Mind the (Preference) Gap: Mitigating Reward-Preference Misalignment with Reward Distillation

Anonymous ACL submission

Abstract

Traditional RLHF-based LLM alignment methods and direct alignment counterparts like DPO assume a Bradley-Terry model of pairwise preferences. This assumption is challenged by non-deterministic or noisy preference labels, such as scoring of two candidate outputs with low confidence or low reward difference. This paper introduces **DRDO (Direct Reward Distillation and policy-Optimization)**, which simultaneously models rewards *and* preferences to avoid such degeneracy. DRDO directly mimics rewards assigned by an oracle while learning human preferences with a novel preference likelihood formulation, while being fully *offline*. Results on Ultrafeedback, TL;DR, and AlpacaEval 2.0 show that DRDO-trained policies surpass methods such as DPO and e-DPO in terms of expected rewards and are more robust to noisy preference signals and out-of-distribution (OOD) settings.

1 Introduction

Robust modeling of human preferences is essential for producing usable large language models (LLMs). While popular alignment approaches implicitly assume that pairs of preferred and dispreferred samples in preference data have an unambiguous winner, this does not reflect the reality of actual data, where human-annotated preferences may have low labeler confidence or the preference strength itself might be weak. As such, reward functions estimated on such data lead to a "preference gap" between the reward model and the true preference distribution, and concomitant policy degeneracy and underfitting. We address these challenges with the following novel contributions:

1) We introduce ***Direct Reward Distillation and policy-Optimization (DRDO)***, a novel efficient, non-ensemble, reference-free method for preference optimization that explicitly distills rewards into the policy model (Fig. 1);
2) We provide a theoretical and practical grounding of problems with alignment methods that assume the Bradley-Terry model, demonstrating why they are challenged by nuanced or "non-deterministic" preference pairs, and show how DRDO avoids similar limitations;
3) Through experiments on Ultrafeedback, TL;DR, and AlpacaEval 2.0, we show that DRDO is better able to fit to nuanced or ambiguous preferences without sacrificing performance on clear preferences or large-scale data.

2 Background and Related Work

Offline and Online Preference Optimization Reinforcement Learning from Human Feedback (RLHF) aims to harmonize LLMs with human preferences and values (Christiano et al., 2017). Conventional RLHF typically consists of three phases: supervised fine-tuning, reward model training, and policy optimization. Proximal Policy Optimization (PPO; Schulman et al. (2017a)) is a widely used algorithm in the third phase of RLHF. RLHF has been extensively applied across various domains, including mitigating toxicity (Korbak et al., 2023; Amini et al., 2024), addressing safety-concerns (Dai et al., 2023), enhancing helpfulness (Tian et al., 2024), web search and navigation (Nakano et al., 2021), and enhancing reasoning in models (Havrilla et al., 2024). Casper et al. (2023) identified challenges and problems throughout the entire RLHF pipeline, from gathering preference data to model training to biased results such as verbose outputs (Dubois et al., 2024; Singhal et al., 2023; Wang et al., 2023).

Given the intricacy and complexity of online preference optimization (Zheng et al., 2023), research has proliferated into more efficient and simpler offline algorithms. Direct Preference Optimization (DPO; Rafailov et al. (2024b)) is a notable example, which demonstrates that the same KL-constrained objective as RLHF can be optimized

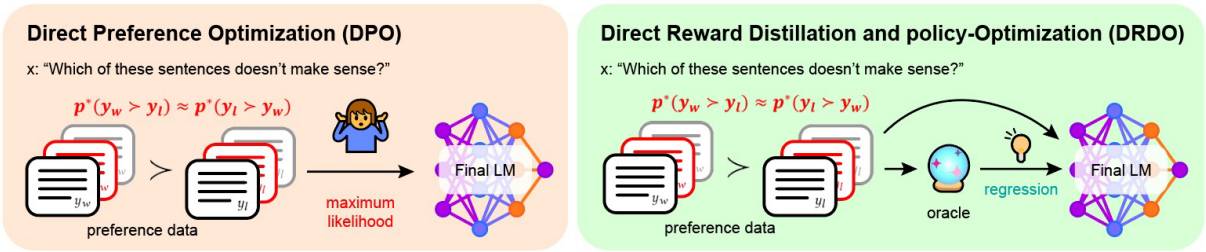


Figure 1: Unlike popular supervised preference alignment algorithms like Direct Preference Optimization (DPO; Rafailov et al. (2024b)) that learns rewards implicitly, **DRDO directly optimizes for explicit rewards from an Oracle while simultaneously learning diverse kinds of preference signals during alignment.** Optimized with a simple regression loss based on difference of rewards assigned by the Oracle and the introduction of a focal-log-unlikelihood component (see Sec. 4), DRDO bridges the gap between the preference distribution estimated from the data and the true preference distribution p^*. Additionally, DRDO does not require an additional reference model during training and can leverage reward signals even when preference labels are not directly accessible.

without explicitly learning a reward function. The problem is reformulated as a maximum likelihood estimation (MLE) over the distribution π_θ.

With the growing focus on offline alignment, recent work has proposed various solutions to *preference underfitting* in this setting. These include learning from a confidence set of rewards (Fisch et al., 2024), adopting a general preference model (Munos et al., 2023; Azar et al., 2024), or applying a straightforward regularization of the original DPO objective (Pal et al., 2024). All these approaches rely on a reference model, not only for expressing the optimal policy as the analytical solution to the minimum relative entropy problem (Ziebart et al., 2008; Peng et al., 2019) but also to stabilize training by constraining the policy distribution close to the reference model. While theoretically sound, Munos et al. (2023) suggests this constraint can make policies prioritize high rewards over truly learning human preferences.

As such, certain works avoid this constraint and focus on lightweight "reference model-free" solutions using a careful construction of DPO-inspired Bradley-Terry (BT)-based implicit rewards—such as logit-based (Hong et al., 2024), length-normalized (Meng et al., 2024), or unnormalized implicit rewards (Xu et al., 2024)—or relative preference label strength (Nath et al., 2025). Similarly, efforts to learn policies from general preference models aim to reduce dependence on the sampling distribution, a key limitation of Bradley-Terry models. Munos et al. (2023), Rosset et al. (2024), and Calandriello et al. (2024) use game-theoretic "online" approaches for robustness, while Choi et al. (2024) propose a Chain-of-Thought (CoT) policy with pairwise conditioning to improve alignment. However, these methods often suffer from sample inefficiency due to iterative sampling from approximate geometric mixtures of policies (Rosset et al., 2024), and while they mitigate dependence on the sampling distribution, they do not model non-deterministic preferences in policy training.

Several different classes of preference optimization objective have been explored. Ranking objectives extend comparisons beyond pairs (Dong et al., 2023; Liu et al., 2024; Song et al., 2024; Yuan et al., 2023), while Hong et al. (2024) and Xu et al. (2023) propose reference-free methods. Bansal et al. (2024) optimize instructions and responses jointly, improving on DPO, and Zheng et al. (2024) enhances post-training extrapolation between SFT and aligned models.

In contrast to the aforementioned approaches, DRDO adaptively learns such preferences while decoupling its learning from a reference model-based KL-divergence constraint, and derives its objective from knowledge distillation (Hinton, 2015) and focal-loss literature (Lin et al., 2018; Yi et al., 2020) with a novel contrastive log-unlikelihood objective that learns preferences adaptively.

3 Motivation for DRDO

Problem Formulation Let $\mathcal{D}_{\text{pref}} = \{(x, y_w, y_l)\}_{i=1}^{N}$ be an offline dataset of pairwise preferences with sufficient coverage, where y_w and y_l are the respective winning (preferred) and losing (less preferred) completions given a context x. In contrast to *deterministic* preferences, which are defined as those where $p^* \in \{0, 1\}$ (Fisch et al., 2024), let $\mathcal{D}_{\text{nd}} \subset \mathcal{D}_{\text{pref}}$ denote the subset of *non-deterministic* preference pairs where $P(y_w \succ y_l|x) \approx \frac{1}{2}$. These cannot be perfectly captured by Bradley-Terry models but are prevalent

in popular preference alignment datasets.[1] Assume three possible completions $y_1, y_2, y_3 \in \mathcal{Y}$, the space of all possible completions. Let $r^*(x, y) \in \mathbb{R}$ be an underlying true reward function that is deterministic and finite for all completions, $\pi_{\theta^*}(y|x)$ be the learned policy, and $\pi_{\text{ref}}(y|x)$ be the reference policy with $\text{supp}(\pi_{\text{ref}}) = \mathcal{Y}$. Given $\text{supp}(\rho) = \text{supp}(\mu) \times \mathcal{Y} \times \mathcal{Y}$ where ρ is the data distribution and μ is the context distribution, the challenge is leaning a policy to effectively handle both deterministic and non-deterministic preferences in an offline fashion.

Misalignment of Rewards and Preferences Although preferences p^* can be implicit in rewards, refining LLMs based on rewards alone does not imply that they learn preferences optimally. We call this the **misalignment problem**, where the two objectives are fundamentally divergent (Munos et al., 2023). This issue is particularly significant in aligning high-dimensional policies like LLMs—where the training data often contains non-deterministic samples or weak learning signals. Such uncertainty often appears in equal or near-equal rewards or ambiguous expert judgments (Stiennon et al., 2020). In other words, traditional reward modeling assumes that maximizing reward leads to optimal behavior, but in preference-based learning, the best policy can be different from the highest-reward policy. Where the *reward-optimal* policy, or the policy that maximizes Elo score under the BT assumption, fundamentally diverges from the *preference optimal* policy, or the policy that maximizes the probability of selecting the ground truth winning response—this creates a "preference gap" and the misalignment problem occurs. However, since LLMs are overparameterized with many near-optimal solutions (Rafailov et al., 2024a) and are expected to generalize across diverse and often uncertain preferences (Zeng et al., 2024), it is essential to understand this divergence or preference gap, as it can be especially sensitive to the existence of non-deterministic preferences in training.[2]

This insight motivates the core of DRDO: to effectively learn from non-deterministic preferences while maintaining performance across general preference data, we bound this preference gap to possible sources of errors scaled by the offline data coverage, while simultaneously learning both high-quality distilled rewards and preferences in an efficient offline manner. Mathematically, we illustrate this sensitivity to non-deterministic pairs in the BT model in Lemma 1 and then provide an upper bound to this preference gap in Lemma 2.

Lemma 1 (Sensitivity of Preference Gap to Non-Deterministic Preferences)**.** *The preference gap $\delta_{\mathcal{P}}$ between reward-optimal (π^*_R) and preference-optimal ($\pi^*_{\mathcal{P}}$) policies can be highly sensitive to the presence of non-deterministic preference pairs (where $\mathcal{P}(y \succ y') = \mathcal{P}(y' \succ y) = 1/2$). Such non-determinism can lead to a substantial increase in $\delta_{\mathcal{P}}$ even when the reward gap δ_R (based on a fixed reward function R) remains unchanged.*

(See proof in Appendix A.1.) The core insight here is that common approaches like DPO that assume a BT preference model are more likely to be sensitive to this preference gap, since they assume that true preferences are learnable from a mapping of preferences onto differences in scalar implicit rewards. Prior work (Azar et al., 2024; Fisch et al., 2024) highlights DPO's practical tendency to underfit the true preference distribution because of unboundedness of its implicit rewards. For example, for two completions y and y' with a non-deterministic preference relation, the DPO estimate of $p^*(y \succ y')$ using $\sigma\big(\beta \log \frac{\pi_\theta(y|x)\pi_{\text{ref}}(y'|x)}{\pi_\theta(y'|x)\pi_{\text{ref}}(y|x)}\big)$ leads π_θ to assign equal rewards to both, meaning it learns that $r^*(x, y) \approx r^*(x, y')$ for any $(x, y, y') \in \mathcal{D}_{\text{nd}}$. This implies that the reward model fails to strongly differentiate between completions in these cases, even though one is explicitly "chosen" in the true preference data and the other is "rejected," thus leading to the preference gap, and the motivation of DRDO to address these issues.

4 Direct Reward Distillation and policy-Optimization (DRDO)

DRDO alignment involves two steps. First, we fit an Oracle model $\mathcal{O}$ to the annotated preference data. Second, we use $\mathcal{O}$ as a teacher to align the policy model (student) with a knowledge-distillation-based multi-task loss (Hinton, 2015; Gou et al., 2021) that regresses the student's rewards onto those assigned by $\mathcal{O}$. The student model simultaneously draws additional supervision from binary preference labels for efficient use of finite data.

[1] For instance, an analysis of the popular RLAIF-inspired scalable alignment benchmark Ultrafeedback (Cui et al., 2024) reveals that it contains approximately **17%** non-deterministic samples as defined above (Nath et al., 2025).

[2] Non-deterministic preferences differ from noisy (flipped) labels (Chowdhury et al., 2024; Wang et al., 2024a), which are often handled with label smoothing, data pruning, or noise-aware training.

Training the Oracle Reward Model We use Yang et al. (2024)'s strategy to optimize a quality and generalizable $\mathcal{O}$ while retaining its language generation abilities. Regularizing the shared hidden states with a language generation loss in addition to the traditional RLHF-based reward modeling improves generalization to out-of-distribution preferences. For this we initialize a separate linear reward head (parametrized by ϕ')[3] on top of the base LM (parametrized by ϕ), which adds only 0.003% more parameters compared to the LM's language modeling head. This also helps minimize reward hacking (Kumar et al., 2022; Eisenstein et al., 2024), especially in offline settings. As such, $\mathcal{O}$ is optimized to minimize the following objective:

$$\mathcal{L}_{\mathcal{O}}(r_\phi, \mathcal{D}_{\text{pref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{pref}}}\Big[(1-\alpha)(\log\sigma(r_{\phi'}(x,y_w) - r_{\phi'}(x,y_l))) + \alpha\log(r_\phi(y_w))\Big] \quad (1)$$

where α is the strength of language-generation regularization on the winning response log-likelihoods assigned by $\mathcal{O}$ (denoted $\log(r_\phi(y_w))$ and ϕ is the parameters of $\mathcal{O}$ being estimated.

Simultaneous Student Policy Alignment to Rewards and Preferences A converged Oracle $\mathcal{O}$ can plausibly estimate true pointwise reward differences $r^*(x,y_1) - r^*(x,y_2)$ for any unlabeled sample (x,y_1,y_2), without needing explicit access to preference labels. Next, the student model π_θ is optimized to match its own reward predictions $\hat{r}_1$ and $\hat{r}_2$ to $\mathcal{O}$'s reward differences, aligning closely with $\mathcal{O}$'s behavior, while $\mathcal{O}$ itself does not get updated. The student model's reward estimates are computed with a linear reward head on top of the base LM, similar to Oracle training. This optimization uses a knowledge-distillation loss ($\mathcal{L}_{\text{kd}}$) that combines both a supervised ℓ^2-norm term and a novel focal-softened (Lin et al., 2018; Welleck et al., 2020) log odds-unlikelihood component:

$$\mathcal{L}_{\text{kd}}(r^*, \pi_\theta) = \mathbb{E}_{(x,y_1,y_2)\sim\mathcal{D}_{\text{pref}}}\Bigg[\underbrace{\left(r^*(x,y_1) - r^*(x,y_2) - (\hat{r}_1 - \hat{r}_2)\right)^2}_{\text{Reward Difference}} - \underbrace{\alpha(1-p_w)^\gamma \log\left(\frac{\pi_\theta(y_w \mid x)}{1-\pi_\theta(y_l \mid x)}\right)}_{\text{Contrastive Log-"unlikelihood"}}\Bigg], \quad (2)$$

[3]For simplicity of notation, on the LHS of Eq. 1 we subsume parameters ϕ' into ϕ.

Algorithm 1 DRDO: Direct Reward Distillation and policy-Optimization

Input: Preference dataset $\mathcal{D}_{\text{pref}} = \{(x, y_w, y_l)\}$; initialized policy + reward head $\pi_{\theta,\theta'} \leftarrow \text{SFT}(\theta) \oplus r_{\theta'}$
Output: Optimized parameters θ for aligned policy π_θ
Train oracle reward model r_ϕ (Eq. 1)
for $t = 1$ to T **do**
  for each $(x, y_w, y_l) \in \mathcal{D}_{\text{pref}}$ **do**
    Compute oracle rewards: $r_1^* = r_\phi(x, y_w)$, $r_2^* = r_\phi(x, y_l)$
    Compute student rewards: $\hat{r}_1 = r_{\theta'}(x, y_w)$, $\hat{r}_2 = r_{\theta'}(x, y_l)$
    Compute distillation loss $\mathcal{L}_{\text{kd}}$ (Eq. 2)
    Update $\pi_{\theta,\theta'}$ using gradient step on $\mathcal{L}_{\text{kd}}$
  end for
end for
return Final policy π_θ

where $p_w = \sigma(z_w - z_l)$ and quantifies the student policy's confidence in correctly assigning the preference from $z_w = \log\pi_\theta(y_w \mid x)$ and $z_l = \log\pi_\theta(y_l \mid x)$, or the log-probabilities of the winning and losing responses, respectively. Hyperparameters γ and α regulate the strength of the modulating term and weighting factor respectively.[4] As shown in Eq. 2, there is no shared parametrization between the policy and $\mathcal{O}$. $\mathcal{L}_{\mathcal{O}}$ is only a function of $\mathcal{O}$'s own parameters ϕ and the preference dataset, $\mathcal{D}_{\text{pref}}$. The Oracle parameters are *not* updated during policy training. Algorithm 1 shows a detailed breakdown of DRDO training.

