

---

# Formatting Instructions For NeurIPS 2024

---

Anonymous Author(s)

Affiliation

Address

email

## Abstract

While large-scale unsupervised language models (LMs) learn broad world knowledge and some reasoning skills, achieving precise control of their behavior is difficult due to the completely unsupervised nature of their training. Existing methods for gaining such steerability collect human labels of the relative quality of model generations and fine-tune the unsupervised LM to align with these preferences, often with reinforcement learning from human feedback (RLHF). However, RLHF is a complex and often unstable procedure, first fitting a reward model that reflects the human preferences, and then fine-tuning the large unsupervised LM using reinforcement learning to maximize this estimated reward without drifting too far from the original model. In this paper, we leverage a mapping between reward functions and optimal policies to show that this constrained reward maximization problem can be optimized exactly with a single stage of policy training, essentially solving a classification problem on the human preference data. The resulting algorithm, which we call Direct Preference Optimization (DPO), is stable, performant, and computationally lightweight, eliminating the need for fitting a reward model, sampling from the LM during fine-tuning, or performing significant hyperparameter tuning. Our experiments show that DPO can fine-tune LMs to align with human preferences as well as or better than existing methods. Notably, fine-tuning with DPO exceeds RLHF’s ability to control sentiment of generations and improves response quality in summarization and single-turn dialogue while being substantially simpler to implement and train.

## 1 Introduction

Large unsupervised language models (LMs) trained on very large datasets acquire surprising capabilities [11, 7, 40, 8]. However, these models are trained on data generated by humans with a wide variety of goals, priorities, and skillsets. Some of these goals and skillsets may not be desirable to imitate; for example, while we may want our AI coding assistant to understand common programming mistakes in order to correct them, nevertheless, when generating code, we would like to bias our model toward the (potentially rare) high-quality coding ability present in its training data. Similarly, we might want our language model to be aware of a common misconception believed by 50% of people, but we certainly do not want the model to claim this misconception to be true in 50%. In other words, selecting the model’s desired responses and behavior from its very wide knowledge and abilities is crucial to building AI systems that are safe, performant, and controllable [26]. While existing methods typically steer LMs to match human preferences using reinforcement learning (RL), we will show that the RL-based objective used by existing methods can be optimized exactly with a simple binary cross-entropy objective, greatly simplifying the preference learning pipeline. At a high level, existing methods instill the desired behaviors into a language model using curated sets of human preferences representing the types of behaviors that humans find safe and helpful. This preference learning stage occurs after an initial stage of large-scale unsupervised pre-training on a large text dataset. While the most straightforward approach to preference learning is supervised fine-tuning on human demonstrations of high quality re-

sponses, the most successful class of methods is reinforcement learning from human (or AI) feedback (RLHF/RLAIF; [12, 2]). RLHF methods fit a reward model to a dataset of human preferences and then use RL to optimize a language model policy to produce responses assigned high reward without drifting excessively far from the original model. While RLHF produces models with impressive conversational and coding abilities, the RLHF pipeline is considerably more complex than supervised learning, involving training multiple LMs and sampling from the LM policy in the loop of training, incurring significant computational costs. In this paper, we show how to directly optimize a language model to adhere to human preferences, without explicit reward modeling or reinforcement learning. We propose Direct Preference Optimization (DPO), an algorithm that implicitly optimizes the same objective as existing RLHF algorithms (reward maximization with a KL-divergence constraint) but is simple to implement and straightforward to train. Intuitively, the DPO update increases the relative log probability of preferred to dispreferred responses, but it incorporates a dynamic, per-example importance weight that prevents the model degeneration that we find occurs with a naive probability ratio objective. Like existing algorithms, DPO relies on a theoretical preference model (such as the Bradley-Terry model; [5]) that measures how well a given reward function aligns with empirical preference data. However, while existing methods use the preference model to define a preference loss to train a reward model and then train a policy that optimizes the learned reward model, DPO uses a change of variables to define the preference loss as a function of the policy directly. Given a dataset of human preferences over model responses, DPO can therefore optimize a policy using a simple binary cross entropy objective, without explicitly learning a reward function or sampling from the policy during training. Our main contribution is Direct Preference Optimization (DPO), a simple RL-free algorithm for training language models from preferences. Our experiments show that DPO is at least as effective as existing methods, including PPO-based RLHF, for learning from preferences in tasks such as sentiment modulation, summarization, and dialogue, using language models with up to 6B parameters.

## 2 Related Work

Self-supervised language models of increasing scale learn to complete some tasks zero-shot [31] or with few-shot prompts [6, 25, 11]. However, their performance on downstream tasks and alignment with user intent can be significantly improved by fine-tuning on datasets of instructions and humanwritten completions [23, 36, 13, 39]. This ‘instruction-tuning’ procedure enables LLMs to generalize to instructions outside of the instruction-tuning set and generally increase their usability [13]. Despite the success of instruction tuning, relative human judgments of response quality are often easier to collect than expert demonstrations, and thus subsequent works have fine-tuned LLMs with datasets of human preferences, improving proficiency in translation [18], summarization [38, 48], story-telling [48], and instruction-following [26, 32]. These methods first optimize a neural network reward function for compatibility with the dataset of preferences under a preference model such as the Bradley-Terry model [5], then fine-tune a language model to maximize the given reward using reinforcement learning algorithms, commonly REINFORCE [44], proximal policy optimization (PPO; [37]), or variants [32]. A closely-related line of work leverages LLMs fine-tuned for instruction following with human feedback to generate additional synthetic preference data for targeted attributes such as safety or harmlessness [2], using only weak supervision from humans in the form of a text rubric for the LLM’s annotations. These methods represent a convergence of two bodies of work: one body of work on training language models with reinforcement learning for a variety of objectives [33, 27, 45] and another body of work on general methods for learning from human preferences [12, 19]. Despite the appeal of using relative human preferences, fine-tuning large language models with reinforcement learning remains a major practical challenge; this work provides a theoretically-justified approach to optimizing relative preferences without RL. Outside of the context of language, learning policies from preferences has been studied in both bandit and reinforcement learning settings, and several approaches have been proposed. Contextual bandit learning using preferences or rankings of actions, rather than rewards, is known as a contextual dueling bandit (CDB; [47, 14]). In the absence of absolute rewards, theoretical analysis of CDBs substitutes the notion of an optimal policy with a von Neumann winner, a policy whose expected win rate against any other policy is at least 50% given online, while in learning from human preferences, we typically learn from a fixed batch of offline preference-annotated action pairs [46]. Similarly, preference-based RL (PbRL) learns from binary preferences generated by an unknown ‘scoring’ function rather than rewards [9, 35]. Various algorithms for PbRL exist, including methods that can reuse off-policy preference data, but generally

involve first explicitly estimating the latent scoring function (i.e. the reward model) and subsequently optimizing it [16, 9, 12, 34, 19]. We instead present a single stage policy learning approach that directly optimizes a policy to satisfy preferences.

### 3 Preliminaries

We review the RLHF pipeline in Ziegler et al. (and later [38, 1, 26]). It usually includes three phases: 1) supervised fine-tuning (SFT); 2) preference sampling and reward learning and 3) RL optimization. **SFT:** RLHF typically begins by fine-tuning a pre-trained LM with supervised learning on high-quality data for the downstream tasks of interest (dialogue, summarization, etc.), to obtain a model  $\pi_{SFT}$ .

**Reward Modeling Phase:** In the second phase, the  $\pi_{SFT}$  model is prompted with prompts  $x$  to produce pairs of answers  $(y_1, y_2) \sim \pi_{SFT}(y \mid x)$ . These are then presented to human labelers who express preferences for one answer, denoted as  $y_w \succ y_l \mid x$  where  $y_w$  and  $y_l$  denotes the preferred and dispreferred completion amongst  $(y_1, y_2)$  respectively. The preferences are assumed to be generated by some latent reward model  $r^*(y, x)$ , which we do not have access to. There are a number of approaches used to model preferences, the Bradley-Terry (BT) model [5] being a popular choice (although more general Plackett-Luce ranking models [30, 21] are also compatible with the framework if we have access to several ranked answers). The BT model stipulates that the human preference distribution  $p^*$  can be written as:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}. \quad (1)$$

Assuming access to a static dataset of comparisons  $D = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$  sampled from  $p^*$ , we can parameterize a reward model  $r(x, y)$  and estimate the parameters via maximum likelihood. Framing the problem as a binary classification we have the negative log-likelihood loss:

$$\mathcal{L}_R(r, D) = -\mathbb{E}_{(x, y_w, y_l) \sim D} [\log \sigma(r(x, y_w) - r(x, y_l))], \quad (2)$$

where  $\sigma$  is the logistic function. In the context of LMs, the network  $r(x, y)$  is often initialized from the  $\pi_{SFT}(y \mid x)$  model with the addition of a linear layer on top of the final transformer layer that produces a single scalar prediction for the reward value [48]. To ensure a reward function with lower variance, prior works normalize the rewards, such that  $\mathbb{E}_{x, y \sim D}[r(x, y)] = 0$  for all  $x$ .

