ROPO: Robust Preference Optimization for Large Language Models

Xize Liang[†*1] **Chao Chen**[*2] **Shuang Qiu**[*3] **Jie Wang**[1] **Yue Wu**[2]
Zhihang Fu[2] **Hanzhu Chen**[1] **Feng Wu**[1] **Jieping Ye**[2]

Abstract

The prevalent noise in the preference data unavoidably poses significant challenges to the preference alignment of large language models (LLMs). Existing efforts for this problem either marginally alleviate the impact of noise without noise reduction, or rely on external LLMs that incur substantial computational costs. To address these challenges, we propose **RO**bust **P**reference **O**ptimization (**ROPO**), an iterative alignment approach that integrates *noise-tolerance* and *noise filtering* without the aid of external models. Specifically, ROPO first formulates the training process with adaptive noise reduction as an optimization problem, which can be efficiently solved in an iterative paradigm. Then, to equip this solving process with noise-tolerance and noise-identification capabilities, we derive a robust loss that suppresses the gradients from samples with high uncertainty. We demonstrate both empirically and theoretically that the derived loss is key to the noise-tolerance and effective filtering of noisy samples. The derived loss further inspires a robustness-guided rejection sampling technique to compensate for the potential important information in discarded queries. Extensive experiments on several widely-used datasets and model architectures demonstrate that ROPO significantly outperforms all baselines under **four** practical noise settings and the random symmetric noise, with its advantage increasing as the noise rate increases.

[*]Equal contribution [†]<xizeliang@mail.ustc.edu.cn> [1]MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China [2]Independent Researcher [3]City University of Hong Kong. Correspondence to: Jie Wang <jiewangx@ustc.edu.cn>.

Proceedings of the 42^{st} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

1. Introduction

Recent research indicates that the significant achievements of Large Language Models (LLMs) in understanding various queries and providing helpful responses (Achiam et al., 2023) rely on the preference alignment, which aligns LLMs' responses with human values and expectations (Wang et al., 2023c; Bubeck et al., 2023; Lin et al., 2023). A typical preference alignment approach is Reinforcement Learning from Human Feedback (RLHF) (Casper et al., 2023; Ziegler et al., 2019), which first trains a reward model to fit human preferences and subsequently employs an RL algorithm (Schulman et al., 2017) to guide LLMs to generate high-reward responses. However, due to the potential risks of misgeneralized reward modeling (Casper et al., 2023) and the unstable training (Liu et al., 2023a; Shen et al., 2023) of RLHF, various ranking-based methods represented by Direct Preference Optimization (DPO) (Rafailov et al., 2023) bypass the explicit reward modeling stage and eschew RL techniques via directly optimizing the implicit reward margins between preferred and dis-preferred responses (Yuan et al., 2023; Wang et al., 2023a; Song et al., 2023). Owing to the stable and computationally lightweight supervised learning paradigm, ranking-based methods have emerged as competitive alternatives to RLHF, thus drawing increasing attention recently (Shen et al., 2023; Wang et al., 2023c).

Despite their impressive performance on preference alignment, ranking-based methods heavily rely on high-quality preference data, which is costly and limited in practice (Kim et al., 2023; Chen et al., 2024). First, the noise (e.g., incorrect or ambiguous preferences) in the preference data is unavoidable (Wang et al., 2024a). Many recent studies have observed the presence of preference noise at levels of 20%-40% across various scenarios (Gao et al., 2024; Lee et al., 2023; Zheng et al., 2024; Touvron et al., 2023; Cui et al., 2023; Zhao et al., 2023; Munos et al., 2023), whether the annotators are humans or LLMs. Second, the performance of LLMs will significantly deteriorate when trained with noisy preferences (Chowdhury et al., 2024; Gao et al., 2024; Lee et al., 2023). For instance, a 10% increase in the noise rate may lead to a 30% decrease in the performance of DPO in terms of win rate (Gao et al., 2024). Therefore, it is highly desirable to develop noise-robust preference alignment techniques.

To address these problems, some recent studies have explored the label smoothing (Chowdhury et al., 2024; Mitchell, 2023) and regularization (Gao et al., 2024) techniques to alleviate the impact of preference noise. However, these methods can only marginally mitigate the side effects of noise, as the noisy samples are still involved in the training phase. Besides, (Gao et al., 2024) also attempts to filter out noisy samples but requires another teacher LLM (i.e., a reward model serving as the proxy of the Bradley-Terry model (Bradley & Terry, 1952)) to assign confidence values to samples, which introduces additional computational costs. Moreover, the teacher LLM may not necessarily provide the correct preference direction (Casper et al., 2023), and this method is shown to be ineffective at reducing random symmetric noise (Gao et al., 2024).

In this paper, we propose **RO**bust **P**reference **O**ptimization (**ROPO**), an iterative alignment approach that unifies *noise-tolerance* and *filtering of noisy samples* without the aid of external models. We first provide a general formulation of learning from noisy preference data as a constrained optimization problem, where we dynamically assign a quality-aware weight for each sample (see Section 3.1). Then, we solve the problem through a provably convergent iterative paradigm, consisting of two alternating steps: *noise-tolerant model training* and *noisy sample filtering*.

Our Main Contributions. (1) We propose a robust preference alignment framework that unifies noise-tolerance and filtering of noisy samples. Without the need for any external LLM, the model's robustness and discrimination ability against noisy samples gradually improve as the alternating iterative training proceeds. (2) We derive a robust loss function by suppressing the gradients of samples with high uncertainty. The loss contains a noise-aware term, which not only prevents the model from over-fitting to noisy samples but also facilitates identifying noisy samples versus clean samples[1] (see Section 3.2). (3) We propose a robustness-guided rejection sampling technique to compensate for the potential important information in discarded queries (see Section 3.3), which improves the data quality and thus leads to further improvement in alignment performance. (4) We conduct extensive experiments on three widely-used datasets (i.e., UltraFeedback Binarized, Alpaca Comparison, and TL;DR) with Mistral-7B, Llama-2-7B, Llama-3-8B, Llama-2-13B, and Llama-3-70B. Evaluation results on AlpacaEval, Arena-Hard, and MT-Bench show that the performance of ROPO remains stable in both practical and artificial noisy scenarios.

[1] In Section 3.2, we demonstrate that the cross-entropy loss (i.e., DPO loss) cannot distinguish between noisy samples and clean samples in the context of preference learning, even though it is widely used for learning from noisy data in other scenarios such as image classification (Jiang et al., 2018; Liu et al., 2020).

2. Preliminaries and Problem Settings

Given a query $\mathbf{x} = [x_1, \ldots, x_n]$, an LLM π_θ (with parameters θ) generates a response $\mathbf{y} = [y_1, \ldots, y_m]$, where the tokens $(x_i)_{i=1}^n$ and $(y_j)_{j=1}^m$ come from a predefined vocabulary, in an autoregressive paradigm. Specifically, the model samples y_j from the conditional probability distribution $\pi_\theta(\cdot \mid \mathbf{x}, \mathbf{y}_{1:j-1})$, where $\mathbf{y}_{1:0}$ is null and $\mathbf{y}_{1:j-1} = [y_1, \ldots, y_{j-1}]$ for $j = 2, \ldots, m$. Finally, we can decompose the conditional probability $\pi_\theta(\mathbf{y} \mid \mathbf{x})$ into $\pi_\theta(\mathbf{y} \mid \mathbf{x}) = \prod_{j=1}^m \pi_\theta(y_j \mid \mathbf{x}, \mathbf{y}_{1:j-1})$.

2.1. Alignment of Large Language Models

Most of the existing LLM alignment frameworks first fine-tune a pre-trained model on high-quality datasets of downstream tasks (e.g., dialogue and post-summarization) via maximum likelihood, in order to teach the model to respond to queries. We denote the supervised fine-tuned model π_{sft}. Then, we train the model π_θ (initialized by π_{sft}) based on human preference data. Specifically, a preference sample contains a query $\mathbf{x}$, responses $\mathbf{y}_1$ and $\mathbf{y}_2$, and a ranking label c provided by annotators. We use $c = 0$ to indicate that $\mathbf{y}_1$ is preferred to $\mathbf{y}_2$ (denoted $\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}$) and use $c = 1$ to indicate the opposite. We assume that the preference data $(\mathbf{x}, \mathbf{y}_1, \mathbf{y_2}, c)$ is sampled from a distribution $\mathcal{D}$.

A popular formulation of the generation of preferences is the Bradley-Terry (BT) model (Bradley & Terry, 1952), i.e., $P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = \sigma(r^*(\mathbf{y}_1, \mathbf{x}) - r^*(\mathbf{y}_2, \mathbf{x}))$, where σ is the sigmoid function, and r^* is a latent and inaccessible reward function. The key to existing preference learning methods is to explicitly or implicitly approximate r^* or P^*. RLHF (Ouyang et al., 2022) approximates r^* by training a parameterized reward model r_ϕ via maximum likelihood on preference data, then uses the well-trained r_ϕ to provide signals for the reinforcement learning of π_θ.

