Direct Preference Optimization with an Offset

Afra Amini **Tim Vieira** **Ryan Cotterell**

{afra.amini, ryan.cotterell}@inf.ethz.ch tim.f.vieira@gmail.com

ETH zürich

Abstract

Direct preference optimization (DPO) is a successful fine-tuning strategy for aligning large language models with human preferences *without* the need to train a reward model or employ reinforcement learning. DPO, as originally formulated, relies on binary preference data and fine-tunes a language model to increase the likelihood of a preferred response over a dispreferred response. However, not all preference pairs are equal. Sometimes, the preferred response is only slightly better than the dispreferred one. In other cases, the preference is much stronger. For instance, if a response contains harmful or toxic content, the annotator will have a strong preference for that response. In this paper, we propose a generalization of DPO, termed **DPO with an offset (ODPO)**, that does not treat every preference pair equally during fine-tuning. Intuitively, ODPO requires the difference between the likelihood of the preferred and dispreferred response to be greater than an offset value. The offset is determined based on the extent to which one response is preferred over another. Our experiments on various tasks suggest that ODPO significantly outperforms DPO in aligning language models, especially when the number of preference pairs is limited.

https://github.com/rycolab/odpo

1 Introduction

Reinforcement learning from human feedback (RLHF) is a key building block in training the most modern large language models. The algorithm aligns the language model's responses to human preferences (Ouyang et al., 2022; Touvron et al., 2023; Anil et al., 2023; OpenAI et al., 2023). A typical implementation of RLHF is as follows. First, humans compare paired responses from a language model, i.e., they determine which of the responses is better, e.g., more helpful or less toxic. Second, a reward model is trained to give higher rewards to the responses preferred by humans. Lastly, a reinforcement learning algorithm is used to update the model's parameters to maximize the expected reward, while not diverging too much from the model's initial parameters.

The aforementioned implementation of RLHF, however, is tedious for two reasons. First, the reward model is usually a model as large as the language model itself, which is expensive to train and store. Second, reinforcement learning algorithms are known to be very sensitive to the choice of hyperparameters (Zheng et al., 2023), and, thus, hard to tune. Therefore, a fruitful line of work attempts to design alternative simpler implementations of RLHF (Welleck et al., 2020; Lu et al., 2022; Zhao et al., 2023).

A highly successful and straightforward alternative to RLHF is **direct preference optimization** (DPO; Rafailov et al., 2023). Unlike the conventional implementation of RLHF, DPO does not train a reward model and sidesteps the use of any reinforcement learning algorithm. Instead, it fine-tunes the language model's weights to maximize the likelihood of the preference data directly using the Bradley–Terry model (Bradley and Terry, 1952). Intuitively, each DPO gradient update to the model's parameters increases the likelihood of the preferred response and decreases the likelihood of the dispreferred response.

However, DPO only takes the ordering between the model's responses into account and not the extent to which one response is preferred over another. In many settings, the difference in quality between the two responses is known—either by asking humans (Stiennon et al., 2020; Touvron et al., 2023)[1] or through point-wise scores, e.g., toxicity scores, sentiment scores, code-based execution scores, etc. In that context, a natural question is the following: how can we make use of this information to better align language models with human preferences?

To answer this question, we propose a new method, which we term DPO with an offset

[1]For example, Touvron et al. (2023) ask humans to specify the extent to which they prefer one output over another, i.e., by choosing from: significantly better, better, slightly better, negligibly better / unsure.

Findings of the Association for Computational Linguistics ACL 2024, pages 9954–9972
August 11-16, 2024

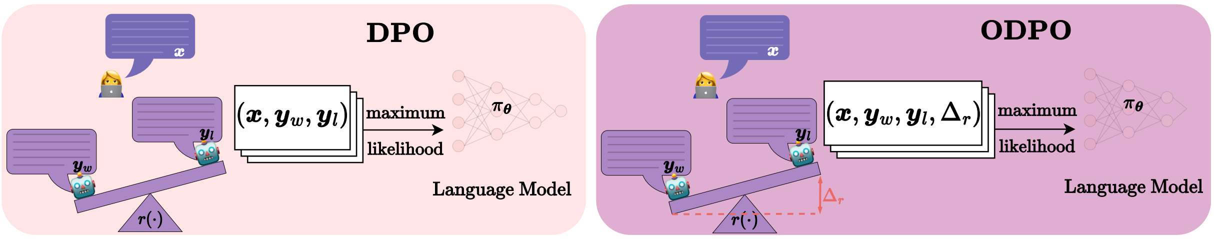


Figure 1: ODPO takes into account the extent to which one output should be preferred over another. The model has to put more probability mass on the preferred output compared to the dispreferred output by an offset that is determined based on how much the winning output is preferred over the losing output.

(ODPO). ODPO is a generalization of DPO that incorporates the difference between responses when modeling preference data. The intuition behind ODPO is simple; it requires the language model to increase the likelihood of the preferred responses compared to the dispreferred responses by an offset that is determined based on the difference between their associated reward values. Therefore, the larger the reward of the preferred response in comparison to the dispreferred response, the higher the likelihood needs to be over the likelihood of the dispreferred response. We further show that when the offset is set to zero, ODPO is equivalent to DPO. This process is illustrated in Fig. 1.

In our experiments, we fine-tune language models of various sizes with DPO and ODPO and compare the two methods. In cases where a ground-truth reward function is given, we measure two competing metrics: (i) the average reward associated with the generations from the fine-tuned model, and (ii) the KL divergence between the language model before and after the fine-tuning. Based on the results of sentiment and toxicity control tasks, we observe that ODPO more often appears on the Pareto frontier of reward and KL compared to DPO. We then apply ODPO to a summarization task, where we use the scores given by humans on a Likert scale to define an offset between the two summaries. We observe that on average and across different sampling temperatures, ODPO results in a higher win rate over human-written summaries compared to DPO.

2 Preliminaries

Given a prompt $\boldsymbol{x} \in \Sigma^*$, a language model $\pi_{\boldsymbol{\theta}}$ is a distribution over the responses $\boldsymbol{y} \in \Sigma^*$, where Σ is the alphabet.[2] As is standard, we parameterize the conditional distribution over responses given a prompt as an autoregressive language model $\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})$ parameterized by $\boldsymbol{\theta}$ as,

$$\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \pi_{\boldsymbol{\theta}}(\text{EOS} \mid \boldsymbol{y}, \boldsymbol{x}) \prod_{t=1}^{T} \pi_{\boldsymbol{\theta}}(y_t \mid \boldsymbol{y}_{<t}, \boldsymbol{x}), \tag{1}$$

where $\text{EOS} \notin \Sigma$ is a distinguished end-of-sequence token. To deploy a language model to a downstream task, it is usually necessary to additionally fine-tune it on high-quality data. Given a dataset of prompts and desired responses $(\boldsymbol{x}, \boldsymbol{y}) \sim \mathcal{D}_{\text{SFT}}$, the standard fine-tuning objective (to be maximized) is simply the log-likelihood of $\mathcal{D}_{\text{SFT}}$

$$\mathcal{J}^{\text{ML}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}_{\text{SFT}}} \Big[\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})\Big]. \tag{2}$$

We use π_{SFT} to refer to the language model fine-tuned with this objective on $\mathcal{D}_{\text{SFT}}$.

Such a fine-tuning approach can be effective for solving downstream tasks with language models. However, there is a disparity between the maximum-likelihood objective and the goal of the downstream task, which is to leverage the language model to generate high-quality responses, as judged by humans. This misalignment arises partly because the maximum-likelihood objective, Eq. (2), fails to distinguish between major and minor errors. For instance, when fine-tuning a language model to perform summarization, a significant error might involve hallucination, while a minor one might involve a failure to employ the appropriate synonyms (Stiennon et al., 2020). Training with the maximum-likelihood objective encourages the model to assign nonzero probability mass to all responses in $\mathcal{D}_{\text{SFT}}$, even those of lower quality.