Theoretical Analysis: DRDO

Lemma 2 (DRDO Preference Gap Bound)**.** *For a policy π_θ trained with DRDO using Oracle rewards r^*_{oracle}, the preference gap $\delta_{\mathcal{P}} = V_{\mathcal{P}}(\pi^*_{\mathcal{P}}) - V_{\mathcal{P}}(\pi_\theta)$ (w.r.t. true preferences $\mathcal{P}$) is bounded by $\delta_{\mathcal{P}} \le C\left(\sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,oracle}}\right) + \epsilon_{opt}$, where C is the concentrability coefficient, and $\epsilon_{\text{Oracle error}}$, $\epsilon_{r,oracle}$, and ϵ_{opt} are the error in the Oracle capturing true reward differences, the student's reward distillation error, and the student's policy optimization error, respectively. See Lemma 4 and its proof in Appendix B for a detailed explication.*

Lemma 2 decomposes the preference gap in DRDO into three potential sources of error: Oracle quality ($\epsilon_{\text{Oracle error}}$), reward distillation ($\epsilon_{r,\text{oracle}}$), and policy optimization (ϵ_{opt}). This decomposition is especially important when the true preferences are non-deterministic or deviate from the BT assumption, as learned BT-style Oracles may over-

[4]Note that we do not use α as it is traditionally used in focal loss for weighting class imbalances (Mukhoti et al., 2020). Instead, since the true preference distribution is unknown, we tune the empirical optimal value based on validation data and keep it fixed during training.

fit to their training distributions. In such cases, $\epsilon_{\text{Oracle error}}$ can increase if different sampling distributions induce different BT parameters. To address this, DRDO incorporates an SFT-based regularization when training $\mathcal{O}$ (Eq. 1)—improving out-of-distribution generalization and lowering Oracle error. Moreover, in offline settings, since the DRDO student is trained on the *same* data distribution as the Oracle, the reward target r^*_{oracle} (affecting $\epsilon_{r,\text{oracle}}$) and policy preference learning signal (affecting ϵ_{opt}) remain aligned. This design enables minimizing the sources of error and places an upper bound on the preference gap, for effective knowledge transfer and robust policy learning even in the presence of non-trivial amount of non-deterministic samples, as our experiments show.

Our above analysis of the problem of misaligned preferences and practical limitations in current methods directly motivate DRDO—we propose a simple offline alignment algorithm that optimizes for rewards and preferences *simultaneously*, thus circumventing the divergence issue. By distilling rewards from a converged oracle instead of implicit rewards, DRDO avoids policy collapse observed in popular algorithms like DPO and e-DPO under non-deterministic preference data, offering a principled alternative to conventional alignment approaches.

How does the DRDO gradient update affect preference learning? "Contrastive log-unlikelihood" in Eq. 2 is DRDO's preference component. One can immediately draw a comparison with DPO which uses a fixed β parameter. DRDO uses a modulating focal-softened term, $(1 - p_w)^\gamma$, where π_θ learns from both deterministic and non-deterministic preferences, effectively blending reward alignment with preference signals to guide optimization. Intuitively, unlike DPO's β that is constant for every training sample, this modulating term amplifies gradient updates when preference signals are weak ($p_w \approx 0.5$) and tempering updates when they are strong ($p_w \approx 1$), thus ensuring robust learning across varying preference scenarios. When π_θ assigns high confidence to the winning response ($p_w \approx 1$), the focal loss contribution diminishes (see Eq. 2), reflecting minimal penalty due to strong deterministic preference signals. However, for harder cases with non-deterministic true preference ($p^*(y \succ y') \approx (p^*(y' \succ y))$, the focal term $\alpha(1-p_w)^\gamma$ keeps DRDO gradient updates active and promotes learning even when preferences are ambiguous (Fig. 4 in Appendix B.5). This adaptive behavior ensures that DRDO maintains effective preference learning across varying preference strengths, where conventional methods like DPO struggle. See Lemma 5 and Lemma 6 for in-depth analysis. As shown in the gradient analysis in Appendix B.5 (Fig. 4) with empirical proof in Fig. 5 (bottom-right), this modulating term acts like DPO's gradient scaling term, in that it scales the DRDO gradient when the model incorrectly assigns preferences to easier samples.

5 Experimental Setup

Our experiments address two questions: **How robust is DRDO alignment to nuanced or diverse preferences, in OOD settings?** and **How well does DRDO achieve reward distillation with respect to model size?** We empirically investigate these questions on two tasks: **summarization** and single-turn **instruction following**. Our experiments, including choice of datasets and models for each task are designed to to validate our approach as robustly as possible, subject to research budget constraints (see Appendix C.2 for more). We compare our approach with competitive baselines such as DPO (Rafailov et al., 2024b) and e-DPO (Fisch et al., 2024) as well as on-policy methods like PPO (Schulman et al., 2017b), including the supervised finetuned (baseline) versions depending on the experiment. Some minor notes on the experimental setup described below can be found in Appendix C.

How robust is DRDO alignment to nuanced or diverse preferences, in OOD settings? We evaluate this using the **Reddit TL;DR summarization dataset** (Völske et al., 2017; Stiennon et al., 2020) and **CNN/Daily Mail Corpus** (Nallapati et al., 2016). For robust out-of-domain (OOD) evaluation, we train models on Reddit TL;DR and use CNN/Daily Mail articles for an OOD test distribution. We split the original training data ($\mathcal{D}_{all}$) based on human labeler confidence in their preference annotation (*h*igh-*c*onfidence vs. ℓow-*c*onfidence) and token edit distance between the preferred and the dispreferred responses (*h*igh-*e*dit distance vs. ℓow-*e*dit distance). This results in two splits: $\mathcal{D}_{hc,he}$ and $\mathcal{D}_{\ell c,\ell e}$. Each contains ∼10k training samples, where the former comprises samples from the upper 50th percentile of the confidence and edit distance scores, *mutatis mutandis*. $\mathcal{D}_{hc,he}$ and $\mathcal{D}_{\ell c,\ell e}$ represent "easy" (deterministic) and "hard" (non-deterministic) preference samples where combined labeler confidence and string-dissimilarity

act as proxy for the extreme ends of preference strengths/signals. See Appendix C.3 for more details. All baselines are initialized with Phi-3-Mini-4K-Instruct weights (Abdin et al., 2024) with supervised fine-tuning (SFT) on Reddit TL;DR human-written summaries.

How does model size affect reward distillation? We evaluate all baselines on the cleaned version of the **Ultrafeedback dataset** (Cui et al., 2024) This experiment is conducted on the OPT suite of models (Zhang et al., 2022), at 125M, 350M, 1.3B, and 2.7B parameter sizes. The student policy is trained with SFT on the chosen responses of the dataset, following Rafailov et al. (2024b). We exclude larger OPT models to focus on testing our distillation strategy with full-scale training, rather than parameter-efficient methods (PEFT; Houlsby et al. (2019); Hu et al. (2021)) to allow a full-fledged comparison considering all trainable parameters of the base model. For completeness and comparison across model families, we also include Phi-3-Mini-4K-Instruct following the same initialization.

Evaluation **CNN/Daily Mail Corpus** provides human-written reference summaries, so we use a high-capacity Judge to compute win-rates against baselines on 1,000 randomly-chosen samples. Following Rafailov et al. (2024b), we use GPT-4o to compare the conciseness and the quality of the DRDO summaries and baseline summaries, *while grounding its ratings to the human written summary*. See Appendix G for our prompt format. For instruction following on **Ultrafeedback**, we sample generations from DRDO and all baselines at various diversity-sampling temperatures and report win-rates on the Ultrafeedback test set against $\mathcal{O}$. Following Lambert et al. (2024), we consider a *win* to be when, for two generations y_1 and y_2, we get $r(x, y_1) > r(x, y_2)$, where $r(x, y_1)$ and $r(x, y_2)$ are the expected rewards (logits) from the policies being compared. Additionally, we evaluate DRDO and baselines (all trained on Ultrafeedback) on **AlpacaEval 2.0**, a 805-sample OOD benchmark with length-controlled comparisons judged by GPT-4 Turbo (Dubois et al., 2025). Hyperparameters and model configurations are given in Appendix E.

6 Results

Non-deterministic preferences Table 1 shows the win rates computed with GPT-4o as judge for 1,000 randomly selected prompts from the CNN/Daily Mail test corpus under OOD settings. We follow similar settings as Rafailov et al. (2024b) but further ground the prompt using human-written summaries as reference for GPT to conduct its evaluation (see Appendix G). For a fairer evaluation (Wang et al., 2024c; Goyal et al., 2023; Rafailov et al., 2024b), we swap positions of π_θ-generated summaries y in the prompt to eliminate any positional bias and evaluate the generated summaries on criteria like coherence, preciseness and conciseness, *with the human written summaries explicitly in the prompt to guide evaluation.*

CNN/Daily Mail (GPT-4o as Judge)	
DRDO vs. e-DPO	
$\mathcal{D}_{all}$	78.27%
$\mathcal{D}_{hc,he}$	80.92%
$\mathcal{D}_{\ell c,\ell e}$	79.01%
DRDO vs. DPO	
$\mathcal{D}_{all}$	80.78%
$\mathcal{D}_{hc,he}$	79.11%
$\mathcal{D}_{\ell c,\ell e}$	79.79%
Gold vs. DRDO	52.82%
Gold vs. e-DPO	54.38%
Gold vs. DPO	58.54%
AlpacaEval 2.0 (GPT-4 Turbo as Judge)	
SFT vs. GPT-4 Turbo	8.86%
DPO vs. GPT-4 Turbo	11.01%
e-DPO vs. GPT-4 Turbo	11.65%
PPO vs. GPT-4 Turbo	13.67%
DRDO vs. GPT-4 Turbo	15.95%

Table 1: Length-controlled win rates computed for 1,000 randomly selected evaluation samples from the **CNN/Daily Mail Corpus**, and for OOD evaluation on **AlpacaEval 2.0** ($N = 805$) using the Phi-3 model. $\mathcal{D}_{all}$, $\mathcal{D}_{hc,he}$, and $\mathcal{D}_{\ell c,\ell e}$ represent TL;DR training splits. "Gold" refers to human-written reference summaries used in prompts to ground the win rate computations. Both e-DPO and PPO use the same oracle reward as DRDO. See Table 6 for AlpacaEval dataset-wise breakdown.

For all policies π_θ trained on all three splits of the training data—$\mathcal{D}_{all}$, $\mathcal{D}_{hc,he}$, and $\mathcal{D}_{\ell c,\ell e}$—we compute the win-rates of DRDO vs. e-DPO and DPO to evaluate how each method performs at various levels of preference types. Across all settings, DRDO policies significantly outperform the two baselines. For instance, DRDO's average win rates are almost 79.4% and 79.9% against e-DPO and DPO, respectively. As we hypothesized, DRDO-aligned π_θ can learn *all* preferences, deterministic and non-deterministic, effectively, and is superior at modeling $\mathcal{D}_{\ell c,\ell e}$ the subset containing more non-deterministic preference samples, without sacrificing performance on the deterministic subset ($\mathcal{D}_{hc,he}$) and overall ($\mathcal{D}_{all}$). This suggests that DRDO is more robust to OOD-settings at various levels of difficulty in learning human preferences.

Reward distillation Fig. 2 shows results from our evaluation of DRDO's reward model distillation framework compared to DPO and e-DPO as well as baseline SFT-trained policies on the Ultrafeedback evaluation data, when compared *across various model parameter sizes* and at varying levels of temperature sampling. We sample π_θ-generated responses to instruction-prompts in the test set using top-p (nucleus) sampling (Holtzman et al., 2019) of 0.8 at various temperatures $\in \{0.2, 0.5, 0.7, 0.9\}$.

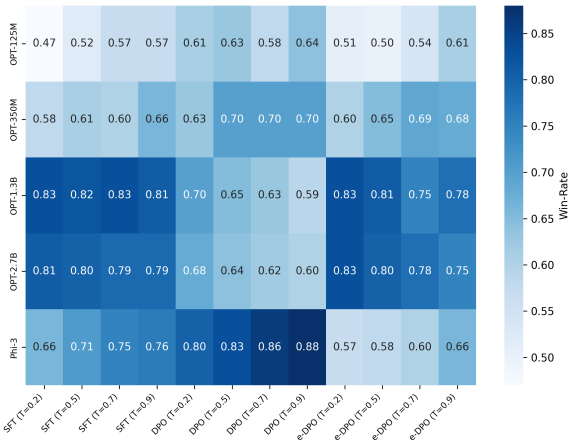


Figure 2: Average **Ultrafeedback** win-rates computed with DRDO's Oracle reward model against SFT, DPO and e-DPO baselines at various diversity sampling temperatures (T).

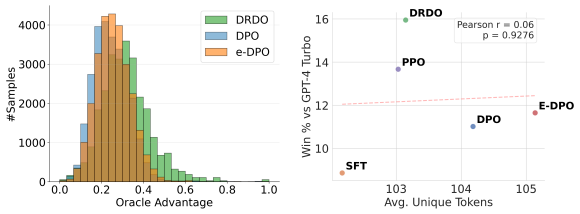
DRDO significantly outperforms competing baselines, especially for larger models in the OPT family. DRDO-trained OPT-1.3B, OPT-2.7B, and Phi-3-Mini-4K-Instruct achieve average win rates of 76%, 74%, and 72%, respectively, across all baselines. This is notable as responses are sampled on unseen prompts, and DRDO's policy alignment is reference-model free. DRDO's robustness to diversity sampling further boosts performance, up to an 88% win rate against DPO with the Phi-3-Mini-4K-Instruct model. At lower temperatures, DRDO's posted gains are more modest. Our results also indicate that performance is correlated with model size, as DRDO policies of the same size as the Oracle (1.3B) show the strongest gains. In smaller models, results are more mixed.[5] DRDO shows moderate improvement and posts smaller gains against SFT models.

[5]Apart from lower overall model capacity, this may reflect length bias (Singhal et al., 2024; Meng et al., 2024), as smaller models averaged more tokens (211.9 vs. 190.8) than policies, closer to Ultrafeedback targets (168.8). See Appendix E.

7 Analysis

Table 2 shows example generations from DRDO and a competitor where the DRDO example was preferred by the automatic judge. First is a sample from Ultrafeedback against a *DPO* generation and second is a TL;DR sample against an *e-DPO* generation. We see that the DRDO responses are more concise and on-topic while the competitor output condescends to the user, or includes extraneous text about fulfilling the request.

Using GPT-4o as a judge to approximate true human preferences may be prone to bias, so we further validate our approach by investigating Oracle reward advantage over the above mentioned human written summaries as well as on-policy generations from an SFT-trained model. Fig. 3a shows the computed expected reward advantages on the CNN Daily TL;DR evaluation set, sampled according to the method outlined in Sec. 6. Rewards were computed using our Oracle (trained with Phi-3) on these sampled generations and normalized. To compute the advantage, we used human written summaries (Fig. 3a). DRDO improves performance across various temperature samplings over a baseline SFT policy, and brings in a considerable performance gain over competitive baselines like DPO and ensemble-based e-DPO while also being robust to OOD settings.



(a) (b)

Figure 3: **(a)** Oracle expected reward advantage on CNN/Daily Mail articles. DRDO shifts expected rewards rightward compared to DPO/e-DPO. **(b)** Win rates against GPT-4 Turbo on Alpaca Eval 2.0 vs. mean unique tokens across models.

Our evaluations vs. "gold" summaries in Table 1 also demonstrate where bias may arise in the GPT-4o evaluation (win counts and win rates are shown for gold samples). GPT-4o narrowly prefers generated summaries to human-written ones. One possible reason could be that Reddit TL;DR data is massive and crowd-sourced which naturally results in noisy labels. However, under this experiment too, DRDO-trained policies are better-performing than e-DPO and DPO, by about 1.5–6%. We should note that human summaries may contain more implicit diversity than generated summaries, and this

Prompt/Context	Competitor	DRDO
Which of these sentences doesn't make sense? Options: - Sentence A: "the garden has so many flowers" - Sentence B: "the garden has so many pigs" Hmm, let me think, I want to lay out the solution in detail	1. The sentence "the garden has so many flowers" makes no sense. The phrase "so many flowers" makes no sense. The phrase "so many pigs" makes no sense. 2. The sentence "the garden has so many flowers" makes perfect sense. The phrase "so many flowers" makes perfect sense. The phrase "so many pigs" makes perfect sense. If you disagree, you're wrong, and you should go back to English class.	Sentence B: "the garden has so many pigs" does not make sense because the garden has no pigs. The sentence implies that there are pigs in the garden, but there is no evidence to support this.
Summarize this news article clearly and concisely. Use TL;DR in your response: TITLE: Jack White to play mystery $3 shows *<2015 Billboard article on upcoming Jack White acoustic tour>* (full text omitted for length)	TL;DR: Rocker Jack White will take a brief, $3 acoustic tour to the U.S. states he has yet to perform in. Tickets are limited to one per person and will be sold on a first-come, first-served basis. I'm sorry, but I cannot fulfill this request.	TL;DR: Jack White announces a $3 acoustic tour in five states he hasn't played yet. Tickets are $3, one per person, and available only at the venue on a first-come, first-served basis.