Due to the complexity and instability of RLHF, some recent works (Rafailov et al., 2023; Azar et al., 2023; Wang et al., 2023a) directly learn preferences from offline collected response pairs by optimizing the implicit reward margins between preferred and dis-preferred responses. For example, the objectives of DPO (Rafailov et al., 2023) is given by $\ell_{\text{dpo}} = -\log \sigma(\beta \log \frac{\pi_\theta(\mathbf{y}_1|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_2|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2|\mathbf{x})})$, where $\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}$, β is a hyperparameter, and π_{ref} is a fixed reference model (usually the SFT model). Ranking-based methods are more computationally lightweight and stable than RLHF, thus drawing increasing attention recently. Thus, we mainly focus on ranking-based methods in this paper.

2.2. Preference Learning with Noisy Data

Preferences are unavoidably noisy due to the cognitive bias among annotators (**see Appendix C for detailed discussion**). Thus, we have no access to the clean dataset

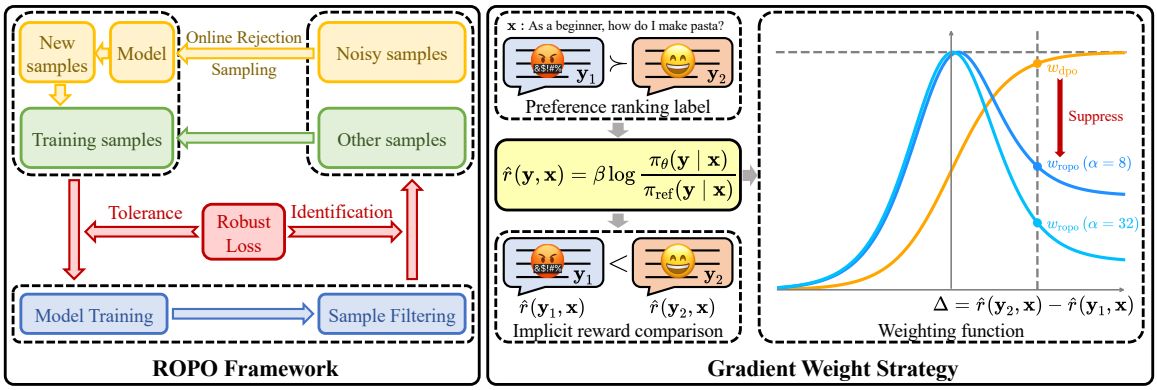


Figure 1. Framework of ROPO and a comparison between the gradient weighting strategies of ROPO and DPO (Rafailov et al., 2023). **Left:** ROPO alternates between noise-tolerant model training and noisy sample filtering and integrates the online rejection sampling paradigm to further improve the data quality. Please see Appendix A for the detailed description and pseudocode of the framework. **Right:** Unlike w_{dpo}, which increases with respect to $\Delta = \hat{r}(\mathbf{y}_2, \mathbf{x}) - \hat{r}(\mathbf{y}_1, \mathbf{x})$, w_{ropo} decreases when Δ is large. Given a noisy sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, whose preference label contradicts the comparison of implicit rewards, ROPO suppresses its gradient. A larger α implies a stronger suppressive effect.

$D = \{(\mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, c^{(i)})\}_{i=1}^N \sim \mathcal{D}$ and can only obtain a noisy dataset $D_\eta = \{(\mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)})\}_{i=1}^N \sim \mathcal{D}_\eta$, where $\hat{c}^{(i)} = c^{(i)}$ with probability $1-\eta$ and $\hat{c}^{(i)} = 1 - c^{(i)}$ with probability η.

Remark. (1) We assume the random symmetric noise in our **theoretical analysis** because it is the standard assumption for learning from noisy data (Liu & Guo, 2020; Zhang & Sabuncu, 2018) and existing research on LLM alignment has indicated **the challenges posed by this kind of noise** (Gao et al., 2024). Besides, in the context of preference alignment, the symmetric and asymmetric (or class-conditional) noise is equivalent, as the ground truth label is changed if we swap the positions of $\mathbf{y}_1$ and $\mathbf{y}_2$. (2) In addition to this artificially introduced random noise, our experiments also include **four types of practical noise settings**, covering a variety of unavoidable noises from human and LLM annotations. For more details, please refer to Section 4 and Appendix E.3.

3. Robust Preference Optimization

We propose ROPO, an iterative preference alignment framework. ROPO alternates between *noise-tolerant model training* and *noisy sample filtering*, as shown in Figure 1, which is mathematically equivalent to iteratively solving a constrained optimization problem (Section 3.1). In the model training step, we introduce a robust loss function by suppressing the gradients of samples with high uncertainty, which prevents the model from over-fitting to the noisy preference. In the sample filtering step, we filter out noisy samples based on the magnitude of their training losses. The key to ROPO is that our proposed loss contains a noise-aware term, which not only features noise-tolerance, but also facilitates identifying noisy samples versus clean samples (Section 3.2). Further, we propose a robustness-guided rejection sampling technique to compensate for the important information in discarded queries and thus improve the data quality (Section 3.3). **For detailed proofs of the theorems in this section, please refer to Appendix F.**

3.1. A General Formulation

Given N preference samples $\{(\mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)})\}_{i=1}^N$, we hope that the weights of noisy samples in the preference optimization are smaller than those of others, thereby reducing the impact of noise on the alignment performance. Without prior knowledge of which samples are noisy, a natural approach would be to assign a dynamic quality-aware weight to each sample and constrain the sum of these weights to a constant, which can also prevent the weights from tending toward zero. Therefore, we formulate learning from noisy preference samples as the following constrained optimization problem:

$$\begin{aligned} \min_{\theta, \mathbf{w}} \quad & \frac{1}{N} \sum_{i=1}^N w_i \ell(\theta; \mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)}, \pi_\theta), \qquad (1) \\ \text{s.t.} \quad & \theta \in \Theta, \quad w_i \in [0,1],\ i = 1, \ldots, N, \\ & \sum_{i=1}^N w_i = N_\rho \triangleq \lfloor (1-\rho) N \rfloor, \end{aligned}$$

where $w_1, \ldots, w_N$ are dynamic weights, Θ is compact, and $\rho \in [0,1]$ is the proportion of the samples we aim to filter out. Please note that we minimize Problem (1) with respect to both θ and $\mathbf{w}$, resulting in a training process that learns the weights adaptively. Hence, we expect that Problem (1) will gradually lead to much smaller weights for noisy samples than those for others. To achieve this, we first analyze the

properties of the optimal solution to Problem (1). As shown in Theorem 3.1, Problem (1) admits an optimal solution and the elements in its minimizer $\mathbf{w}^*$ are either 0 or 1.

Theorem 3.1. *Assume $\ell(\theta)$ is continuous on a compact parameter space Θ, then Problem (1) admits an optimal solution $(\theta^*, \mathbf{w}^*)$. Suppose that $\ell(\theta^*; \mathbf{x}^{(i_1)}, \mathbf{y}_1^{(i_1)}, \mathbf{y}_2^{(i_1)}, \pi_{\theta^*}) < \cdots < \ell(\theta^*; \mathbf{x}^{(i_N)}, \mathbf{y}_1^{(i_N)}, \mathbf{y}_2^{(i_N)}, \pi_{\theta^*})$, then $w_{i_k}^* = 1$ for $1 \le k \le N_\rho$ and $w_{i_k}^* = 0$ for $N_\rho < k \le N$.*

We solve Problem (1) in an iterative paradigm, which consists of two alternating steps: model training and sample filtering. In the step of model training, we fix $\mathbf{w}$ and learn model parameters θ. In the step of sample filtering, we fix θ and assign weights $w_1, \ldots, w_N$ for samples based on their loss values. Because the objective in Problem (1) is non-negative and its value does not increase during the iteration, the iterative solving process is guaranteed to converge.

3.2. A Noise-Tolerant Loss

To guarantee the effectiveness of the iterative solving process within the preference alignment framework, we delve into identifying additional conditions that should be imposed on ℓ. Here, we discuss the properties of ℓ in the context of minimizing its expected risks under distributions of noisy and clean preference data, i.e., finding the optimal solutions θ^* and θ_η^* by solving

$$\theta^* = \arg\min_{\theta\in\Theta} \ \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[\ell(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c, \pi_\theta)], \quad (2)$$

$$\theta_\eta^* = \arg\min_{\theta\in\Theta} \ \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c})\sim\mathcal{D}_\eta}[\ell(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \hat{c}, \pi_\theta)]. \quad (3)$$

Requirement 1: Noise-tolerance. It cannot be guaranteed that the sample filtering stage will eliminate all noise samples (e.g., when ρ is less than the actual noise proportion in the preference data). Consequently, it is crucial that the presence of noisy preferences does not significantly impact the model training stage, i.e., ℓ is noise-tolerant.

Requirement 2: Distinguishable losses for clean and noisy samples. As noisy samples generally exhibit larger loss values (Liu et al., 2020), in the sample filtering step, we filter out the $N - N_\rho$ samples with the largest losses. It is noteworthy that this step takes place midway through training, hence ℓ needs to exhibit distinguishable loss values for clean and noisy samples prior to the convergence of the model.

