.77
+DPO (UltraFeedback)	**81.73**	33.64	45.50	65.72	59.16	**66.89**	58.77
+DPO (Reward-Augmented)	80.67	**36.16**	**48.68**	**67.39**	58.88	65.53	**59.55**

Table 3: Performance comparison between the LLMs after DPO on UltraFeedback, on reward-augmented UltraFeedback, and their base models on academic multi-choice QA benchmarks in standard zero-shot, few-shot, and CoT settings. Here, n-s refers to n-shot, the **bold** texts represent the best results in each family of models.

It can be observed that performing DPO on the reward-augmented preference data leads to better average academic scores for most families of models compared to models fine-tuned on the original UltraFeedback dataset and the base models. Besides, we didn't observe severe alignment tax phenomenons (Askell et al., 2021; Noukhovitch et al., 2024; Li et al., 2024a) after DPO, and our method is able to improve the base models on most of the benchmarks.

6.2 Ablation Studies

	LC WR	WR	MT	Arena
SPPO	55.60	49.61	8.40	47.6
+DPO (UF)	52.75	40.58	8.41	40.4
+DPO (RA)	**60.97**	**66.41**	**8.73**	**49.0**

Table 4: SPPO can be improved with DPO by performing reward augmentation on the same data.

Our Method Improves the Utility of Preference Data. We provide two pieces of evidence that our method can get more juice out of the preference data compared to directly applying DPO. Firstly, we evaluate SPPO (Wu et al., 2024) fine-tuned with DPO on UltraFeedback (UF). The results are shown in Table 4. Since the SPPO model is already trained on UltraFeedback from Gemma-2-9B-It, an additional round of DPO training with the same data significantly degrades its performance. In contrast, performing DPO on Reward-Augmented (RA) UltraFeedback results in substantial performance gains for SPPO, indicating that our method enhances the utility of the preference data.

The second evidence is that after DPO, the implicit reward can be used to relabel and augment the same preference data. Specifically, after training Qwen2-7B-Instruct with DPO on UltraFeedback, we leverage the resulting model π_{DPO} to calculate the implicit reward for each prompt x and response y, i.e., $\widehat{r} = \beta(\log \pi_{\text{DPO}}(y \mid x) - \log \pi_{\text{Qwen}}(y \mid x))$. Then we perform DPO on Qwen2-7B-Instruct using the Implicit-Reward-Augmented (IRA) UltraFeedback dataset. The results are shown in Table 5. We observe that augmenting the data with the implicit reward from the DPO (UF) model leads to superior performance even compared to augmenting the data with reward scores from the LLM judge, i.e., DPO (RA). This result highlights that DPO does not fully exploit the potential of the data. Moreover, this ablation demonstrates that our method is compatible with binarized preference datasets that only contain chosen and rejected response pairs, bypassing the need for reward scores from judge models.

	LC WR	WR	MT	Arena
Qwen2-7B-It	20.93	18.22	7.90	24.3
+DPO (UF)	21.46	19.35	8.33	21.9
+DPO (RA)	31.17	27.58	8.47	**30.1**
+DPO (IRA)	**32.61**	**29.15**	**8.49**	28.3

Table 5: A second round of DPO on the reward-augmented data, i.e., DPO (IRA), relabeled with the implicit reward from the DPO model at the first round, i.e., DPO (UF), significantly improves it. Our method helps get more juice out of the *binarized* (i.e., without judge model rewards) preference data.

Reward-Augmented Data is Superior Not Just Due to Its Increased Size. In this part, we show that the success of our method is not merely due to the increased size of the training dataset. To illustrate this, we perform DPO on the dataset where reward augmentation is applied to the first half of the UltraFeedback data, which we denote as DPO (Half RA). By doing so, the reward-augmented data is of the same size as the original dataset, but with only half of the prompts and the corresponding responses being utilized. It can be observed from Table 6 that DPO (Half RA) outperforms fine-tuning on the whole UltraFeedback (UF) by a large margin and achieves comparable performance to applying reward augmentation across the entire UF dataset, which is denoted as DPO (RA).

	LC WR	WR	MT	Arena
Qwen2-7B-It	20.93	18.22	7.90	24.3
+DPO (UF)	21.46	19.35	8.33	21.9
+DPO (RA)	**31.17**	27.58	**8.47**	**30.1**
+DPO (Half RA)	29.56	**28.30**	8.33	26.9
Gemma-2-9B-It	49.20	37.58	8.54	42.8
+DPO (UF)	50.70	35.02	8.54	35.8
+DPO (RA)	**59.27**	**54.56**	8.59	**43.9**
+DPO (Half RA)	53.12	43.74	**8.66**	41.3

Table 6: DPO trained on only half of the data with reward augmentation outperforms the baseline.