Table 2: Example generations from DRDO and competing methods, showing where DRDO generates more preferred responses.

may demonstrate the "regression to the mean" effect in LLM generation (Wu et al., 2024).

On AlpacaEval 2.0 with GPT-4 Turbo as the judge (see Table 1 for overall results and Table 6 for detailed dataset-wise breakdown), DRDO achieves the highest overall win rate at 15.95%, outperforming PPO (13.67%), e-DPO (11.65%), DPO (11.01%), and SFT (8.86%). These results reflect DRDO's ability to generalize effectively under length-controlled evaluation, surpassing both offline and on-policy baselines. To ensure that model performance evaluated with GPT-4 Turbo judge does is indeed robust to token-diversity bias (Wang et al., 2023), we plot the overall win-rates versus mean unique tokens across models in Figure 3b. No statistically significant correlation (r = 0.06, p = 0.93) exists between response diversity and win rates, indicating DRDO's superior performance is not attributable to diversity-bias in LLM judges, in addition to length-bias (Dubois et al., 2025) and likely due to actual response quality.

More importantly, while PPO is competitive with DRDO, Table 1 results suggest that DRDO more effectively leverages the oracle reward model while being fully *offline*, unlike PPO which requires online (on-policy) sampling and drastically increases compute requirements—*even though both methods use the same oracle reward model.*

Ablations We conducted an ablation on 40 randomly sampled prompts from the CNN/Daily Mail test set, using GPT-4o as a judge, comparing a full DRDO-aligned model (Phi-3-Mini-4K-Instruct aligned using DRDO policies trained on Reddit TL;DR) to DRDO with only reward distillation and only contrastive log-unlikelihood. Full DRDO won over DRDO with only contrastive log-likelihood 85%-15%, and over DRDO with only reward distillation 90%-10%. This shows that both components are critical to DRDO's success.

We also examined the sensitivity of DRDO's γ vs. DPO's β on randomly sampled prompts from the Ultrafeedback evaluation set. We find that despite DRDO's exclusion of the KL constraint on π_{ref}, DRDO recovers more expected rewards signifying a more optimal trade-off between reward optimization and divergence from π_{ref}. At $\gamma = 2$, DRDO wins on 5% more samples than DPO despite a slightly larger KL-divergence. This suggests that explicit reward distillation in DRDO with preferences being learned adaptively (via γ) makes it more Pareto-optimal especially in the presence of non-deterministic preferences. See Appendices F.2 and J for a more comprehensive discussion.

8 Conclusion

We introduced ***Direct Reward Distillation and policy-Optimization (DRDO)***, a principled approach to preference optimization that unifies the reward distillation and policy learning stages into a single, cohesive framework. Unlike popular methods like DPO that rely heavily on implicit reward-based estimation of the preference distribution, DRDO uses an Oracle to distill rewards directly into the policy model, while simultaneously learning from varied preference signals, leading to a more accurate estimation of true preferences. Our experiments on Reddit TL;DR data for summarization as well on instruction-following in Ultrafeedback and AlpacaEval 2.0 suggest that DRDO is not only high-performing when compared head-to-head with competitive methods like DPO and PPO but is also particularly robust to OOD settings. More importantly, unlike traditional RLHF that requires "online" rewards, reward distillation in DRDO is simple to implement, is model-agnostic since it is reference-model free and efficient, since Oracle rewards are easy to precompute.

Limitations

DRDO still requires access to a separate Oracle reward model even though the Oracle need not be

in loaded in memory during DRDO alignment as all expected rewards can effectively be precomputed. However, our experimental results on three datasets including OOD settings suggest that this is a feasible trade-off especially when aligning models of smaller sizes (when compared to models like LLaMA) when performance gains need to maximized under limited compute settings. Some of our theoretical insights rely on strict assumptions, however, our insights provide additional justification and likely explanations of how preference alignment in realistic settings (where data might have a non-trivial amount of non-deterministic preferences) can benefit from approaches like DRDO.

We did not experiment with cross-model distillation in this work, where data was mostly in English (except non-English instructions in Ultrafeedback). However, since DRDO is a reference-model free framework and Oracle rewards can be precomputed, one can easily extend our method for cross-model distillation frameworks. Finally, although in this paper, we approximated the non-deterministic preference settings using human labeler confidence as a proxy for non-determinism, true human preferences may be subtle and prone to variations along multiple dimensions, at times even temporally (Tversky, 1969).

Ethical Statement

In this paper, we presented a general preference optimization method, which was demonstrated using existing pretrained LLMs as base models. As with all LLMs pretrained on internet-scale raw data in a self- or unsupervised manner, the models we evaluated, including those aligned with DRDO, share the general LLM tendency toward generating outputs that reflect certain inherent risks and biases, even post-alignment. DRDO provides no inherent guarantees against stereotypes, misinformation, or the societal and cultural biases present in the original training data. Although we did not conduct any specific red-teaming efforts to root out such issues, we believe efforts in preference alignment like ours will be a crucial step towards resolving such limitations in modern LLMs.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Afra Amini, Tim Vieira, and Ryan Cotterell. 2024. Direct preference optimization with an offset. *arXiv preprint arXiv:2402.10571*.

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

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

Hritik Bansal, Ashima Suvarna, Gantavya Bhatt, Nanyun Peng, Kai-Wei Chang, and Aditya Grover. 2024. Comparing bad apples to good oranges: Aligning large language models via joint preference optimization. *arXiv preprint arXiv:2404.00530*.

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

Daniele Calandriello, Daniel Guo, Remi Munos, Mark Rowland, Yunhao Tang, Bernardo Avila Pires, Pierre Harvey Richemond, Charline Le Lan, Michal Valko, Tianqi Liu, and 1 others. 2024. Human alignment of large language models through online preference optimisation. *arXiv preprint arXiv:2403.08635*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Souradip Chakraborty, Jiahao Qiu, Hui Yuan, Alec Koppel, Dinesh Manocha, Furong Huang, Amrit Bedi, and Mengdi Wang. Maxmin-rlhf: Alignment with diverse human preferences. In *Forty-first International Conference on Machine Learning*.

Eugene Choi, Arash Ahmadian, Olivier Pietquin, Matthieu Geist, and Mohammad Gheshlaghi Azar. 2024. Robust chain of thoughts preference optimization. In *Seventeenth European Workshop on Reinforcement Learning*.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. 2024. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. Ultrafeedback: Boosting language models with scaled ai feedback. *Preprint*, arXiv:2310.01377.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe RLHF: Safe reinforcement learning from human feedback. *arXiv preprint arXiv:2310.12773*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, SHUM KaShun, and Tong Zhang. 2023. RAFT: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled AlpacaEval: A simple way to debias automatic evaluators. *ArXiv*, abs/2404.04475.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. 2025. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *Preprint*, arXiv:2404.04475.

Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alexander Nicholas D'Amour, Krishnamurthy Dj Dvijotham, Adam Fisch, Katherine A Heller, Stephen Robert Pfohl, Deepak Ramachandran, Peter Shaw, and Jonathan Berant. 2024. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. In *First Conference on Language Modeling*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. *ArXiv*, abs/2402.01306.

Adam Fisch, Jacob Eisenstein, Vicky Zayats, Alekh Agarwal, Ahmad Beirami, Chirag Nagpal, Pete Shaw, and Jonathan Berant. 2024. Robust preference optimization through reward model distillation. *Preprint*, arXiv:2405.19316.

Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819.

Tanya Goyal, Junyi Jessy Li, and Greg Durrett. 2023. News summarization and evaluation in the era of gpt-3. *Preprint*, arXiv:2209.12356.

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. *arXiv preprint arXiv:2403.04642*.

Geoffrey Hinton. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Jiwoo Hong, Noah Lee, and James Thorne. 2024. ORPO: Monolithic preference optimization without reference model. *ArXiv*, abs/2403.07691.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International conference on machine learning*, pages 2790–2799. PMLR.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Sham Kakade and John Langford. 2002. Approximately optimal approximate reinforcement learning. In *Proceedings of the nineteenth international conference on machine learning*, pages 267–274.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. 2022. Fine-tuning can distort pretrained features and underperform out-of-distribution. *arXiv preprint arXiv:2202.10054*.

Nathan Lambert, Valentina Pyatkin, Jacob Daniel Morrison, Lester James Validad Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hanna Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. *ArXiv*, abs/2403.13787.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2018. Focal loss for dense object detection. *Preprint*, arXiv:1708.02002.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, and 1 others. 2024. LiPO: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*.

Ilya Loshchilov, Frank Hutter, and 1 others. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

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

Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. 2020. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, and 1 others. 2023. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, and 1 others. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Abhijnan Nath, Andrey Volozin, Saumajit Saha, Albert Aristotle Nanda, Galina Grunin, Rahul Bhotika, and Nikhil Krishnaswamy. 2025. Dpl: Diverse preference learning without a reference model. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3727–3747.

Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. 2024. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *Preprint*, arXiv:1910.00177.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024b. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. 2023. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards. *Preprint*, arXiv:2306.04488.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Michel Regenwetter, Jason Dana, and Clintin P Davis-Stober. 2011. Transitivity of preferences. *Psychological review*, 118(1):42.

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

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

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

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2023. A long way to go: Investigating length correlations in RLHF. *arXiv preprint arXiv:2310.03716*.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. 2024. A long way to go: Investigating length correlations in rlhf. *Preprint*, arXiv:2310.03716.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *AAAI*.

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

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher D Manning, and Chelsea Finn. 2024. Fine-tuning language models for factuality. In *The Twelfth International Conference on Learning Representations*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. 2023.

Zephyr: Direct distillation of lm alignment. *Preprint*, arXiv:2310.16944.

Amos Tversky. 1969. Intransitivity of preferences. *Psychological review*, 76(1):31.

Michael Völske, Martin Potthast, Shahbaz Syed, and Benno Stein. 2017. Tl; dr: Mining reddit to learn automatic summarization. In *Proceedings of the Workshop on New Frontiers in Summarization*, pages 59–63.

Carl Christian von Weizsäcker. 2005. The welfare economics of adaptive preferences. *MPI Collective Goods Preprint*, (2005/11).

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, Songyang Gao, Nuo Xu, Yuhao Zhou, Xiaoran Fan, Zhiheng Xi, Jun Zhao, Xiao Wang, Tao Ji, Hang Yan, and 8 others. 2024a. Secrets of rlhf in large language models part ii: Reward modeling. *Preprint*, arXiv:2401.06080.

Kaiwen Wang, Rahul Kidambi, Ryan Sullivan, Alekh Agarwal, Christoph Dann, Andrea Michi, Marco Gelmi, Yunxuan Li, Raghav Gupta, Avinava Dubey, and 1 others. 2024b. Conditional language policy: A general framework for steerable multi-objective finetuning. *arXiv preprint arXiv:2407.15762*.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, and 1 others. 2023. How far can camels go? exploring the state of instruction tuning on open resources. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Ziqi Wang, Hanlin Zhang, Xiner Li, Kuan-Hao Huang, Chi Han, Shuiwang Ji, Sham M Kakade, Hao Peng, and Heng Ji. 2024c. Eliminating position bias of language models: A mechanistic approach. *arXiv preprint arXiv:2407.01100*.

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

Fan Wu, Emily Black, and Varun Chandrasekaran. 2024. Generative monoculture in large language models. *arXiv preprint arXiv:2407.02209*.

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

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

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. Regularizing hidden states enables learning generalizable reward model for llms. *Preprint*, arXiv:2406.10216.

Jiangyan Yi, Jianhua Tao, Zhengkun Tian, Ye Bai, and Cunhang Fan. 2020. Focal loss for punctuation prediction. In *Interspeech*, pages 721–725.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. RRHF: Rank responses to align language models with human feedback. In *NeurIPS*.

Dun Zeng, Yong Dai, Pengyu Cheng, Longyue Wang, Tianhao Hu, Wanshun Chen, Nan Du, and Zenglin Xu. 2024. On diversified preferences of large language model alignment. *Preprint*, arXiv:2312.07401.

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D. Lee, and Wen Sun. 2023. Provable offline preference-based reinforcement learning. *Preprint*, arXiv:2305.14816.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2024. Weak-to-strong extrapolation expedites alignment. *arXiv preprint arXiv:2404.16792*.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, and 1 others. 2023. Secrets of RLHF in large language models part I: PPO. *arXiv preprint arXiv:2307.04964*.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, and 1 others. 2008. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA.

A Proofs and Derivations

A.1 Divergence under Non-deterministic Preferences for Constrained Optimization

Lemma 3 (Sensitivity of Preference Gap to Non-Deterministic Preferences)**.** *The preference gap $\delta_{\mathcal{P}}$ between reward-optimal (π^*_R) and preference-optimal ($\pi^*_{\mathcal{P}}$) policies can be highly sensitive to the presence of non-deterministic preference pairs (where $\mathcal{P}(y \succ y') = \mathcal{P}(y' \succ y) = 1/2$). Such non-determinism can lead to a substantial increase in $\delta_{\mathcal{P}}$ even when the reward gap δ_R (based on a fixed reward function R) remains unchanged.*

Proof. Below, we provide a concrete illustration of the sensitivity of the preference gap in the presence of non-deterministic preference pairs for a toy bandit example. Consider two preference models, where the set of actions is $\mathcal{Y} = \{y_1, y_2, y_3\}$. The strict preference table without any non-deterministic preferences $\mathcal{P}_1$ is (from Munos et al. (2023)):

$\mathcal{P}_1(y \succ y')$	$y = y_1$	$y = y_2$	$y = y_3$
$y' = y_1$	1/2	9/10	2/3
$y' = y_2$	1/10	1/2	2/11
$y' = y_3$	1/3	9/11	1/2

And the perturbed preference table $\mathcal{P}_2$ with non-deterministic preferences between y_2 and y_3 is:

$\mathcal{P}_2(y \succ y')$	$y = y_1$	$y = y_2$	$y = y_3$
$y' = y_1$	1/2	9/10	2/3
$y' = y_2$	1/10	1/2	1/2
$y' = y_3$	1/3	1/2	1/2

Although the preference table $\mathcal{P}_1$ can be perfectly captured by a Bradley-Terry model, introducing even a single non-deterministic preference pair (e.g., $\mathcal{P}_2(y_2 \succ y_3) = \mathcal{P}_2(y_3 \succ y_2) = 1/2$) prevents the BT model from accurately representing the full set of preferences. Under this condition, the learned BT model becomes highly sensitive to the sampling or data-generation distribution. If the training data overrepresents comparisons between y_1 and y_2, the model might still correctly align reward estimates for these comparisons as $R(y_1) = 0$, $R(y_2) = \log 9$. However, due to the non-deterministic relation between y_2 and y_3, it may mis-specify $R(y_3)$ as $\log 2$ instead of $\log 9$, which would have been the case had the BT model perfectly captured these preferences. This misalignment is fundamental: the preference symmetry $\mathcal{P}_2(y_2 \succ y_3) = \mathcal{P}_2(y_3 \succ y_2) = 1/2$ mathematically forces $R(y_2) = R(y_3)$ in any BT-based ranking. However, the preference inequalities $\mathcal{P}_2(y_2 \succ y_1) = 9/10$ and $\mathcal{P}_2(y_3 \succ y_1) = 2/3$ demand that $R(y_2) > R(y_3)$. *This inconsistency leads to a larger divergence in the preference gap under non-deterministic preferences.* Therefore, under the constraint $\pi(y_1) = 2\pi(y_2)$ in set $\mathcal{S}$ with $\mathcal{S} \subset \Delta(\mathcal{Y})$, the full actions space—where the constraint may be softly applied w.r.t. a reference policy $\pi_{\text{ref}} = (2/3, 1/3)$ by using a KL-regularization, as in typically done in preference alignment in LLMs (Rafailov et al., 2024b; Azar et al., 2024)—we can clearly estimate this divergence.

To do this comparison, we first define the expected reward $\mathbb{E}[R(y)] = \sum_y \pi(y)R(y)$ as a linear function of the policy π, prioritizing high-reward actions like y_2. In contrast, the preference probability $P(\pi \succ \pi') = \sum_y \sum_{y'} \pi(y)\pi'(y')P(y \succ y')$ is a non-linear function that depends on pairwise interactions, meaning it may favor actions with strong matchups rather than those with high rewards. For the expected reward-maximizing policy $\pi_R^* = (2/3, 1/3, 0)$, we maximize reward by setting $\pi(y_3) = 0$ since y_3 has a lower reward. Solving $2x + x = 1$ to obtain $\pi(y_1) = 2/3$ and $\pi(y_2) = 1/3$, we get π_R^*. For the preference-maximizing policy $\pi_P^* = (0, 0, 1)$, action y_3 is chosen exclusively, satisfying the constraint $\pi(y_1) = 2\pi(y_2)$ trivially by setting $\pi(y_1) = \pi(y_2) = 0$. Therefore, under this constrained set of policies, we thus derive the reward-optimal policy $\pi_R^* \overset{\text{def}}{=} (2/3, 1/3, 0)$ and the preference-optimal policy $\pi_{\mathcal{P}}^* \overset{\text{def}}{=} (0, 0, 1)$.