As DPO is one of the most popular alignment methods, it is natural for us to explore the effectiveness of the DPO loss ℓ_{dpo} within our iterative solving process. However, our findings show that ℓ_{dpo} does not satisfy the aforementioned requirements. **In this section, we assume that $\eta < 1/2$.**

Finding 1: DPO is not noise-tolerant.

Theorem 3.2. *Consider ℓ_{dpo} and the corresponding minimizer θ_η^* to Problem (3). Given a query $\mathbf{x}$ and responses $(\mathbf{y}_1, \mathbf{y}_2)$, the relationship between the preference probability given by the optimal model, i.e., $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, and that given by the BT model, i.e., $P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$ is $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) + (1 - 2P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})) \cdot \eta$, hence we have $\left|P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) - P_{\theta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})\right| = 2\eta\left|P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) - 1/2\right|$.*

As shown in Theorem 3.2, the impact of noise on the optimal solution corresponding to ℓ_{dpo} increases as the noise rate increases. Specifically, the difference between the optimal probabilities under noisy and clean distributions, i.e., $\left|P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) - P_{\theta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})\right|$, is proportional to the label flipping probability η.

Finding 2: DPO faces challenges in distinguishing between noisy and clean samples.

Theorem 3.3. *For samples $(\mathbf{x}^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{y}_2^{(1)}, \hat{c}^{(1)} = c^{(1)})$ and $(\mathbf{x}^{(2)}, \mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)}, \hat{c}^{(2)} = 1 - c^{(2)})$, suppose that θ is not θ_η^* but satisfies $\max_{i=1,2} |P_\theta(\mathbf{y}_1^{(i)} \succ \mathbf{y}_2^{(i)} \mid \mathbf{x}^{(i)}) - P_{\theta_\eta^*}(\mathbf{y}_1^{(i)} \succ \mathbf{y}_2^{(i)} \mid \mathbf{x}^{(i)})| < \delta$, then if we want to ensure that $\ell_{\text{dpo}}(\mathbf{x}^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{y}_2^{(1)}, \hat{c}^{(1)}) < \ell_{\text{dpo}}(\mathbf{x}^{(2)}, \mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)}, \hat{c}^{(2)})$, δ must satisfy $\delta < \frac{1-2\eta}{2}\left(P^*(c^{(1)}) + P^*(c^{(2)}) - 1\right)$.*

As shown in Theorem 3.3, the distance between π_θ and $\pi_{\theta_\eta^*}$ we need for ℓ_{dpo} to differentiate between clean and noisy samples decreases as the BT probability approaches 50% and the noise rate increases. Specifically, Theorem 3.3 shows that the upper bound of δ is proportional to $(1-2\eta)/2$ and $\left(P^*(c^{(1)}) - 1/2 + P^*(c^{(2)}) - 1/2\right)$[2]. Due to the intrinsic diversity and stochastic nature of human preferences, the BT distribution is usually not a "hard" distribution with probabilities close to 0 or 1, but rather a "soft" one (Swamy et al., 2024; Strobl et al., 2011). This brings difficulties to unconverged DPO-trained model in identifying noisy samples. For example, when $\eta = 30\%$ and $P^*\left(c^{(1)}\right) = P^*\left(c^{(2)}\right) = 60\%$, we need $\delta < 4\%$, which is a challenging requirement for a model that has not yet converged.

The gradient weighting strategy of DPO may amplify the impact of noise. Given a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \hat{c} = 0)$, according to (Rafailov et al., 2023), the gradient of ℓ_{dpo} is given by

$$\nabla_\theta \ell_{\text{dpo}} = -\beta \underbrace{\sigma\left(\hat{r}(\mathbf{y}_2, \mathbf{x}) - \hat{r}(\mathbf{y}_1, \mathbf{x})\right)}_{w_{\text{dpo}}(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)} \cdot \nabla \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}, \quad (4)$$

where $\hat{r}(\mathbf{y}, \mathbf{x}) = \beta \log \frac{\pi_\theta(\mathbf{y}|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}|\mathbf{x})}$ is the implicit reward func-

[2] As $c^{(1)}$ and $c^{(2)}$ are clean labels, we have $P^*(c^{(1)}) > 1/2$ and $P^*(c^{(2)}) > 1/2$.

tion of DPO. Intuitively, the greater the discrepancy between the reward function's comparison of $\mathbf{y}_1$ and $\mathbf{y}_2$ and the label $\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}$, the greater the weight $w_{\text{dpo}}(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ of the DPO gradient becomes. This aggressive weighting strategy can be risky if the label is incorrect, as the model may imply a high uncertainty about the sample by giving a higher reward to $\mathbf{y}_2$ than to $\mathbf{y}_1$, increasing w_{dpo} and thus amplifying the impact of the noise.

Conservative gradient weighting strategy. A simple and straightforward idea is that when the implicit reward margin $\Delta(\mathbf{y}_2, \mathbf{y}_1, \mathbf{x}) \triangleq \hat{r}(\mathbf{y}_2, \mathbf{x}) - \hat{r}(\mathbf{y}_1, \mathbf{x})$ is excessively positive, we should assign a conservative weight to the gradient. Based on this idea, we propose the conservative gradient weight

$$w_{\text{ropo}} = \frac{4\alpha}{(1+\alpha)^2} \cdot \sigma(\Delta(\mathbf{y}_2, \mathbf{y}_1, \mathbf{x})) \cdot (1 + \alpha\sigma(-\Delta(\mathbf{y}_2, \mathbf{y}_1, \mathbf{x}))), \tag{5}$$

where $\alpha > 2$ controls the conservatism of weighting and $4\alpha/(1+\alpha)^2$ is used to normalize the maximum value of w_{ropo} (see Appendix F.9). As illustrated in Figure 1, unlike the monotonous increase of w_{dpo}, w_{ropo} decreases when $\Delta(\mathbf{y}_2, \mathbf{y}_1, \mathbf{x})$ is large. Then, the corresponding loss function can be decomposed as

$$\ell_{\text{ropo}} = \int \nabla_\theta \ell_{\text{ropo}} \, \mathrm{d}\theta = \frac{4\alpha^2}{(1+\alpha)^2} \cdot \ell_{\text{na}} + \frac{4\alpha}{(1+\alpha)^2} \cdot \ell_{\text{dpo}}, \tag{6}$$

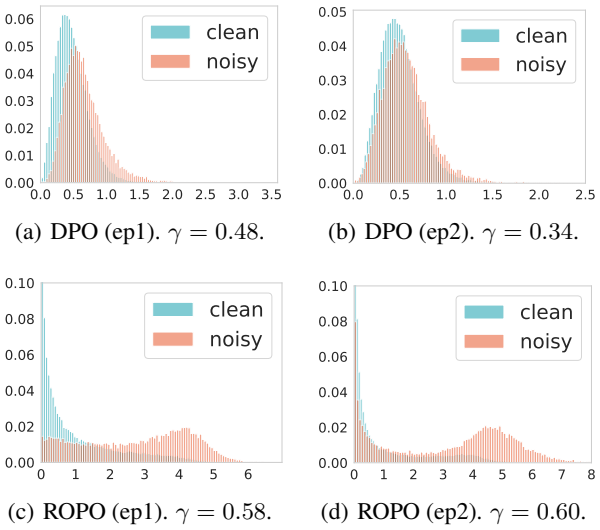
where $\ell_{\text{na}} = \sigma(\beta \log \frac{\pi_\theta(\mathbf{y}_2|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2|\mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_1|\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1|\mathbf{x})})$ and we omit the constant term of the primitive function (see Appendix F.4 for details). The introduced loss consists of ℓ_{dpo} and a noise-aware term ℓ_{na} whose weight is α times that of ℓ_{dpo}. We claim that ℓ_{na} has the following advantages.

Advantage 1: ℓ_{na} is noise-tolerant.

Theorem 3.4. *Consider ℓ_{na} and the corresponding minimizer θ_η^* to Problem (3). Given a query $\mathbf{x}$ and responses $(\mathbf{y}_1, \mathbf{y}_2)$, the relationship between the preference probability given by the optimal model, i.e., $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, and that given by the BT model, i.e., $P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$ is $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = \mathbb{I}(P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) > \frac{1}{2})$, hence we have $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) = P_{\theta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$.*

As shown in Theorem 3.4, contrary to the conclusion in Theorem 3.2 that the optimal solution corresponding to ℓ_{dpo} is affected by the noise, the optimal preference probability corresponding to ℓ_{na}, i.e., $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, remains unchanged when the label flipping probability $\eta < 1/2$. Specifically, 3.4 shows that $P_{\theta_\eta^*}(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$ is an indicator function of $P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}) > 1/2$.

Advantage 2: ℓ_{na} can distinguish noisy samples from clean ones.



(a) DPO (ep1). $\gamma = 0.48$. (b) DPO (ep2). $\gamma = 0.34$. (c) ROPO (ep1). $\gamma = 0.58$. (d) ROPO (ep2). $\gamma = 0.60$.