Reward-Augmented Data Mitigates the Unlearning Issue. We first demonstrate that DPO suffers from the limitation of unnecessarily unlearning high-quality rejected responses, as discussed in Section 3.1. Specifically, on the test set of UltraFeedback, we calculate the log probability of each rejected response for the Qwen2-7B-Instruct model, its DPO (UF) model, and our method DPO (RA). In Figure 3, we plot the expected log probability for rejected responses with reward scores ≥ 5. We find that DPO substantially decreases the probability of these high-quality rejected responses, confirming that the unlearning issue arises in practice. In contrast, our method alleviates this issue, although the probability is still slightly lower than the base model, which is proven to be the feature of DPO (Rafailov et al., 2024a; Zhang et al., 2024; Xu et al., 2024b).

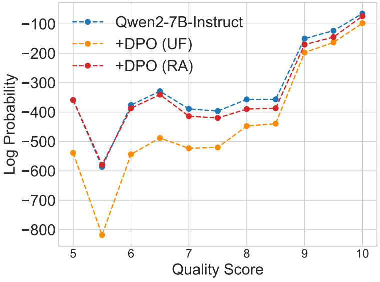


Figure 3: Our method helps mitigate the unlearning issue of DPO.

Controllable Generation with Prompt. In Table 7, we ablate how generations differ when changing the goal rewards in the system prompt. We observe that the AlpacaEval 2.0 scores of the Qwen2-7B-It+DPO (RA) model change accordingly as g varies. However, using the same $g = 10$

prompt during inference for the Qwen2-7B-It+DPO (UF) model fails to give competitive results, indicating that our method is superior not only because of the additional system prompt.

	$g = 10$	$g = 8$	$g = 6$	UF ($g = 10$)
LC WR	**31.17**	28.66	25.56	24.44
WR	**27.58**	25.57	18.88	20.75

Table 7: Different goal reward in the inference prompt.

Impact of the Accuracy of AI Feedback.
We consider the 19.8k prompts from a 1/3 subset of UltraFeedback following the setup from Snorkel (Hoang Tran, 2024). Five on-policy responses are first generated from Llama-3-8B-Instruct[7]. An external reward model is followed to rank these responses. We choose the best and worst responses as the chosen and rejected ones. DPO is then performed on the resulting preference pairs and the reward-augmented pairs. To ablate how our method will be impacted by the accuracy of AI feedback, we experiment with two reward models as the ranker: PairRM (Jiang et al., 2023b) and ArmoRM (Wang et al., 2024b). PairRM is a small-sized (0.4B) pairwise reward model, while ArmoRM is a 8B model that is state-of-the-art on RewardBench (Lambert et al., 2024) and much stronger than PairRM. We implement a variant (denoted as RA+) of the proposed reward augmentation method that only conditions on the goal rewards of the chosen responses, not those of the rejected ones, leading to datasets that have the same size.

	Llama-3-	PairRM (0.4B)			ArmoRM (8B)	
	8B-Instruct	DPO (UF)	DPO (RA+)	DPO (RA)	DPO (UF)	DPO (RA+)
LC WR	22.92	41.76	44.72	48.20	42.32	**48.73**
WR	23.15	45.79	44.70	**53.17**	42.79	45.36

Table 8: Ablation on the impact of AI feedback quality on the AlpacaEval 2.0 benchmark.

The results in Table 8 demonstrate that training on augmented data conditioned on both chosen and rejected rewards is necessary for PairRM feedback, while relabeling with only the chosen rewards is sufficient to achieve strong performance for ArmoRM feedback. This aligns with our motivation outlined in Section 3.1: in noisy preference data, rejected responses may actually be of high quality, unlearning which can degrade performance. Similarly, low-quality chosen responses may also be reinforced. This issue does not arise with strong reward models that provide accurate preferences.

	SLiC-HF	ORPO	CPO	RRHF	KTO	IPO	RPO	R-DPO	SimPO	Ours
LC WR	26.9	28.5	28.9	31.3	33.1	35.6	40.8	41.1	44.7	**48.2**
WR	27.5	27.4	32.2	28.4	31.8	35.6	41.7	37.8	40.5	**53.2**

Table 9: Comparison between our method, i.e., Llama-3-8B-Instruct+DPO (RA) and baselines fine-tuned on the same model and on-policy data ranked by PairRM.