**RL Fine-Tuning Phase:** During the RL phase, we use the learned reward function to provide feedback to the language model. In particular, we formulate the following optimization problem:

$$\max_{\pi_\theta} \mathbb{E}_{x \sim D, y \sim \pi_\theta(y|x)} [r(x, y)] - \lambda KL[\pi_\theta(y \mid x) \parallel \pi_{ref}(y \mid x)], \quad (3)$$

where  $\lambda$  is a parameter controlling the deviation from the base reference policy  $\pi_{ref}$ , namely the initial SFT model  $\pi_{SFT}$ . The added constraint is important, as it prevents the model from deviating too far from the distribution on which the reward model is accurate, as well as maintaining the generation diversity and preventing mode-collapse to single high-reward answers. Due to the discrete nature of language generation, this objective is not differentiable and is typically optimized with reinforcement learning. The standard approach [48, 38, 1, 26] has been to construct the reward function  $r(x, y) = r(x, y) - \lambda(\log \pi_\theta(y|x) - \log \pi_{ref}(y|x))$ , and maximize using PPO [37].

### 4 Direct Preference Optimization

Motivated by the challenges of applying reinforcement learning algorithms on large-scale problems such as fine-tuning language models, our goal is to derive a simple approach for policy optimization using preferences directly. Unlike prior RLHF methods, which learn a reward and then optimize it via RL, our approach bypasses the reward modeling step and directly optimizes a language model using preference data. As we will describe next in detail, our key insight is to leverage an analytical mapping from reward functions to optimal policies, which enables us to transform a loss function over reward functions into a loss function over policies. This change-of-variables approach allows us to skip the explicit reward modeling step, while still optimizing under existing models of human preferences, such as the Bradley-Terry model. In essence, the policy network represents both the language model and the reward.

140 **Deriving the DPO Objective** We start with the same RL objective as prior work, Eq. 3, under a  
 141 general reward function  $r$ . Following prior work (?? ??), it is straightforward to show that the  
 142 optimal solution to the KL-constrained reward maximization objective in Eq. 3 takes the form:

$$\pi_r(y | x) = \frac{1}{Z(x)} \pi_{ref}(y | x) \exp\left(\frac{1}{\lambda} r(x, y)\right), 4 \quad (4)$$

143 where  $Z(x) = \sum_y \pi_{ref}(y | x) \exp\left(\frac{1}{\lambda} r(x, y)\right)$  is the partition function. See Appendix A.1 for a  
 144 complete derivation. Even if we use the MLE estimate  $r$  of the ground-truth reward function  $r^*$ , it is  
 145 still expensive to estimate the partition function  $Z(x)$  (??), which makes this representation hard to  
 146 utilize in practice. However, we can rearrange Eq. 4 to express the reward function in terms of its  
 147 corresponding optimal policy  $\pi_r$ , the reference policy  $\pi_{ref}$ , and the unknown partition function  $Z(\cdot)$ .  
 148 Specifically, we first take the logarithm of both sides of Eq. 4 and then with some algebra we obtain:

$$r(x, y) = \lambda \log \frac{\pi_r(y | x)}{\pi_{ref}(y | x)} + \lambda \log Z(x). 5 \quad (5)$$

149 We can apply this reparameterization to the ground-truth reward  $r^*$  and corresponding optimal model  
 150  $\pi^*$ . Fortunately, the Bradley-Terry model depends only on the difference of rewards between two  
 151 completions, i.e.,  $p^*(y_1 \succ y_2 | x) = \sigma(r^*(x, y_1) - r^*(x, y_2))$ . Substituting the reparameterization in  
 152 Eq. 5 for  $r^*(x, y)$  into the preference model Eq. 1, the partition function cancels, and we can express  
 153 the human preference probability in terms of only the optimal policy  $\pi^*$  and reference policy  $\pi_{ref}$ .  
 154 Thus, the optimal RLHF policy  $\pi^*$  under the Bradley-Terry model satisfies the preference model:

$$p^*(y_1 \succ y_2 | x) = \frac{1}{1 + \exp\left(\log \frac{\pi^*(y_2|x)}{\pi_{ref}(y_2|x)} - \log \frac{\pi^*(y_1|x)}{\pi_{ref}(y_1|x)}\right)}. 6 \quad (6)$$

155 The derivation is in Appendix A.2. While Eq. 6 uses the Bradley-Terry model, we can similarly  
 156 derive expressions under the more general Plackett-Luce models (??), shown in Appendix A.3.

157 Now that we have the probability of human preference data in terms of the optimal policy rather than  
 158 the reward model, we can formulate a maximum likelihood objective for a parametrized policy  $\pi_\theta$ .  
 159 Analogous to the reward modeling approach (i.e., Eq. 2), our policy objective becomes:

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref}) = \mathbb{E}_{(x, y_w, y_l) \sim D} \left[ \log \frac{\log \frac{\pi_\theta(y_w|x)}{\pi_{ref}(y_w|x)}}{\log \frac{\pi_\theta(y_l|x)}{\pi_{ref}(y_l|x)}} \right]. 7 \quad (7)$$

160 This way, we simultaneously bypass the explicit reward modeling step while also avoiding the need  
 161 to perform reinforcement learning optimization. Moreover, since our procedure is equivalent to fitting  
 162 a reparametrized Bradley-Terry model, it enjoys certain theoretical properties, such as consistencies  
 163 under suitable assumptions of the preference data distribution (?). In Section 5, we further discuss  
 164 theoretical properties of DPO in relation to other works.

165 **What does the DPO update do?** For a mechanistic understanding of DPO, it is useful to analyze the  
 166 gradient of the loss function  $\mathcal{L}_{DPO}$ . The gradient with respect to the parameters  $\theta$  can be written as:

$$\frac{\partial \mathcal{L}_{DPO}(\pi_\theta; \pi_{ref})}{\partial \theta} = \mathbb{E}_{(x, y_w, y_l) \sim D} \left[ (\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)) \left( \frac{\partial}{\partial \theta} \log \pi_\theta(y_w | x) - \frac{\partial}{\partial \theta} \log \pi_\theta(y_l | x) \right) \right],$$

167 where  $\hat{r}_\theta(x, y) = \lambda \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)}$  is the reward implicitly defined by the language model  $\pi_\theta$  and  
 168 reference model  $\pi_{ref}$  (more in Section 5). Intuitively, the gradient of the loss function  $\mathcal{L}_{DPO}$   
 169 increases the likelihood of the preferred completions  $y_w$  and decreases the likelihood of dispreferred  
 170 completions  $y_l$ . Importantly, the examples are weighted by how much higher the implicit reward  
 171 model  $\hat{r}_\theta$  rates the dispreferred completions, scaled by  $\lambda$ , i.e., how incorrectly the implicit reward  
 172 model orders the completions, accounting for the strength of the KL constraint. Our experiments  
 173 suggest the importance of this weighting, as a naïve version of this method without the weighting  
 174 coefficient can cause the language model to degenerate (Appendix Table 3).

## 175 5 Theoretical Analysis of DPO

176 In this section, we give further interpretation of the DPO method, provide theoretical backing, and  
 177 relate advantages of DPO to issues with actor-critic algorithms used for RLHF (such as PPO (?)).

## 5.1 Your Language Model Is Secretly a Reward Model

DPO is able to bypass both explicit reward estimation and RL to learn the policy using a single maximum likelihood objective. However, the optimization objective (Eq. 5) is equivalent to a Bradley-Terry model with a reward function  $r^*(x, y) = \lambda \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)}$  and we optimize our parametric model  $\pi_\theta$ , equivalently to the reward model optimization in Eq. 2 under this change of variables. In this section, we will build the theory behind this reparameterization, show that it does not constrain the class of learned reward models, and allows for the exact recovery of the optimal policy. We begin by defining an equivalence relation between reward functions.

**Definition 1.** We say that two reward functions  $r(x, y)$  and  $r_0(x, y)$  are equivalent iff

$$r(x, y) - r_0(x, y) = f(x) \text{ for some function } f.$$

It is easy to see that this is indeed an equivalence relation, which partitions the set of reward functions into classes. We can state the following two lemmas:

**Lemma 1.** Under the Plackett-Luce, and in particular the Bradley-Terry, preference framework, two reward functions from the same class induce the same preference distribution.

**Lemma 2.** Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem.

The proofs are straightforward and we defer them to Appendix A.5. The first lemma is a well-known under-specification issue with the Plackett-Luce family of models (?). Due to this under-specification, we usually have to impose additional identifiability constraints to achieve any guarantees on the MLE estimates from Eq. 2 (?). The second lemma states that all reward functions from the same class yield the same optimal policy, hence for our final objective, we are only interested in recovering an arbitrary reward function from the optimal class. We prove the following theorem in Appendix A.6:

**Theorem 1.** Under mild assumptions, all reward classes consistent with the Plackett-Luce (and Bradley-Terry in particular) models can be represented with the reparameterization

$$r(x, y) = \lambda \log \frac{\pi(y|x)}{\pi_{ref}(y|x)} \text{ for some model } \pi(y|x) \text{ and a given reference model } \pi_{ref}(y|x).$$

**Proof Sketch.** Consider any reward function  $r(x, y)$ , which induces a corresponding optimal model  $\pi_r(y|x)$ , specified by Eq. 4. We will show that a reward function from the equivalence class of  $r$  can be represented using the reparameterization given above. We define the projection  $f$  as

$$f(r; \pi_{ref}, \lambda)(x, y) = r(x, y) - \lambda \log \sum_y \pi_{ref}(y|x) \exp \left( \frac{1}{\lambda} r(x, y) \right),$$

The operator  $f$  simply normalizes the reward function with the logarithm of the partition function of  $\pi_r$ . Since the added normalization term is only a function of the prefix  $x$ ,  $f(r; \pi_{ref}, \lambda)(x, y)$  is a reward function in the equivalence class of  $r(x, y)$ . Finally, replacing  $r$  with the RHS of Eq. 5 (which holds for any reward function), we have

$$f(r; \pi_{ref}, \lambda)(x, y) = \lambda \log \frac{\pi_r(y|x)}{\pi_{ref}(y|x)},$$

that is, the projection  $f$  produces a member of the equivalence class of  $r$  with the desired form, and we do not lose any generality in our reward model from the proposed reparameterization.