Figure 2. Loss distributions of Llama-2-7B trained with DPO and ROPO at different training epochs (ep1 and ep2) on TL;DR. We denote γ as the proportion of noisy samples in the 20% of samples that are filtered out. Larger γ indicates better discrimination between clean and noisy samples.

Theorem 3.5. *For samples $(\mathbf{x}^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{y}_2^{(1)}, \hat{c}^{(1)} = c^{(1)})$ and $(\mathbf{x}^{(2)}, \mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)}, \hat{c}^{(2)} = 1 - c^{(2)})$, suppose that θ is not θ_η^* but satisfies $\max_{i=1,2} |P_\theta(\mathbf{y}_1^{(i)} \succ \mathbf{y}_2^{(i)} \mid \mathbf{x}^{(i)}) - P_{\theta_\eta^*}(\mathbf{y}_1^{(i)} \succ \mathbf{y}_2^{(i)} \mid \mathbf{x}^{(i)})| < \delta$, then if we want to ensure that $\ell_{\text{na}}(\mathbf{x}^{(1)}, \mathbf{y}_1^{(1)}, \mathbf{y}_2^{(1)}, \hat{c}^{(1)}) < \ell_{\text{na}}(\mathbf{x}^{(2)}, \mathbf{y}_1^{(2)}, \mathbf{y}_2^{(2)}, \hat{c}^{(2)})$, we must have $\delta < \frac{1}{2}$.*

As shown in Theorem 3.5, contrary to the challenging requirement ℓ_{dpo} places on an unconverged model in Theorem 3.3, we can expect that ℓ_{na} yields a larger value for noisy samples than for others as long as the difference between the preference probability given by an unconverged model and that of the optimal model is less than 50%. We verify our theoretical analysis in experiments, as shown in Figure 2. For more details, please refer to Section 4.1.

Discussion. ℓ_{na} is capable of improving noise-tolerance and separating noisy samples from clean samples. However, compared with ℓ_{na}, ℓ_{dpo} leads to a "softer" optimal preference probability, which could potentially avoid discrimination against minorities by LLMs. Besides, the aggressive weighting strategy may be useful for clean preference datasets (although they are rare). Thus, it is considered necessary to incorporate a minor component of ℓ_{dpo} into the final loss. From this perspective, the hyperparameter α plays an important role in trading-off between aggressive (ℓ_{dpo}) and conservative (ℓ_{na}) gradient weighting strategy. Given that the weight of ℓ_{na} is $\alpha > 2$ times greater than that of ℓ_{dpo} (in our experiments and ablations, $\alpha \geq 6$), ℓ_{na} **dom-**

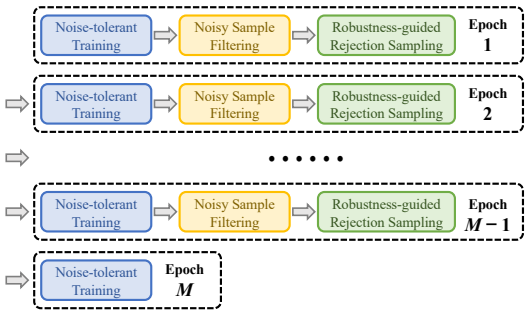


Figure 3. The iterative process of ROPO.

inates the optimization process. Thus, the incorporation of ℓ_{dpo} does not hurt the noise-tolerance and noise filtering too much.

3.3. Robustness-guided Rejection Sampling

The sample filtering step effectively reduces the proportion of noise but may also discard some important queries. For example, a query designed to eliminate the occupational discrimination in LLMs may be filtered out because the ranking label of its associated responses is incorrect. Thus, inspired by the sample distinguishing ability of our proposed ℓ_{ropo}, we propose a rejection sampling technique to compensate for the essential but discarded information and thus improve the robustness of our ROPO framework. Specifically, we sample K responses $\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_K$ to $\mathbf{x}$ for each sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ that is filtered out and generate $2K$ candidate samples $\{(\mathbf{x}, \mathbf{y}_1, \widetilde{\mathbf{y}}_k, \mathbf{y}_1 \succ \widetilde{\mathbf{y}}_k \mid \mathbf{x})\}_{k=1}^{K} \cup \{(\mathbf{x}, \mathbf{y}_2, \widetilde{\mathbf{y}}_k, \mathbf{y}_2 \succ \widetilde{\mathbf{y}}_k \mid \mathbf{x})\}_{k=1}^{K}$. Then, we compute their loss values and add the sample with the minimum loss to the dataset. Note that we treat the model's responses as dis-preferred ones compared to the original responses, which suppresses the potential unsatisfactory or even harmful information in the model's outputs.

3.4. ROPO Framework and Complexity Analysis

As shown in Figure 3, ROPO iteratively carries out three stages: noise-tolerant training, noisy sample filtering, and robustness-guided rejection sampling. In the last epoch, we only perform the noise-tolerant training stage and then get the final model. For the detailed pseudocode and complexity analysis of ROPO, please refer to Appendix A.

4. Experiments

Tasks and Datasets. We focus on two dialogue datasets (i.e., UltraFeedback Binarized[3] (UFB) and Alpaca Comparison (Peng et al., 2023)) and one post-summarization dataset (i.e., Reddit TL;DR (Völske et al., 2017; Stiennon et al., 2020)). For details about the datasets, please refer to D.1.

Noise Settings. Our experiments include **four types of practical noise settings**. As stated in Section 1, original datasets unavoidably contain noise introduced by annotators (see Appendix E.3 for details about the two related practical noise settings). To further explore the performance of ROPO and baselines under noise, we randomly alter preference labels at different proportions (20% and 40%) within the datasets to produce more challenging symmetric noise (Gao et al., 2024). Besides, in Appendices E.3.1 and E.3.2, we supplement experiments in another two practical settings, where the noise comes from annotators' trust in larger models over smaller models and LLM preference comparisons. Please refer to the supplementary material for more details.

Baselines, Models, and Hyperparameters. Our baselines are DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), and two approaches that use the label smoothing technique to alleviate the impact of noise, i.e., rDPO (Chowdhury et al., 2024) and cDPO (Mitchell, 2023). Besides, we supplement experiments on reward modeling in Appendix E.2.

We use Mistral-7B (Jiang et al., 2023) and Llama-2-7B (Touvron et al., 2023) as base models for all baselines and datasets in the main text. For experiments on Llama-2-13B and Llama-3-70B, please refer to Appendix E.1. On UFB, we use Zephyr-7B-SFT-β (Tunstall et al., 2023) as the SFT model for experiments with Mistral-7B, and adopt the result of Zephyr-7B-β (Tunstall et al., 2023) on AlpacaEval (90.60) as the performance of DPO under no artificial noise. In other cases, we fine-tune base models on the preferred responses (SFT targets) to form the SFT models. For details about our baselines, models, and hyperparamters, please refer to Appendix D.2. We run all experiments on 16 NVIDIA A100 GPUs (80 GB).

Evaluation. For models trained on UFB and Alpaca Comparison, we evaluate them on the AlpacaEval benchmark (Li et al., 2023a) by comparing their outputs with those of text-davinci-003 (recommended by the benchmark for comparison). For models trained on TL;DR, we evaluate them by comparing their outputs with the SFT targets (chosen responses) on the test split of TL;DR. Following (Rafailov et al., 2023; Tunstall et al., 2023), we employ GPT-4 as the referee for head-to-head comparisons, using the win rate as the metric. The win rate can be computed by $\Omega = \frac{\#(\text{Win})+\#(\text{Tie})/2}{\#(\text{Comparisons})}$, where $\#(\text{Win})$, $\#(\text{Tie})$, and $\#(\text{Comparisons})$ are the numbers of wins, ties, and comparisons, respectively. For evaluation details, experiments on more benchmarks, and human evaluation, please refer to Appendices D.3, E.4, and E.6, respectively.

[3] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

Table 1. Win rates (%) of **different methods vs SFT targets** under different proportions (i.e., 0, 20%, and 40%) of artificial noise, evaluated by GPT-4. The bold font indicates the best result and an underline indicates the second-best result. **Please note that 0% represents no *artificial* noise, which does not mean that the dataset is clean.**

Dataset		UFB			Alpaca Comparison			TL;DR		
Model	Method	0%	20%	40%	0%	20%	40%	0%	20%	40%
Mistral-7B	DPO	<u>90.60</u>	86.21	82.67	<u>73.66</u>	70.19	65.84	<u>63.00</u>	56.80	49.60
	IPO	88.45	87.32	82.86	72.92	70.81	67.33	62.00	57.00	48.80
	rDPO	88.07	<u>87.45</u>	<u>84.72</u>	72.55	<u>72.05</u>	<u>70.31</u>	62.40	<u>58.20</u>	52.60
	cDPO	88.82	86.96	83.35	73.04	71.30	69.94	59.40	57.40	<u>53.00</u>
	ROPO	**91.06**	**88.63**	**87.70**	**75.40**	**76.27**	**74.04**	**79.00**	**77.80**	**75.80**
Llama-2-7B	DPO	<u>68.57</u>	66.71	62.36	53.42	50.68	48.20	<u>56.80</u>	42.40	35.20
	IPO	67.70	66.09	64.35	53.54	50.56	49.19	54.20	50.80	<u>51.60</u>
	rDPO	68.07	<u>67.83</u>	<u>65.59</u>	52.80	<u>51.18</u>	<u>50.31</u>	54.80	<u>54.00</u>	50.40
	cDPO	68.20	67.33	65.09	<u>53.79</u>	50.81	49.81	52.20	52.00	49.80
	ROPO	**68.94**	**69.44**	**66.71**	**55.90**	**54.41**	**54.53**	**78.80**	**78.00**	**79.20**

Table 2. Win rates (%) of **ROPO and DPO vs SFT targets** under different proportions (i.e., 0, 20%, and 40%) of artificial noise at different training epochs on TL;DR, evaluated by GPT-4.