Moreover, in Table 9, we compare our method and various baselines under the same setting on the AlpacaEval 2.0 benchmark, including SLiC-HF (Zhao et al., 2023), ORPO (Hong et al., 2024), CPO (Xu et al., 2024a), RRHF (Yuan et al., 2024), KTO (Ethayarajh et al., 2024), IPO (Azar et al., 2023), R-DPO (Park et al., 2024), and SimPO (Meng et al., 2024), where the results are from Meng et al. (2024), as well as the RPO (Adler et al., 2024) baseline that we implement. Our method outperforms the above algorithms by a considerable margin.

Conditioning on Multi-Attribute Rewards Enables SOTA Models. In previous parts, our method is implemented by conditioning on the scalar reward values given by the judge models, either LLMs or reward models. We find that our approach is generalizable to settings of multi-dimensional rewards that correspond to different attributes, such as helpfulness and truthfulness. Specifically, we follow the setting from last part to construct the

	AlpacaEval 2.0		
	LC Win Rate	Win Rate	Avg. Len.
Ours	**56.57**	**52.19**	1840
SimPO	53.70	47.50	1777
OpenChat	17.48	11.36	1362

Table 10: Our method trained with DPO achieves SOTA when conditioning on 5-dim rewards.

[7] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

preference dataset by applying the ArmoRM reward model on the on-policy responses generated by Llama-3-8B-Instruct. Since ArmoRM is a multi-objective model that not only gives a scalar reward value but also predicts human-interpretable fine-grained attributes, we first select 5 attributes (namely complexity, instruction following, honesty, helpfulness, and intelligence depth) that have the highest average coefficients on the UltraFeedback data. Then we relabel the data by conditioning on the 5-dim reward and follow the implementation of using ArmoRM described in the last part. The resulting model achieves state-of-the-art within the Llama-3-8B-Instruct model family, surpassing the strong baselines including SimPO (Meng et al., 2024) that is trained also on on-policy data ranked by ArmoRM, and OpenChat (Wang et al., 2023a) fine-tuned with Conditioned-RLFT from the same Llama-3-8B-Instruct model.

Comparison with Conditional Fine-Tuning Baselines. We further compared with additional conditional post-training baselines on the offline UltraFeedback dataset (i.e., without on-policy data), including DPA (Wang et al., 2024a) that performs preference-conditioned rejection sampling fine-tuning, and SteerLM (Dong et al., 2023b) that proposed attribute conditioned SFT. Since both baselines aim to optimize a user-controllable attribute-conditioned LLM that is optimal under diverse preference profiles with different coefficients of the reward's attributes, in Figure 4, we plot the win rate curves of these methods under varying preference profiles, such as adjusting verbosity preferences as considered in Wang et al. (2024a). It can be observed that fine-tuned from Zephyr-SFT[8], our method achieves the best AlpacaEval 2.0 win rate. In addition to the comparison with the implemented RPO (Adler et al., 2024) in Table 9, we also report the performance of RPO fine-tuned on additional models including Qwen2-7B-Instruct and Gemma2-9B-It. As shown in Table 11, the implemented RPO is outperformed by our method across these models.

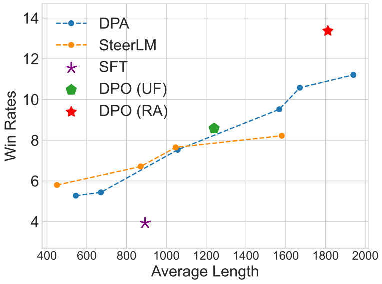


Figure 4: Comparison with DPA and SteerLM on Zephyr-SFT.

	AlpacaEval 2.0		
	LC Win Rate	Win Rate	Avg. Len.
Qwen+RPO	20.29	17.34	1704
Qwen+DPO (RA)	**31.17**	**27.58**	1789
Gemma+RPO	43.14	30.93	1413
Gemma+DPO (RA)	**59.27**	**54.56**	1872

Table 11: Comparison on AlpacaEval 2.0 between our method and RPO fine-tuned from the Qwen2-7B-Instruct and Gemma2-9B-It models. Our method consistently outperforms RPO across these fine-tuned models.