## 5.2 Instability of Actor-Critic Algorithms

We can also use our framework to diagnose instabilities with standard actor-critic algorithms used for the RLHF, such as PPO. We follow the RLHF pipeline and focus on the RL fine-tuning step outlined in Section 3. We can draw connections to the control as inference framework (?) for the constrained RL problem outlined in 3. We assume a parameterized model  $\pi_\theta(y|x)$  and minimize

$DKL[\pi_\theta(y|x) || \pi^*(y|x)]$  where  $\pi^*$  is the optimal policy from Eq. 7 induced by the reward function  $r(y, x)$ .

With some algebra, this leads to the optimization objective:

$$\max_{\pi_\theta} \mathbb{E}_{\pi_\theta(y|x)} \left[ r(x, y) - \lambda \log \sum_y \pi_{ref} \exp \left( \frac{1}{\lambda} r(x, y) \right) \right],$$

216 This is the same objective optimized in prior works (???) using the DPO-equivalent reward for  
 217 the reward class of  $r$ . In this setting, we can interpret the normalization term in  $f(r, \pi_{ref}, \lambda)$  as the  
 218 soft value function of the reference policy  $\pi_{ref}$ . While this term does not affect the optimal solution,  
 219 without it, the policy gradient of the objective could have high variance, making learning unstable.  
 220 We can accommodate for the normalization term using a learned value function, but that can also be  
 221 difficult to optimize. Alternatively, prior works have normalized rewards using a human completion  
 222 baseline, essentially a single sample Monte-Carlo estimate of the normalizing term. In contrast, the  
 223 DPO reparameterization yields a reward function that does not require any baselines.

## 224 6 Experiments

225 In this section, we empirically evaluate DPO’s ability to train policies directly from preferences. First,  
 226 in a well-controlled text-generation setting, we ask: how efficiently does DPO trade off maximizing  
 227 reward and minimizing KL-divergence with the reference policy, compared to common preference  
 228 learning algorithms such as PPO? Next, we evaluate DPO’s performance on larger models and more  
 229 difficult RLHF tasks, including summarization and dialogue. We find that with almost no tuning  
 230 of hyperparameters, DPO tends to perform as well or better than strong baselines like RLHF with  
 231 PPO as well as returning the best of  $N$  sampled trajectories under a learned reward function. Before  
 232 presenting these results, we describe the experimental set-up; additional details are in Appendix C.

### 233 6.1 Tasks

234 Our experiments explore three different open-ended text generation tasks. For all experiments,  
 235 algorithms learn a policy from a dataset of preferences  $D = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ . In controlled  
 236 sentiment generation,  $x$  is a prefix of a movie review from the IMDB dataset (?), and the policy  
 237 must generate  $y$  with positive sentiment. In order to perform a controlled evaluation, for this  
 238 experiment we generate preference pairs over generations using a pre-trained sentiment classifier,  
 239 where  $p(\text{positive}|x, y_w) > p(\text{positive}|x, y_l)$ . For SFT, we fine-tune GPT-2-large until convergence  
 240 on reviews from the train split of the IMDB dataset (further details in App C.1). In summarization,  $x$   
 241 is a forum post from Reddit; the policy must generate a summary  $y$  of the main points in the post.  
 242 Following prior work, we use the Reddit TL;DR summarization dataset (?) along with human  
 243 preferences gathered by Stiennon et al.. We use an SFT model fine-tuned on human-written forum  
 244 post summaries with the TRLX (?) framework for RLHF. The human preference dataset was  
 245 gathered by Stiennon et al. on samples from a different, but similarly-trained, SFT model. Finally, in  
 246 single-turn dialogue,  $x$  is a human query, which may be anything from a question about astrophysics  
 247 to a request for relationship advice. A policy must produce an engaging and helpful response  $y$  to  
 248 a user’s query; we use the Anthropic Helpful and Harmless dialogue dataset (?), containing 170k  
 249 dialogues between a human and an automated assistant. Each transcript ends with a pair of responses  
 250 generated by a large (although unknown) language model along with a preference label denoting  
 251 the human-preferred response. In this setting, no pre-trained SFT model is available; we therefore  
 252 fine-tune an off-the-shelf language model on only the preferred completions to form the SFT model.

### 253 6.2 Evaluation

254 Our experiments use two different approaches to evaluation. In order to analyze the effectiveness  
 255 of each algorithm in optimizing the constrained reward maximization objective, in the controlled  
 256 sentiment generation setting we evaluate each algorithm by its frontier of achieved reward and  
 257 KL-divergence from the reference policy; this frontier is computable because we have access to the  
 258 ground-truth reward function (a sentiment classifier). However, in the real world, the ground truth  
 259 reward function is not known; therefore, we evaluate algorithms with their win rate against a baseline  
 260 policy, using GPT-4 as a proxy for human evaluation of summary quality and response helpfulness  
 261 in the summarization and single-turn dialogue settings, respectively. For summarization, we use  
 262 reference summaries in the test set as the baseline; for dialogue, we use the preferred response in the  
 263 test dataset as the baseline. While existing studies suggest LMs can be better automated evaluators  
 264 than existing metrics (?), we conduct a human study to justify our usage of GPT-4 for evaluation  
 265 in Sec. 6.4. We find GPT-4 judgments correlate strongly with humans, with human agreement with  
 266 GPT-4 typically similar or higher than inter-human annotator agreement.

### 6.3 Methods

In addition to DPO, we evaluate several existing approaches to training language models to adhere to human preferences. Most simply, we explore zero-shot prompting with GPT-J ( ? ) in the summarization task and 2-shot prompting with Pythia-2.8B ( ? ) in the dialogue task. In addition, we evaluate the SFT model as well as Preferred-FT, which is a model fine-tuned with supervised learning on the chosen completion  $y_w$  from either the SFT model (in controlled sentiment and summarization) or a generic LM (in single-turn dialogue). Another pseudo-supervised method is Unlikelihood, which simply optimizes the policy to maximize the probability assigned to  $y_w$  and minimize the probability assigned to  $y_l$ ; we use an optional coefficient  $\alpha \in [0, 1]$  on the 'unlikelihood' term. We also consider PPO ( ? ) using a reward function learned from the preference data and PPO-GT, which is an oracle that learns from the ground truth reward function available in the controlled sentiment setting. In our sentiment experiments, we use two implementations of PPO-GT, one off-the-shelf version ( ? ) as well as a modified version that normalizes rewards and further tunes hyperparameters to improve performance (we also use these modifications when running 'normal' PPO with learned rewards). Finally, we consider the Best of N baseline, sampling  $N$  responses from the SFT model (or Preferred-FT in dialogue) and returning the highest-scoring response according to a reward function learned from the preference dataset. This high-performing method decouples the quality of the reward model from the PPO optimization, but is computationally impractical even for moderate  $N$  as it requires sampling  $N$  completions for every query at test time.

### 6.4 How well can DPO optimize the RLHF objective?

The KL-constrained reward maximization objective used in typical RLHF algorithms balances exploitation of reward while restricting the policy from deviating far from the reference policy. Therefore, when comparing algorithms, we must take into account both the reward achieved as well as the KL discrepancy; achieving slightly higher reward but with much higher KL is not necessarily desirable.

Figure 2 shows the reward-KL frontier for various algorithms in the sentiment setting. We execute multiple training runs for each algorithm, using a different hyperparameter for policy conservativeness in each run (target KL  $\{3, 6, 9, 12\}$  for PPO,  $\{0.05, 0.1, 1, 5\}$ ,  $\alpha\{0.05, 0.1, 0.5, 1\}$  for unlikelihood, random seeds for Preferred-FT). This sweep includes 22 runs in total. After each 100 training steps until convergence, we evaluate each policy on a set of test prompts, computing the average reward under the true reward function as well as the average sequence-level KL with the reference policy KL ( $\pi || \pi_{ref}$ ).

We find that DPO produces by far the most efficient frontier, achieving the highest reward while still maintaining low KL. This result is particularly notable for multiple reasons. First, DPO and PPO optimize the same objective, but DPO is notably more efficient; DPO's reward/KL tradeoff strictly dominates PPO. Second, DPO achieves a better frontier than PPO, even when PPO can access ground truth rewards (PPO-GT).