To address this issue, one can optimize the expected reward of language model generations instead of using the maximum-likelihood objective.

[2] An alphabet is a finite, non-empty set.

Consider a reward function $r(\boldsymbol{x}, \boldsymbol{y})$, estimated from quality assessments performed by humans. The reward function assigns real values to prompt–response pairs; we will discuss in §3 how such a reward model can be learned from human feedback. Our objective is to make sure that the responses generated by the language model have high quality. Therefore, we can directly optimize the expected reward of the responses generated by the language model,

$$\mathcal{J}^{\mathrm{RL}}(\boldsymbol{\theta}) = \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \Big[r(\boldsymbol{x}, \boldsymbol{y})\Big], \tag{3}$$

where $\mathcal{D} = \{\boldsymbol{x}^{(n)}\}_{n=1}^{N}$ is a multiset of prompts. To prevent reward hacking (Amodei et al., 2016) and to make sure that we do not diverge too much from the supervised fine-tuned model π_{SFT}, a regularization term is often added to the objective (Stiennon et al., 2020),

$$\begin{aligned} \mathcal{J}^{\mathrm{RL}}(\boldsymbol{\theta}) = &\mathop{\mathbb{E}}_{\boldsymbol{x}\sim\mathcal{D}, \boldsymbol{y}\sim\pi_{\boldsymbol{\theta}}(\cdot|\boldsymbol{x})} \Big[r(\boldsymbol{x}, \boldsymbol{y})\Big] \\ &- \beta\, D_{\mathrm{KL}}\Big[\pi_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) \,\|\, \pi_{\mathrm{SFT}}(\boldsymbol{y} \mid \boldsymbol{x})\Big]. \end{aligned} \tag{4}$$

The above objective is optimized using proximal policy optimization (PPO; Schulman et al., 2017) or another actor–critic algorithm (Mnih et al., 2016; Glaese et al., 2022).

3 Reward Modeling

Pointwise Rewards. A key component in Eq. (4) is the task-dependent reward function that assigns pointwise real-valued rewards to each output. In many tasks, learning such reward functions is straightforward. For example, in open-ended text generation, a desired attribute could be the presence of indicators of positive sentiment in the generated text, while an undesired attribute could be toxicity. In such cases, the reward model might take the form of a classifier that assesses responses based on their sentiment or toxicity. Similarly, in code generation tasks, the quality of the code can be automatically evaluated, providing another straightforward example.

Pairwise Preferences. Learning a point-wise reward function for tasks like summarization or dialogue generation is more complex. Judging the absolute quality of a summary can depend on several factors, e.g., coherence, faithfulness, and conciseness, which makes it hard to collect human feedback datasets for reward model training. An effective strategy in such cases is to collect human preferences instead of point-wise judgments (Ziegler et al., 2020; Wu et al., 2021; Ouyang et al., 2022). In this setup, humans are shown two (or more) responses to a prompt $\boldsymbol{x}$ and are asked to select the response they prefer. Therefore, a datapoint in a human feedback dataset is a triple $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) \sim \mathcal{D}_{\mathrm{HF}}$, where $\boldsymbol{y}_w$ is preferred over $\boldsymbol{y}_l$. Given such a dataset, one needs to learn the point-wise reward function. It is common to assume that the preference data can be modeled by a Bradley–Terry model (Bradley and Terry, 1952),

$$\begin{aligned} &p_{\mathrm{BT}}(\boldsymbol{y}_w \succ \boldsymbol{y}_l \mid \boldsymbol{x}) \\ &\quad= \frac{\exp\big(r(\boldsymbol{x}, \boldsymbol{y}_w)\big)}{\exp\big(r(\boldsymbol{x}, \boldsymbol{y}_w)\big) + \exp\big(r(\boldsymbol{x}, \boldsymbol{y}_l)\big)} &\text{(5a)} \\ &\quad= \sigma\big(r(\boldsymbol{x}, \boldsymbol{y}_w) - r(\boldsymbol{x}, \boldsymbol{y}_l)\big), &\text{(5b)} \end{aligned}$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$ is the sigmoid function. Assuming that $\mathcal{D}_{\mathrm{HF}} \sim p_{\mathrm{BT}}$, we can train a reward model $r_{\boldsymbol{\phi}}$, parameterized by $\boldsymbol{\phi}$, as a binary classifier that maximizes the following log-likelihood

$$\mathbb{E}\Big[\log\sigma\big(r_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{y}_w) - r_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{y}_l)\big)\Big], \tag{6}$$

where the expectation is over $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l) \sim \mathcal{D}_{\mathrm{HF}}$. Intuitively, the reward function should assign higher reward values to the responses that are preferred by humans. The estimated reward function is then plugged into Eq. (4) to enable policy estimation.

It is often the case that directly optimizing Eq. (4) is complex and resource-intensive. Next, we will elaborate on an alternative method to using RLHF that has demonstrated comparable or even superior effectiveness to PPO.

4 Direct Preference Optimization

Rafailov et al. (2023) introduce a method to avoid reward model training and, thus, to directly optimize the language model. Their method, termed direct preference optimization (DPO) works as follows. The critical observation is that the optimal solution that maximizes the Eq. (4) is,

$$\begin{aligned} &\pi^{\star}_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x}) = \\ &\quad \frac{1}{Z(\boldsymbol{x})}\pi_{\mathrm{SFT}}(\boldsymbol{y} \mid \boldsymbol{x}) \exp\Big(\frac{1}{\beta} r(\boldsymbol{x}, \boldsymbol{y})\Big), \end{aligned} \tag{8}$$

where

$$Z(\boldsymbol{x}) = \sum_{\boldsymbol{y}\in\Sigma^*} \pi_{\mathrm{SFT}}(\boldsymbol{y} \mid \boldsymbol{x}) \exp\Big(\frac{1}{\beta} r(\boldsymbol{x}, \boldsymbol{y})\Big), \tag{9}$$

$$\mathcal{L}^{\text{DPO}}(\boldsymbol{\theta}) = -\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}_{\text{HF}}}\left[\log\sigma\left(\beta\log\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y}_w \mid \boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}_w \mid \boldsymbol{x})} - \beta\log\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y}_l \mid \boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}_l \mid \boldsymbol{x})}\right)\right] \tag{7a}$$

$$= -\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}_{\text{HF}}}\left[\log\sigma\left(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w) - \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l)\right)\right] \tag{7b}$$

$$\mathcal{L}^{\text{ODPO}}(\boldsymbol{\theta}) = -\mathop{\mathbb{E}}_{(\boldsymbol{x},\boldsymbol{y}_w,\boldsymbol{y}_l)\sim\mathcal{D}_{\text{HF}}}\left[\log\sigma\left(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w) - \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l) - \Delta_r\right)\right] \tag{7c}$$

Note that $Z(\boldsymbol{x})$ is, in general, difficult to compute as it involves a sum over a countably infinite set. Nevertheless, we can write the reward as a function of $\pi^\star_{\boldsymbol{\theta}}$ by rearranging the terms,

$$r(\boldsymbol{x},\boldsymbol{y}) = \beta\log\frac{\pi^\star_{\boldsymbol{\theta}}(\boldsymbol{y} \mid \boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y} \mid \boldsymbol{x})} + \beta\log Z(\boldsymbol{x}). \tag{10}$$

Under the assumption that the preference data is well-modeled by the Bradley–Terry model (Eq. (5a)), we substitute the reward in Eq. (6) with Eq. (10), and formulate the loss function in terms of the language model parameters. Therefore, we directly optimize the language model by maximizing the likelihood of the preference data in Eq. (7b). In this equation, $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}) = \beta\log\frac{\pi_{\boldsymbol{\theta}}(\boldsymbol{y}|\boldsymbol{x})}{\pi_{\text{SFT}}(\boldsymbol{y}|\boldsymbol{x})}$ is called the **estimated reward**. Note that the intractable normalization factor $Z(\boldsymbol{x})$ cancels out.