7 Conclusion

In this paper, we first investigate the limitations of direct alignment algorithms, which arise from focusing solely on relative preferences while neglecting the qualities of the responses and their gaps. Specifically, since many rejected responses are only slightly worse than the chosen ones, striving to maximize the reparameterized reward gap will cause overfitting and unnecessarily unlearning the high-quality rejected response. Moreover, the directly aligned LLMs often struggle to differentiate between responses of varying quality, indiscriminately learning the low-quality chosen responses and failing to generalize effectively to more optimal responses that are sparse in the preference data. To resolve the above limitations, we introduce a straightforward solution—learning reward-conditioned policies. By optimizing the LLM to generate responses conditioned on their qualities, it can better differentiate between quality levels and learn from the entire spectrum. Motivated by this, we propose a data relabeling method that constructs reward-augmented datasets by conditioning on the quality of responses as the goal quality. In experiments, we fine-tune various LLMs by applying DPO on our reward-augmented data. The results demonstrate that our approach consistently delivers significant performance improvements across various instruction-following benchmarks and increases the average accuracy on academic benchmarks. Comprehensive ablation studies demonstrate that our method not only enhances the utility of preference data but also addresses issues beyond simple dataset expansion, such as mitigating the unlearning limitation. Our method also achieves state-of-the-art results on AlpacaEval 2.0 when applied to on-policy data.

[8] https://huggingface.co/alignment-handbook/zephyr-7b-sft-full

References

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.

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

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv 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, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022a.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022b.

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, et al. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*, 2024.

Shicong Cen, Jincheng Mei, Katayoon Goshvadi, Hanjun Dai, Tong Yang, Sherry Yang, Dale Schuurmans, Yuejie Chi, and Bo Dai. Value-incentivized preference optimization: A unified approach to online and offline rlhf. *arXiv preprint arXiv:2405.19320*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023a.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv e-prints*, pp. arXiv–2405, 2024.

Yi Dong, Zhilin Wang, Makesh Narsimhan Sreedhar, Xianchao Wu, and Oleksii Kuchaiev. Steerlm: Attribute conditioned sft as an (user-steerable) alternative to rlhf. *arXiv preprint arXiv:2310.05344*, 2023b.

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

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

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

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*, 2023.

Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792*, 2024.

Braden Hancock Hoang Tran, Chris Glaze. Snorkel-mistral-pairrm-dpo. 2024. URL https://huggingface.co/snorkelai/Snorkel-Mistral-PairRM-DPO.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A Smith, Iz Beltagy, et al. Camels in a changing climate: Enhancing lm adaptation with tulu 2. *arXiv preprint arXiv:2311.10702*, 2023.

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. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023a.

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

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787*, 2024.

Shengzhi Li, Rongyu Lin, and Shichao Pei. Multi-modal preference alignment remedies regression of visual instruction tuning on language model. *arXiv preprint arXiv:2402.10884*, 2024a.

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

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Forty-first International Conference on Machine Learning*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. *arXiv preprint arXiv:2405.16436*, 2024.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Eric Mitchell. A note on dpo with noisy preferences & relationship to ipo, 2023.

Michael Noukhovitch, Samuel Lavoie, Florian Strub, and Aaron C Courville. Language model alignment with elastic reset. *Advances in Neural Information Processing Systems*, 36, 2024.

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. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

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

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

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. *Advances in Neural Information Processing Systems*, 36, 2024b.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

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

Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. Musr: Testing the limits of chain-of-thought with multistep soft reasoning. *arXiv preprint arXiv:2310.16049*, 2023.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alvaro Bartolome, Alexander M. Rush, and Thomas Wolf. The Alignment Handbook. URL `https://github.com/huggingface/alignment-handbook`.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. The alignment handbook. `https://github.com/huggingface/alignment-handbook`, 2023a.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023b.

Guan Wang, Sijie Cheng, Xianyuan Zhan, Xiangang Li, Sen Song, and Yang Liu. Openchat: Advancing open-source language models with mixed-quality data. *arXiv preprint arXiv:2309.11235*, 2023a.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong Zhang. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*, 2024a.

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

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36, 2024c.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528*, 2023b.

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

Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

Tengyang Xie, Dylan J Foster, Akshay Krishnamurthy, Corby Rosset, Ahmed Awadallah, and Alexander Rakhlin. Exploratory preference optimization: Harnessing implicit q*-approximation for sample-efficient rlhf. *arXiv preprint arXiv:2405.21046*, 2024.

Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*, 2023.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In *Forty-first International Conference on Machine Learning*, 2024.

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

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

Shusheng Xu, Wei Fu, Jiaxuan Gao, Wenjie Ye, Weilin Liu, Zhiyu Mei, Guangju Wang, Chao Yu, and Yi Wu. Is dpo superior to ppo for llm alignment? a comprehensive study. *arXiv preprint arXiv:2404.10719*, 2024b.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*, 2023.