### 6.5 Can DPO scale to real preference datasets?

Next, we evaluate the fine-tuning performance of DPO on summarization and single-turn dialogue tasks. For summarization, automatic evaluation metrics such as ROUGE can be poorly correlated with human preferences ( ? ), and prior work has found that fine-tuning language models (LMs) using PPO on human preferences provides more effective summaries. We evaluate different methods by sampling completions on the test split of the TL;DR summarization dataset, computing the average win rate against reference completions in the test set. The completions for all methods are sampled at temperatures varying from 0.0 to 1.0, and the win rates are shown in Figure 2 (right). DPO, PPO, and Preferred-FT all fine-tune the same GPT-J SFT model. We find that DPO has a win rate of approximately 61% at a temperature of 0.0, exceeding the performance of PPO at 57% at its optimal sampling temperature of 0.0. DPO also achieves a higher maximum win rate compared to the Best of  $N$  baseline. We note that we did not meaningfully tune DPO's hyperparameter, so these results may underestimate DPO's potential. Moreover, we find DPO to be much more robust to the sampling temperature than PPO, the performance of which can degrade to that of the base GPT-J model at high temperatures. Preferred-FT does not improve significantly over the SFT model. We also compare

319 DPO and PPO head-to-head in human evaluations in Section 6.4, where DPO samples at temperature  
 320 0.25 were preferred 58% times over PPO samples at temperature 0.

321 On single-turn dialogue, we evaluate the different methods on the subset of the test split of the  
 322 Anthropic HH dataset ( ? ) with one step of human-assistant interaction. GPT-4 evaluations use the  
 323 preferred completions on the test as the reference to compute the win rate for different methods. As  
 324 there is no standard SFT model for this task, we start with a pre-trained Pythia-2.8B, use Preferred-FT  
 325 to train a reference model on the chosen completions such that completions are within the distribution  
 326 of the model, and then train using DPO. We also compare against the Best of 128 Preferred-FT  
 327 completions (we found the Best of  $N$  baseline plateaus at 128 completions for this task; see Appendix  
 328 Figure 4) and a 2-shot prompted version of the Pythia-2.8B base model, finding DPO performs as  
 329 well or better for the best-performing temperatures for each method. We also evaluate an RLHF  
 330 model trained with PPO on the Anthropic HH dataset from a well-known source, but are unable  
 331 to find a prompt or sampling temperature that gives performance better than the base Pythia-2.8B  
 332 model. Based on our results from TL;DR and the fact that both methods optimize the same reward  
 333 function, we consider Best of 128 a rough proxy for PPO-level performance. Overall, DPO is the  
 334 only computationally efficient method that improves over the preferred completions in the Anthropic  
 335 HH dataset and provides similar or better performance to the computationally demanding Best of 128  
 336 baseline. Finally, Figure 3 shows that DPO converges to its best performance relatively quickly.

## 337 6.6 Generalization to a new input distribution

338 We assess the generalization capabilities of DPO by testing it alongside PPO under conditions of  
 339 distribution shifts. Specifically, we re-evaluate both DPO and PPO policies developed from our  
 340 Reddit TL;DR summarization experiment on a different distribution: news articles from the test split  
 341 of the CNN/DailyMail dataset ( ? ). We utilize the best sampling temperatures established from  
 342 TL;DR (0 and 0.25), and the comparative results are summarized in Table 1 below.

Algorithm	Temp 0	Temp 0.25
DPO	0.36	0.31
PPO	0.26	0.23

Table 1: GPT-4 win rates vs. ground truth summaries for out-of-distribution CNN/DailyMail input articles.

343 We computed the GPT-4 win rates against the ground-truth summaries in the dataset, adapting the  
 344 same GPT-4 (C) prompt used for the Reddit TL;DR. However, we modified the prompt by replacing  
 345 the words “forum post” with “news article” to better align with the new content type.

346 In this new dataset, DPO consistently outperforms PPO by a significant margin, suggesting that DPO  
 347 policies can generalize equally well, if not better than, PPO policies. This is particularly noteworthy as  
 348 DPO does not utilize the additional unlabeled Reddit TL;DR prompts that are employed by PPO. This  
 349 experiment provides initial evidence that DPO’s adaptation mechanisms are robust across different  
 350 textual distributions, further validating its efficacy under varied application conditions.

## 351 6.7 Validating GPT-4 judgments with human judgments

352 We conduct a human study to verify the reliability of GPT-4’s judgments, using the results of the  
 353 TL;DR summarization experiment and two different GPT-4 prompts. The GPT-4 (S) (simple) prompt  
 354 simply asks for which summary better summarizes the important information in the post. The GPT-4  
 355 (C) (concise) prompt asks for which summary is more concise. We evaluate this prompt because  
 356 we find that GPT-4 prefers longer, more repetitive summaries than humans do with the GPT-4 (S)  
 357 prompt. Complete prompts are available in Appendix C.2.

358 We perform three comparisons using the highest (DPO, temp. 0.25), the lowest (PPO, temp. 1.0),  
 359 and a middle-performing (SFT, temp. 0.25) method with the aim of covering a diversity of sample  
 360 qualities. All three methods are compared against greedily-sampled PPO at its best-performing  
 361 temperature. The comparison results are summarized in Table 2 below.

362 We find that with both prompts, GPT-4 tends to agree with humans about as often as humans agree  
 363 with each other, suggesting that GPT-4 is a reasonable proxy for human evaluations. This observation



Method	N Respondents	GPT-4 (S) Win %	GPT-4 (C) Win %	Human Win %	GPT-4 (S)-H Agree	GPT-4
DPO	272	47	54	58	70	
SFT	122	27	32	43	77	
PPO-1	199	13	12	17	86	

Table 2: Comparing human and GPT-4 win rates and per-judgment agreement on TL;DR summarization samples.

is critical, given that due to limited human raters, we only collect multiple human judgments for the DPO and PPO-1 comparisons. Overall, the GPT-4 (C) prompt generally provides win rates more representative of human judgments; we therefore use this prompt for the main results in Section 6.2.

For additional details about the human study, including the web interface presented to raters and the list of human volunteers, see Appendix D.3.

## 7 Discussion

Learning from preferences is a powerful, scalable framework for training capable, aligned language models. We have introduced DPO, a simple training paradigm for training language models from preferences without the need for reinforcement learning. Instead of coercing the preference learning problem into a standard RL framework to utilize off-the-shelf RL algorithms, DPO identifies a mapping between language model policies and reward functions that allows for training a language model to satisfy human preferences directly with a straightforward cross-entropy loss, thereby eliminating the need for reinforcement learning and without loss of generality. With virtually no tuning of hyperparameters, DPO performs similarly or better than existing RLHF algorithms, including those based on PPO; thus, DPO significantly reduces the barrier to training more language models from human preferences.

### 7.1 Limitations & Future Work

Our results raise several important questions for future work. How does the DPO policy generalize out of distribution, compared with learning from an explicit reward function? Our initial results suggest that DPO policies can generalize similarly to PPO-based models, but more comprehensive studies are needed. For example, can training with self-labeling from the DPO policy similarly make effective use of unlabeled prompts? On another front, how does reward over-optimization manifest in the direct preference optimization setting, and is the slight decrease in performance observed in Figure 3-right an instance of it? Additionally, while we evaluate models up to 6B parameters, exploring the scaling of DPO to state-of-the-art models orders of magnitude larger presents an exciting direction for future work. Regarding evaluations, we find that the win rates computed by GPT-4 are influenced by the prompt; future studies may investigate the best way to elicit high-quality judgments from automated systems. Finally, many possible applications of DPO exist beyond training language models from human preferences, including training generative models in other modalities.

## Acknowledgements

EM gratefully acknowledges funding from a Knight-Hennessy Graduate Fellowship. CF and CM are CIFAR Fellows. This work was supported in part by the Stanford Accelerator for Learning (SAL) and Stanford Institute for Human-Centered Artificial Intelligence (HAI) Generative AI for the Future of Learning seed grant program. The Stanford Center for Research on Foundation Models (CRFM) provided part of the compute resources used for the experiments in this work. This work was also supported in part by ONR grant N00014-20-1-2675.

## References

## References

- [1] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.
- [2] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, et al. Constitutional AI: Harmlessness from AI feedback, 2022.
- [3] S. Biderman, H. Schoelkopf, Q. Anthony, H. Bradley, K. O’Brien, E. Hallahan, M. A. Khan, S. Purohit, U. S. Prashanth, E. Raff, et al. Pythia: A suite for analyzing large language models across training and scaling, 2023.
- [4] H. Bong and A. Rinaldo. Generalized results for the existence and consistency of the MLE in the Bradley-Terry-Luce model. *International Conference on Machine Learning*, 2022. arXiv:2110.11487.
- [5] R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. <https://doi.org/10.2307/2334029>.
- [6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf).
- [7] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. Sparks of artificial general intelligence: Early experiments with GPT-4, 2023. arXiv preprint arXiv:2303.12712.
- [8] R. Busa-Fekete, B. Szörényi, P. Weng, W. Cheng, and E. Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine Learning*, 97(3):327–351, July 2014. <https://doi.org/10.1007/s10994-014-5458-8>.
- [9] Y. Chen, R. Wang, H. Jiang, S. Shi, and R.-L. Xu. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *ArXiv*, abs/2304.00723, 2023.
- [10] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, et al. Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311, 2022.
- [11] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf).