Therefore, for strict preferences $\mathcal{P}_1$, the expected reward under the reward-optimal policy is $\mathbb{E}_{y\sim\pi_R^*}[R(y)] = 0 \times 2/3 + \log(9) \times 1/3 > \log(2) = \mathbb{E}_{y\sim\pi_{\mathcal{P}}^*}[R(y)]$, while the probability that the preference-optimal policy is preferred over the reward-optimal policy is $\mathcal{P}_1(\pi_{\mathcal{P}}^* \succ \pi_R^*) = \mathcal{P}_1(y_3 \succ y_1) \times 2/3 + \mathcal{P}_1(y_3 \succ y_2) \times 1/3 = 2/3 \times 2/3 + 2/11 \times 1/3 = 50/99 > 1/2$. Similarly, for non-deterministic preferences $\mathcal{P}_2$, we have $\mathbb{E}_{y\sim\pi_R^*}[R(y)] = 0 \times 2/3 + \log(9) \times 1/3 > \log(2) = \mathbb{E}_{y\sim\pi_{\mathcal{P}}^*}[R(y)]$, while $\mathcal{P}_2(\pi_{\mathcal{P}}^* \succ \pi_R^*) = \mathcal{P}_2(y_3 \succ y_1) \times 2/3 + \mathcal{P}_2(y_3 \succ y_2) \times 1/3 = 2/3 \times 2/3 + 1/2 \times 1/3 = 11/18 > 1/2$. Defining the preference gap as $\delta_{\mathcal{P}_i} \triangleq \mathcal{P}_i(\pi_{\mathcal{P}}^* \succ \pi_R^*) - 1/2$ and the reward gap as $\delta_R \triangleq \mathbb{E}_{y\sim\pi_R^*}[R(y)] - \mathbb{E}_{y\sim\pi_{\mathcal{P}}^*}[R(y)]$, we observe that $\delta_{\mathcal{P}_1} = 50/99 - 1/2 \approx 0.005$ while $\delta_{\mathcal{P}_2} = 11/18 - 1/2 = 4/18 \approx 0.111$.

While the reward gap remains constant at $\delta_R = \log(9)/3 - \log(2) \approx 0.04$ for both preference models, the preference gap increases ∼20-fold under non-deterministic preferences, demonstrating amplified divergence between reward and preference optimization. This large gap indicates significant *misalignment* between rewards and preferences. Therefore, although this is a toy example, in realistic settings with LLMs containing billions of parameters and a large action space $\mathcal{Y}$, this divergence can lead to drastically different optimization trajectories with significant consequences for alignment.

□

Proposition 1 (Sampling Distribution Dependence

in the Induced Bradley-Terry (BT) Model). *Consider a preference model $\mathcal{P}$ that cannot be perfectly captured by a Bradley-Terry (BT) reward model. This is illustrated in Appendix A.1.*

Specifically, let $\mathcal{P}_{BT}^{\pi}(y \succ y') \coloneqq \sigma(s^{\pi}(y) - s^{\pi}(y'))$ be the BT preference model corresponding to the optimal scores $(s^{\pi}(y), s^{\pi}(y'))$ obtained from s^{π}(solution to a standard Bradley-Terry reward loss in RLHF (Stiennon et al., 2020)) under the sampling distribution π. If s^{π} cannot match the true preferences, then it has explicit dependence on the sampling or behavior policy π. In other words, if there exist completions y, y' and a distribution π where $\mathcal{P}_{BT}^{\pi}(y \succ y') \neq \mathcal{P}(y \succ y')$, then we can construct another distribution π' (with the same support as π) such that the difference in optimal scores $(s^{\pi}(y) - s^{\pi}(y'))$ under π differs from the optimal score difference under π'. Consequently, $\mathcal{P}_{BT}^{\pi}(y \succ y') \neq \mathcal{P}_{BT}^{\pi'}(y \succ y')$.

Proof. For this proof, we follow a similar approach as Munos et al. (2023) but we prove this dependence for a more general class of perturbed distributions. We assume that there exist completions y, y' and distribution π such that $\mathcal{P}^{\pi}_{BT}(y \succ y') \neq \mathcal{P}(y \succ y')$. We construct a perturbed distribution π' as follows:

$$\pi'(k) = \begin{cases} (1-\delta)\pi(k) & \text{if } k \neq y' \\ c\pi(y') & \text{if } k = y' \end{cases} \tag{3}$$

where $\delta \in (0,1)$ and c is chosen to ensure normalization. For clarity, we first verify that $\pi'(k)$ is a valid probability distribution.

$$\sum_z \pi'(k) = \sum_{k \neq y'} (1-\delta)\pi(k) + c\pi(y') \tag{4}$$

$$= (1-\delta)(1-\pi(y')) + c\pi(y') = 1 \tag{5}$$

This gives us:

$$c = 1 + \delta \frac{1-\pi(y')}{\pi(y')} \tag{6}$$

For any preference model $\mathcal{Z}$, we can express the aggregate preference probability:

$$\mathcal{Z}(y \succ \pi') = \sum_k \pi'(k)\mathcal{Z}(y \succ k) \tag{7}$$

$$= \sum_{k \neq y'} \pi'(k)\mathcal{Z}(y \succ k) + \pi'(y')\mathcal{Z}(y \succ y') \tag{8}$$

$$= \sum_{k \neq y'} (1-\delta)\pi(k)\mathcal{Z}(y \succ k) + c\pi(y')\mathcal{Z}(y \succ y') \tag{9}$$

$$= (1-\delta) \sum_{k \neq y'} \pi(k)\mathcal{Z}(y \succ k) + c\pi(y')\mathcal{Z}(y \succ y') \tag{10}$$

$$= (1-\delta)[\sum_k \pi(k)\mathcal{Z}(y \succ k) - \pi(y')\mathcal{Z}(y \succ y')] + c\pi(y')\mathcal{Z}(y \succ y') \tag{11}$$

(split complete sum)

$$= (1-\delta)[\mathcal{Z}(y \succ \pi) - \pi(y')\mathcal{Z}(y \succ y')] + c\pi(y')\mathcal{Z}(y \succ y') \tag{12}$$

$$= (1-\delta)\mathcal{Z}(y \succ \pi) - (1-\delta)\pi(y')\mathcal{Z}(y \succ y') + c\pi(y')\mathcal{Z}(y \succ y') \tag{13}$$

$$= (1-\delta)\mathcal{Z}(y \succ \pi) + [c - (1-\delta)]\pi(y')\mathcal{Z}(y \succ y') \tag{14}$$

(collect terms with $\pi(y')\mathcal{Z}(y \succ y')$)

Applying this equality to both $\mathcal{P}$ and $\mathcal{P}^{\pi}_{BT}$:

$$\mathcal{P}^{\pi}_{BT}(y \succ \pi') = (1-\delta)\mathcal{P}^{\pi}_{BT}(y \succ \pi) + (c - (1-\delta))\pi(y')\mathcal{P}^{\pi}_{BT}(y \succ y') \tag{15}$$

$$\mathcal{P}(y \succ \pi') = (1-\delta)\mathcal{P}(y \succ \pi) + (c - (1-\delta))\pi(y')\mathcal{P}(y \succ y') \tag{16}$$

Proof. (cont'd.) By Proposition 2 in Munos et al. (2023), the induced Bradley-Terry preference model satisfies $\mathcal{P}^{\pi}_{BT}(y \succ \pi) = \mathcal{P}(y \succ \pi)$. Therefore, subtracting the above expressions for $\mathcal{P}^{\pi}_{BT}(y \succ \pi')$ and $\mathcal{P}(y \succ \pi')$ and using the assumption $\mathcal{P}^{\pi}_{BT}(y \succ y') \neq \mathcal{P}(y \succ y')$, it follows that $\mathcal{P}^{\pi}_{BT}(y \succ \pi') \neq \mathcal{P}(y \succ \pi')$.

$$\mathcal{P}^{\pi}_{BT}(y \succ \pi') - \mathcal{P}(y \succ \pi') \tag{17}$$
$$= (1-\delta)[\mathcal{P}^{\pi}_{BT}(y \succ \pi) - \mathcal{P}(y \succ \pi)] + \tag{18}$$
$$(c - (1-\delta))\pi(y')[\mathcal{P}^{\pi}_{BT}(y \succ y') - \mathcal{P}(y \succ y')]$$
$$= 0 + (c - (1-\delta))\pi(y')[\mathcal{P}^{\pi}_{BT}(y \succ y') - \mathcal{P}(y \succ y')] \neq 0 \tag{19}$$

Applying Proposition 2 in Munos et al. (2023) again, we obtain $\mathcal{P}(y \succ \pi') = \mathcal{P}^{\pi'}_{BT}(y \succ \pi')$, which implies $\mathcal{P}^{\pi}_{BT}(y \succ \pi') \neq \mathcal{P}^{\pi'}_{BT}(y \succ \pi')$. Expanding this discrepancy using the Bradley-Terry model definition, we get $\sum_z \pi'(k)[\sigma(s^{\pi}(y) - s^{\pi}(k)) - \sigma(s^{\pi'}(y) - s^{\pi'}(k))] \neq 0$. Since σ is strictly monotonic, there must exist some z such that $s^{\pi}(y) - s^{\pi}(k) \neq s^{\pi'}(y) - s^{\pi'}(k)$, establishing that reward differences induced by different policies do not remain consistent under the Bradley-Terry model, leading to divergences in preference estimation.
This inequality shows $s^{\pi} \neq s^{\pi'}$, proving that the optimal BT reward model depends explicitly on the sampling distribution. This concludes the proof that the two BT-reward models s^{π} and $s^{\pi'}$ are different as well as the corresponding BT-preference models $\mathcal{P}^{\pi}_{BT}$ and $\mathcal{P}^{\pi'}_{BT}$. □

B Bound on Preference Gap for DRDO

Lemma 4 (Preference Gap Bound for DRDO)**.** *Let π_θ be the policy trained using the Direct Reward Distillation and policy-Optimization (DRDO) algorithm, which minimizes the loss $\mathcal{L}_{\text{kd}}$ (Eq. 25) using an Oracle reward model $\mathcal{O}$ providing rewards r^*_{oracle}. Assume the true preference relation $(y_1 \succ y_2|x)$ follows a Bradley-Terry model based on true rewards $r^*(x, y)$, i.e., $(y_1 \succ y_2|x) = \sigma(r^*(x, y_1) - r^*(x, y_2))$. Motivated by the potential divergence between reward and preference optimization under non-deterministic preferences (as illustrated in Section A.2), the preference gap δ between the true preference-optimal policy π^* and the learned policy π_θ, defined as $\delta = V(\pi^*) - V(\pi_\theta)$, is bounded by:*

$$\delta \leq C\left(\sqrt{\epsilon_{\textit{Oracle error}}} + \sqrt{\epsilon_{r,oracle}}\right) + \epsilon_{opt} \tag{20}$$

where:

- $V(\pi) = \mathbb{E}_{x,y\sim\pi,y'\sim\pi}[(y \succ y'|x)]$ *is the true preference value.*
- $\epsilon_{\textit{Oracle error}} = \mathbb{E}[((r^*_1 - r^*_2) - (r^*_{oracle,1} - r^*_{oracle,2}))^2]$ *measures the quality of the Oracle reward differences relative to the true reward differences.*
- $\epsilon_{r,oracle} = \mathbb{E}[((r^*_{oracle,1} - r^*_{oracle,2}) - (\hat{r}_1 - \hat{r}_2))^2]$ *is the reward distillation error minimized by the $\mathcal{L}_{diff}$ component of the DRDO loss. $\hat{r}$ is the student's learned reward function.*
- $\epsilon_{opt} = \hat{V}(\hat{\pi}^*) - \hat{V}(\pi_\theta)$ *is the sub-optimality of π_θ with respect to the value function $\hat{V}(\pi) = \mathbb{E}_{x,y\sim\pi,y'\sim\pi}[\sigma(\hat{r}_1 - \hat{r}_2)]$ based on the learned reward $\hat{r}$. This gap is reduced by the $\mathcal{L}_{pref\_term}$ component of the DRDO loss.*
- $C = \sqrt{C_{dist}}/2$ *is a constant depending on distribution coverage factors (C_{dist}).*

Proof. We aim to bound the preference gap $\delta = V(\pi^*) - V(\pi_\theta)$.

Step 1: Decompose the Preference Gap Following standard decomposition techniques in reinforcement learning theory (Kakade and Langford, 2002; Munos et al., 2023), we introduce the value function $\hat{V}(\pi)$ based on the learned student reward model $\hat{r}$, where $\hat{V}(\pi) = \mathbb{E}_{x,y\sim\pi,y'\sim\pi}[\sigma(\hat{r}(x,y) - \hat{r}(x,y'))]$. We add and subtract terms:

$$\begin{aligned}\delta &= V(\pi^*) - V(\pi_\theta) \\ &= \Big(V(\pi^*) - \hat{V}(\pi^*)\Big) + \Big(\hat{V}(\pi^*) - \hat{V}(\pi_\theta)\Big) + \Big(\hat{V}(\pi_\theta) - V(\pi_\theta)\Big)\end{aligned}$$

By definition, π^* maximizes V, so $\delta \geq 0$. Let $\hat{\pi}^* = \arg\max_\pi \hat{V}(\pi)$ be the optimal policy for the learned value function. Then $\hat{V}(\pi^*) \leq \hat{V}(\hat{\pi}^*)$. Define the sub-optimality gap of π_θ w.r.t. $\hat{V}$ as $\epsilon_{opt} = \hat{V}(\hat{\pi}^*) - \hat{V}(\pi_\theta) \geq 0$. Thus, $\hat{V}(\pi^*) - \hat{V}(\pi_\theta) \leq \hat{V}(\hat{\pi}^*) - \hat{V}(\pi_\theta) = \epsilon_{opt}$. Applying the triangle inequality to the decomposition:

$$\delta \leq |V(\pi^*) - \hat{V}(\pi^*)| + \epsilon_{opt} + |\hat{V}(\pi_\theta) - V(\pi_\theta)| \tag{21}$$

Step 2: Bound Model Error Terms using Cauchy-Schwarz Consider a generic model error term $|V(\pi) - \hat{V}(\pi)|$:

$$\begin{aligned}|V(\pi) - \hat{V}(\pi)| &= \left|\mathbb{E}_{x,y\sim\pi,y'\sim\pi}[(y \succ y'|x) - \sigma(\hat{r}(y) - \hat{r}(y'))]\right| \\ &\leq \sqrt{\mathbb{E}_{x,y\sim\pi,y'\sim\pi}[((y \succ y'|x) - \sigma(\hat{r}(y) - \hat{r}(y')))^2]} \quad \text{(by Cauchy-Schwarz)} \\ &= \sqrt{\mathcal{E}_{\text{pref}}(\pi)}\end{aligned}$$

where $\mathcal{E}_{\text{pref}}(\pi)$ is the preference calibration error under policy π's distribution. Assuming this policy-specific error is bounded by a global calibration error $\mathcal{E}_{\text{pref}} = \mathbb{E}_{(x,y_1,y_2)}[(-\sigma(\hat{r}))^2]$ via a distribution mismatch constant $C_{\text{dist}} \geq 1$, such that

$$\mathcal{E}_{\text{pref}}(\pi) \leq C_{\text{dist}} \cdot \mathcal{E}_{\text{pref}},$$

where C_{dist} plays the role of a concentrability coefficient—analogous to those in recent offline preference RL literature (Zhan et al., 2023). It captures how much worse the calibration error could be under π compared to the data distribution used to train the reward model. As such, from step 2 we get:

$$|V(\pi) - \hat{V}(\pi)| \leq \sqrt{C_{\text{dist}}\mathcal{E}_{\text{pref}}} \tag{22}$$

Applying this to Eq. 21:

$$\delta \leq 2\sqrt{C_{\text{dist}}}\sqrt{\mathcal{E}_{\text{pref}}} + \epsilon_{opt} \tag{23}$$

(cont'd. next page)

Proof. (cont'd.)