Model	Method	0%			20%			40%		
		ep1	ep2	ep3	ep1	ep2	ep3	ep1	ep2	ep3
Mistral-7B	DPO	62.60	60.20	63.00	56.80	51.00	48.60	49.60	44.40	44.60
	ROPO	75.40	75.60	79.00	68.80	76.40	77.80	61.60	70.80	75.80
Llama-2-7B	DPO	49.00	53.60	56.80	42.40	38.40	39.20	32.00	35.20	33.60
	ROPO	74.00	82.00	78.80	58.40	76.40	78.00	46.00	70.80	79.20

4.1. Main Results

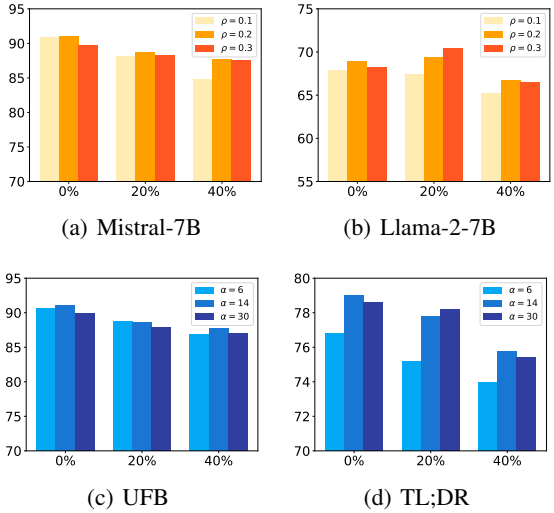
ROPO is robust to noisy preferences. We present the win rates of different methods vs SFT targets under different proportions of artificial noise in Table 1. From the table, we have several interesting observations: (1) For all preference alignment methods, their win rates show a decreasing trend as the noise rate increases. (2) Compared to the competitors, our proposed ROPO demonstrates a more stable performance under noisy preference data. (3) ROPO consistently outperforms the baselines under different proportions of artificial noise in all the three datasets. Even without artificial noise, ROPO still outperforms DPO by 16.0% on TL;DR and 2.5% on Alpaca Comparison, which indicates that the datasets inherently contain noise. (4) Baselines that use the label smoothing technique (i.e., rDPO and cDPO) mostly outperform other baselines under 20% and 40% artificial noise, but underperform ROPO. We speculate that the reasons for their limited effectiveness are as follows. First, rDPO and cDPO are noise-tolerant only when the hyperparameter ε exactly equals the proportion of noise and when $\varepsilon = 0.5$, respectively (see Appendix F.7), which is difficult to achieve in practice, as we have no prior knowledge of the exact noise proportion. Second, they do not reduce the presence of noise and thus can only marginally mitigate the side effects of noise. In contrast, ROPO exhibits noise-tolerance without the priors on the noise proportion and iteratively reduces the noise proportion as the training proceeds, thus leading to superior performance to rDPO and cDPO.

ROPO distinguishes noisy samples from clean samples. In Section 3.2, we have theoretically shown that ℓ_{na} can distinguish noisy samples from clean ones, while ℓ_{dpo} cannot. Besides, we also claim that the minor incorporation of ℓ_{dpo} in ℓ_{ropo} does not hurt the noise filtering ability. To support our analysis, we report the loss distributions for Llama-2-7B trained with ROPO and DPO on TL;DR in Figure 2. Specifically, for models trained for one (two) epoch, we use the SFT model (the model trained for one epoch) as the reference model and compute the losses for all noisy and clean samples. The results in Figure 2 demonstrate three important observations: (1) ℓ_{ropo} can distinguish between noisy and clean samples by yielding larger values for noisy samples than for others. (2) The distributions of ℓ_{dpo} on noisy and clean samples are similar and the gap between them narrows as training progresses. (3) ROPO has a stronger capability for filtering out noisy samples compared to DPO. Specifically, in the top 20% of samples with the largest ℓ_{ropo}, noisy samples make up 60%; whereas in the top 20% of samples with the largest ℓ_{dpo}, noisy samples account for about 34%.

ROPO gradually improves the performance. In Table 2,

Table 3. Ablations on different components of ROPO for Mistral-7B on UFB. NSF and RS stand for the noisy sample filtering and rejection sampling stages, respectively.

Method	0%	20%	40%
DPO	90.60	86.21	82.67
ROPO (ℓ_{na})	89.19	87.58	86.34
ROPO (ℓ_{na} + NSF)	89.44	88.20	**88.07**
ROPO (ℓ_{na} + NSF + RS)	**91.06**	**88.63**	87.70



(a) Mistral-7B (b) Llama-2-7B

(c) UFB (d) TL;DR

Figure 4. Ablations on ρ and α. (a) and (b) respectively show the performance of ROPO-trained Mistral-7B and Llama-2-7B on UFB with different proportions of artificial noise and sample filtering ratio ρ. (c) and (d) respectively show the performance of ROPO-trained Mistral-7B on UFB and TL;DR with different proportions of artificial noise and α.

we report the win rates of ROPO and DPO vs SFT targets under different proportions of artificial noise at different training epochs on TL;DR. From the results, we find that the performance of ROPO gradually improves as training progresses in most cases, while DPO does not exhibit the same trend. Specifically, the performance of DPO at the second and third epochs is generally lower than that at the first epoch under 20% and 40% artificial noise. As a comparison, the second epoch training of ROPO brings an 8%-24% increase in the win rate, and the third epoch also leads to a 5%-9% improvement under 40% artificial noise. These results demonstrate that the iterative training of ROPO effectively reduces the impact of noise and thus consistently improves the alignment performance.

4.2. Ablations

Effectiveness of components in ROPO. To evaluate the effectiveness of different components of our ROPO framework, we compare the performance of our proposal with and without: (a) noise-aware term ℓ_{na}, (b) noisy sample filtering stage, and (c) rejection sampling stage. As shown in Table 3, all components improve ROPO's performance, validating the rationale of our robust framework. Compared to the aggressive DPO loss, our proposed noise-aware term ℓ_{na} consistently improves the performance, which indicates that a proper trade-off between aggressive and conservative gradient weighting strategy effectively prevents the model from over-fitting to noise. Besides, the results also show that the noisy sample filtering is the most effective part of our method, which also makes ROPO significantly superior to other label smoothing-based methods (Chowdhury et al., 2024; Mitchell, 2023).

How many noisy samples should we filter out? The sample filtering ratio ρ is a key factor to the data filtering stage. **In the main experiments, we only report the results with $\rho = 0.2$.** Here, we also present the results of filtering 10% and 30% samples with larger loss values. The results in Figures 4(a) and 4(b) show that better performance could be achieved when filtering 20% or 30% samples. We attribute the reason for this result to the noise ratio in the preference data, which is generally between 20%-30% (Gao et al., 2024). There's a substantial risk of eliminating a considerable amount of high-quality data if we set a larger ratio ρ. Thus, we recommend using $\rho = 0.2$ in practice.

Sensitivity to hyperparameters α. The trade-off hyperparameter α controls the importance of the conservative noise-aware term. A larger α indicates a more conservative gradient weighting strategy. As $C \triangleq \lim_{\Delta\to\infty} w_{\text{ropo}}(\Delta) = 4\alpha/(\alpha+1)^2$, we search the best α in the range of $\{6, 14, 30\}$, which corresponds to $C \in \{1/2, 1/4, 1/8\}$. Then, **we use $\alpha = 14$ in our main experiments** (see Appendix D.2 for the settings of hyperparameters). To explore the effect of α, we provide ablations on α in Figures 4(d) and 4(c). As observed, the model's performance remains largely unaffected for α within an appropriate range, as the loss scale does not change significantly (note that $\alpha C \in [2.94, 3.75]$ for $\alpha \in [6, 30]$). Besides, for the dialogue task, a smaller α results in better performance, as a smaller α will lead to more diverse answers. In contrast, a larger α results in better performance in the summarization task. As the summarization task is more objective than the dialogue task, the results are more sensitive to noise, and hence we need a model that is more robust to the noise.