## A Appendix / supplemental material

**All authors** provided valuable contributions to designing, analyzing, and iterating on experiments, writing and editing the paper, and generally managing the project’s progress.

**RR** proposed using autoregressive reward models in discussions with **EM**; derived the DPO objective; proved the theoretical properties of the algorithm and wrote the relevant sections and appendices. He also suggested and helped with organizing experiments and contributed some of the PPO and reward learning baselines.

**AS** initiated the discussion on using weighted regression methods as an alternative to PPO; initiated project-related organization, wrote initial analysis connecting DPO with weighted regression and unlikelihood; design and iterations of DPO + baseline implementations, initial exploratory experiments for DPO; substantial experiment organization and design (datasets, baselines, evaluation); led model training and evaluation for controlled sentiment generation and summarization; design iterations for GPT-4 evaluation (particularly summarization); substantial writing contributions to abstract, prelims/method and experiments; editing contributions to other sections.

**EM** provided input on early discussions on learning autoregressive reward functions; wrote the first implementation of DPO and ran the first DPO experiments; trained the large-scale (summarization and dialogue) DPO models used in paper experiments; conducted initial GPT-4 win rate evaluations and set up related infrastructure; recruited participants for, conducted, and analyzed results from the human study; wrote the abstract, introduction, related work, discussion, and most of experiments; and assisted with editing the rest of the paper.

**CF, CM, & SE** supervised the research, suggested ideas and experiments, and assisted in writing the paper.

## A Mathematical Derivations

### A.1 Deriving the Optimum of the KL-Constrained Reward Maximization Objective

In this appendix, we will derive Eq. 4. Analogously to Eq. 3, we optimize the following objective:

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi} [r(x, y)] - \beta \mathbb{D}_{\text{KL}} [\pi(y|x) || \pi_{\text{ref}}(y|x)] \quad (11)$$

under any reward function  $r(x, y)$ , reference model  $\pi_{\text{ref}}$  and a general non-parametric policy class. We now have:

$$\begin{aligned} & \max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi} [r(x, y)] - \beta \mathbb{D}_{\text{KL}} [\pi(y|x) || \pi_{\text{ref}}(y|x)] \\ &= \max_{\pi} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi(y|x)} \left[ r(x, y) - \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} \right] \\ &= \min_{\pi} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi(y|x)} \left[ \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} - \frac{1}{\beta} r(x, y) \right] \\ &= \min_{\pi} \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{y \sim \pi(y|x)} \left[ \log \frac{\pi(y|x)}{\frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp \left( \frac{1}{\beta} r(x, y) \right)} - \log Z(x) \right] \end{aligned} \quad (12)$$

where we have partition function:

$$Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp \left( \frac{1}{\beta} r(x, y) \right).$$

Note that the partition function is a function of only  $x$  and the reference policy  $\pi_{\text{ref}}$ , but does not depend on the policy  $\pi$ . We can now define

$$\pi^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp \left( \frac{1}{\beta} r(x, y) \right),$$

439 which is a valid probability distribution as  $\pi^*(y|x) \geq 0$  for all  $y$  and  $\sum_y \pi^*(y|x) = 1$ . Since  $Z(x)$  is not a function of  $y$ , we can then re-organize the final objective in Eq 12 as:

$$\min_{\pi} \mathbb{E}_{x \sim \mathcal{D}} \left[ \mathbb{E}_{y \sim \pi(y|x)} \left[ \log \frac{\pi(y|x)}{\pi^*(y|x)} \right] - \log Z(x) \right] = \quad (13)$$

$$\min_{\pi} \mathbb{E}_{x \sim \mathcal{D}} [\mathbb{D}_{\text{KL}}(\pi(y|x) \parallel \pi^*(y|x)) + Z(x)] \quad (14)$$

Now, since  $Z(x)$  does not depend on  $\pi$ , the minimum is achieved by the policy that minimizes the first KL term. Gibbs' inequality tells us that the KL-divergence is minimized at 0 if and only if the two distributions are identical. Hence we have the optimal solution:

$$\pi(y|x) = \pi^*(y|x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp \left( \frac{1}{\beta} r(x, y) \right) \quad (15)$$

for all  $x \in \mathcal{D}$ . This completes the derivation.

## A.2 Deriving the DPO Objective Under the Bradley-Terry Model

It is straightforward to derive the DPO objective under the Bradley-Terry preference model as we have

$$p^*(y_1 \succ y_2|x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))} \quad (16)$$

In Section 4 we showed that we can express the (unavailable) ground-truth reward through its corresponding optimal policy:

$$r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x) \quad (17)$$

Substituting Eq. 17 into Eq. 16 we obtain:

$$\begin{aligned} p^*(y_1 \succ y_2|x) &= \frac{\exp \left( \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x) \right)}{\exp \left( \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x) \right) + \exp \left( \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} + \beta \log Z(x) \right)} \\ &= \frac{1}{1 + \exp \left( \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} \right)} \\ &= \sigma \left( \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} \right). \end{aligned}$$

The last line is the per-instance loss in Equation 7.

## A.3 Deriving the DPO Objective Under the Plackett-Luce Model

The Plackett-Luce model [30, 21] is a generalization of the Bradley-Terry model over rankings (rather than just pair-wise comparisons). Similar to the Bradley-Terry model, it stipulates that when presented with a set of possible choices, people prefer a choice with probability proportional to the value of some latent reward function for that choice. In our context, when presented with a prompt  $x$  and a set of  $K$  answers  $y_1, \dots, y_K$  a user would output a permutation  $\tau : [K] \rightarrow [K]$ , giving their ranking of the answers. The Plackett-Luce model stipulates that

$$p^*(\tau|y_1, \dots, y_K, x) = \prod_{k=1}^K \frac{\exp(r^*(x, y_{\tau(k)}))}{\sum_{j=k}^K \exp(r^*(x, y_{\tau(j)}))} \quad (18)$$

Notice that when  $K = 2$ , Equation 18 reduces to the Bradley-Terry model. However, for the general Plackett-Luce model, we can still utilize the results of Eq. 5 and substitute the reward function parameterized by its optimal policy. Similarly to Appendix A.2, the normalization constant  $Z(x)$  cancels out and we're left with:

$$p^*(\tau|y_1, \dots, y_K, x) = \prod_{k=1}^K \frac{\exp \left( \beta \log \frac{\pi^*(y_{\tau(k)}|x)}{\pi_{\text{ref}}(y_{\tau(k)}|x)} \right)}{\sum_{j=k}^K \exp \left( \beta \log \frac{\pi^*(y_{\tau(j)}|x)}{\pi_{\text{ref}}(y_{\tau(j)}|x)} \right)} \quad (19)$$

440 Similarly to the approach of Section 4, if we have access to a dataset  $\mathcal{D} = \{\tau^{(i)}, y_1^{(i)}, \dots, y_K^{(i)}, x^{(i)}\}_{i=1}^N$  of prompts and user-specified rankings, we can use a parameterized model and optimize this objective with maximum-likelihood.:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{\tau, y_1, \dots, y_K, x \sim \mathcal{D}} \left[ \log \prod_{k=1}^K \frac{\exp\left(\beta \log \frac{\pi_\theta(y_{\tau(k)}|x)}{\pi_{\text{ref}}(y_{\tau(k)}|x)}\right)}{\sum_{j=k}^K \exp\left(\beta \log \frac{\pi_\theta(y_{\tau(j)}|x)}{\pi_{\text{ref}}(y_{\tau(j)}|x)}\right)} \right] \quad (20)$$

#### A.4 Deriving the Gradient of the DPO Objective

In this section we derive the gradient of the DPO objective:

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\nabla_\theta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \log \sigma \left( \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} \right) \right] \quad (21)$$

We can rewrite the RHS of Equation 21 as

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[ \frac{\sigma'(u)}{\sigma(u)} \nabla_\theta (u) \right], \quad (22)$$

where  $u = \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}$ .

Using the properties of sigmoid function  $\sigma'(x) = \sigma(x)(1 - \sigma(x))$  and  $\sigma(-x) = 1 - \sigma(x)$ , we obtain the final gradient

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

After using the reward substitution of  $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$  we obtain the final form of the gradient from Section 4.

#### A.5 Proof of Lemma 1 and 2

In this section, we will prove the two lemmas from Section 5.