Simply put, Eq. (7b) requires the estimated reward for the preferred response to be larger than the estimated reward for the dispreferred response. A strong assumption behind the DPO's loss is that the preference data is well-modeled by a Bradley–Terry model. However, the Bradley–Terry model only tells us the probability that one response is preferred over another, and not the *extent* to which this preference will hold. To enhance DPO, we will introduce our modification to DPO, which generalizes DPO and introduces an offset between the responses in its loss function.

5 DPO with an Offset

The intuition behind ODPO is simple: we want the estimated reward for the preferred response to be larger than the estimated reward for the dispreferred response, by an offset that is a function of the actual reward differences assigned to the two responses. To begin our exposition of ODPO, we first discuss the connection between the Bradley–Terry model and Gumbel random variables, which we then use to construct the loss function for ODPO.

Theorem 1. *Let $\boldsymbol{y}_w$ and $\boldsymbol{y}_l$ be two responses to a prompt $\boldsymbol{x}$, and let $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w)$ and $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l)$ be their associated estimated rewards. Finally, let $R_w \sim \text{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w),1)$ and $R_l \sim \text{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l),1)$ be Gumbel random variables. Then, we have,*

$$\begin{aligned}\mathbb{P}(R_w - R_l > 0) &= p_{\text{BT}}(\boldsymbol{y}_w \succ \boldsymbol{y}_l \mid \boldsymbol{x}) \\ &= \sigma(\Delta_{\widehat{r}_{\boldsymbol{\theta}}}),\end{aligned} \tag{11}$$

where $p_{\text{BT}}(\boldsymbol{y}_w \succ \boldsymbol{y}_l \mid \boldsymbol{x})$ is a Bradley–Terry model Eq. (5a) parameterized by $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w)$ and $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l)$, and $\Delta_{\widehat{r}_{\boldsymbol{\theta}}} \stackrel{\text{def}}{=} \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w) - \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l)$ is the difference between the estimated rewards.

Proof. The proof is simple and follows directly from the Gumbel-max trick (Maddison and Tarlow, 2017). See App. A for details. ■

Theorem 2. *Let $\boldsymbol{y}_w$ and $\boldsymbol{y}_l$ be two responses for a prompt $\boldsymbol{x}$. Given a desired offset Δ_r between the responses' reward values,*

$$\mathbb{P}\big(R_w - R_l > \Delta_r\big) = \sigma(\Delta_{\widehat{r}_{\boldsymbol{\theta}}} - \Delta_r). \tag{12}$$

where $\Delta_{\widehat{r}_{\boldsymbol{\theta}}} \stackrel{\text{def}}{=} \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w) - \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l)$.

Proof. The difference between two independent Gumbel variables $R_w \sim \text{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_w),1)$ and $R_l \sim \text{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{y}_l),1)$ follows the logistic distribution. Therefore, the probability of the difference between the noisy estimated rewards being greater than Δ_r is,

$$\mathbb{P}\big(R_w - R_l > \Delta_r\big) = 1 - \mathcal{F}(\Delta_r), \tag{13}$$

where $\mathcal{F}(\cdot)$ is the CDF for logistic distribution. The mean of this distribution is $\Delta_{\widehat{r}_{\boldsymbol{\theta}}}$, and the variance is 1. Substituting the CDF in Eq. (13) with its definition,

$$\mathbb{P}\big(R_w - R_l > \Delta_r\big) = 1 - \mathcal{F}(\Delta_r) \tag{14a}$$

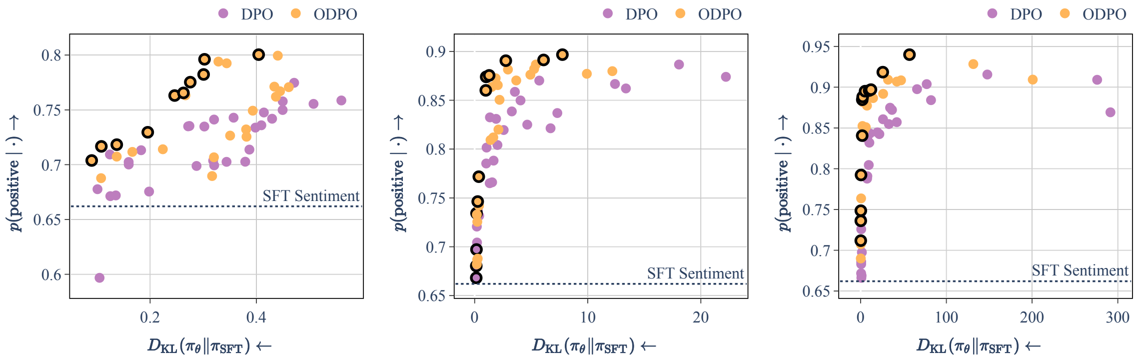
$$= \frac{1}{2} - \frac{1}{2}\tanh\Big(\frac{1}{2}(\Delta_r - \Delta_{\widehat{r}_{\boldsymbol{\theta}}})\Big) \tag{14b}$$

$$= \frac{1}{2} - \frac{1}{2}\Big(2\sigma(\Delta_r - \Delta_{\widehat{r}_{\boldsymbol{\theta}}}) - 1\Big) \tag{14c}$$

$$= 1 - \sigma(\Delta_r - \Delta_{\widehat{r}_{\boldsymbol{\theta}}}) \tag{14d}$$

$$= \sigma(\Delta_{\widehat{r}_{\boldsymbol{\theta}}} - \Delta_r). \tag{14e}$$

■



(a) $|\mathcal{D}_{\mathrm{HF}}| = 5000$, 100% of the points on the Pareto front belong to ODPO.

(b) $|\mathcal{D}_{\mathrm{HF}}| = 7500$, 66% of the points on the Pareto front belong to ODPO.

(c) $|\mathcal{D}_{\mathrm{HF}}| = 10000$, 100% of the points on the Pareto front belong to ODPO.

Figure 2: Steering generated movie reviews towards positive sentiment. Points on the Pareto front are highlighted with a black border. We observe that in all 3 settings, most (if not all) points on the Pareto front belong to ODPO.

Similar to DPO, we then maximize the likelihood of the preference data, as shown in Eq. (7c). Intuitively, minimizing the above loss function enforces the estimated reward for the preferred response $\boldsymbol{y}_w$ to be higher than $\boldsymbol{y}_l$ by the offset Δ_r. Comparing Eq. (7b) and Eq. (7c), when $\Delta_r = 0$, ODPO will be equivalent to DPO.

Designing Δ_r. We choose to model the offset as a monotonically increasing function $\mathbf{f}(\cdot)$ of the difference between the scores associated with the responses:

$$\Delta_r = \alpha\, \mathbf{f}\big(\text{score}(\boldsymbol{x}, \boldsymbol{y}_w) - \text{score}(\boldsymbol{x}, \boldsymbol{y}_l)\big). \quad (15)$$

where α is a hyperparameter that controls the extent to which an offset should be enforced. When $\alpha = 0$, ODPO becomes equivalent to DPO. In tasks where a ground-truth reward model $r(\cdot)$ is given we set $\text{score}(\cdot) = r(\cdot)$. In other tasks, one can use the quality scores given by humans to responses as score (see §6.3), or ask humans directly for the difference between the quality of the responses (Touvron et al., 2023). We ablate $\mathbf{f}$ in §7.1 and α in §7.2.