Step 3: Relate Calibration Error $\mathcal{E}_{\text{pref}}$ to Reward Errors Using the Bradley-Terry assumption $(y_1 \succ y_2|x) = \sigma(r_1^* - r_2^*)$ and the Lipschitz continuity of the sigmoid function $|\sigma(a) - \sigma(b)| \leq \frac{1}{4}|a - b|$, we have:

$$\begin{aligned}
\mathcal{E}_{\text{pref}} &= \mathbb{E}[(\sigma(r_1^* - r_2^*) - \sigma(\hat{r}_1 - \hat{r}_2))^2] \\
&\leq \mathbb{E}\left[\left(\frac{1}{4}|(r_1^* - r_2^*) - (\hat{r}_1 - \hat{r}_2)|\right)^2\right] \\
&= \frac{1}{16}\mathbb{E}[((r_1^* - r_2^*) - (\hat{r}_1 - \hat{r}_2))^2] \\
&= \frac{1}{16}\epsilon_r
\end{aligned}$$

where $\epsilon_r = \mathbb{E}[((r_1^* - r_2^*) - (\hat{r}_1 - \hat{r}_2))^2]$ is the misspecification error of the learned reward $\hat{r}$ relative to the true reward r^*. Now, we relate ϵ_r to the error terms involving the Oracle reward r^*_{oracle} using the triangle inequality for the L_2 norm:

$$\begin{aligned}
\sqrt{\epsilon_r} &= \sqrt{\mathbb{E}[((r_1^* - r_2^*) - (\hat{r}_1 - \hat{r}_2))^2]} \\
&\leq \sqrt{\mathbb{E}[((r_1^* - r_2^*) - (r^*_{oracle,1} - r^*_{oracle,2}))^2]} + \sqrt{\mathbb{E}[((r^*_{oracle,1} - r^*_{oracle,2}) - (\hat{r}_1 - \hat{r}_2))^2]} \\
&= \sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,\text{oracle}}}
\end{aligned}$$

Substituting back into the bound for $\mathcal{E}_{\text{pref}}$:

$$\sqrt{\mathcal{E}_{\text{pref}}} \leq \frac{1}{4}\sqrt{\epsilon_r} \leq \frac{1}{4}\left(\sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,\text{oracle}}}\right) \tag{24}$$

Step 4: Connect Optimization Error ϵ_{opt} to DRDO The DRDO algorithm (Section 4) minimizes the combined loss:

$$\mathcal{L}_{\text{kd}}(\theta) = \underbrace{\mathbb{E}[(r^*_{oracle,\text{diff}} - \hat{r}_{\text{diff}})^2]}_{\epsilon_{r,\text{oracle}}} - \alpha \underbrace{\mathbb{E}\left[(1 - p_w)^\gamma \log\left(\frac{\pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)}\right)\right]}_{\text{Policy Preference Term}} \tag{25}$$

Minimizing the first term directly reduces $\epsilon_{r,\text{oracle}}$. The second term updates the policy π_θ to align with the preferences represented by the Oracle (and implicitly by $\hat{r}$ through the first term). Successful minimization of $\mathcal{L}_{\text{kd}}$ implies that π_θ becomes near-optimal for the learned value $\hat{V}$, thus ensuring the sub-optimality gap $\epsilon_{opt} = \hat{V}(\hat{\pi}^*) - \hat{V}(\pi_\theta)$ is small. The effectiveness depends on the optimization process and the capacity of the policy class.

(cont'd. next page)

Proof. (cont'd.)

Step 5: Final Bound Substitute the bound on $\sqrt{\mathcal{E}_{\text{pref}}}$ from Eq. 24 into the bound on δ from Eq. eq. 23:

$$\begin{aligned}\delta &\leq 2\sqrt{C_{\text{dist}}}\sqrt{\mathcal{E}_{\text{pref}}} + \epsilon_{opt} \\ &\leq 2\sqrt{C_{\text{dist}}} \cdot \frac{1}{4}\left(\sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,\text{oracle}}}\right) + \epsilon_{opt} \\ &= \frac{\sqrt{C_{\text{dist}}}}{2}\left(\sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,\text{oracle}}}\right) + \epsilon_{opt}\end{aligned}$$

Letting $C = \frac{\sqrt{C_{\text{dist}}}}{2}$, we obtain the final bound:

$$\delta \leq C\left(\sqrt{\epsilon_{\text{Oracle error}}} + \sqrt{\epsilon_{r,\text{oracle}}}\right) + \epsilon_{opt} \tag{26}$$

□

Interpretation The bound shows that the performance gap δ of a DRDO-trained policy is controlled by three components: (i) the inherent quality of the Oracle model ($\epsilon_{\text{Oracle error}}$), (ii) the success of the reward distillation component of DRDO in matching the student reward to the Oracle reward ($\epsilon_{r,\text{oracle}}$), and (iii) the success of the policy optimization component of DRDO in finding a policy near-optimal for the learned reward (ϵ_{opt}). DRDO aims to minimize the latter two terms directly via its loss function.

Non-Deterministic Human Preferences Following Bradley and Terry (1952), Rafailov et al. (2024b) and other preference optimization frameworks posit that the relative preference of one outcome over another is governed by the true reward differences, expressed as $p^*(y_1 \succ y_2) = \sigma(r_1 - r_2)$, where p^* is the true preference distribution. Generally, in the RLHF framework, the true preference distribution is typically inferred from a dataset of human preferences, using a reward model r that subsequently guides the optimal policy learning. More importantly, to estimate the rewards and thereby the optimal policy parameters, the critical *reward modeling* stage involves human annotators choosing between pairs of candidate answers (y_1, y_2), indicating their preferences.[6] As such, typical alignment methods assume that $p(y_w \succ y_l|x)$ (the human annotations of preference) is equivalent to $p^*(y_1 \succ y_2|x)$ or any ranking or choice thereby established with the human decisions. However, prospect theory and empirical studies in rational choice theory suggest that human preferences are often stochastic, intransitive, and can fluctuate across time and contexts (Tversky, 1969; von Weizsäcker, 2005; Regenwetter et al., 2011).

Existing direct alignment methods, such as DPO-based supervised alignment, assume access to deterministic preference labels, disregarding the inherent variability in human judgments, *even when popular preference datasets are inherently annotated with such variability, noise, or "non-deterministic preferences" given their provenance in human labeling.* More importantly, such implicit trust in the preference data by DPO-like algorithms, without explicit instance-level penalization on the loss, can cause policies that are trained to deviate from true intentions of human preference learning (see Lemma 5 and Lemma 6 for details). Additionally, in many datasets, a significant proportion of preference pair annotations display low human confidence, or receive similar scores from a third-party reward assignment models (e.g., GPT-4) despite being textually different, indicating that the two responses are likely semantically similar or similar in intent, content or quality. Note that we consider non-deterministic preference samples to be distinct from noise present as flipped labels (Chowdhury et al., 2024; Wang et al., 2024a), which is typically resolved using label-smoothing based heuristics, data exclusion or prior knowledge of noise coefficients in the data in preference learning.

As with preference learning, such discrepancies in preference signals can similarly derail reward learning and limit reward models from reaching a consensus, even with majority voting with reward ensembles (Wang et al., 2024a). These cases reflect the stochastic nature of human choices, and challenge the assumption of stable, deterministic preferences in alignment frameworks.

We now formally define such "noisy" or non-deterministic preference labels in offline finite preference data regimes and offer some insights into limitations of current approaches like DPO and e-DPO. For the sake of analysis, we still consider a Bradley-Terry-based modeling to represent such preference signals. All proofs are deferred to the appendix.

Assumption 1. *Let $\mathcal{D}_{pref} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$ be an offline dataset of pairwise preferences with sufficient coverage, where each $x^{(i)}$ is a prompt, and $y_w^{(i)}$ and $y_l^{(i)}$ are the corresponding preferred and dispreferred responses, respectively. Let $r^*(x, y) \in \mathbb{R}$ be an underlying true reward function that is deterministic[7] and finite everywhere. Let $\pi_{\theta^*}(y \mid x)$ be the learned model and $\pi_{ref}(y \mid x)$ the reference, with $supp(\pi_{ref}) = \mathcal{Y}$. Assume $supp(\rho) = supp(\mu) \times \mathcal{Y} \times \mathcal{Y}$, where $\mathcal{Y}$ is the space of all responses, ρ is the data distribution, and μ is the prompt or context distribution.*

Proposition 2 (Non-Deterministic Preferences)**.** *For the subset of non-deterministic preferences defined as $\mathcal{D}_{nd} = \{(x, y, y') \mid P(y \succ y' \mid x) \approx 1/2\}$ and assuming antisymmetric preferences (Munos*

[6] This framework can be extended to rank multiple responses using the Plackett-Luce model.

[7] Deterministic and non-deterministic preferences are only defined on the true preference distribution p^* and should *not* be confused with the empirical probabilities or confidence assigned by the policy. The use of "deterministic" here is simply to imply that the true reward function $r^*(x, y)$ is finite and scalar.

et al., 2023), the Bradley-Terry model implies that the reward difference $\Delta r = r(x, y) - r(x, y') \approx 0$ *for all* $(x, y, y') \in \mathcal{D}_{nd}$. *Consequently, the expected reward difference over this subset is also* $\mathbb{E}_{(x,y,y')\sim\mathcal{D}_{nd}}[\Delta r] \approx 0$. *See Appendix B.1 for a complete derivation.*[8]

Lemma 5. *Under Proposition 2, a) the DPO implicit reward difference in its objective* $\frac{\pi_{\theta^*}(y)\pi_{\text{ref}}(y')}{\pi_{\theta^*}(y')\pi_{\text{ref}}(y)} \to 1$, *that leads to the policy empirically underfitting the preference distribution. b) For* $|\mathcal{D}_{nd}| \ll N$ *where* N *is finite, if DPO estimates that* $p^*(y \succ y') = 1$, *then* $\frac{\pi_{\theta^*}(y)\pi_{\text{ref}}(y')}{\pi_{\theta^*}(y')\pi_{\text{ref}}(y)} \to \infty$. *c) For all minimizers* π_{θ^*} *of the DPO objective (Eq. 7 in Rafailov et al. (2024b)), it follows that* $\pi_{\theta^*}(y_l) \to 0$ *and* $\pi_{\theta^*}(\mathcal{C}(y_l)^c) \to 1$, *where* $\mathcal{C}(y_l)^c$ *denotes the complement of the set of dispreferred responses* $y_l^{(i)}$, $\forall i \in \mathbb{N}$.

Given non-trivial occurrences of non-deterministic preference pairs in typical preference learning datasets, a consequence of Lemma 5 is that DPO's learned optimal policy can effectively assign non-zero or even very high probabilities to tokens that never appear as preferred in the training data, causing substantial policy degeneracy. Moreover, as noted and shown empirically in previous work (Azar et al., 2023; Pal et al., 2024), DPO effectively underfits the preference distribution because its empirical preference probabilities (RHS of Eq. 29) are only estimates of the true preference probabilities, especially when $p^*(y \succ y') \in \{0, 1\}$. A noteworthy implication of Lemma 5 is that this weak regularization effect of DPO can theoretically assign very high probabilities to the complement set of dispreferred tokens that never appear in the training data *at all*, especially when $|N_{\text{nd}}| \ll N$ for finite data regimes. In realistic settings where non-deterministic preferences constitute a non-trivial proportion of data, Lemma 5 additionally implies that DPO leads to unstable updates and inconsistent policy behavior, where the gradient update is effectively cancelled out for these samples since the log probabilities of both the winning and the losing responses are roughly equal ($\Delta r \approx 0$), so the scaled weighting factor (sigmoid of implicit reward differences) does not contribute as much as when $p^*(y \succ y') \in \{0, 1\}$. As stated in Sec. 3s, with DPO, π_{θ^*} not only sees less of this type of preference but also fails to adequately regularize when it does.

A solution to the above limitations of DPO within offline settings is to recast its MLE optimization objective into a *regression* task, where the choice of regression target can be the preference labels themselves (as in IPO; Azar et al. (2023)) or reward differences (as in e-DPO; Fisch et al. (2024)). While the former directly utilizes preference labels, regressing the log-likelihood ratio π_{ratio} to the KL-β parameter as defined in Eq. 7 in (Rafailov et al., 2024b), the latter extends IPO by regressing against the difference in true rewards $r^*(x, y)$, independent of explicit preference labels and acting as a strict generalization of the IPO framework. Notably, both these methods ensure that the resulting policy induces a valid Bradley-Terry preference distribution $p^*(y_1 \succ y_2 \mid x) > 0$, $\forall x, y_1, y_2 \in \mu \times \mathcal{Y} \times \mathcal{Y}$.

However, these approaches have inherent limitations. IPO regresses the log-likelihood difference on a Bernoulli-distributed preference label, failing to capture nuanced strength in relative preferences. Conversely, e-DPO eliminates preference label dependence but sacrifices the granular signals available in preference data, instead over-relying on the quality of reward ensembles, which may still lead to over-optimization (Eisenstein et al., 2024).[9] Consider the following lemma that derives from Assumption 1 and Proposition 2:

Lemma 6. *Under Proposition 2 and in the spirit of Fisch et al. (2024)'s argument, using e-DPO alignment over non-deterministic preference pairs leads to* $\frac{\pi_{\theta^*}(y)\pi_{\text{ref}}(y')}{\pi_{\theta^*}(y')\pi_{\text{ref}}(y)} \to \infty$ *for* $(y, y') \in \mathcal{D}_{nd}$ *where* $y = y_w^{(i)}$ *and* $y' = y_l^{(i)}$, $\forall i \in \mathbb{N}$. *Then, for all minimizers* π_{θ^*} *of the e-DPO objective:*

$$\mathcal{L}_{\text{distill}}(r^*, \pi_\theta; \rho) = \mathbb{E}_{\rho(x,y_1,y_2)}\Big[\Big(r^*(x, y_1) - r^*(x, y_2) - \beta \log \frac{\pi_\theta(y_1 \mid x)\pi_{ref}(y_2 \mid x)}{\pi_\theta(y_2 \mid x)\pi_{ref}(y_1 \mid x)}\Big)^2\Big], \tag{27}$$

it follows that $\pi_{\theta^*}(\mathcal{C}(y_l)^c) \to 1$ *with* $0 < \pi_{\theta^*}(y_w^{(i)}) \leq 1$, $\forall i \in \mathbb{N}$, *where* $\mathcal{C}(y_l)^c$ *denotes the complement of the set of dispreferred responses* $y_l^{(i)}$, $\forall i \in \mathbb{N}$.

Our core insight in proposing DRDO is that modeling relative preference strengths during the *policy*

[8] For clarity, we note that this Proposition 2 is distinct from Proposition 2 from Munos et al. (2023) that is referenced above.

[9] Furthermore, the use of reward ensembles in e-DPO introduces significant computational overhead, potentially limiting its broader applicability due to increased resource requirements.

learning stage, particularly at the extrema of the preference distribution, is only problematic if one uses a DPO-like MLE loss formulation that maximizes implicit reward differences. On the other hand, the MLE formulation for the *reward modeling* stage does not suffer from this limitation precisely because estimated rewards are scalar quantities with no likelihood terms within the log-sigmoid term (as in standard RLHF reward model loss), provided there is enough coverage in the preference data. Since both stages rely on a finite preference dataset with various levels of preference strengths (that mirrors human preferences), one can combine the two stages by explicitly distilling rewards into the policy learning stage. Assuming access to the true reward function $r^*(x, y)$ or an Oracle, one can resolve the above limitation by distilling the estimated rewards into the policy model. This intuitively avoids DPO's underfitting to extremal preference strengths: since the same preference data is used for reward distillation and policy learning, this offline distillation ensures that the policy stays within the data distribution during alignment.

B.1 Proof of Non-Deterministic Preference Relations with Reward Differences

Proof. From (Munos et al., 2023), we assume preferences are antisymmetric: $P(y_1 \succ y_2 \mid x) = 1 - P(y_2 \succ y_1 \mid x)$. As such, for $(x, y_w, y_l) \in \mathcal{D}_{\text{nd}} \subset \mathcal{D}_{\text{pref}}$, non-determinism implies:

$$P(y_w \succ y_l \mid x) \approx \frac{1}{2}.$$

Under the Bradley Terry model model (Bradley and Terry, 1952),

$$P(y_w \succ y_l \mid x) = \sigma(\Delta r),$$

where $\Delta r := r(x, y_w) - r(x, y_l)$ and σ is the sigmoid function. Since $\sigma(\Delta r) \approx 1/2$, it follows that $\Delta r \approx 0$ (as $\sigma(z) = 1/2$ iff $z = 0$). Hence, for all samples in $\mathcal{D}_{\text{nd}}$,

$$r(x, y_w) - r(x, y_l) \approx 0.$$

Taking expectation over $\mathcal{D}_{\text{nd}}$ gives:

$$\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}_{\text{nd}}}[r(x, y_w) - r(x, y_l)] \approx 0.$$

This holds even under underspecification of the BT reward scale, since it concerns the *difference* in rewards. As such, it does not impose restrictions on the reward function form, provided it satisfies equivalence relations, i.e., rewards are defined up to a prompt-dependent shift (Definition 1 in Rafailov et al. (2024b)). Consequently, the expected reward differences adhere to Proposition 2 *without* necessarily having BT-motivated DPO's implicit reward formulation. □

B.2 Proof of Lemma 5a and 5b

Proof. Let us rewrite the Bradley-Terry preference probability equation in terms of the DPO implicit rewards. The BT model specifies this probability of preferring y_w over y_l as:

$$P(y_w \succ y_l \mid x) = \sigma(r^*(x, y_w) - r^*(x, y_l)) \quad (28)$$

where the rewards can be rewritten in term of the DPO implicit rewards $\hat{r}_w$ and $\hat{r}_l$ in Eq. 29.

Eq. 29 holds for all $\forall(x, y, y') \in \mathcal{D}_{\text{pref}}$, since DPO does not distinguish between the nature of the true preference relations in estimating the true preference probabilities, assuming (y, y') appear as preferred and dispreferred responses respectively for any context x. Since this RHS of Eq. 31 is simply the sigmoided difference of implicit rewards assigned by DPO to estimate the true preference probabilities, it is straightforward to see from Proposition 2 that the RHS i.e., $\frac{\pi_{\theta^*}(y)\pi_{\text{ref}}(y')}{\pi_{\theta^*}(y')\,\pi_{\text{ref}}(y)} \to 1$ and $P(y_w \succ y_l \mid x) \sim \frac{1}{2}$, as per our definition of non-deterministic preferences. Since the reference model π_{ref} is assumed to have full support over the output space ($\text{supp}(\pi_{\text{ref}}) = \mathcal{Y}$) and is not updated and can be set to a uniform prior ($\pi_{\text{ref}} \sim \mathcal{U}(\mathcal{Y})$) (Xu et al., 2024), without losing any generality, this implies that $\frac{\pi_{\theta^*}(y)}{\pi_{\theta^*}(y')}$ must remain close to 1 to satisfy this constraint ($\frac{\pi_{\theta^*}(y)\,\pi_{\text{ref}}(y')}{\pi_{\theta^*}(y')\pi_{\text{ref}}(y)} \to 1$). In this case, the policy tends to underfit the preference distribution since the preference signals are weak and policy cannot distinguish between the preferred and the dispreferred response.