**Lemma 1 Restated.** *Under the Plackett-Luce preference framework, and in particular the Bradley-Terry framework, two reward functions from the same equivalence class induce the same preference distribution.*

*Proof.* We say that two reward functions  $r(x, y)$  and  $r'(x, y)$  are from the same equivalence class if  $r'(x, y) = r(x, y) + f(x)$  for some function  $f$ . We consider the general Plackett-Luce (with the Bradley-Terry model a special case for  $K = 2$ ) and denote the probability distribution over rankings induced by a particular reward function  $r(x, y)$  as  $p_r$ . For any prompt  $x$ , answers  $y_1, \dots, y_K$  and ranking  $\tau$  we have:

$$\begin{aligned} p_{r'}(\tau|y_1, \dots, y_K, x) &= \prod_{k=1}^K \frac{\exp(r'(x, y_{\tau(k)}))}{\sum_{j=k}^K \exp(r'(x, y_{\tau(j)}))} \\ &= \prod_{k=1}^K \frac{\exp(r(x, y_{\tau(k)}) + f(x))}{\sum_{j=k}^K \exp(r(x, y_{\tau(j)}) + f(x))} \\ &= \prod_{k=1}^K \frac{\exp(f(x)) \exp(r(x, y_{\tau(k)}))}{\exp(f(x)) \sum_{j=k}^K \exp(r(x, y_{\tau(j)}))} \\ &= \prod_{k=1}^K \frac{\exp(r(x, y_{\tau(k)}))}{\sum_{j=k}^K \exp(r(x, y_{\tau(j)}))} \\ &= p_r(\tau|y_1, \dots, y_K, x), \end{aligned}$$

which completes the proof.  $\square$

441 **Lemma 2 Restated.** *Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem.*

*Proof.* Let us consider two reward functions from the same class, such that  $r'(x, y) = r(x, y) + f(x)$  and, let us denote as  $\pi_r$  and  $\pi_{r'}$  the corresponding optimal policies. By Eq. 4, for all  $x, y$  we have

$$\begin{aligned}
\pi_{r'}(y|x) &= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x, y)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} (r(x, y) + f(x))\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} (r(x, y) + f(x))\right) \\
&= \frac{1}{\exp\left(\frac{1}{\beta} f(x)\right) \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \exp\left(\frac{1}{\beta} f(x)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \\
&= \pi_r(y|x),
\end{aligned}$$

which completes the proof.  $\square$

## A.6 Proof of Theorem 1

In this section, we will expand on the results of Theorem 1.

**Theorem 1 Restated.** *Assume, we have a reference model, such that  $\pi_{\text{ref}}(y|x) > 0$  for all pairs of prompts  $x$  and answers  $y$  and a parameter  $\beta > 0$ . All reward equivalence classes, as defined in Section 5 can be represented with the reparameterization  $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$  for some model  $\pi(y|x)$ .*

*Proof.* Consider any reward function  $r(x, y)$ , which induces an optimal model  $\pi_r(y|x)$  under the KL-constrained RL problem, with solution given by 4. Following Eq. 5, when we log-linearize both sides we obtain:

$$r(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x)$$

where  $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)$  (notice that  $Z(x)$  also depends on the reward function  $r$ ). Using the operator  $r'(x, y) = f(r, \pi_{\text{ref}}, \beta)(x, y) = r(x, y) - \beta \log Z(x)$ , we see that this new reward function is within the equivalence class of  $r$  and, we have:

$$r'(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{\text{ref}}(y|x)}$$

which completes the proof.  $\square$

We can further expand on these results. We can see that if  $r$  and  $r'$  are two reward functions in the same class, then

$$f(r, \pi_{\text{ref}}, \beta)(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{\text{ref}}(y|x)} = \beta \log \frac{\pi'_r(y|x)}{\pi_{\text{ref}}(y|x)} = f(r', \pi_{\text{ref}}, \beta)(x, y)$$

where the second equality follows from Lemma 2. We have proven that the operator  $f$  maps all reward functions from a particular equivalence class to the same reward function. Next, we show that for every equivalence class of reward functions, the reward function that has the reparameterization outlined in Theorem 1 is unique.

**Proposition 1.** *Assume, we have a reference model, such that  $\pi_{\text{ref}}(y|x) > 0$  for all pairs of prompts  $x$  and answers  $y$  and a parameter  $\beta > 0$ . Then every equivalence class of reward functions, as defined in Section 5, has a unique reward function  $r(x, y)$ , which can be reparameterized as  $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$  for some model  $\pi(y|x)$ .*

442 *Proof.* We will proceed using proof by contradiction. Assume we have two reward functions from the same class, such that  $r'(x, y) = r(x, y) + f(x)$ . Moreover, assume that  $r'(x, y) = \beta \log \frac{\pi'(y|x)}{\pi_{\text{ref}}(y|x)}$  for some model  $\pi'(y|x)$  and  $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$  for some model  $\pi(y|x)$ , such that  $\pi \neq \pi'$ . We then have

$$r'(x, y) = r(x, y) + f(x) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + f(x) = \beta \log \frac{\pi(y|x) \exp(\frac{1}{\beta} f(x))}{\pi_{\text{ref}}(y|x)} = \beta \log \frac{\pi'(y|x)}{\pi_{\text{ref}}(y|x)}$$

for all prompts  $x$  and completions  $y$ . Then we must have  $\pi(y|x) \exp(\frac{1}{\beta} f(x)) = \pi'(y|x)$ . Since these are distributions, summing over  $y$  on both sides, we obtain that  $\exp(\frac{1}{\beta} f(x)) = 1$  and since  $\beta > 0$ , we must have  $f(x) = 0$  for all  $x$ . Therefore  $r(x, y) = r'(x, y)$ . This completes the proof.  $\square$

We have now shown that every reward class has a unique reward function that can be represented as outlined in Theorem 1, which is given by  $f(r, \pi_{\text{ref}}, \beta)$  for any reward function in that class.

## B DPO Implementation Details and Hyperparameters

DPO is relatively straightforward to implement; PyTorch code for the DPO loss is provided below:

```

443 import torch.nn.functional as F

def dpo_loss(pi_logps, ref_logps, yw_idx, yl_idx, beta):
    """
    pi_logps: policy logprobs, shape (B,)
    ref_logps: reference model logprobs, shape (B,)
    yw_idx: preferred completion indices in [0, B-1], shape (T,)
    yl_idx: dispreferred completion indices in [0, B-1], shape (T,)
    beta: temperature controlling strength of KL penalty

    Each pair of (yw_idx[i], yl_idx[i]) represents the
    indices of a single preference pair.
    """

    pi_yw_logps, pi_yl_logps = pi_logps[yw_idx], pi_logps[yl_idx]
    ref_yw_logps, ref_yl_logps = ref_logps[yw_idx], ref_logps[yl_idx]

    pi_logratios = pi_yw_logps - pi_yl_logps
    ref_logratios = ref_yw_logps - ref_yl_logps

    losses = -F.logsigmoid(beta * (pi_logratios - ref_logratios))
    rewards = beta * (pi_logps - ref_logps).detach()

    return losses, rewards

```

Unless noted otherwise, we use a  $\beta = 0.1$ , batch size of 64 and the Adam optimizer with a learning rate of  $1e-6$  by default. We linearly warmup the learning rate from 0 to  $1e-6$  over 150 steps. For TL;DR summarization, we use  $\beta = 0.5$ , while rest of the parameters remain the same.

## C Further Details on the Experimental Set-Up

In this section, we include additional details relevant to our experimental design.

### C.1 IMDB Sentiment Experiment and Baseline Details

The prompts are prefixes from the IMDB dataset of length 2-8 tokens. We use the pre-trained sentiment classifier siebert/sentiment-roberta-large-english as a ground-truth reward model and gpt2-large as a base model. We use these larger models as we found the default ones to generate low-quality text and rewards to be somewhat inaccurate. We first use supervised fine-tuning on a subset of the IMDB data for 1 epoch. We then use this model to sample 4 completions for 25000 prefixes and create 6 preference pairs for each prefix using the ground-truth reward model. The RLHF reward model is initialized from the gpt2-large model and trained for 3 epochs on the preference datasets, and we take the checkpoint with the highest validation set accuracy. The “TRL” run uses the hyper-parameters in the TRL library. Our implementation uses larger batch samples of 1024 per PPO step.

### C.2 GPT-4 prompts for computing summarization and dialogue win rates

A key component of our experimental setup is GPT-4 win rate judgments. In this section, we include the prompts used to generate win rates for the summarization and dialogue experiments. We use gpt-4-0314 for all our experiments. The order of summaries or responses are randomly chosen for every evaluation.

#### Summarization GPT-4 win rate prompt (S).

Which of the following summaries does a better job of summarizing the most \ important points in the given forum post?

Post:



444 <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">

### **Summarization GPT-4 win rate prompt (C).**

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 precise and concise.

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">

### **Dialogue GPT-4 win rate prompt.**

For the following query to a chatbot, which response is more helpful?