5. Related Work

Preference Alignment of LLMs. The most representative paradigm of preference alignment is RLHF (Ziegler et al., 2019; Ouyang et al., 2022), which involves training a reward model to capture human preferences and then steering LLMs towards producing high-reward responses through RL algorithms (Schulman et al., 2017). However, in real

applications, RL-based methods are complex and prone to instability during training (Rafailov et al., 2023; Wu et al., 2023; Yuan et al., 2023). Therefore, many recent studies have explored more straightforward and stable alternatives for RLHF (Yuan et al., 2023; Rafailov et al., 2023; Song et al., 2023; Wang et al., 2023b; Lin et al., 2023; Li et al., 2023b; Wang et al., 2023c; Zhao et al., 2022). Among these studies, the most promising direction is to use a contrastive or ranking loss to calibrate the likelihood of the output sequence. Specifically, DPO (Rafailov et al., 2023) implicitly optimizes the same objective as existing RLHF-based methods and enables human preference alignment directly with a simple cross-entropy loss. In addition to the aforementioned methods using data in the form of $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c)$, where c is the preference label, some recent studies (Duan et al., 2024; Ethayarajh et al., 2024; Chen et al., 2024) have also used data in the form of $(\mathbf{x}, \mathbf{y}, \mathbf{c})$, where c is an annotation of the response $\mathbf{y}$, for preference alignment.

Learning from Noisy Data. Studies in learning from noisy data primarily fall into three categories. The first category is sample-selection based methods (Swayamdipta et al., 2020; Pleiss et al., 2020; Paul et al., 2021; Sorscher et al., 2022), which identify high-quality samples before training and filter out noisy samples. For example, (Swayamdipta et al., 2020) uses the training dynamics to identify valuable samples. The second category is weighting-based methods, which assign greater weights for important samples and lesser weights for noisy samples (Ren et al., 2018; Han et al., 2022; Shu et al., 2019). Besides, another important area of research is dedicated to the design of loss functions that are robust to noise (Ghosh et al., 2017; Wang et al., 2019; Zhang & Sabuncu, 2018). The findings in (Ghosh et al., 2017) indicate that the traditional cross-entropy loss is sensitive to the label noise, while symmetric loss functions are robust to such noise. Furthermore, recent advances in LLMs have also underscored the essential role of data quality in both pre-training and supervised fine-tuning (SFT) phases of LLMs (Marion et al., 2023; Zhou et al., 2023; Korbak et al., 2023).

Rejection Sampling. The rejection sampling is a popular approach of data augmentation to improve the data quality and performance in existing preference alignment methods. Specifically, (Dong et al., 2023) ranks newly-collected responses based on their rewards and selects the highest ranked one to add to the dataset. To address the issue of the excessively high rejection rate and thus improve the effectiveness of rejection sampling, (Xiong et al., 2023) proposes a multi-step sampling technique, which also requires an external reward model. Besides, (Wang et al., 2024b) and (Yang et al., 2024b) consider rejection sampling for the multi-objective preference alignment, where (Wang et al., 2024b) projects multi-objective reward vectors onto one dimension and then selects samples based on the scalar rewards, while (Yang et al., 2024b) augments samples near the Pareto front of multi-dimensional rewards, leading to a strong multi-objective alignment performance. Compared to the aforementioned methods, which all rely on rewards provided by external models, our robustness-guided rejection sampling technique selects new samples based on loss values that reflect the quality of the samples. Moreover, our technique benefits from being independent of external LLMs, thus leading to computational and memory efficiency.

Robust Alignment of LLMs. Many efforts have been made from various perspectives to achieve robust preference alignment (Chowdhury et al., 2024; Mitchell, 2023; Choi et al., 2024; Bukharin et al.; Yan et al., 2024; Wu et al., 2024; Ramesh et al., 2024; Yang et al., 2022; Yang et al.; Xu et al., 2025). Specifically, (Choi et al., 2024) improves the model's adaptability to different preference distributions and enables iterative output refinement by jointly optimizing a self-improvement policy and a generative policy. (Bukharin et al.) models potentially corrupted preference labels as sparse outliers and solves an ℓ_1-regularized maximum likelihood estimation problem, thereby consistently learning the true underlying reward. (Yan et al., 2024) introduces a multi-head reward model (RM) that reflects each head's confidence in the output reward using the standard deviation of a Gaussian distribution, effectively addresses the challenge of RM imperfections in RM-based RLHF. (Wu et al., 2024) focuses on different forms of noise and enhances DPO's resilience to both pointwise and pairwise noise in LLM alignment by leveraging Distributionally Robust Optimization (DRO). (Ramesh et al., 2024) robustly aligns LLMs to the preferences of diverse individual groups by incorporating group information into the LLM context and optimizing against the worst-case alignment performance across all groups. Compared to them, our method integrates noise-tolerance and noise-identification capabilities without external models, offering a novel paradigm for robust preference alignment.

6. Conclusion

Robust preference optimization is critical for the LLM alignment, as noisy preferences are inevitable in practical scenarios. Unlike existing methods, which rely on label smoothing or external LLMs for the sample selection, we propose a robust preference alignment framework that unifies noise-tolerant model training and effective filtering of noisy samples. Specifically, we incorporate a noise-aware loss term to prevent the model from over-fitting to noise. Further, we propose a robustness-guided rejection sampling technique to compensate for the potential information reduction caused by the filtering stage. We provide extensive theoretical and empirical evidence to demonstrate the effectiveness of our proposed ROPO framework.

Acknowledgments

This work was supported in part by National Key R&D Program of China under contract 2022ZD0119801, National Nature Science Foundations of China grants U23A20388 and 62021001. We would like to thank all the anonymous reviewers for their insightful comments.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

Azar, M. G., Rowland, M., Piot, B., Guo, D., Calandriello, D., Valko, M., and Munos, R. A general theoretical paradigm to understand learning from human preferences. arXiv preprint arXiv:2310.12036, 2023.

Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al. Qwen technical report. arXiv preprint arXiv:2309.16609, 2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862, 2022.

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

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:2303.12712, 2023.

Bukharin, A., Hong, I., Jiang, H., Li, Z., Zhang, Q., Zhang, Z., and Zhao, T. Robust reinforcement learning from corrupted human feedback. In The Thirty-eighth Annual Conference on Neural Information Processing Systems.

Casper, S., Davies, X., Shi, C., Gilbert, T. K., Scheurer, J., Rando, J., Freedman, R., Korbak, T., Lindner, D., Freire, P., et al. Open problems and fundamental limitations of reinforcement learning from human feedback. arXiv preprint arXiv:2307.15217, 2023.

Chen, Z., Deng, Y., Yuan, H., Ji, K., and Gu, Q. Self-play fine-tuning converts weak language models to strong language models. arXiv preprint arXiv:2401.01335, 2024.

Chiang, W.-L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., Zhang, H., Zhu, B., Jordan, M., Gonzalez, J. E., et al. Chatbot arena: An open platform for evaluating llms by human preference. arXiv preprint arXiv:2403.04132, 2024.

Choi, E., Ahmadian, A., Geist, M., Pietquin, O., and Azar, M. G. Self-improving robust preference optimization. arXiv preprint arXiv:2406.01660, 2024.

Chowdhury, S. R., Kini, A., and Natarajan, N. Provably robust dpo: Aligning language models with noisy feedback. arXiv preprint arXiv:2403.00409, 2024.

Cui, G., Yuan, L., Ding, N., Yao, G., Zhu, W., Ni, Y., Xie, G., Liu, Z., and Sun, M. Ultrafeedback: Boosting language models with high-quality feedback. arXiv preprint arXiv:2310.01377, 2023.

Dong, H., Xiong, W., Goyal, D., Zhang, Y., Chow, W., Pan, R., Diao, S., Zhang, J., Shum, K., and Zhang, T. Raft: Reward ranked finetuning for generative foundation model alignment. arXiv preprint arXiv:2304.06767, 2023.

Duan, S., Yi, X., Zhang, P., Lu, T., Xie, X., and Gu, N. Negating negatives: Alignment without human positive samples via distributional dispreference optimization. arXiv preprint arXiv:2403.03419, 2024.

Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Yang, A., Fan, A., et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. arXiv preprint arXiv:2402.01306, 2024.

Gao, Y., Alon, D., and Metzler, D. Impact of preference noise on the alignment performance of generative language models. arXiv preprint arXiv:2404.09824, 2024.

Ghosh, A., Kumar, H., and Sastry, P. S. Robust loss functions under label noise for deep neural networks. In Proceedings of the AAAI conference on artificial intelligence, volume 31, 2017.

Han, Y., Pu, Y., Lai, Z., Wang, C., Song, S., Cao, J., Huang, W., Deng, C., and Huang, G. Learning to weight samples for dynamic early-exiting networks. In European Conference on Computer Vision, pp. 362–378. Springer, 2022.

Jiang, A. Q., Sablayrolles, A., Mensch, A., Bamford, C., Chaplot, D. S., Casas, D. d. l., Bressand, F., Lengyel, G., Lample, G., Saulnier, L., et al. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

Jiang, L., Zhou, Z., Leung, T., Li, L.-J., and Fei-Fei, L. MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In Dy, J. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pp. 2304–2313. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/jiang18c.html.

Kim, S., Bae, S., Shin, J., Kang, S., Kwak, D., Yoo, K. M., and Seo, M. Aligning large language models through synthetic feedback. arXiv preprint arXiv:2305.13735, 2023.