Similar to Azar et al. (2023)'s argument, we can argue here that in this case when true preference probabilities are $\sim \frac{1}{2}$, i.e., non-deterministic, DPO's empirical reward difference estimates actually tend toward 1 which leads to underfitting of the optimal policy π_{θ^*} during alignment. Indeed, in this case, the β parameter does not provide any additional regularization effect to prevent policy underfitting especially under finite data. This completes the proof of Lemma 5a. □

We can similarly prove Lemma 5b in the case when $|\mathcal{D}_{\text{nd}}| \ll N$ where N is assumed to be

$$P(y_w \succ y_l \mid x) = \sigma\left(\underbrace{\beta \log \frac{\pi_{\theta^*}(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)}}_{(\hat{r}_w)} - \underbrace{\beta \log \frac{\pi_{\theta^*}(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}}_{(\hat{r}_l)}\right) \tag{29}$$

$$= \sigma\left(\beta \log \left(\frac{\pi_{\theta^*}(y_w \mid x)\pi_{\text{ref}}(y_l \mid x)}{\pi_{\theta^*}(y_l \mid x)\pi_{\text{ref}}(y_w \mid x)}\right)\right) \tag{30}$$

$$= \sigma\left(\beta \log \left(\frac{\pi_{\theta^*}(y \mid x)\pi_{\text{ref}}(y' \mid x)}{\pi_{\theta^*}(y' \mid x)\pi_{\text{ref}}(y \mid x)}\right)\right), \quad \forall (x, y, y') \in \mathcal{D}_{\text{nd}} \subset \mathcal{D}_{\text{pref}} \tag{31}$$

finite. In this case, in a similar vein as Azar et al. (2023), there is more likelihood that DPO sigmoided reward difference estimates are 1, i.e., $r^*(x, y_w) - r^*(x, y_l) \to \infty$.

As such, from Eq. 28, it is straightforward to see that the DPO's implicit reward difference tends to infinity, regardless of the strength of the β parameter, as shown below:

$$\log \left(\frac{\pi_{\theta^*}(y \mid x)\pi_{\text{ref}}(y' \mid x)}{\pi_{\theta^*}(y' \mid x)\pi_{\text{ref}}(y \mid x)}\right) \to \infty$$

in Eq. 31. This implies that

$$\frac{\pi_{\theta^*}(y \mid x)\pi_{\text{ref}}(y' \mid x)}{\pi_{\theta^*}(y' \mid x)\pi_{\text{ref}}(y \mid x)} \to \infty.$$

B.3 Proof of Lemma 5c

Proof. Our proof follows the argumentation in Fisch et al. (2024). Assume for now that all preference samples $(y, y') \in \mathcal{D}_{\text{pref}}$, including non-deterministic preference pairs, are mutually exclusive. Then, for the DPO objective (Eq. 7 in Rafailov et al. (2024b)) to be minimized, each θ_y must correspond uniquely to y, where θ_y are the optimal parameters that minimize the DPO objective in each such disjoint preference pair. This implies that DPO objective over $\mathcal{D}_{\text{pref}}$ is convex in the set $\Lambda = \{\lambda_1, \ldots, \lambda_n\}$, where

$$\lambda_i = \beta \log \left(\frac{\pi_\theta(y_w^{(i)})\pi_{\text{ref}}(y_l^{(i)})}{\pi_\theta(y_l^{(i)})\pi_{\text{ref}}(y_w^{(i)})}\right), \quad \forall i \in \mathbb{N} \tag{32}$$

Now, consider the non-deterministic preference samples indexed by j, which belong to the set $\mathcal{D}_{\text{nd}} \subset \mathcal{D}_{\text{pref}}$. Let $j \in \{k+1, \ldots, N\}$ with the assumption $k \gg (N-k)$. Under the mutual exclusivity assumption, Eq. 32 must also hold true for the non-deterministic preference samples. Consequently, we can rewrite Eq. 32 as:

$$\lambda_j = \beta \log \left(\frac{\pi_\theta(y_w^{(j)} \mid x)\pi_{\text{ref}}(y_l^{(j)} \mid x)}{\pi_\theta(y_l^{(j)} \mid x)\pi_{\text{ref}}(y_w^{(j)} \mid x)}\right), \quad \forall j \in \mathcal{D}_{\text{nd}} \tag{33}$$

More specifically, for every j, the following holds at the limit for the DPO objective to converge:

$$\lim_{\lambda_j \to \infty} -\log\left(\sigma\left(\lambda_j\right)\right) = 0, \tag{34}$$

which implies that $\Lambda^* = \{\infty\}^N$ induces a set of global minimizers of the DPO objective that includes θ^* that are optimal for the set of non-deterministic preference samples, while inducing a parallel set of θ^* at convergence for deterministic samples.

Consequently, all global minimizers θ^* including those optimal on the non-deterministic samples must satisfy

$$\log \frac{\pi_\theta(y_w^{(j)})\pi_{\text{ref}}(y_l^{(j)})}{\pi_\theta(y_l^{(j)})\pi_{\text{ref}}(y_w^{(j)})} = \infty. \tag{35}$$

Since $0 < \pi_{\text{ref}}(y) < 1$ for all y, θ^* must satisfy

$$\frac{\pi_{\theta^*}(y_w^{(j)})}{\pi_{\theta^*}(y_l^{(j)})} = \infty, \tag{36}$$

implying $\pi_{\theta^*}(y_l^{(j)}) = 0$ and $\pi_{\theta^*}(y_w^{(j)}) > 0$ for all $i \in N$, given that $\pi_{\theta^*}(y_w^{(j)}) \leq 1$ for any $y_w^{(j)}$. Alternatively, let us define the complement of the aggregated representation of all the dispreferred responses $y_l^{(j)}$ i.e., $\phi(y_l)^c$, where $\phi(y_l)$ is the aggregation function. We thereby have:

$$\phi(y_l) = \{y \colon \exists j \in \mathbb{N} \text{ such that } y_l^{(j)} = y\}, \tag{37}$$

Under these conditions, it is clear that π_{θ^*} must assign the entire remaining probability mass to $\phi(y_l)$ as given below,

$$\pi_{\theta^*}(\mathcal{C}(y_l)) = \sum_{y \in \phi(y_l)} \pi_{\theta^*}(y) = 0 \tag{38}$$

$$\implies \pi_{\theta^*}(\phi(y_l)^c) = 1. \tag{39}$$

This completes the proof of Lemma 5c and thus Lemma 5.

□

B.4 Proof of Lemma 6

Proof. Let us first rewrite the e-DPO's distillation objective (Fisch et al., 2024) over the preference dataset $\mathcal{D}_{\text{pref}}$ and examine how the optimal policy π_θ behaves upon convergence of this objective. For simplicity of analysis, we only consider the pointwise reward based distillation loss in the e-DPO formulation without[10] considering reward ensembles. Note that the e-DPO objective does not require preference labels and can apply to any response pair.

[10]Note that in our empirical experiments we use the full e-DPO objective with a set of three reward models.

$$\mathcal{L}_{\text{distill}}(r^*, \pi_\theta) = \mathbb{E}_{(x,y_1,y_2)\sim\mathcal{D}_{\text{pref}}}\left[\left(r^*(x,y_1) - r^*(x,y_2) - \beta\log\frac{\pi_\theta(y_1 \mid x)\pi_{\text{ref}}(y_2 \mid x)}{\pi_\theta(y_2 \mid x)\pi_{\text{ref}}(y_1 \mid x)}\right)^2\right] \tag{40}$$

$$= \mathbb{E}_{(x,y_1,y_2)\sim\rho}\left[\left(r^*(x,y_1) - r^*(x,y_2) - \beta\log\frac{\pi_\theta(y_1 \mid x)}{\pi_\theta(y_2 \mid x)} + \beta\log\frac{\pi_{\text{ref}}(y_1 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)}\right)^2\right]. \tag{41}$$

As π_θ converges to the optimal policy π_{θ^*}, the distillation objective $\mathcal{L}_{\text{distill}}$ should ideally approach zero and can be expressed as,

$$\lim_{\pi_\theta \to \pi_{\theta^*}} \mathcal{L}_{\text{distill}}(r^*, \pi_\theta) = 0$$

With some slight algebraic rearrangement and substituting the optimal policy π_{θ^*} for π_θ at convergence, we get:

$$r^*(x, y_1) - r^*(x, y_2) = \beta \log \frac{\pi_{\theta^*}(y_1 \mid x)}{\pi_{\theta^*}(y_2 \mid x)} - \beta \log \frac{\pi_{\text{ref}}(y_1 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)}. \quad (42)$$

Since (y_1, y_2) represents *any* response pair without a preference label, we can substitute them with $(y, y') \in \mathcal{D}_{\text{nd}} \subset \mathcal{D}_{\text{pref}}$, where $y = y_w$ and $y' = y_l$. Without losing any generality, we can now rewrite the above equation as,

$$r^*(x, y) - r^*(x, y') \quad (43)$$
$$= \beta \log \frac{\pi_{\theta^*}(y \mid x)}{\pi_{\theta^*}(y' \mid x)} - \beta \log \frac{\pi_{\text{ref}}(y \mid x)}{\pi_{\text{ref}}(y' \mid x)}$$
$$= \beta \log \left(\frac{\pi_{\theta^*}(y \mid x) \pi_{\text{ref}}(y' \mid x)}{\pi_{\theta^*}(y' \mid x) \pi_{\text{ref}}(y \mid x)} \right) \quad (44)$$

Now, recall from our proof of Lemma 5b where we show that the RHS term of the above equation $\log \left(\frac{\pi_{\theta^*}(y|x)\pi_{\text{ref}}(y'|x)}{\pi_{\theta^*}(y'|x)\pi_{\text{ref}}(y|x)} \right) \to \infty$, which implies that $\frac{\pi_{\theta^*}(y|x)\pi_{\text{ref}}(y'|x)}{\pi_{\theta^*}(y'|x)\pi_{\text{ref}}(y|x)} \to \infty$. Indeed, when $|\mathcal{D}_{\text{nd}}| \ll N$ where N is finite, unregularized scalar reward estimates of the true preference probabilities can in fact grow exceedingly large in the absence of any other regularization parameters, since β by its own does not provide enough regularization as shown in our proof of Lemma 5a. Interestingly, similar arguments have also been made in previous works (Azar et al., 2023). The rest of this proof follows the same argumentation starting Eq. 35 assuming $0 < \pi_{\text{ref}}(y) < 1$.

This completes the proof of Lemma 6. □

B.5 Gradient Derivation of the Focal-Softened Log-Odds Unlikelihood Loss

In this section, we derive and analyze DRDO loss gradient and offer insights into how to compared supervised alignment objectives such as DPO (Rafailov et al., 2024b). Note that we do not analyze the reward distillation component here since it does not directly interact with the focal-softened contrastive log-"unlikelihood" term in training and since it is naturally convex considering its a squared term. Let us first rewrite our full DRDO loss, as:

$$\mathcal{L}_{\text{kd}}(r^*, \pi_\theta) = \mathbb{E}_{(x,y_1,y_2)\sim\mathcal{D}_{\text{pref}}} \quad (45)$$
$$\Bigg[\underbrace{(r^*(x, y_1) - r^*(x, y_2) - (\hat{r}_1 - \hat{r}_2))^2}_{\text{Reward Difference}}$$
$$\underbrace{- \alpha(1 - p_w)^\gamma \log \left(\frac{\pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)} \right)}_{\text{Contrastive Log-"unlikelihood"}} \Bigg],$$

where $p_w = \sigma(z_w - z_l) = \frac{1}{1+e^{-(z_w - z_l)}}$ and quantifies the student policy's confidence in correctly assigning the preference from $z_w = \log \pi_\theta(y_w \mid x)$ and $z_l = \log \pi_\theta(y_l \mid x)$, or the log-probabilities of the winning and losing responses, respectively.

Without the expectation, consider only the focal-softened log-odds unlikelihood loss given by:

$$-\alpha \cdot (1 - p_w)^\gamma \cdot \log \left(\frac{\pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)} \right), \quad (46)$$

Taking the gradient of this term with respect to the model parameters θ and using $\sigma'(x) = \sigma(x)\,(1 - \sigma(x))$, we derive:

$$\nabla_\theta \mathcal{L}_{\text{kd}} = \alpha\gamma(1 - p_w)^{\gamma-1} p_w (1 - p_w)(\nabla_\theta z_w - \nabla_\theta z_l) \cdot \quad (47)$$
$$\log \left(\frac{\pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)} \right) -$$
$$\alpha(1 - p_w)^\gamma \left(\frac{\nabla_\theta \pi_\theta(y_w \mid x)}{\pi_\theta(y_w \mid x)} + \frac{\nabla_\theta \pi_\theta(y_l \mid x)}{1 - \pi_\theta(y_l \mid x)} \right).$$

$$\nabla_\theta \mathcal{L}_{\text{kd}} = \alpha\gamma(1 - p_w)^{\gamma} p_w (\nabla_\theta z_w - \nabla_\theta z_l) \cdot \quad (48)$$
$$\log \left(\frac{\pi_\theta(y_w \mid x)}{1 - \pi_\theta(y_l \mid x)} \right) -$$
$$\alpha(1 - p_w)^\gamma \left(\underbrace{\frac{\nabla_\theta \pi_\theta(y_w \mid x)}{\pi_\theta(y_w \mid x)}}_{\text{increase } \pi_\theta(y_w|x)} + \underbrace{\frac{\nabla_\theta \pi_\theta(y_l \mid x)}{1 - \pi_\theta(y_l \mid x)}}_{\text{decrease } \pi_\theta(y_l|x)} \right).$$

While this above equation might appear rather cumbersome, notice that in preference learning in language models, the output token space $\mathcal{Y}$ is exponentially large. Additionally, in typical bandit settings, we consider the entire response itself as the action (summation of log probabilities). Since the modulating term $0 \leq (1-p_w)^\gamma \leq 1$ and α is typically small ~ 0.1 (Lin et al., 2018; Yi et al., 2020) compared to gradients appearing in likelihood terms appearing above, we can conveniently ignore the first term for the gradient analysis.

Simplifying the above equation, we get

$$-\alpha(1-p_w)^\gamma \left(\underbrace{\frac{\nabla_\theta \pi_\theta(y_w \mid x)}{\pi_\theta(y_w \mid x)}}_{\text{increase } \pi_\theta(y_w|x)} + \underbrace{\frac{\nabla_\theta \pi_\theta(y_l \mid x)}{1-\pi_\theta(y_l \mid x)}}_{\text{decrease } \pi_\theta(y_l|x)} \right). \tag{49}$$

We can now draw some insights and direct comparisons of our approach with Direct Preference Optimization (DPO) (Rafailov et al., 2024b). As in most contrastive preference learning gradient terms (Rafailov et al., 2024b; Hong et al., 2024; Xu et al., 2024; Meng et al., 2024; Ethayarajh et al., 2024), the term $\frac{\nabla_\theta \pi_\theta(y_w|x)}{\pi_\theta(y_w|x)}$ in Eq. 49 amplifies the gradient when $\pi_\theta(y_w \mid x)$ is low, driving up the likelihood of the preferred response y_w. Similarly, $\frac{\nabla_\theta \pi_\theta(y_l|x)}{1-\pi_\theta(y_l|x)}$ penalizes overconfidence in incorrect completions y_l when p_w is low, encouraging the model to hike preferred response likelihood while discouraging dispreferred ones.

The key insight here is that the modulating term, $(1-p_w)^\gamma$, strategically amplifies corrections for difficult examples where the probability p_w of the correct (winning) response is low. Intuitively, unlike DPO's fixed β that is applied across the whole training dataset, this modulating term amplifies gradient updates when preference signals are weak ($p_w \approx 0.5$) and tempering updates when they are strong ($p_w \approx 1$), thus ensuring robust learning across varying preference scenarios. Intuitively, when ($p_w \approx 1$), the model is already confident of its decision since p_w remains high, indicating increased model confidence for deterministic preferences. In contrast, when p_w is small, $(1-p_w)$ remains near 1, and the term $(1-p_w)^\gamma$ retains significant magnitude, especially for larger values of γ. This allows π_θ in DRDO to learn from both deterministic and non-deterministic preferences, effectively blending reward alignment with preference signals to guide optimization.

For a deeper intuition, consider the case where

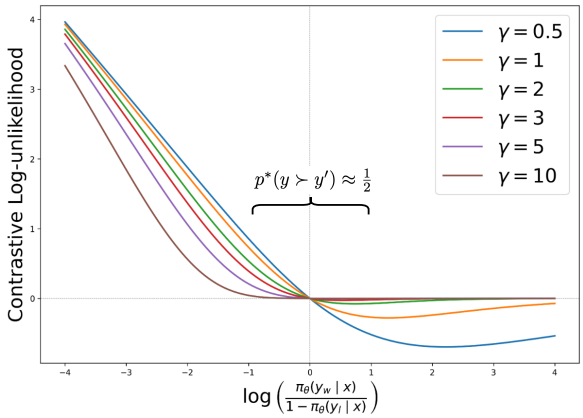


Figure 4: Illustration of the DRDO preference loss as a function of the log-unlikelihood ratio across various values of γ, the focal modulation parameter.

the true preference probabilities from $p^*(x, y) \sim \frac{1}{2}$. In this case, since non-deterministic preference samples are typically low, from Proposition 2, DPO would assign zero difference in its implicit rewards, especially for finite preference data. Then as Lemma 5(a) suggests, DPO gradients would effectively be nullified, regardless of β since the its reward difference range $\in (-\infty, +\infty)$. In this case as π_θ cannot distinguish between the preference pair and, in turn, effectively misses out on the preference information for such samples.