Connection to Softmax Margin. We now show how ODPO is connected to softmax margin (Gimpel and Smith, 2010). The idea behind the softmax margin is to augment the softmax with a cost function, such that high-cost responses get penalized more heavily. For two responses per prompt, we maximize

$$\mathop{\mathbb{E}}_{(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)} \left[\log \sigma \Big(\Delta_{\hat{r}_{\boldsymbol{\theta}}} - \text{cost}(\boldsymbol{y}_w, \boldsymbol{y}_l) \Big) \right]. \quad (16)$$

Importantly, the cost function is by definition a non-negative function. Therefore, ODPO loss Eq. (7c) is only equivalent to softmax-margin loss if we restrict $\Delta_r \geq 0$.

6 Experiments

In this section, we empirically compare the performance of ODPO with DPO across different tasks. We refer the reader to App. C for the computational budget used for conducting the experiments.

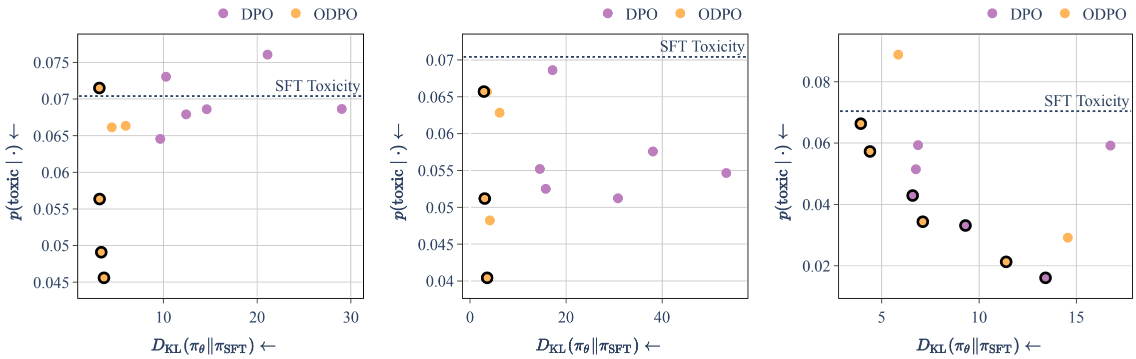
6.1 Sentiment Control

In this experiment, we steer generations of `GPT2-Large` (Radford et al., 2019) model towards positive sentiment. Following the experimental setup in (Rafailov et al., 2023), we fine-tune `GPT2-Large` on the train split of IMDB dataset (Maas et al., 2011) until convergence. This gives us the SFT checkpoint π_{SFT}.

Reward Model. We use a state-of-the-art binary sentiment classifier[3] with sentiments $\{\text{POS}, \text{NEG}\}$ as the reward model. Concretely if the sentiment of the response is negative, we set $r(\boldsymbol{x}, \boldsymbol{y})$ to $1 - p(\text{NEG} \mid \cdot)$, and if the sentiment is positive $r(\boldsymbol{x}, \boldsymbol{y}) = 1 + p(\text{POS} \mid \cdot)$, where p is given by the classifier. For notational ease, we show $r(\boldsymbol{x}, \boldsymbol{y})$ with $r(\boldsymbol{y})$.

Bootstraping the Preference Data. DPO, as opposed to RLHF, only works on top of preference data. Therefore, in tasks that a ground-truth reward model is given, the preference data needs to be bootstraped. We follow Rafailov et al. (2023) to bootstrap the preference dataset ($\mathcal{D}_{\mathrm{HF}}$) from point-wise rewards. Given the reward function, which in this experiment is given by the sentiment classifier, and a dataset of prompts that are prefixes of movie reviews, we proceed as follows. First,

[3] Specifically, we use `https://huggingface.co/lvwerra/distilbert-imdb`.



(a) $|\mathcal{D}_{\mathrm{HF}}| = 8000$, 100% of the points on the Pareto front belong to ODPO.
(b) $|\mathcal{D}_{\mathrm{HF}}| = 9000$, 100% of the points on the Pareto front belong to ODPO.
(c) $|\mathcal{D}_{\mathrm{HF}}| = 10000$, 57% of the points on the Pareto front belong to ODPO.

Figure 3: Steering generations away from toxic content. We highlight points on the Pareto front with a black border. We observe that, especially when the size of the dataset is small, ODPO manages to reduce the toxicity better than DPO while not diverging too far from the SFT model.

for each prompt $\boldsymbol{x} \in \mathcal{D}$, M responses are sampled from the language model $\pi_{\mathrm{SFT}}(\cdot \mid \boldsymbol{x})$[4] to form a multiset $Y_{\boldsymbol{x}} = \{\boldsymbol{y}_i\}_{m=1}^{M}$,[5] where $\boldsymbol{y}_m$ is the m^{th} response generated for prompt $\boldsymbol{x}$. Next, for each $i \neq j \in \{1, \ldots, M\}$ a tuple $(\boldsymbol{x}, \boldsymbol{y}_w, \boldsymbol{y}_l)$ is added to $\mathcal{D}_{\mathrm{HF}}$ if $r(\boldsymbol{y}_i) \neq r(\boldsymbol{y}_j)$, where

$$(\boldsymbol{y}_w, \boldsymbol{y}_l) = \begin{cases} (\boldsymbol{y}_i, \boldsymbol{y}_j) & \textbf{if } r(\boldsymbol{y}_i) > r(\boldsymbol{y}_j) \\ (\boldsymbol{y}_j, \boldsymbol{y}_i) & \textbf{if } r(\boldsymbol{y}_i) < r(\boldsymbol{y}_j) \end{cases}. \quad (17)$$

While DPO only uses the *order* of responses' rewards, with ODPO we further use the reward model to determine the offset between the two responses and set $\Delta_r = \log\big(r(\boldsymbol{y}_w) - r(\boldsymbol{y}_l)\big)$. Following Eq. (15), this is equivalent to choosing $\mathbf{f}(\cdot)$ as the log function and setting $\alpha = 1$.

We vary the KL regularization term β in $\{0.1, 0.2, \ldots, 1\} \cup \{1, 2, 3, 4, 5\}$[6] and sample from the fine-tuned language model $\pi_{\boldsymbol{\theta}}$ two times with two different random seeds. This gives us 28 different samples for each method. We use these samples to approximate the average probability of the generations having positive sentiment as well as the KL divergence between the fine-tuned model $\pi_{\boldsymbol{\theta}}$ and the initial model π_{SFT}. Ideally, we want not to diverge too much from the SFT policy, while generating movie reviews with a positive sentiment.

To capture the tradeoff between the KL divergence and achieved reward, we evaluate the two methods based on the Pareto frontier of achieved reward and KL divergence. Concretely, we report the percentage of points on the Pareto set that belong to each method. We compare the performance of ODPO and DPO in 3 different settings by varying the number of data points in the preference dataset, i.e., $|\mathcal{D}_{\mathrm{HF}}|$. As depicted in Fig. 2, in all experimental setups, ODPO is more effective in generating more samples with positive sentiment, while not diverging too far from π_{SFT}.[7] The difference between the two methods is more pronounced with smaller datasets, i.e., when $|\mathcal{D}_{\mathrm{HF}}| = 5000$.[8]

6.2 Toxicity Control

In this task, our goal is to reduce the toxicity of the generations. We use `GPT-neo-2.7b` (Black et al., 2021)[9] as the SFT checkpoint, π_{SFT}. We adversarially sample the prompts from REALTOXICITYPROMPTS (Gehman et al., 2020), where we sample 10000 prompts that have toxicity scores of more than 0.3. We generate two preference pairs from π_{SFT} for each prompt and compute their rewards using a toxicity classifier.[10] Similar to the previous experiment, we choose the offset as the log scaled differences between the rewards of preferred and dispreferred responses.

[4] One can also compare the responses generated by the model with human written responses (Stiennon et al., 2020).

[5] In this experiment, we generate two responses per prompt.