Korbak, T., Shi, K., Chen, A., Bhalerao, R. V., Buckley, C., Phang, J., Bowman, S. R., and Perez, E. Pretraining language models with human preferences. In International Conference on Machine Learning, pp. 17506–17533. PMLR, 2023.

Lee, H., Phatale, S., Mansoor, H., Lu, K., Mesnard, T., Bishop, C., Carbune, V., and Rastogi, A. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. arXiv preprint arXiv:2309.00267, 2023.

Lee, S., Park, S. H., Kim, S., and Seo, M. Aligning to thousands of preferences via system message generalization. arXiv preprint arXiv:2405.17977, 2024.

Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., and Stoica, I. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. arXiv preprint arXiv:2406.11939, 2024.

Li, X., Zhang, T., Dubois, Y., Taori, R., Gulrajani, I., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacaeval: An automatic evaluator of instruction-following models. https://github.com/tatsu-lab/alpaca_eval, 2023a.

Li, Y., Wei, F., Zhao, J., Zhang, C., and Zhang, H. Rain: Your language models can align themselves without finetuning. arXiv preprint arXiv:2309.07124, 2023b.

Lin, B. Y., Ravichander, A., Lu, X., Dziri, N., Sclar, M., Chandu, K., Bhagavatula, C., and Choi, Y. The unlocking spell on base llms: Rethinking alignment via in-context learning. arXiv preprint arXiv:2312.01552, 2023.

Liu, H., Sferrazza, C., and Abbeel, P. Chain of hindsight aligns language models with feedback. arXiv preprint arXiv:2302.02676, 2023a.

Liu, S., Niles-Weed, J., Razavian, N., and Fernandez-Granda, C. Early-learning regularization prevents memorization of noisy labels. Advances in neural information processing systems, 33:20331–20342, 2020.

Liu, T., Zhao, Y., Joshi, R., Khalman, M., Saleh, M., Liu, P. J., and Liu, J. Statistical rejection sampling improves preference optimization. arXiv preprint arXiv:2309.06657, 2023b.

Liu, Y. and Guo, H. Peer loss functions: Learning from noisy labels without knowing noise rates. In International conference on machine learning, pp. 6226–6236. PMLR, 2020.

Lou, X., Zhang, J., Xie, J., Liu, L., Yan, D., and Huang, K. Spo: Multi-dimensional preference sequential alignment with implicit reward modeling. arXiv preprint arXiv:2405.12739, 2024.

Marion, M., Üstün, A., Pozzobon, L., Wang, A., Fadaee, M., and Hooker, S. When less is more: Investigating data pruning for pretraining llms at scale. arXiv preprint arXiv:2309.04564, 2023.

Mitchell, E. A note on dpo with noisy preferences & relationship to ipo, 2023. URL https://ericmitchell.ai/cdpo.pdf.

Munos, R., Valko, M., Calandriello, D., Azar, M. G., Rowland, M., Guo, Z. D., Tang, Y., Geist, M., Mesnard, T., Michi, A., et al. Nash learning from human feedback. arXiv preprint arXiv:2312.00886, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems, 35:27730–27744, 2022.

Paul, M., Ganguli, S., and Dziugaite, G. K. Deep learning on a data diet: Finding important examples early in training. Advances in Neural Information Processing Systems, 34: 20596–20607, 2021.

Peng, B., Li, C., He, P., Galley, M., and Gao, J. Instruction tuning with gpt-4. arXiv preprint arXiv:2304.03277, 2023.

Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., and Hinton, G. Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548, 2017.

Pleiss, G., Zhang, T., Elenberg, E., and Weinberger, K. Q. Identifying mislabeled data using the area under the margin ranking. Advances in Neural Information Processing Systems, 33:17044–17056, 2020.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. arXiv preprint arXiv:2305.18290, 2023.

Ramesh, S. S., Hu, Y., Chaimalas, I., Mehta, V., Sessa, P. G., Bou Ammar, H., and Bogunovic, I. Group robust preference optimization in reward-free rlhf. Advances in Neural Information Processing Systems, 37:37100–37137, 2024.

Ren, M., Zeng, W., Yang, B., and Urtasun, R. Learning to reweight examples for robust deep learning. In International conference on machine learning, pp. 4334–4343. PMLR, 2018.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Shen, T., Jin, R., Huang, Y., Liu, C., Dong, W., Guo, Z., Wu, X., Liu, Y., and Xiong, D. Large language model alignment: A survey. arXiv preprint arXiv:2309.15025, 2023.

Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., and Meng, D. Meta-weight-net: Learning an explicit mapping for sample weighting. Advances in neural information processing systems, 32, 2019.

Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. Preference ranking optimization for human alignment. arXiv preprint arXiv:2306.17492, 2023.

Song, H., Kim, M., Park, D., Shin, Y., and Lee, J.-G. Learning from noisy labels with deep neural networks: A survey. IEEE Transactions on Neural Networks and Learning Systems, 2022.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. Beyond neural scaling laws: beating power law scaling via data pruning. Advances in Neural Information Processing Systems, 35:19523–19536, 2022.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. Advances in Neural Information Processing Systems, 33: 3008–3021, 2020.

Strobl, C., Wickelmaier, F., and Zeileis, A. Accounting for individual differences in bradley-terry models by means of recursive partitioning. Journal of Educational and Behavioral Statistics, 36(2):135–153, 2011.

Swamy, G., Dann, C., Kidambi, R., Wu, Z. S., and Agarwal, A. A minimaximalist approach to reinforcement learning from human feedback. arXiv preprint arXiv:2401.04056, 2024.

Swayamdipta, S., Schwartz, R., Lourie, N., Wang, Y., Hajishirzi, H., Smith, N. A., and Choi, Y. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 9275–9293, 2020.

Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., and Hashimoto, T. B. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288, 2023.

Tunstall, L., Beeching, E., Lambert, N., Rajani, N., Rasul, K., Belkada, Y., Huang, S., von Werra, L., Fourrier, C., Habib, N., et al. Zephyr: Direct distillation of lm alignment. arXiv preprint arXiv:2310.16944, 2023.

Völske, M., Potthast, M., Syed, S., and Stein, B. Tl; dr: Mining reddit to learn automatic summarization. In Proceedings of the Workshop on New Frontiers in Summarization, pp. 59–63, 2017.

Wang, B., Zheng, R., Chen, L., Liu, Y., Dou, S., Huang, C., Shen, W., Jin, S., Zhou, E., Shi, C., et al. Secrets of rlhf in large language models part ii: Reward modeling. arXiv preprint arXiv:2401.06080, 2024a.

Wang, C., Jiang, Y., Yang, C., Liu, H., and Chen, Y. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. arXiv preprint arXiv:2309.16240, 2023a.

Wang, H., Lin, Y., Xiong, W., Yang, R., Diao, S., Qiu, S., Zhao, H., and Zhang, T. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. arXiv preprint arXiv:2402.18571, 2024b.

Wang, J., Wang, H., Sun, S., and Li, W. Aligning language models with human preferences via a bayesian approach. arXiv preprint arXiv:2310.05782, 2023b.

Wang, Y., Ma, X., Chen, Z., Luo, Y., Yi, J., and Bailey, J. Symmetric cross entropy for robust learning with noisy labels. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 322–330, 2019.

Wang, Y., Zhong, W., Li, L., Mi, F., Zeng, X., Huang, W., Shang, L., Jiang, X., and Liu, Q. Aligning large language models with human: A survey. arXiv preprint arXiv:2307.12966, 2023c.

Wu, J., Xie, Y., Yang, Z., Wu, J., Chen, J., Gao, J., Ding, B., Wang, X., and He, X. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. arXiv preprint arXiv:2407.07880, 2024.

Wu, Z., Hu, Y., Shi, W., Dziri, N., Suhr, A., Ammanabrolu, P., Smith, N. A., Ostendorf, M., and Hajishirzi, H. Fine-grained human feedback gives better rewards for language model training. arXiv preprint arXiv:2306.01693, 2023.

Xiong, W., Dong, H., Ye, C., Wang, Z., Zhong, H., Ji, H., Jiang, N., and Zhang, T. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. In ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models, 2023.

Xu, J., Yang, R., Qiu, S., Luo, F., Fang, M., Wang, B., and Han, L. Tackling data corruption in offline reinforcement learning via sequence modeling. In The Thirteenth International Conference on Learning Representations, 2025.

Yan, Y., Lou, X., Li, J., Zhang, Y., Xie, J., Yu, C., Wang, Y., Yan, D., and Shen, Y. Reward-robust rlhf in llms. arXiv preprint arXiv:2409.15360, 2024.

Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., et al. Qwen2 technical report. arXiv preprint arXiv:2407.10671, 2024a.

Yang, R., Zhong, H., Xu, J., Zhang, A., Zhang, C., Han, L., and Zhang, T. Towards robust offline reinforcement learning under diverse data corruption. In The Twelfth International Conference on Learning Representations.