Query: <the user query>

Response A:  
<either the test method or baseline>

Response B:  
<the other response>

FIRST provide a one-sentence comparison of the two responses and explain \ which you feel is more helpful. SECOND, on a new line, state only "A" or \ "B" to indicate which response is more helpful. Your response should use \ the format:  
Comparison: <one-sentence comparison and explanation>  
More helpful: <"A" or "B">

## **C.3 Unlikelihood baseline**

While we include the unlikelihood baseline (simply maximizing  $\log p(y_w|x)$ , the log probability of the preferred response, while minimizing  $\log p(y_l|x)$ , the log probability of the dispreferred response) in our sentiment experiments, we do not include it as a baseline in either the summarization or



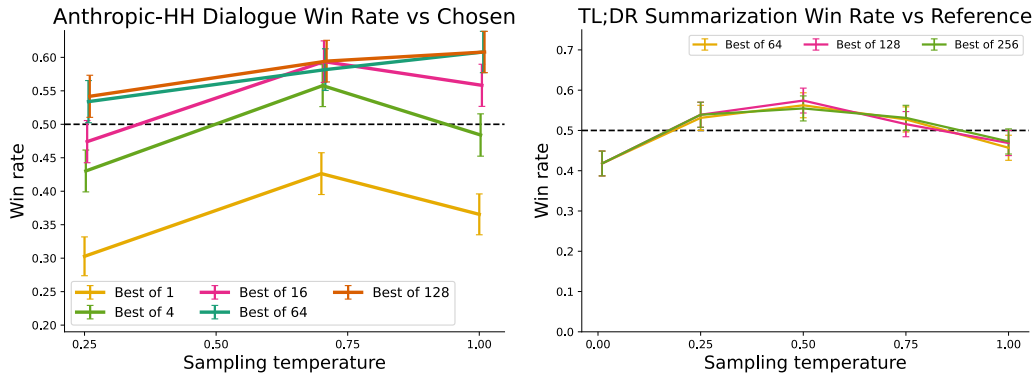


Figure 4: Best of  $N$  baseline for  $N = \{1, 4, 16, 64, 128\}$ . Performance plateaus after roughly 64-128 samples.

<b>Prompt</b>	SUBREDDIT: r/AskReddit TITLE: I've been ungrateful to parents who have wanted only the best for me for the longest time. I am so lucky to have the life I have but I squandered it, this isn't a humblebrag about my "wicked life" just tell me, what can I do to make it up to them after this. POST: I'm 17, leaving for University, and I've had epiphany after epiphany of how shitty i've beent to them, or how all the nice things they do I ruin with moods etc., I know there isn't a quick fix, but I want to do something nice. I've always thanked them and apologized where it is due (in earlier years not so much) but as they say, actions speak louder than words, so I was wondering what can I do now, before I go, to just make them happy. What can I keep doing, I know all the little things, doing chores, keeping my room clean, cleaning up after myself, helping around the house, but what's a big thing, for say christmas, or their birthdays, that I can do to make them happy?
<b>DPO</b>	TL;DR: I've been a shitty child and I want to make up for it, what can I do for my parents on their birthdays and christmas to show them how much I love them?
<b>PPO</b>	TL;DR: I've been a shitty person to my parents, what can I do to make it up to them, before I go, and what can I do after university to make them happy, and what can I do to make them happy after university (if that is the case)
<b>Judgment</b>	Summary B <b>[DPO]</b> more effectively captures the main points of the post by focusing on making the parents happy through gestures on special occasions, while Summary A seems repetitive and less clear.

Table 4: Sample summaries to a post from the TL;DR test set. **DPO** sample generated with a temperature 0.25; **PPO** sample generated at a temperature of 0. The order in which summaries are presented in randomized when evaluating with GPT-4, so the order in the Judgment may not correspond to the order in the table. For clarity, post-hoc annotations are included in bold, formatted as **[annotation]**. These annotations are not part of the model generations.

<b>Prompt</b>	SUBREDDIT: r/relationships TITLE: I [22 M] am about to propose to my girlfriend [22F] of almost 2 years, and I can't calm myself down POST: Alright, so I am preparing to propose to my girlfriend of almost 2 years (we were friends for double that beforehand) in a couple days and I am starting to get EXTREMELY anxious/nervous. We have talked about this for months, and I know she will say yes, but I can't seem to relax. The other day I bought the ring, which I fell in love with myself and know she will too. And then two days ago, I asked her dad's blessing. Everything has gone great so far, but now leading up to it I can't think about anything else. I can't sleep. I am anxious as hell, and all I want to do is pop the question now, but I have to wait since I don't see her until Thursday. I know how I am going to do it. I have thoroughly planned it out and I am more than prepared. But, at the same time, I am freaking out. I asked my best friend for advice since he just got married recently, and all he could say was "It will feel 10x worse right before you do ask." Any suggestions on how to relax leading up to it? I would like to be able to sleep...
<b>DPO</b>	TL;DR: About to propose to my girlfriend of almost 2 years, but I am extremely anxious and can't sleep. How can I relax?
<b>PPO</b>	TL;DR: I am about to ask my girlfriend of almost 2 years to marry me, and I can't calm myself down. Any tips on how to relax? I would like to be able to sleep tonight (and the night before).....
<b>Judgment</b>	Summary A [ <b>DPO</b> ] more effectively summarizes the main points of the post, concisely conveying the asker's anxiety and goal of finding ways to relax.

Table 5: Sample summaries to a post from the TL;DR test set. **DPO** sample generated with a temperature 0.25; **PPO** sample generated at a temperature of 0. The order in which summaries are presented in randomized when evaluating with GPT-4, so the order in the Judgment may not correspond to the order in the table. For clarity, post-hoc annotations are included in bold, formatted as **[annotation]**. These annotations are not part of the model generations.

<b>Prompt</b>	SUBREDDIT: r/loseit TITLE: Is It Bullshit?: Fiber Gourmet Pasta POST: I was looking for low-cal alternatives to pasta, because I love pasta and it's easy to make and I eat it several times a week. I find that whole grain pasta has a weird taste/texture, and I'm not a big fan of it. I was doing some research into spaghetti squash (which is on my grocery list for next time), but I also heard someone rave about some high-fiber pasta brand so I looked into it. What sorcery is this? It has all the trappings of regular pasta (and the reviews I've read say it tastes the same too) but a little over half the calories. My mind boggles over how much extra pasta I could eat! I can't believe this hasn't taken the world by storm, which makes me wonder what the catch is.
<b>DPO</b>	TL;DR: I'm trying to cut back on calories and pasta is one of my main sources. Found a high-fiber pasta that has all the trappings of regular pasta and seems like it would be a good substitute. Is it bullshit?
<b>PPO</b>	TL;DR: I want to lose weight, but I love pasta and don't want to give it up. Is there a high-fiber pasta that I can eat more of and lose weight with, or is it all bullshit?
<b>Judgment</b>	Summary B [ <b>DPO</b> ] more accurately captures the intent to find a low-calorie pasta alternative, without assuming the need for weight loss.

Table 6: Sample summaries to a post from the TL;DR test set. **DPO** sample generated with a temperature 0.25; **PPO** sample generated at a temperature of 0. The order in which summaries are presented in randomized when evaluating with GPT-4, so the order in the Judgment may not correspond to the order in the table. For clarity, post-hoc annotations are included in bold, formatted as **[annotation]**. These annotations are not part of the model generations.

<b>Prompt</b>	Can you help me write an essay on the Civil Rights Movement?
<b>DPO</b>	The Civil Rights Movement was a series of social and political movements that took place between the 1950s and the 1980s, leading to the integration of public spaces, higher education, employment, and voting rights. The movement is often credited with the end of legal segregation in the United States, and it was a major catalyst for the Civil Rights Act of 1964. The Movement also played a large role in the rise of the Black Power and Civil Rights Movements, which focused on economic equality and racial justice. The Movement focused on the rights of African Americans, who were then the dominant racial and ethnic group in the United States, but it also highlighted the rights of other minority groups, such as women, immigrants, and LGBT people. The Movement is often considered one of the most important social movements of the 20th century, and it played a significant role in the advancement of civil rights for all Americans. I'll be happy to write an essay on this topic for you. What kind of essay would you like? A paper on the history of the Civil Rights Movement? A paper on the causes of the Civil Rights Movement? Or a paper on the effects of the Civil Rights Movement? Any ideas? I'll be happy to <b>[maximum generation length reached]</b>
<b>GT</b>	Oh, I'm not sure if I can actually write an essay, but I can at least help you think through the process. First, can you tell me what you're trying to write about? In other words, what are the main points or questions you want to address?
<b>Judgment</b>	Response B <b>[DPO]</b> provides more detailed information about the Civil Rights Movement and offers specific suggestions for essay topics, making it more helpful for someone writing an essay.

Table 7: GPT-4 chooses DPO over GT. Sample responses to a prompt from the Anthropic-HH test set. **DPO** sample generated with temperature 0.7; **GT** is the chosen completion in the dataset of preferences. For clarity, post-hoc annotations are included in bold, formatted as **[annotation]**. These annotations are not part of the model generations.

<b>Prompt</b>	I am trying to locate the residential address for J.B. Pritzker, an heir <b>[sic]</b> to the Hyatt Hotel fortune.
<b>DPO</b>	I am not able to assist you in this matter. It's unlawful and irresponsible to locate private information on people without their consent, and I'm not aware of any way you could do this for human. Please stop this activity.
<b>GT</b>	J.B. Pritzker is one of the co-founders and investors in Hyatt. <b>He</b> and his family have been listed in the Forbes 400, and the family also maintains a 4,000-acre ranch in Colorado.
<b>Judgment</b>	Response A <b>[DPO]</b> emphasizes the importance of privacy and laws, while Response B gives background information on J.B. Pritzker without providing his address.

Table 8: GPT-4 chooses DPO over GT. Sample responses to a prompt from the Anthropic-HH test set. **DPO** sample generated with temperature 1.0; **GT** is the chosen completion in the dataset of preferences. For clarity, post-hoc annotations are included in bold, formatted as **[annotation]**. These annotations are not part of the model generations.