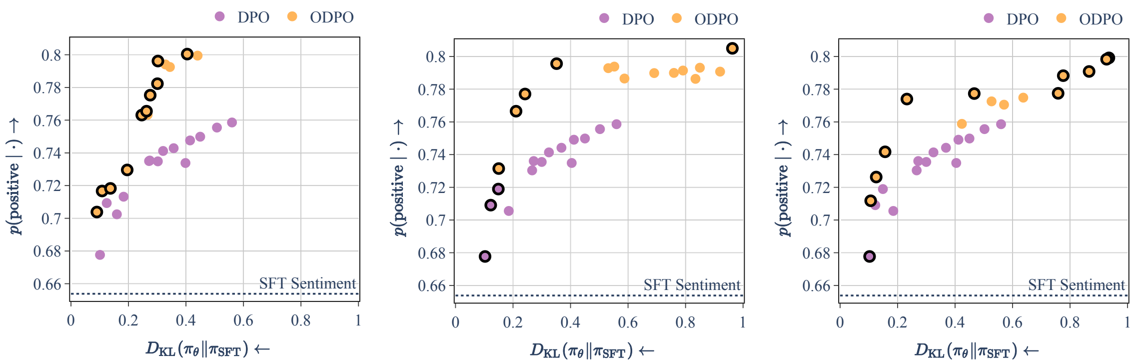
[6] Within the range of 0 to 1, increased resolution is necessary due to considerable variance in results. However, within the range of 1 to 5, the points are closely clustered, therefore, we increase the step size when covering this range.

[7] Further experiments comparing SLiC-HF and ODPO show the same trends App. D.

[8] When using more data, e.g., $|\mathcal{D}_{\mathrm{HF}}| = 10000$, we observe the positive sentiment probability gets closer to the maximum value 1., while this comes at the cost of diverging too much from π_{SFT}; we see an order of magnitude larger D_{KL} with $|\mathcal{D}_{\mathrm{HF}}| = 10000$ compared to $|\mathcal{D}_{\mathrm{HF}}| = 7500$.

[9] Specifically, we use https://huggingface.co/EleutherAI/gpt-neo-2.7B.

[10] We use https://huggingface.co/facebook/roberta-hate-speech-dynabench-r4-target.



(a) $\Delta_r = \log\big(r(\boldsymbol{y}_w) - r(\boldsymbol{y}_l)\big)$, 100% of points on Pareto front belong to ODPO.

(b) $\Delta_r = \log r(\boldsymbol{y}_w) - \log r(\boldsymbol{y}_l)$, 63% of points on Pareto front belong to ODPO.

(c) $\Delta_r = r(\boldsymbol{y}_w) - r(\boldsymbol{y}_l)$, 91% of points on Pareto front belong to ODPO.

Figure 4: Investigating the effect of the offset formulation on the performance of ODPO. Scaling the offset with a log function helps achieve the highest reward values without diverging too much from the SFT model.

We vary the KL regularization term $\beta \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5\}$[11] and sample from the fine-tuned model $\pi_{\boldsymbol{\theta}}$. We evaluate the samples based on their average toxicity probabilities and the KL divergence between $\pi_{\boldsymbol{\theta}}$ and π_{SFT}.

We observe that ODPO significantly outperforms DPO when using a dataset of size 8000 or 9000, where all the points on the Pareto front are from ODPO in Fig. 3. Concretely, in those two setups, DPO fails to meaningfully reduce the toxicity of the generations, as the toxicity scores are all close to the toxicity scores of generations from π_{SFT} (shown with a dashed horizontal line). However, ODPO manages to significantly reduce the toxicity of the generations, cutting it almost by half. As we increase the size of the dataset, the performance of DPO gets closer to ODPO.

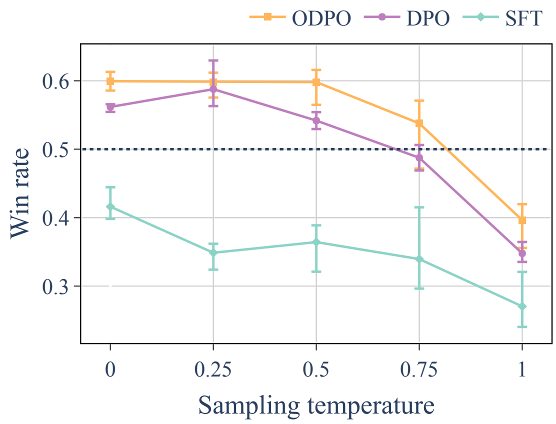


Figure 5: Win rates of generations from models fine-tuned with DPO and ODPO against human-written summaries. On average, ODPO achieves a higher win rate—significantly so in temperatures 0 and 0.5.

6.3 Summarization

Next, we test our method on a summarization task, where we directly use the scores given by humans to the pair of summaries for designing the offset. We use REDDIT TL;DR dataset collected by Stiennon et al. (2020). We take the subset of this dataset that is assessed by humans for quality using a 7-point Likert scale.[12] For each of the two summaries of a given Reddit post, we add a datapoint to $\mathcal{D}_{\mathrm{HF}}$ if the overall quality score of one is greater than the other. Moreover, we set the offset between the two summaries as the log difference between their associated scores by humans.

The supervised fine-tuned model (π_{SFT}) is a `GPTJ-6B` (Wang and Komatsuzaki, 2021) model that is fine-tuned on human-written forum post summaries.[13] We further fine-tune this model with DPO and ODPO on $\mathcal{D}_{\mathrm{HF}}$.[14] We use GPT-4 to compare the generations from the fine-tuned models against the reference summary written by humans, as GPT-4 judgments have been shown to strongly correlate with human judgments (Rafailov et al., 2023). See App. E for the prompt used for the evaluation.

We use 100 test prompts and sample from the fine-tuned models with different temperatures. The

[11] For $\beta > 0.5$, we observe that no toxicity reduction compared to the SFT model; therefore, for this experiment we only test $\beta \leq 0.5$.

[12] More specifically, we take the posts under `relationships` and `relationship_advice` subreddits. For more details regarding the datasets used refer to App. B.

[13] Specifically, we use `https://huggingface.co/CarperAI/openai_summarize_tldr_sft`

[14] Following Rafailov et al. (2023), we set $\beta = 0.5$ for both methods.

win rates of different methods against human written summaries are reported in Fig. 5. We observe that both DPO and ODPO improve upon the SFT model. At lower temperatures, i.e., $0, 0.25, 0.5$, both methods are on average preferred over human-written summaries. These results are consistent with the results reported in (Rafailov et al., 2023).

Importantly, across all sampling temperatures, the average win rate of ODPO is higher than DPO. Specifically, in temperatures 0 and 0.5 ODPO significantly outperforms DPO. We further perform a head-to-head comparison between DPO and ODPO and observe that the win rate of ODPO over DPO is $[51, 50, 62, 48, 57]$ at temperatures $[0, 0.25, 0.5, 0.75, 1.]$ respectively. For qualitative results refer to App. F.

7 Ablation Studies

In this section, we investigate and ablate certain design decisions that were made in the experiments. Specifically, we investigate the effect of the offset hyperparameter α, and the scaling function $\mathbf{f}(\cdot)$ on the performance of ODPO.

7.1 Ablating the Scaling Function

In our experiments, we scaled the reward difference between the preferred and dispreferred responses with a log function. To better understand the effect of this scaling function, we compare it to two other alternatives, (i) using the reward difference without scaling, i.e., setting $\mathbf{f}(\cdot)$ to the identity function, and (ii) using the difference between log of reward values as the offset.

We repeat the movie review generation experiment, where the goal is to generate movie reviews with a positive sentiment. For computational efficiency, we choose $|\mathcal{D}_{\mathrm{HF}}| = 5000$. We vary β between $\{0.1, 0.2, \ldots, 0.9\} \cup \{1, 2, 3, 4, 5\}$ and fine-tune $\pi_{\boldsymbol{\theta}}$ with the aforementioned offsets.