Shenao Zhang, Donghan Yu, Hiteshi Sharma, Ziyi Yang, Shuohang Wang, Hany Hassan, and Zhaoran Wang. Self-exploring language models: Active preference elicitation for online alignment. *arXiv preprint arXiv:2405.19332*, 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.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36, 2024.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness and harmlessness with rlaif, November 2023.

A Experiment Details

A.1 Setup

We use the following prompt during training. Here, the reward values are the quality scores given by the judge models that exist in the preference dataset. The prompt is set as the system prompt whenever the LLM supports, such as Qwen2-7B-Instruct and Llama-3.1-8B-Instruct, and it is prefixed before the original prompt when the LLM doesn't support system prompting, such as Mistral-7B-Instruct-v0.3 and Gemma-2-9B-It.

Training prompt

```
You are an assistant that generates responses for the instruction
while implicitly achieving the following target score (on a scale of
1-10, where 1 is lowest and 10 is highest):
Overall score: {reward_value}.
```

At inference time, we use almost the same prompt, except that the goal score is the highest one, i.e., the overall score is 10.

Inference prompt

```
You are an assistant that generates responses for the instruction
while implicitly achieving the following target score (on a scale of
1-10, where 1 is lowest and 10 is highest):
Overall score: 10.
```

In our experiments using UltraFeedback, we directly leverage the LLM-as-Judge scores provided by GPT-4 in the dataset, which range from 1 to 10. For our method that is applied to on-policy data ranked by external reward models, including PairRM and ArmoRM, we apply linear transformations to normalize the resulting reward scores, ensuring they are scaled within the same 1 to 10 range.

For hyperparameters, we utilize a KL regularization coefficient of $\beta = 0.01$ in DPO, and we adopt the AdamW optimizer (Loshchilov, 2017). The batch size is set to 128, with a learning rate of $5e-7$ and a warmup ratio of 0.1. Furthermore, we observe that for models such as Qwen2-7B-Instruct and Gemma-2-9B-It on UltraFeedback, as well as Llama-3-8B-Instruct on on-policy data, both DPO and our proposed method yield improved performance when employing the conservative DPO (cDPO) technique (Mitchell, 2023). Consequently, for these models, we set the label smoothing hyperparameter from the Alignment Handbook (Tunstall et al., 2023a) to 0.3, while keeping it at 0 for the remaining models.

A.2 Full Results

In Table 12, we present the full results on instruction-following benchmarks, which correspond to the performance illustrated in Figure 2 in the main text.

	AlpacaEval 2.0			MT-Bench			Arena-Hard-Auto	
	LC WR	WR	Avg. Len.	Avg.	1st	2nd	Score	Avg. Len.
Mistral-7B-Instruct-v0.3	19.65	15.40	1503	7.67	8.00	7.34	17.0	494
+DPO (UltraFeedback)	18.76	16.93	1643	7.66	7.92	**7.40**	17.6	504
+DPO (Reward-Augmented)	**25.99**	**28.36**	2270	**7.69**	**8.02**	7.36	**18.3**	883
Qwen2-7B-Instruct	20.93	18.22	1788	7.90	8.23	7.56	24.3	617
+DPO (UltraFeedback)	21.46	19.35	1797	8.33	8.72	7.93	21.9	553
+DPO (Reward-Augmented)	**31.17**	**27.58**	1789	**8.47**	**8.93**	**7.97**	**30.1**	644
Llama-3.1-8B-Instruct	24.79	27.38	2081	8.44	8.99	7.90	26.9	831
+DPO (UltraFeedback)	28.67	30.21	2053	8.47	**9.01**	7.93	33.0	1070
+DPO (Reward-Augmented)	**31.20**	**35.93**	2006	**8.47**	8.91	**8.03**	**34.4**	824
Gemma-2-9B-It	49.20	37.58	1572	8.54	8.81	8.28	42.8	541
+DPO (UltraFeedback)	50.70	35.02	1464	8.54	8.70	**8.37**	35.8	456
+DPO (Reward-Augmented)	**59.27**	**54.56**	1872	**8.59**	**8.93**	8.25	**43.9**	611
SPPO	55.60	49.61	1822	8.40	8.53	8.26	47.6	578
+DPO (UltraFeedback)	52.75	40.58	1544	8.41	8.78	8.04	40.4	457
+DPO (Reward-Augmented)	**60.97**	**66.41**	2543	**8.73**	**9.06**	**8.41**	**49.0**	761

Table 12: Results of the DPO models fine-tuned on UltraFeedback and on reward-augmented UltraFeedback. We evaluate on the instruction-following benchmarks including AlpacaEval 2.0, MT-Bench, and Arena-Hard-Auto.