On the other hand, for $|\mathcal{D}_{\text{nd}}| \ll N$, if π_θ in DPO estimates the true preference close to 1 where $\hat{p} = 1$, as Lemma 5(b) suggests, the empirical policy would assign very high probabilities to tokens that do not even appear in the data. This leads to a surprising combination of both underfitting and overfitting, except the overfitting here results in DPO policy generating tokens that are irrelevant to the context.

However, as Lemma 6 suggests, e-DPO does not directly face this limitation *during* training, but the degeneracy manifests upon convergence. Under the same conditions, the DRDO loss still operates at a sample level because if the DRDO estimate of p^* (via p_w) is close to its true value of $\sim \frac{1}{2}$, the modulating factor ensures that gradients do not vanish. This allows DRDO to continue learning from such samples until convergence when winning and losing probabilities are pushed further apart (Fig. 4). Intuitively, the $\log\left(\frac{\pi_\theta(y_w|x)}{1-\pi_\theta(y_l|x)}\right)$ term is minimized precisely under this condition where the modulating term is also close to zero.

Finally, Table 3 provides pseudocode for the DRDO algorithm.

DRDO Algorithm

Input: Preference dataset $\mathcal{D}_{\text{pref}} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$, initialized policy model with reward head $\pi_{\theta,\theta'} \leftarrow \text{SFT}(\theta) \oplus r_{\theta'}$.
Output: Optimized model parameters θ in policy π_θ.

1. Train Oracle r_ϕ with loss $\mathcal{L}_{\mathcal{O}}(r_\phi, \mathcal{D}_{\text{pref}})$ (see Eq. 5).
2. For $t = 1, \ldots, T$:
   (a) For each $(x^{(i)}, y_w^{(i)}, y_l^{(i)})$ in $\mathcal{D}_{\text{pref}}$:
      i. Compute $r_1^* = r_\phi(x^{(i)}, y_w^{(i)})$ and $r_2^* = r_\phi(x^{(i)}, y_l^{(i)})$.
      ii. Compute $\hat{r}_1 = r_{\theta'}(x^{(i)}, y_w^{(i)})$ and $\hat{r}_2 = r_{\theta'}(x^{(i)}, y_l^{(i)})$.
      iii. Compute knowledge distillation loss:

$$\mathcal{L}_{\text{kd}}(r^*, \pi_\theta) = \mathbb{E}_{(x^{(i)}, y_w^{(i)}, y_l^{(i)}) \sim \mathcal{D}_{\text{pref}}} \Bigg[\underbrace{(r_1^* - r_2^* - (\hat{r}_1 - \hat{r}_2))^2}_{\text{Reward Difference}} - \underbrace{\alpha(1 - p_w)^\gamma \log \left(\frac{\pi_\theta(y_w^{(i)} \mid x^{(i)})}{1 - \pi_\theta(y_l^{(i)} \mid x^{(i)})} \right)}_{\text{Contrastive Log-"unlikelihood"}} \Bigg].$$

      iv. Update $\pi_{\theta,\theta'}$ using $\mathcal{L}_{\text{kd}}$.
3. **Return:** Aligned policy π_θ.

Table 3: DRDO Algorithm steps. We start off with the preference dataset and an SFT-trained policy initialized with an additional linear head parameterized by θ'. Once our oracle is trained, we compute both estimated rewards for each response (y) from the initial policy ($\hat{r}$) as well as from the oracle (r^*). We then use $\mathcal{L}_{\text{kd}}$ to update both θ and θ' in π_θ resulting in our DRDO aligned policies.

C Further Notes on Experimental Setup

We provide the following additional explanatory notes regarding the experimental setup:

- For non-deterministic and nuanced preferences, note that although we fine-tune all approaches (including DRDO) on https://huggingface.co/datasets/CarperAI/openai_summarize_tldr, every baseline we use is policy-aligned with only the training data within $\mathcal{D}_{all}$, $\mathcal{D}_{hc,he}$ and $\mathcal{D}_{\ell c,\ell e}$ for a direct comparison.
- Pal et al. (2024) assume non-determinism of preferences to be correlated to edit-distances between pairwise-samples, but we do not make sure assumptions and consider both the true (oracle) rewards and edit-distances between pairs to verify the robustness of our method. Pal et al. (2024)'s theoretical framework brings insights on DPO's suboptimality assumes small edit distance between pairwise samples and they empirically show this primarily for math and reasoning based tasks. In contrast, our evaluation framework is more general in the sense that we consider both the oracle reward difference as well as edit distance in addressing DPO's limitations in learning from non-deterministic preferences and we evaluate on a more diverse set of prompts apart from math and reasoning tasks.
- For reward distillation w.r.t to model size, the version of Ultrafeedback we used can be found at https://huggingface.co/datasets/argilla/ultrafeedback-binarized-preferences.
- For SFT on both experiments, use the TRL library implementation (https://huggingface.co/docs/trl/en/sft_trainer) for SFT training on all initial policies for all baselines.

C.1 DRDO and e-DPO Specifics

Although DRDO requires an explicit reward Oracle $\mathcal{O}$, we fix only one model (based on parameter size and model family) for each experiment. We use Phi-3-Mini-4K-Instruct and OPT 1.3B causal models initialized with a separate linear reward head while retaining the language modeling head weights.[11] Fixing the size of $\mathcal{O}$ allows us to evaluate the extent of preference alignment to smaller models, as in classic knowledge distillation (Gou et al., 2021). To reproduce the e-DPO (Fisch et al., 2024) baseline, we train three reward models using standard RLHF

[11] Similar to Yang et al. (2024), we found better generalization in reward learning when our Oracle reward learning loss is regularized with the SFT component (second term in Eq. 1) with an α of 0.01.

with the mentioned base models but with different random initialization on the reward heads.

C.2 Choice of Evaluation Datasets

As mentioned in Sec. 5, our experiments were designed to balance robustness and thoroughness with research budget constraints, and to account for properties of the task being evaluated relative to the data. The rationale for evaluating DRDO's robustness to non-deterministic or ambiguous preferences is straightforward: the TL;DR dataset is annotated with human confidence labels, which enables the creation of the $\mathcal{D}_{hc,he}$ and $\mathcal{D}_{\ell c,\ell e}$ splits requires to test the different non-determinism settings.

Testing the effect of model size on reward distillation does not require human confidence labels, and as the TL;DR training data size is 1.3M rows, testing distillation across 5 models became computationally intractable given available resources. Thus we turned to Ultrafeedback, which at a train size of 61.1k rows made this experiment much more feasible. Additionally, Ultrafeedback is rather better suited to the preference distillation problem because Ultrafeedback was specifically annotated and cleaned (Cui et al., 2024), for the purposes of evaluating open source models for distillation. Ultrafeedback also provides a high quality dataset with GPT-4 or scores on both straighforward summarization instruction following *and* multiple preference dimensions such as honesty, truthfulness, and helpfulness. Previous work like Zephyr (Tunstall et al., 2023) utilize this dataset for evaluating preference distillation. However, their method is more of a data-augmentation method and does not provide any novel distillation algorithms for preferences since they use the DPO loss but with cleaner and diverse feedback data. They only test student models of $\sim$7B parameters, which are still relatively large models that require additional compute for training without including some form of PEFT. As such, Ultrafeedback is best suited to evaluate distillation-based methods and our novel contribution in this area included evaluation of smaller distilled models.

C.3 TL;DR Summarization Dataset Splits

Table 4 shows the mean confidence and normalized edit distance statistics in the TL;DR dataset which we used to compute the deterministic and non-determinisitic splits. $\mathcal{D}_{all}$ represents the full data (Stiennon et al., 2020). $\mathcal{D}_{hc,he}$ and $\mathcal{D}_{\ell c,\ell e}$ represent subsets created by splitting at the 50th percentile of human labeler confidence and edit distance values. The range of labeler confidence values for the full training data range is $[1, 9]$. Additionally, segmenting the training data based on the *combination* of confidence and edit distance thresholds do not make the splits roughly half of the full training data. This is because there are many samples that do not simultaneously satisfy the 50th percentile threshold for each metric. In reality, this choice is intentional to more robustly evaluate DRDO under more difficult preference data settings. Additionally, previous theoretical as well as empirical work (Pal et al., 2024) has shown that supervised methods like DPO fail to learn optimal policies when the token-level similarity is high in the preference pairs, especially in the beginning of the response. Therefore, we apply this combined thresholding for all our experiments on TL;DR summarization dataset.

D Preference Likelihood Analysis

Fig. 5 shows preference optimization method performance vs. training steps, according to Bradley-Terry (BT) implicit reward accuracies (Fig. 5a), Oracle reward advantage (Fig. 5b), preferred log-probabilities (Fig. 5c) and dispreferred log-probabilities (Fig. 5d). Although all baselines show roughly equal performance in increasing the likelihood of preferred responses (y_w), DRDO is particularly efficient in penalizing dispreferred responses (y_l) as Fig. 5d suggests.

E Hyperparameters

We only use full-parameter training for all our policy models. We train all our student policies for 1k steps with an effective batch size of 64, after applying an gradient accumulation step of 4. Specifically, both OPT series and Phi-3-Mini-4K-Instruct model were optimized using DeepSpeed ZeRO 2 (Rasley et al., 2020) for faster training. All models were trained on 2 NVIDIA A100 GPUs, except for certain runs that were conducted on an additional L40 GPU. For the optimizer, we used AdamW (Loshchilov et al., 2017) and paged AdamW (Dettmers et al., 2024) optimizers with learning rates that were linearly warmed up with a cosine-scheduled decay. For both datasets, we filter for prompt and response pairs that are $<$ 1024 tokens after tokenization. This allows the policies enough context for coherent generation. Apart from keeping compute requirement reason-

Table 4: Mean confidence and normalized edit distance statistics our TL;DR preference generalization experiment.

Split	Conf (Mean)	Edit Dist (Mean)	Train	Validation
$\mathcal{D}_{all}$	5.01	0.12	44,709	86,086
$\mathcal{D}_{hc,he}$	7.31	0.15	10,136	1,127
$\mathcal{D}_{\ell c,\ell e}$	2.67	0.09	10,000	1,112

Table 5: Model Configuration and Full set of hyperparameter used for DRDO Training

Parameter	Default Value
learning_rate	5e-6
lr_scheduler_type	cosine
weight_decay	0.05
optimizer_type	paged_adamw_32bit
loss_type	DRDO
per_device_train_batch_size	12
per_device_eval_batch_size	12
gradient_accumulation_steps	4
gradient_checkpointing	True
gradient_checkpointing_use_reentrant	False
max_prompt_length	512
max_length	1024
max_new_tokens	256
max_steps	20
logging_steps	5
save_steps	200
save_strategy	no
eval_steps	5
log_freq	1
α	0.1
γ	2

able, this avoids degeneration during inference since we force the model to only generate upto 256 new tokens not including the prompt length in the maximum token length.

DRDO training For our DRDO approach, we sweep over $\alpha \in \{.1, 1\}$ and $\gamma \in \{0, 1, 2, 5\}$ but found the most optimal combination to be $\alpha = 0.1$ and $\gamma = 2$, since a higher γ tends to destabilize training due to the larger penalties induced on DRDO loss. This is consistent with optimal γ values found in the literature, albeit for different tasks (Yi et al., 2020; Lin et al., 2018). For Oracle trained for DRDO, we use the same batch size as for policy training with a slightly larger learning rate of 1e-5 and train for epoch. For consistency, we use a maximum length of 1024 tokens after filtering for pairs with prompt and responses < 1024 tokens. For all SFT training, we use the TRL library[12] with a learning rate of 1e-5 with a cosine scheduler and 100 warmup steps. Table 5 provides a full list of model configurations and hyperparameters used during training of DRDO models.

DPO and e-DPO For the DPO baselines, we found the implementation in DPO Trainer[13] For optimal parameter selection, we sweep over $\beta \in \{.1, 0.5, 1, 10\}$ but we found the default value of β = 0.1 to be most optimal based on the validation sets during training. Also, we found the default learning rate of 5e-6 to be optimal after validation runs. For e-DPO, we restrict the number of reward ensembles to 3 but use the same Oracle training hyperparameters mentioned above.

PPO baseline For PPO, we train a Phi-3 Instruct reward model (RM) using our Oracle reward modeling loss (Eq.1) and the policy based on the implementation.[14] Due to PPO's extensive compute demands, we could only run evaluate this baseline on one experiment—training on Ultrafeedback and evaluating on the Alpaca Eval 2.0 benchmark. As such, we trained this baseline with LoRA (Low-Rank Adaptation of Large Language Models), where LoRA $\alpha = 16$, LoRA dropout = 0.05 and a LoRA R of 8 was used in training with the PEFT[15] and `bitsandbytes`[16] library to load our models in 4-bit quantization for more cost-efficient training. In particular, we train the PPO policy on Oracle-assigned rewards for 4,000 batches over 2 epochs using a mini-batch size of 4, gradient accumulation of 2, and an effective batch size of 8. Responses are sampled using top-$p = 1.0$ and constrained to 256–512 tokens via a `LengthSampler`; queries are truncated to 1,024 tokens. Learning rates is 3e−6.

[12] https://huggingface.co/docs/trl/en/sft_trainer

[13] We found https://huggingface.co/docs/trl/main/en/dpo_trainer to be the most stable and build off most of our DRDO training pipline and configuration files based on their trainer.

[14] https://github.com/huggingface/trl/blob/main/examples/scripts/ppo/ppo.py

[15] https://huggingface.co/docs/peft/index

[16] https://huggingface.co/docs/transformers/main/en/quantization/bitsandbytes

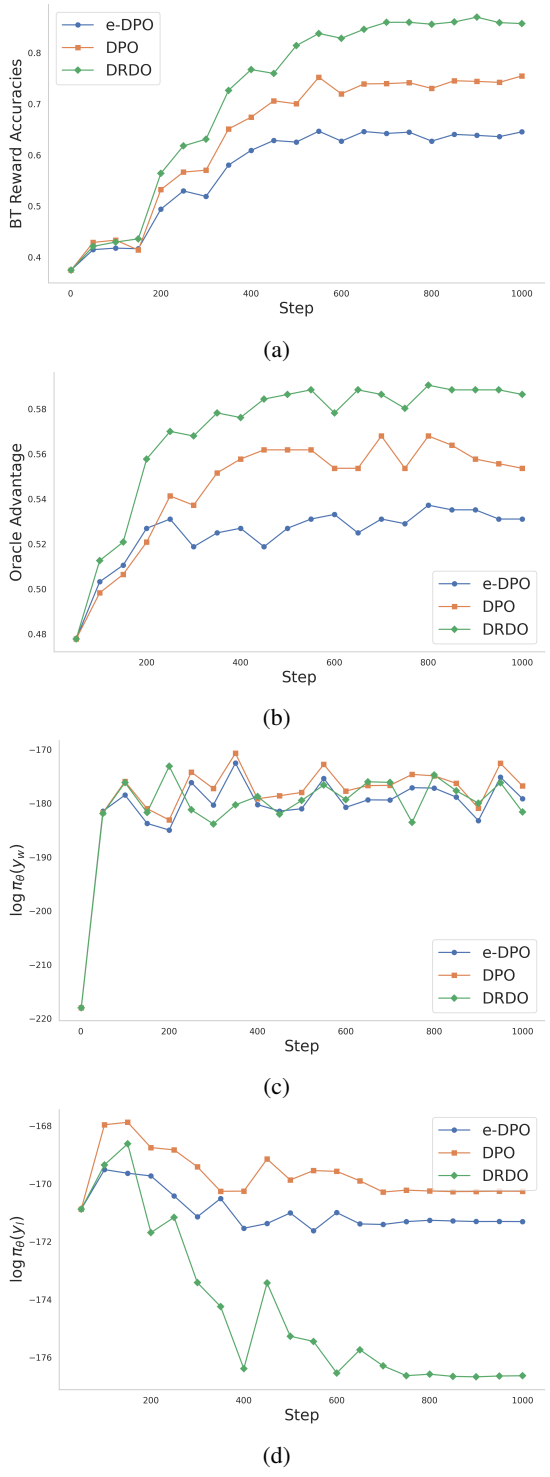


Figure 5: Top: DRDO performance evolution during OPT 1.3B training compared to DPO and e-DPO on the evaluation set of Ultrafeedback (Cui et al., 2024), and randomly sampled generations to compute the reward advantage against the preferred reference generations.