The results are depicted in Fig. 4. Overall, ODPO outperforms DPO with all three choices of the offset. However, there exist some notable differences between the three offsets. Scaling the offset with the log function makes the model not diverge too far from π_{SFT}, while achieving high rewards: While the models trained with log scaled offset (left plot) achieve high rewards (around 0.8) with KL of 0.4, models without log scaling reach to 0.8 rewards only when the KL between $\pi_{\boldsymbol{\theta}}$ and π_{SFT} is around 1 (right plot). Scaling each reward value separately lies somewhere in between (middle plot).

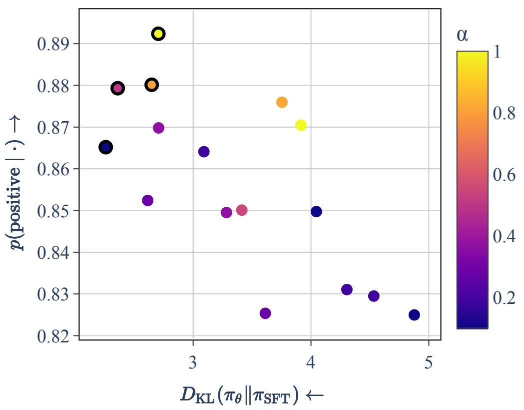


Figure 6: Ablating alpha values for movie review generation. The offset is set to $\Delta_r = \alpha \log\big(r(\boldsymbol{y}_w) - r(\boldsymbol{y}_l)\big)$. The highest rewards are associated with models trained with $\alpha = 1$, while smaller α leads to lower D_{KL}.

7.2 Ablating the Offset Hyperparameter

In the previous experiment, we set $\alpha = 1$. To understand the effect of enforcing an offset in the loss function we experiment with different values of α. Repeating the movie review experiment, we vary $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.8, 1.0\}$ and set the offset to $\Delta_r = \alpha \log\big(r(\boldsymbol{y}_w) - r(\boldsymbol{y}_l)\big)$. Note that when $\alpha = 0$ ODPO loss is exactly equal to DPO loss. After fine-tuning $\pi_{\boldsymbol{\theta}}$ on a dataset of size 7500 with the ODPO loss Eq. (7c) (with $\beta = 0.5$), we sample from the models two times with two different random seeds and report the results in Fig. 6.

We observe that higher values of α can lead to higher reward values at the expense of diverging from the SFT model. Lower values of α on average lead to lower reward values. On the Pareto front of the Fig. 6, points with the highest rewards are associated with models trained with $\alpha = 1$, while points with lower D_{KL} are fine-tuned with smaller α values.

8 Related Work

In this section, we review alternative methods to RLHF for aligning language models to human preferences.

Sequence-Level Contrastive Methods. The intuition that is shared by these methods is simple: they encourage the language model $\pi_{\boldsymbol{\theta}}$ to assign more probability mass to the preferred response compared to the dispreferred response(s). Zhao et al. (SLiC; 2023) employ a rank calibration loss that requires $\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}_w \mid \boldsymbol{x})$ to be greater than $\log \pi_{\boldsymbol{\theta}}(\boldsymbol{y}_l \mid \boldsymbol{x})$, by an offset δ. Importantly, in their formulation δ is a hyperparameter and *does not* depend on the the responses $\boldsymbol{y}_w, \boldsymbol{y}_l$. Similarly, Yuan et al. (RRHF; 2023) uses a ranking loss without the offset. While intuitive, the objective that contrastive losses are optimizing for, and its connection to the RLHF objective (maximizing the expected reward) is unclear.

DPO and Variants. DPO proposes an alternative method to optimize the regularized expected rewards without using RL algorithms. Importantly, DPO shares *the objective* with conventional methods for RLHF (Stiennon et al., 2020), but the optimization is done without training a separate reward model nor using RL algorithms. Since the introduction of DPO, several follow-up studies attempted to improve DPO along different dimensions. To prevent DPO from overfitting to the preference dataset, Azar et al. (2023) introduce Identity Preference Optimization (IPO). IPO replaces the unbounded function of preference probabilities in DPO loss formulation with the bounded identity function. Ethayarajh et al. (2023) propose a method called Kahneman-Tversky Optimisation (KTO) that dispenses the need for paired preference data altogether. KTO's loss function relies on *unpaired* examples that are labeled either as "good" or "bad". Zhou et al. (2024) suggest another variant of DPO specifically for multi-objective alignment.

Our Approach. ODPO attempts to solve another shortcoming of DPO, which is to treat every preference pair equally and not take into account the extent to which the two responses differ from each other. ODPO's loss requires the estimated reward for the preferred response to be larger than the dispreferred response by an offset that depends on the difference between the quality of the responses.

9 Conclusion

We propose ODPO, a generalization of DPO for aligning language models with human preferences. Just as with DPO, ODPO does not rely on a pretrained reward model and does not require an RL algorithm. However, in contrast to DPO, ODPO does not treat every preference pair equally and incorporates the extent to which one response should be preferred over another in its loss function. Experiments on a variety of tasks suggest that ODPO is more effective than DPO in aligning language models to human preferences.

Limitations

Human Preference Data. Not all datasets with human feedback contain judgments regarding the extent to which one response is preferred over another. In our experiments, we focused on tasks and datasets where we had either access to such information or point-wise reward functions. The results presented in the paper provide strong motivation to ask humans to indicate their degree of preference when collecting human feedback data.

Offset values. Deciding how to scale offset values can depend on the task. In this study, we experimented with offsets based on Likert scores and classifier probabilities. We defer extending ODPO to different tasks for future work.

Ethical Considerations

We foresee two main ethical concerns regarding the use of direct preference optimization and, in general, any RLHF method for aligning language models with human preferences. First, as with DPO and other RLHF methods, malicious actors can use ODPO to steer the responses to generate harmful or toxic content. However, we must note that we foresee no particular reason for DPO to be more suitable for malicious use cases compared to DPO or other RLHF methods. Second, reward functions might inherit unwanted biases from the datasets that they were trained on. Therefore, fine-tuning to increase the expected reward can then inject such biases into the language models' generations. We view ODPO as a fine-tuning strategy that can work with any given reward function or preference dataset. Therefore, as we develop more accurate and less biased reward functions, or find more inclusive ways of collecting human feedback, less unwanted bias will propagate to the fine-tuning process.

Acknowledgements

We thank Alizée Pace for her feedback on an early version of this paper. Afra Amini is supported by ETH AI Center doctoral fellowship.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in AI safety. *Computing Research Repository*, arXiv:1606.06565.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. PaLM 2 technical report. Technical report, Google.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A general theoretical paradigm to understand learning from human preferences. *Computing Research Repository*, arXiv:2310.12036.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

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

Kawin Ethayarajh, Winnie Xu, Dan Jurafsky, and Douwe Kiela. 2023. Human-centered loss functions (HALOs). Technical report, Contextual AI.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Kevin Gimpel and Noah A. Smith. 2010. Softmax-margin CRFs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736, Los Angeles, California. Association for Computational Linguistics.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. 2022. Improving alignment of dialogue agents via targeted human judgements. *Computing Research Repository*, arXiv:2209.14375.

Ximing Lu, Sean Welleck, Jack Hessel, Liwei Jiang, Lianhui Qin, Peter West, Prithviraj Ammanabrolu, and Yejin Choi. 2022. QUARK: Controllable Text generation with Reinforced Unlearning. In *Advances in Neural Information Processing Systems*, volume 35, pages 27591–27609. Curran Associates, Inc.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Chris J. Maddison and Danny Tarlow. Gumbel machinery [online]. 2017.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA. PMLR.

OpenAI, :, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mo Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2023. GPT-4 technical report. Technical report, OpenAI.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36. Curran Associates, Inc.

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

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal 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. 2023. Llama 2: Open foundation and fine-tuned chat models. Technical report, Meta.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2020. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

Jeff Wu, Long Ouyang, Daniel M. Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. 2021. Recursively summarizing books with human feedback. *Computing Research Repository*, arXiv:2109.10862.