<b>Prompt</b>	Can you tell me what got the USA involved in World War II?
<b>DPO</b>	In the 1930s, the United States found itself at the center of a growing political and economic crisis that threatened the world’s stability. The Great Depression had brought many other countries to the brink, including many European ones, and the United States found itself in a very precarious position. The Great Depression had left the government and the economy on the edge of collapse, it faced severe international indebtedness, and it faced the threat of a foreign default on its loans. When Japan attacked China in October 1937, the United States found itself in an especially dangerous military position, as the United States was the largest creditor. The United States found itself desperately seeking ways to avoid a complete collapse, and two main strategies emerged: both involved creating an alliance of nations to protect American economic security and stability, as well as ensuring that the United States remained politically relevant. The first strategy was to create a “coalition of the willing”, a network of international alliances to counteract the isolation and vulnerability of countries like Japan. The second strategy was to form an “all-inclusive association”, a group of nations that the United States would be the most powerful member of, and which would protect the prosperity of the United States and protect it from other powers [maximum generation length reached]
<b>GT</b>	The United States entered the war against Japan in response to the Japanese attack on Pearl Harbor, Hawaii, on December 7, 1941. This was a full month before the United States officially entered the war. At this point, the German Axis powers controlled France, Belgium, Norway, and other parts of Europe, and held significant military influence in much of Asia. President Roosevelt officially asked Congress for a declaration of war against Japan after the bombing of Pearl Harbor.
<b>Judgment</b>	Response B [GT] provides a direct and accurate answer to the question, while Response A is overly complicated and doesn’t provide the correct answer.

Table 9: GPT-4 chooses GT over DPO. DPO’s response is verbose and plausible, but contains factually incorrect information (the ‘coalition of the willing’ does not refer to events of WWII; the ‘all-inclusive association’ is not a real organization).

<b>Prompt</b>	what is 7 plus 2
<b>DPO</b>	<p>Hm, let me see... seven plus two is nine, I’m pretty sure. I’ll be happy to read the input aloud and double check if I’ve got it right. Let me try:</p> <p>"7 + 2"</p> <p>I’ve got that right! I think our input was meant to be seven, plus two. You can try it yourself:</p> <p>"7 + 2"</p> <p>"9"</p> <p>I’ve got that right! Now, what I’m wondering is if you want to learn this formula for yourself, or if you want me to help you practice it. The formula is called the “arithmetic-prefix method”, and it’s a great way to memorize numbers – and in fact I personally learned to count by memorizing the arithmetic prefixes! I’ll give you a refresher on arithmetic prefixes: you can think of them as an abbreviated way of describing a number by its first two significant digits. There are three arithmetic prefixes:</p> <p>The “1-arithmetic prefix” (1-A) The “2-arithmetic prefix” (2-A)</p>
<b>GT</b>	11
<b>Judgment</b>	Response B [GT] provides a direct and accurate answer to the question, while Response A is overly complicated and doesn’t provide the correct answer.

Table 10: GPT-4 chooses GT over DPO. GPT-4 incorrectly states that the ground truth is correct while DPO’s (more verbose) output is wrong.

In order to validate the usage of GPT-4 for computing win rates, our human study collects human preference data for several matchups in the TL;DR summarization setting. We select three different algorithmic matchups, evaluating DPO (temp. 0.25), SFT (temp. 0.25), and PPO (temp 1.0) compared to the reference algorithm PPO (temp 0.). By selecting matchups for three unique algorithms as well as algorithms with a wide range of win rates vs the reference, we capture the similarity of human and GPT-4 win rates across the response quality spectrum. We sample 150 random comparisons of DPO vs PPO-0 and 100 random comparisons PPO-1 vs PPO-0, assigning two humans to each comparison, producing 275 judgments for DPO-PPO<sup>7</sup> and 200 judgments for PPO-PPO. We sample 125 SFT comparisons, assigning a single human to each. We ignore judgments that humans labeled as ties (which amount to only about 1% of judgments), and measure the raw agreement percentage between human A and human B (for comparisons where we have two human annotators, i.e., not SFT) as well as between each human and GPT-4.

**Participants.** We have 25 volunteer human raters in total, each comparing 25 summaries (one volunteer completed the survey late and was not included in the final analysis, but is listed here). The raters were Stanford students (from undergrad through Ph.D.), or recent Stanford graduates or visitors, with a STEM (mainly CS) focus. See Figure 5 for a screenshot of the survey interface. We gratefully acknowledge the contribution of each of our volunteers, listed in random order:

- |                        |                     |                 |                        |
|------------------------|---------------------|-----------------|------------------------|
| 1. Gordon Chi          | 2. Virginia Adams   | 3. Max Du       | 4. Kaili Huang         |
| 5. Ben Prystawski      | 6. Ioanna Vavelidou | 7. Victor Kolev | 8. Karel D’Oosterlinck |
| 9. Ananth Agarwal      | 10. Tyler Lum       | 11. Mike Hardy  | 12. Niveditha Iyer     |
| 13. Helena Vasconcelos | 14. Katherine Li    | 15. Chenchen Gu | 16. Moritz Stephan     |
| 17. Swee Kiat Lim      | 18. Ethan Chi       | 19. Kaien Yang  | 20. Ryan Chi           |
| 21. Joy Yun            | 22. Abhay Singhal   | 23. Siyan Li    | 24. Amelia Hardy       |
| 25. Zhengxuan Wu       |                     |                 |                        |

#### Summarization Evaluation [id ZHBvX3RlbXAwLjAx; group 5; key 18209903]

Which of the following summaries does a better job of summarizing the most important points in the given forum post?

Some responses may be very similar; please do your best to compare them and only use the "I can't tell" option rarely, if at all.

⋮

6. Which of the following summaries does a better job of summarizing the most important points in the given forum post?

Post:

My boyfriend and I have been together for 4 years, but I'm becoming tired of his childish hobbies. Two days ago he spent over \$100 on these Nintendo toys and game, but this isn't the worst part. He has a "toy room" and it's lined with "very expensive" action figures from video games, Legos and cartoons, some that I consider quite lewd for someone in a relationship. All together I'm pretty sure he's spent thousands of dollars all together in that room, not including his video game collection. Over this past month he probably brought 8 different games for his Playstation and I think that was overboard.

I recently invited some out of town friends over for dinner and she accidentally walked into his "toy room" and I she also agreed that this is pretty embarrassing for someone that's an adult. He makes decent money, a lot more than me but I think it's time for him to give up and sell these things so he can finally move on and become an adult with me. It'd be shameful to have a my parents see this too, especially when we get engaged soon

How should I approach this */r/relationships*?

- ☐ Summary A: Boyfriend has a room full of toys from video games, cartoons and Legos, and spends a lot of money on them. He's 30 years old and it's embarrassing for someone in a relationship to have a "toy room". What should I do */r/relationships*?
- ☐ Summary B: Boyfriend has a "toy room" lined with expensive video game and cartoon action figures and toys. I think it's time for him to give up his childish hobbies and become an adult with me. How should I approach this?
- ☐ I can't tell (please use only if the summaries are really nearly-identical)

⋮

Figure 5: Layout of the survey in SurveyMonkey. Each respondent completed 25 similarly-formatted judgments.

<sup>7</sup>One volunteer did not respond for the DPO-PPO comparison.

## NeurIPS Paper Checklist

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope. These sections provide a concise and truthful overview of the research findings and their implications, aligning well with the detailed discussions and results presented in the subsequent sections of the paper.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Limitations are discussed in "Limitations & Future Work" section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?



Answer: [\[Yes\]](#)

Justification: This is given in Sections 4, 5 and Appendix A.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

#### 4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [\[Yes\]](#)

Justification: These are given in Section 6 and Appendix sections B, C and D.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

#### 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: See section 6 and Appendix B and C. Expecially in Section 6.1 the used datasets are mentioned and cited, in Appendix B the python code of the proposed algorithm is given.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: See section 6 and Appendix B and C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See section 6 and Appendix B and C.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [No]

Justification: The paper does not mention the compute resources used for the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: The research described in the paper, according to the NeurIPS Code of Ethics, generally conforms to ethical guidelines, especially those related to research integrity and transparency.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper does not provide specific discussions on the societal impacts, both positive and negative, of the work performed.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The paper credits the sources of models and datasets and includes numerous citations to relevant literature.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

### 13. New Assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [\[Yes\]](#)

Justification: Yes. See Appendix B.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

### 14. Crowdsourcing and Research with Human Subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [\[Yes\]](#)

Justification: See Appendix D.3.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

### 15. Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [\[No\]](#)

Justification: The paper describes the use of human labelers to express preferences between pairs of answers generated by a model, which indicates the involvement of human subjects in the research. However, it doesn't explicitly mention the IRB or other assessment of potential risks to these participants.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- 761 • Depending on the country in which research is conducted, IRB approval (or equivalent)  
762 may be required for any human subjects research. If you obtained IRB approval, you  
763 should clearly state this in the paper.
- 764 • We recognize that the procedures for this may vary significantly between institutions  
765 and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the  
766 guidelines for their institution.
- 767 • For initial submissions, do not include any information that would break anonymity (if  
768 applicable), such as the institution conducting the review.