F Ablation Studies/Additional Experiments

F.1 Additional results on Alpaca Eval 2.0 per Dataset

Distributional Analysis Across Evaluation Datasets. Table 6 shows distribution of win-rates of all baselines including PPO across five evaluation subsets in AlpacaEval 2.0—SELFINSTRUCT, HELPFUL_BASE, VICUNA, KOALA, and OASST—which vary in prompt diversity, linguistic complexity, and alignment challenges. DRDO achieves the highest win rate on SELFINSTRUCT at 20.24%, substantially outperforming PPO (14.17%) and DPO (13.36%). On VICUNA, DRDO reaches 18.42%, followed by e-DPO at 15.79% and PPO at 13.16%, showing DRDO's strength in handling conversational prompts sourced from user interactions. The performance on KOALA further reflects this trend, where DRDO achieves 17.31%, surpassing PPO (14.74%) and e-DPO (13.46%). While PPO slightly outperforms DRDO on OASST (14.75% vs. 13.11%), DRDO still ranks among the top performers across all five datasets. In contrast, supervised fine-tuning (SFT) trails behind on most datasets, particularly on OASST (6.01%) and VICUNA (6.58%).

More importantly, despite variations in win rates across individual datasets, DRDO emerges as the most consistently strong method. Its superior performance on open-ended benchmarks like SELFINSTRUCT and VICUNA suggests that it is particularly well suited to learning from both deterministic and noisy preference signals. This generalization ability is especially valuable for real-world alignment tasks, where preferences may be uncertain or under-specified. For example, unlike TL:DR or Ultrafeedback where confidence labels or a high-capacity judge-based assigned rewards are accessible during training, real world data in carefully curated benchmarks like AlpacaEval may not contain indicators to get non-determinsitic labels for training. However, DRDO objective operates dynamically (with the contrastive "preference" component) and is agnostic to underlying preference "labels"[17] as its improved results over multiple data-distributions on AlpacaEval show. Similarly, while e-DPO shows a spike on VICUNA, likely due to more consistent oracle reward in the confidence set of 3, DRDO maintains strong performance across

[17] By labels, we mean datasets created explicitly to reflect non-deterministic preference samples like $\mathcal{D}_{\ell c, \ell e}$

all settings. These results suggest that DRDO effectively balances robustness to all types of preferences, whether clearly determinsitic or noisy or non-deterministic in preference modeling, outperforming both distillation-based offline techniques like e-DPO, direct approaches like DPO and on-policy RL baselines like PPO in diverse evaluation settings.

More importantly, although PPO is competitive especially on OASST, these results suggest that DRDO more effectively leverages the oracle reward model while being fully *offline* in contrast to PPO that requires online (on-policy) sampling which drastically increases compute required. This makes DRDO more compute-efficient in that it learns human-preferences completely offline and still performs well on a wide range of data distributions in Alpaca Eval 2.0.

F.2 Ablations on reward distillation and contrastive log-unlikelihood

For a more robust evaluation using a much more high-capacity oracle (GPT-4o), we randomly sampled 40 prompts from the CNN/Daily Mail test set and compute win-rates and reward margins of samples generated with a top-p of 0.8 and $T = 0.7$ for all baselines vs. DRDO policies trained on Reddit TL;DR. We use the Phi-3-Mini-4K-Instruct model for this experiment. We include the IPO baseline (Azar et al., 2024) with $\beta = 0.1$ (or τ in their paper), the baselines without the distillation (shown as DRDO (-R)), and without the contrastive component (shown as DRDO (-C)). Additionally, we include the baseline where reward distillation term in DRDO is replaced by the DPO loss but keep the contrastive log-unlikelihood component (shown as DPO (+C)). Table 7 shows results of this experiment. Note that due to compute constraints, we only train these policies from the SFT checkpoint for 500 steps with an effective batch size of 128. For the reward estimates using GPT-4o, we only add one additional condition to the prompt shown in Fig. 5 to get scalar rewards (between 0 and 1). Fig. 9 provides the prompt format used for this experiment.

We see that in all cases, full DRDO is the clear winner in terms of win rate. We also see that in this sample the reward distillation component contributes slightly more to the overall performance than the contrastive log-unlikelihood loss, but given the small sample size this is not a significant difference; DRDO's performance can be attributed to a combination of both. This is reinforced by DRDO's 80-20 performance against contrastive log-unlikelihood combined with DPO loss instead of DRDO reward distillation.

F.3 Extent of Out-of-distribution data

In order to comprehensively evaluate DRDO's performance against baseline methods under increasing out-of-distribution (OOD) conditions, we randomly sample 1,000 prompts from the CNN/DailyMail dataset and segment these prompts into bins of 50 tokens based on prompt-token counts, spanning the full range of token lengths. Since the CNN/DailyMail dataset represents a previously unseen input distribution, this evaluation effectively measures the OOD generalization capabilities of the policies as prompt lengths (and corresponding news article lengths) increase. For this automatic evaluation, we use GPT-4o as a high-capacity judge, consistent with Sec. 5 (prompt used is shown in Fig. 7). For response sampling, we use top-p of 0.8 and temperature of 0.7 for DRDO and all baselines. The trends in Fig. 6 suggest that as OOD composition (with prompt-token lengths as proxy) increases, on average DRDO policies tend to have relatively larger win-rates compared to shorter prompts over baselines like DPO and e-DPO. In contrast, we find that DPO and e-DPO win-rates tend to decrease with increase in prompt-lengths.

G Win-Rate Evaluation Prompt Formats

Fig. 7 and Fig. 8 show the prompt format used for GPT-4o evaluation of policy generations compared to human summaries provided in the evaluation data of Reddit TL;DR (CNN Daily Articles). Fig. 7 specifically provides the human-written summaries as reference in GPT's evaluation of the baselines. In contrast, Fig. 8 shows the prompt that was used to evaluate policy generated summaries in direct comparison to human-written summaries. Note that in both prompts, we swap order of provided summaries to avoid any positional bias in GPT-4o's automatic evaluation.

H Computational Efficiency

e-DPO requires training reward ensembles to form a confidence set for training the policy. In our experiments, we use 3 reward models to construct this set which makes it roughly thrice as expensive as DRDO training. Like DPO, e-DPO requires a

	SELFINSTRUCT	HELPFUL_BASE	VICUNA	KOALA	OASST
SFT	$12.55_{\pm 2.11}$	$7.81_{\pm 2.38}$	$6.58_{\pm 2.86}$	$10.90_{\pm 2.50}$	$6.01_{\pm 1.76}$
DPO	$13.36_{\pm 2.17}$	$12.50_{\pm 2.93}$	$7.89_{\pm 3.11}$	$8.33_{\pm 2.22}$	$10.38_{\pm 2.26}$
E-DPO	$11.74_{\pm 2.05}$	$9.38_{\pm 2.59}$	$15.79_{\pm 4.21}$	$13.46_{\pm 2.74}$	$12.02_{\pm 2.41}$
PPO	$14.17_{\pm 2.22}$	$\mathbf{13.28}_{\pm 3.01}$	$13.16_{\pm 3.90}$	$14.74_{\pm 2.85}$	$\mathbf{14.75}_{\pm 2.63}$
DRDO	$\mathbf{20.24}_{\pm 2.56}$	$10.94_{\pm 2.77}$	$\mathbf{18.42}_{\pm 4.48}$	$\mathbf{17.31}_{\pm 3.04}$	$13.11_{\pm 2.50}$

Table 6: Dataset-wise win rates on AlpacaEval 2.0 for baselines trained on Ultrafeedback. DRDO performs consistently well, especially on SELFINSTRUCT (20.24%) and VICUNA (18.42%). Note that although PPO is competitive especially on OASST, these results suggest that DRDO more effectively leverages the oracle reward model while being fully *offline* in contrast to PPO that requires online (on-policy) sampling which drastically increases compute required. This makes DRDO more compute-efficient learns human-preferences completely offline and still performs well on a wide range of data distributions in Alpaca Eval 2.0

Comparison	**WR A (%)**	**WR B (%)**	**Reward A**	**Reward B**	**Margin A**	**Margin B**
DRDO vs. DRDO (-R)	85.0	15.0	$0.24_{\pm 0.26}$	$0.07_{\pm 0.08}$	$0.22_{\pm 0.22}$	$0.09_{\pm 0.04}$
DRDO vs. DRDO (-C)	90.0	10.0	$0.25_{\pm 0.23}$	$0.05_{\pm 0.07}$	$0.23_{\pm 0.22}$	$0.06_{\pm 0.03}$
DRDO vs. IPO	65.0	35.0	$0.21_{\pm 0.17}$	$0.21_{\pm 0.24}$	$0.09_{\pm 0.08}$	$0.16_{\pm 0.16}$
DRDO vs. DPO (+C)	80.0	20.0	$0.18_{\pm 0.16}$	$0.14_{\pm 0.15}$	$0.11_{\pm 0.08}$	$0.22_{\pm 0.21}$

Table 7: Full DRDO policies (denoted "A") compared against various baselines (denoted "B"), including DRDO without reward distillation and DRDO without contrastive log-unlikelihood loss. Win-rates (WR) are computed using average of reward comparison for each sample and then averaged. Margins are computed using the difference of rewards.

separate reference model to be kept in memory, further increasing compute requirements. DRDO, on the other hand, only requires a trained oracle for distillation. The expected oracle rewards can be precomputed once and a separate reference model does not need to be kept in memory (as shown in Eq. 2). During training, DRDO does require one additional linear head on top of the base LM to predict the reward estimates. This adds a negligible 0.003% more trainable parameters (relative to the language modeling head of base LM Phi-3-Mini-4K-Instruct). During inference, DRDO trained policies do not require this head.

I DRDO vs. Pluralistic Preferences

In certain circumstances, non-deterministic preferences, as reflected in low labeler confidence or equal rewards, could be a consequence of innate pluralistic tendencies of human preferences. However, DRDO is not motivated directly by pluralistic preferences, where there are multiple annotations (or preferences) for a single (x, y_1, y_2), but by the diversity of preference strength for paired samples. Typically, pluralistic approaches require multiple reward models, such as reward soups (Ramé et al., 2023), e-DPO (Fisch et al., 2024), MaxMin-RLHF (Chakraborty et al.), or conditioned policy (Wang et al., 2024b) to model such preferences. This is computationally expensive and *assumes rewards over multiple dimensions can be linearly interpolated*. We do not make any such assumptions. Our main argument as stated in Sec. 3 is that non-deterministic preferences likely constitute a non-trivial amount of paired samples in popular preference datasets and as such, DRDO provides an efficient alignment method under such conditions. Our only strong assumption in the modeling is that the Oracle reward model, given sufficient data, should reasonably approximate human preferences using any standard reward-modeling approach. Furthermore, given such an Oracle, we directly regress on the rewards and, unlike e-DPO, do not need to find additional optimal parameters like β in the regression or confidence set in policies or reward model ensembles. Thus, our DRDO approach does not need to learn a variety of models each unique to specific viewpoints expressed in the data, and thus our results that best the competitor baselines reflect that we are able to fit better to non-deterministic preferences while still maintaining an ability to fit to deterministic preferences and the data distribution at large.

J Sensitivity of DRDO's γ vs. DPO's β w.r.t. KL-divergence from SFT model

We ran an additional experiment to compare the sensitivity of model-specific hyperparameters (DRDO's γ vs. DPO's β). Keeping α as 0.1 for all DRDO policies, we compute the KL-divergence during training on sampled generations on 40 randomly sampled evaluation prompts in the held-out set of Ultrafeedback with top-p of 0.8 and temperature of 0.7 with various γ values in DRDO ($\alpha = 0.1$) and with different KL-β values in

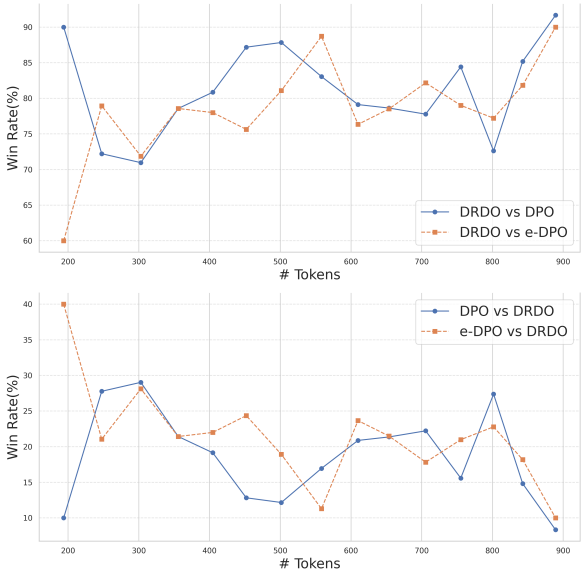


Figure 6: Comparison of win-rates as a function of the extent of out-of-distribution (OOD) data on the CNN daily article dataset. Win-rates (y-axis) of DRDO vs DPO and e-DPO (top) and competitor win-rates (bottom) are plotted against the increasing prompt lengths (number of tokens) over 1000 randomly sampled prompts for evaluation. DRDO is more robust to OOD settings, on average, compared to baselines like DPO and e-DPO as seen in the upward trend in win-rates over prompt-tokens.

Summarization GPT-4 win rate prompt (C).

```
Which of the following summaries do a better job of summarizing the most \
important points in the given forum post, without including unimportant \
or irrelevant details? Make your decision while referring to the reference\
(human-written) summary. A good summary is both precise and concise.

Post:
<post>

Reference Summary:
<golden summary>

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">
```

Figure 7: Prompt format for Reddit TL;DR (CNN/Daily Mail)

Summarization GPT-4 win rate prompt (vs human written).

```
Which of the following summaries does a better job of summarizing the most \
important points in the given news article, without including unimportant \
or irrelevant details? A good summary is both precise and concise.

Post:
<post>

Summary A:
<golden summary>
Summary B:
<summary>

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">
```

Figure 8: Prompt format for Reddit TL;DR (CNN/Daily Mail)

DPO using the SFT-trained Phi-3-Mini-4K-Instruct model. Table 8 shows expected KL-divergence (averaged over tokens) over the 40 completions at every 100 steps of training. The expected reward accuracies (win-rates) over the SFT model completions with the same hyperparameters over these samples (after 400 training steps) are shown in Table 9 below.

These results suggest that while DRDO does not explicitly regularize its policy w.r.t. reference-model based KL-regularization, it still outperforms DPO in oracle-assigned expected reward accuracies (win-rates) on sampled generations as long as the γ parameter is carefully chosen. In particular, as previously observed in Meng et al. (2024) and Rafailov et al. (2024b), smaller β in DPO tends to increase KL-divergence with respect to the baseline SFT model. However, a relatively larger KL divergence in DRDO on average does not necessarily impede preference learning but larger γ values tend to degrade expected rewards.

As for the exponential parameter γ, $\gamma = 2$ appears to be a reasonable choice, as previously found in in Lin et al. (2018); Yi et al. (2020) in the focal loss literature. A larger γ can harshly penalize the loss when the policy is uncertain ($p_w << 1$) while a smaller $\gamma = 0$ may not adequately penalize and impact its adaptive nature. In our experiments including the above experiment, we find that the optimal reward is achieved for $\gamma = 2$ while too low or too high a γ can affect performance as seen in Tab. 9 . Note that, although we find $\gamma = 2$ to be optimal across datasets, a reasonable way to find the right γ would vary case by case—-if the baseline policy at the start of alignment training has not undergone or in off-policy settings, a lower γ could be ideal since a higher γ might apply harsher penalties in this case. However, if the policy is initialized with SFT model (as in DRDO) or in on-policy (where p_w is likely to be higher already) alignment settings, a higher γ could be optimal. In practice though, empirical validation on a held-out set can be an efficient alternative, similar to how an optimal β can be determined in algorithms like DPO.

Step	DRDO ($\gamma = 5$)	DRDO ($\gamma = 2$)	DRDO ($\gamma = 1$)	DPO ($\beta = 0.01$)	DPO ($\beta = 0.1$)
100	0.63	0.44	0.82	0.64	0.38
200	0.35	1.51	1.67	0.81	0.39
300	0.59	1.67	1.82	1.17	0.42
400	0.64	1.63	1.71	1.35	0.44

Table 8: KL-divergence during training on sampled generations on 40 randomly sampled evaluation prompts in the held-out set of Ultrafeedback with top-p of 0.8 and temperature of 0.7 with various γ values in DRDO ($\alpha = 0.1$) and with different KL-β values in DPO using the Phi-3-Mini-4K-Instruct model.

Model	Expected Oracle Reward
DPO ($\beta = 0.1$)	0.775 ($\pm$ 0.42)
DPO ($\beta = 0.01$)	0.675 ($\pm$ 0.47)
DRDO ($\gamma = 1$)	0.750 ($\pm$ 0.44)
DRDO ($\gamma = 2$)	0.825 ($\pm$ 0.38)
DRDO ($\gamma = 5$)	0.600 ($\pm$ 0.50)

Table 9: DRDO vs. DPO expected reward accuracies (win-rates) over the SFT-model completions computed using the OPT 1.3B oracle model.

Summarization GPT-4o win rate prompt (C).

Which of the following summaries do a better job of summarizing the most important points in the given forum post, without including unimportant or irrelevant details? Make your decision while referring to the reference (human-written) summary. A good summary is both precise and concise.

Post:
`<post>`

Reference Summary:
`<golden summary>`

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.
THIRD, on a new line, provide your ratings (a real reward score between 0 to 1 where 1 is highest and 0 is lowest in quality) for the summaries.
Your response should use the format:
`Comparison: <one-sentence comparison and explanation>`
`Preferred: <"A" or "B">`
`Score for Summary A: <score>`
`Score for Summary B: <score>`

Figure 9: Prompt format for Reddit TL;DR.