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

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

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, Limao Xiong, Lu Chen, Zhiheng Xi, Nuo Xu, Wenbin Lai, Minghao Zhu, Cheng Chang, Zhangyue Yin, Rongxiang Weng, Wensen Cheng, Haoran Huang, Tianxiang Sun, Hang Yan, Tao Gui, Qi Zhang, Xipeng Qiu, and Xuanjing Huang. 2023. Secrets of RLHF in large language models part i: PPO. *Computing Research Repository*, arXiv:2307.04964.

Zhanhui Zhou, Jie Liu, Chao Yang, Jing Shao, Yu Liu, Xiangyu Yue, Wanli Ouyang, and Yu Qiao. 2024. Beyond one-preference-for-all: Multi-objective direct preference optimization. *Computing Research Repository*, arXiv:2310.03708.

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

A Proof of Thm. 1

Theorem 1. *Let $\boldsymbol{y}_w$ and $\boldsymbol{y}_l$ be two responses to a prompt $\boldsymbol{x}$, and let $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w)$ and $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$ be their associated estimated rewards. Finally, let $R_w \sim \mathrm{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w), 1)$ and $R_l \sim \mathrm{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l), 1)$ be Gumbel random variables. Then, we have,*

$$\begin{aligned} \mathbb{P}(R_w - R_l > 0) &= p_{\mathrm{BT}}(\boldsymbol{y}_w \succ \boldsymbol{y}_l \mid \boldsymbol{x}) \\ &= \sigma(\Delta_{\widehat{r}_{\boldsymbol{\theta}}}), \end{aligned} \tag{11}$$

where $p_{\mathrm{BT}}(\boldsymbol{y}_w \succ \boldsymbol{y}_l \mid \boldsymbol{x})$ is a Bradley–Terry model Eq. (5a) parameterized by $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w)$ and $\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$, and $\Delta_{\widehat{r}_{\boldsymbol{\theta}}} \stackrel{\text{def}}{=} \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w) - \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$ is the difference between the estimated rewards.

Proof. We define a random variable I as,

$$I \stackrel{\text{def}}{=} \underset{l,w}{\mathrm{argmax}}\{R_l, R_w\} \tag{18}$$

To prove the theorem, it suffices to prove,

$$\mathbb{P}(I = w) = \frac{\exp\left(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w)\right)}{\exp\left(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w)\right) + \exp\left(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)\right)} \tag{19}$$

For ease of notation, we define $g_{\widehat{r}_w} \stackrel{\text{def}}{=} \mathrm{Gumbel}(\widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w), 1)$, $\widehat{r}_w \stackrel{\text{def}}{=} \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_w)$, and $\widehat{r}_l \stackrel{\text{def}}{=} \widehat{r}_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{y}_l)$. Then, consider the following manipulation

$$\begin{aligned} \mathbb{P}(I = w) &= \mathop{\mathbb{E}}_{m \sim g_{\widehat{r}_w}}\left[\mathbb{P}(R_l < m)\right] && \text{(20a)} \\ &= \int_{-\infty}^{+\infty} g_{\widehat{r}_w}(m) \exp\big(-\exp(\widehat{r}_l - m)\big)\, \mathrm{d}m && \text{(20b)} \\ &= \int_{-\infty}^{+\infty} \exp\big(\widehat{r}_w - m - \exp(\widehat{r}_w - m)\big) \exp\big(-\exp(\widehat{r}_l - m)\big)\, \mathrm{d}m && \text{(20c)} \\ &= \int_{-\infty}^{+\infty} \exp(\widehat{r}_w) \exp(-m) \exp\Big(-\exp(-m)\underbrace{\big(\exp(\widehat{r}_w) + \exp(\widehat{r}_l)\big)}_{:=Z}\Big)\, \mathrm{d}m && \text{(20d)} \\ &= \exp(\widehat{r}_w) \int_{-\infty}^{+\infty} \exp(-m) \exp\Big(-Z\, \exp(-m)\Big)\, \mathrm{d}m && \text{(20e)} \\ &= \frac{\exp(\widehat{r}_w)}{\exp(\widehat{r}_w) + \exp(\widehat{r}_l)}. && \text{(20f)} \end{aligned}$$

■

B Datasets Statistics

We used the following datasets for either fine-tuning or prompting language models, which is consistent with the intended use case of the datasets. All the datasets are in English. We refer to the corresponding papers for data collection and postprocessing procedures, as well as the demographics of human annotators.

Sentiment Control. We train the SFT model on the train set of IMDB dataset (Maas et al., 2011), which consists of 25000 movie reviews. For fine-tuning with DPO and ODPO, we sample 10000 prompts from the train set and use the language model to generate continuations. For evaluation, we sample 256 prompts from the test set of IMDB dataset.

Toxicity Control. We sample 10000 prompts for fine-tuning and 256 for evaluation from REALTOXICITYPROMPTS (Gehman et al., 2020). The dataset is released under Apache-2.0 license.

	$\lvert\mathcal{D}_{\text{HF}}\rvert = 5000$		$\lvert\mathcal{D}_{\text{HF}}\rvert = 7500$		$\lvert\mathcal{D}_{\text{HF}}\rvert = 10000$	
Method	$D_{\text{KL}}(\pi_{\boldsymbol{\theta}} \Vert \pi_{\text{SFT}}) \downarrow$	Reward $\uparrow$	$D_{\text{KL}}(\pi_{\boldsymbol{\theta}} \Vert \pi_{\text{SFT}}) \downarrow$	Reward $\uparrow$	$D_{\text{KL}}(\pi_{\boldsymbol{\theta}} \Vert \pi_{\text{SFT}}) \downarrow$	Reward $\uparrow$
SLiC	$0.55[0.52, 0.56]$	$0.71[0.71, 0.72]$	$3.56[3.24, 3.88]$	$0.81[0.81, 0.82]$	$209.42[85.94, 333.7]$	$0.85[0.84, 0.86]$
DPO	$0.39[0.37, 0.4]$	$0.72[0.71, 0.73]$	4.13 [3.42, 5.27]	0.83 [0.82, 0.84]	32.32 [27.57, 36.68]	0.86 [0.85, 0.86]
ODPO	$\mathbf{0.35}[0.32, 0.39]$	$\mathbf{0.76}[0.75, 0.78]$	$\mathbf{3.09}[2.43, 4.04]$	$\mathbf{0.87}[0.87, 0.88]$	$\mathbf{18.73}[12.15, 29.65]$	$\mathbf{0.89}[0.88, 0.89]$

Table 1: Comparing ODPO to SLiC-HF on sentiment control task. In all 3 experimental setups ODPO achieves lower KL values and higher rewards compared to SLiC.

Summarization. We take 20000 posts under `relationships` and `relationship_advice` subreddits in REDDIT TL;DR dataset (Stiennon et al., 2020) and 100 posts for evaluation. The dataset is released under a modified MIT license.

C Computational Budget

For sentiment control experiments, a single fine-tuning and evaluation run takes approximately 20 minutes on 2 `rtx_4090` GPUs. For toxicity control experiments, a single fine-tuning and evaluation run takes approximately 2 hours on 2 `a100_40gb` GPUs. For the summarization task, a single fine-tuning and evaluation run takes approximately 15 hours on 2 `a100_80gb` GPUs. Notably, the reported runtimes are the same for DPO and ODPO, and there is no extra computation cost for ODPO compared to DPO.

D Comparison to SLiC-HF

We compare DPO and ODPO to SLiC-HF (Zhao et al., 2023) on the sentiment control task. Following Zhao et al. (2023), we set the margin hyperparameter to 1 for SLiC and β to 0.5 for DPO and ODPO. We observe that none of the SLiC runs end up on the Pareto frontier of KL divergence and reward. We report the mean values and 0.9 confidence intervals for $D_{\text{KL}}(\pi_{\boldsymbol{\theta}} \parallel \pi_{\text{SFT}})$ and reward in Tab. 1 for 3 dataset sizes.

E GPT-4 Evaluation Prompt for the Summarization Task

Following Rafailov et al. (2023) we use the prompt below to evaluate the win rates of generated summaries against human-written summaries.

```
Which of the following summaries does a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details? A good summary is both concise and precise.
Post:
<post>
Summary A:
<summary_a>
Summary B:
<summary_b>
FIRST provide a one-sentence comparison of the two summaries, explaining which you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your choice. Your response should use the format:
Comparison: <one-sentence comparison and explanation>
Preferred: <"A" or "B">
```

F Qualitative Results

Post	We have been dating for two and a half months, but we were friends in high school. We are both divorced, but her kids' dad takes them maybe one day a week. By their bedtime she's too tired to stay up for time for herself. I have more free time because I only have my kids for half the time. What can I say to her to help her get over the mentality she has to take care of all her problems alone? I'm willing to invest whatever it takes because we get along so well. Ask me any info that I may have left out.
DPO Summary	dating with benefits is giving my girlfriend anxiety and a mentality of taking care of her problems alone even though she said she wanted a relationship, not a one-man show. Do I need a break-up? Ask me anything.
ODPO Summary	Girl is tired when hanging out with me. What can I say to her to help her get over the mentality she has to take care of all her problems alone? We get along so well. Any info that I may have mentioned/left out.
Judgment	Summary A **[DPO]** brings up unnecessary details and confusion while Summary B **[ODPO]** is concise, straightforward and better delivers the post's message.

Table 2: An example of two summaries sampled with temperature 1, and its corresponding judgment from GPT-4. Summaries are shown to the model in random order and are tagged with A or B. For clarity, we add the method in brackets.

Post	I don't really have a question or anything. I just need to vent and maybe reach out to those who have been in similar situations. My sister told me yesterday that she's pregnant. At first I was thrilled. But then reality hit me. *I can't have children.* I'm so happy for my sister and my brother-in-law, I really am. But they weren't planning on having children for at least a few more years. My sister's never really wanted to be a mom. Me, on the other hand, have always lived for it. As a kid when people asked what I want to be when I grow up, I always said, "a mom." But, 3 years ago I found out I can never have children. I was devastated but have tried to tough it out, tried to believe it isn't true for years. I'm in a serious relationship and we're planning on getting married in a little over a year and starting to **try** having kids immediately, but we will probably end up adopting. While adopting is wonderful and that thought excites me, I still so badly want to be pregnant one day. My sister being pregnant is like a kick in the side. A reminder that I'll never be pregnant. I don't want to be around her, I don't want to be happy for her. I'm jealous. I don't know how to deal with this. I never expected to feel anger towards my sister for getting pregnant. But really, how am I expected to feel about this? I would never dare tell my sister or anyone else in my family how I feel about this. I wouldn't want to upset anyone and I really don't want to be selfish and make this about me. It's just so hard.
DPO Summary	I have a sister who's pregnant and I feel immensely jealous and sad and I don't know how to deal with my feelings. I don't want to have children.
ODPO Summary	My sister is pregnant and I'm jealous and sad even though I'm happy for her and I would love to be pregnant one day. I don't know how to feel. I really just want to vent.
Judgment	Summary A **[ODPO]** better captures the overall emotion, situation, and dilemma of the original post, while Summary B **[DPO]** falsely claims that the poster does not want children, which contradicts the information provided in the original post.

Table 3: An example of two summaries sampled with temperature 0.75, and its corresponding judgment from GPT-4. Summaries are shown to the model in random order and are tagged with A or B. For clarity, we add the method in brackets.

Post	We have been dating for two and a half months, but we were friends in high school. We are both divorced, but her kids' dad takes them maybe one day a week. By their bedtime she's too tired to stay up for time for herself. I have more free time because I only have my kids for half the time. What can I say to her to help her get over the mentality she has to take care of all her problems alone? I'm willing to invest whatever it takes because we get along so well. Ask me any info that I may have left out.
DPO Summary	My girlfriend has to take care of her kids and her problems because she doesn't have time to do anything for herself. How can I help her?
ODPO Summary	My girlfriend of two months is constantly tired from taking care of her kids. How can I help her get over the mentality she has to take care of everything alone?
Judgment	Summary B **[ODPO]** is more concise and accurate as it mentions the key point about the poster wanting to help his girlfriend overcome her mentality of taking care of everything by herself, compared to Summary A **[DPO]** which does not convey this aspect clearly.

Table 4: An example of two summaries sampled with temperature 0.5, and its corresponding judgment from GPT-4. Summaries are shown to the model in random order and are tagged with A or B. For clarity, we add the method in brackets.

Post	I am currently in college and I met this wonderful girl. We have gone on a few dates and everything appeared to be going great. She wanted to taking things quite slow because she was getting over a recent ex boyfriend. However when I met her roommate she brought up that this girl has a boyfriend who has been acting rather shady towards her. He goes to another college and he has been spending the night "on the couch" at this other girls place frequently. He also has recently been posting on her wall and now I'm rather confused. I know something is up but I really don't know how to handle this. I would really like to be straight forward with her but she seems dodgy whenever I do. I really am beginning to over think things and I almost feel like I was just there to shove in his face. I really hate this situation because I feel like I could have a legitimate relationship with this girl.
DPO Summary	Met a girl who has a boyfriend who is spending the night at her place frequently. She has been posting on his wall and I feel like she is hiding something. I don't know how to proceed.
ODPO Summary	met girl, she has a boyfriend who is acting shady towards her, she wants to take things slow, I'm confused as to what is going on.
Judgment	Summary A **[ODPO]** provides more accurate information about the situation, specifically pointing out that the girl wants to take things slow, and expresses the poster's confusion more effectively, whereas, Summary B **[DPO]** provides false information about the girl posting on the boyfriend's wall.

Table 5: An example of two summaries sampled with temperature 0.25, and its corresponding judgment from GPT-4. Summaries are shown to the model in random order and are tagged with A or B. For clarity, we add the method in brackets.

Post	My best friend and I have been friends for around 5 years now? She and her boyfriend has been together for a few years alr too. She is the kind of person who always put relationship over friendship, and because she's my best friend, I understand her and never said anything. I understand when she stop replying me for days when she stays over at her boyfriend's house. I understand when she ask me to leave early because she wanted to spend time with her boyfriend. I understand when she said I'm not allowed to be in contact with her boyfriend.(One time she got angry and ignored me for a month because I called her boyfriend to come over when they had a fight) I'm really okay with her being like this. Her boyfriend is an important part of her life and I can accept that. But recently, I can't even get a reply from her. Her text replies are always short and cold. We go to different schools so I don't get to talk/see her everyday. I'm always the one looking for a topic to talk about with her. She only comes looking for me if she and her boyfriend has a fight and she needed somebody to talk to or when her boyfriend is too busy to talk or go out with her. I had confronted her about it, but she said that it's not like that, she didn't mean to make me feel this way. But after that it's always back to the same situation and I always end up feeling used and neglected. Any advice?
DPO Summary	My best friend doesn't reply me anymore and I feel like she doesn't care about me anymore. She always put her boyfriend over me. I'm okay with her being like this, but I'm not okay with feeling like she doesn't care about me anymore. Any advice?
ODPO Summary	Best friend of 5 years doesn't reply to my texts anymore and always puts her boyfriend over me. I feel like she doesn't care about me anymore. Any advice?
Judgment	Summary B **[ODPO]** is preferred as it encapsulates all the necessary details more concisely and accurately without any unnecessary repetition.

Table 6: An example of two summaries generated with greedy decoding, and its corresponding judgment from GPT-4. Summaries are shown to the model in random order and are tagged with A or B. For clarity, we add the method in brackets.