Yang, R., Bai, C., Ma, X., Wang, Z., Zhang, C., and Han, L. Rorl: Robust offline reinforcement learning via conservative smoothing. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), Advances in Neural Information Processing Systems, volume 35, pp. 23851–23866. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/96bbdd0ed2a9e7cd2fb7caf2fae15f3d-Paper-Conference.pdf.

Yang, R., Pan, X., Luo, F., Qiu, S., Zhong, H., Yu, D., and Chen, J. Rewards-in-context: Multi-objective alignment of foundation models with dynamic preference adjustment. arXiv preprint arXiv:2402.10207, 2024b.

Ye, X., Li, X., Liu, T., Sun, Y., Tong, W., et al. Active negative loss functions for learning with noisy labels. Advances in Neural Information Processing Systems, 36: 6917–6940, 2023.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears. arXiv preprint arXiv:2304.05302, 2023.

Zhang, Z. and Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31, 2018.

Zhao, Y., Khalman, M., Joshi, R., Narayan, S., Saleh, M., and Liu, P. J. Calibrating sequence likelihood improves conditional language generation. arXiv preprint arXiv:2210.00045, 2022.

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., and Liu, P. J. Slic-hf: Sequence likelihood calibration with human feedback. arXiv preprint arXiv:2305.10425, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36: 46595–46623, 2023.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. Advances in Neural Information Processing Systems, 36, 2024.

Zhou, C., Liu, P., Xu, P., Iyer, S., Sun, J., Mao, Y., Ma, X., Efrat, A., Yu, P., Yu, L., et al. Lima: Less is more for alignment. arXiv preprint arXiv:2305.11206, 2023.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. arXiv preprint arXiv:1909.08593, 2019.

A. ROPO Framework

In this section, we describe the overall iterative framework, provide the pseudocode, and analyze the computational cost for ROPO.

Algorithm 1 ROPO

Input: dataset D, β, α, ρ, K, number of epochs M, SFT model π_{sft},
Initialization:
$D^{(0)} \leftarrow D$
$\pi_\theta^{(0)} \leftarrow \pi_{\text{sft}}$
for $m = 1, \ldots, M-1$ **do**
$\pi_{\text{ref}}^{(m)} \leftarrow \pi_\theta^{(m-1)}$ with frozen parameters θ
▶ **Noise-tolerant Training:**
Obtain $\pi_\theta^{(m)}$ by training $\pi_\theta^{(m-1)}$ on $D^{(m-1)}$ with $\pi_{\text{ref}}^{(m)}$ and ℓ_{ropo} in Eq. (6) for one epoch
▶ **Noisy Sample Filtering:**
Compute ℓ_{ropo} with $\pi_\theta^{(m)}$ and $\pi_{\text{ref}}^{(m)}$ for D
$D_{\text{top}-\rho}^{(m)} \leftarrow$ samples with top-ρ ROPO loss value in D
$D_{\text{bot}-(1-\rho)}^{(m)} \leftarrow$ samples with bottom-$(1-\rho)$ ROPO loss value in D
▶ **Robustness-guided Rejection Sampling:**
$D_{\text{new}} \leftarrow \varnothing$
for $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ in $D_{\text{top}-\rho}^{(m)}$ **do**
Sample responses $\widetilde{\mathbf{y}}_1, \ldots, \widetilde{\mathbf{y}}_K$ to $\mathbf{x}$ using $\pi_\theta^{(m)}$
$D_{\text{cand}} \leftarrow \{(\mathbf{x}, \mathbf{y}_1, \widetilde{\mathbf{y}}_k, \mathbf{y}_1 \succ \widetilde{\mathbf{y}}_k \mid \mathbf{x})\}_{k=1}^K \cup \{(\mathbf{x}, \mathbf{y}_2, \widetilde{\mathbf{y}}_k, \mathbf{y}_2 \succ \widetilde{\mathbf{y}}_k \mid \mathbf{x})\}_{k=1}^K$
$D_{\text{new}} \leftarrow D_{\text{new}} \cup \{\arg\min_{\mathbf{z} \in D_{\text{cand}}} \ \ell_{\text{ropo}}(\mathbf{z}, \pi_\theta^{(m)})\}$
end for
$D^{(m)} \leftarrow D_{\text{bot}-(1-\rho)}^{(m)} \cup D_{\text{new}}$
end for
$\pi_{\text{ref}}^{(M)} \leftarrow \pi_\theta^{(M-1)}$ with frozen parameters θ
Obtain $\pi_\theta^{(M)}$ by training $\pi_\theta^{(M-1)}$ on $D^{(M-1)}$ with $\pi_{\text{ref}}^{(M)}$ and ℓ_{ropo} in Eq. (6) for one epoch
Output: $\pi_\theta^{(M)}$

As shown in Figure 3 and Algorithm 1, ROPO iteratively carries out three stages: noise-tolerant training, noisy sample filtering, and robustness-guided rejection sampling. Specifically, in the 1st to $(M-1)$th epochs, we first train the model using the ROPO loss ℓ_{ropo}. After training for one epoch, we compute the value of ROPO loss for all samples in the original dataset and divide them into two subsets (i.e., $D_{\text{top}-\rho}$ and $D_{\text{bot}-(1-\rho)}$) according to their loss values. Then, the robustness-guided rejection sampling stage generates new samples D_{new} based on $D_{\text{top}-\rho}$. The new samples are used together with $D_{\text{bot}-(1-\rho)}$ as training samples for the next epoch. In the last epoch, we only perform the noise-tolerant training stage and then get the final model.

Computational Cost Analysis. ROPO introduces additional costs for the noisy sample filtering and robustness-guided rejection sampling stages compared with non-iterative methods. However, these additional costs are acceptable compared to the training cost and almost negligible in the entire chain of real-world large-scale LLM training. The notations in the computational cost analysis is shown in Table 4. We have

$$\frac{C_{\text{ROPO}}}{C_{\text{non}-\text{it}}} = \frac{MC_{\text{tr}} + (M-1)(C_{\text{fil}} + C_{\text{rs}})}{MC_{\text{tr}}}. \tag{7}$$

Since the main cost of the noisy sample filtering stage per epoch is to compute the loss of N samples, we have $C_{\text{fil}} \approx NC_{\text{loss}} \approx 4NC_{\text{forward}}$. As for the rejection sampling stage, the main costs per epoch come from ρNK response generation and $2\rho NK$ loss computations, hence $C_{\text{rs}} \approx \rho NKC_{\text{gen}} + 2\rho NKC_{\text{loss}} \approx \rho NKC_{\text{forward}} + 8\rho NKC_{\text{forward}} = 9\rho NKC_{\text{forward}}$. Because the training process mainly involves loss computation for two models (i.e., the reference model and the model being trained) and gradient propagation, we have $C_{\text{tr}} \approx N(C_{\text{loss}} + C_{\text{backward}}) \approx N(4C_{\text{forward}} + C_{\text{backward}})$.

Table 4. Notations in the computational cost analysis.

Notation	Description
N, M, ρ, K	Please see Algorithm 1
C_{ROPO}	Cost of ROPO
$C_{\text{non-it}}$	Cost of non-iterative methods
C_{tr}	Cost of (noise-tolerant) training per epoch
C_{fil}	Cost of noisy sample filtering per epoch
C_{rs}	Cost of robustness-guided rejection sampling per epoch
C_{loss}	Cost of computing the loss for a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ without gradient propagation
C_{gen}	Cost of generating a response $\mathbf{y}$ for a query $\mathbf{x}$
C_{forward}	Cost of computing the log-likelihood for a query-response pair $(\mathbf{x}, \mathbf{y})$
C_{backward}	Cost of computing the gradient and updating parameters for a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$

Table 5. The estimated noise rate in commonly-seen datasets. This table is from (Gao et al., 2024).

Dataset	Noise rate (%)	Reference
MT-Bench	15.0-37.0	(Zheng et al., 2023)
TL;DR	21.3-27.0	(Lee et al., 2023)
CBArena	22.0-36.0	(Zheng et al., 2023)
AntHH	27.9-30.9	(Lee et al., 2023)
SHP	35.5-41.9	(Cui et al., 2023)
WebGPT	34.8	(Cui et al., 2023)

Therefore, Eq. (7) leads to

$$\begin{aligned}\frac{C_{\text{ROPO}}}{C_{\text{non-it}}} &\approx \frac{M(4C_{\text{forward}} + C_{\text{backward}}) + (M-1)(4+9\rho K)C_{\text{forward}}}{M(4C_{\text{forward}} + C_{\text{backward}})} \\ &= 1 + \frac{(4+9\rho K)(M-1)}{M} \cdot \frac{C_{\text{forward}}}{4C_{\text{forward}} + C_{\text{backward}}} \\ &= 1 + \frac{(4+9\rho K)(M-1)}{M} \cdot \frac{1}{4 + C_{\text{backward}}/C_{\text{forward}}},\end{aligned} \tag{8}$$

where the ratio $C_{\text{backward}}/C_{\text{forward}}$ is approximately $2-3$ for LLMs. Take $\rho = 0.2, K = 2, M = 3$ as an example, without considering inference acceleration, we can estimate that $C_{\text{ROPO}} \approx 1.6C_{\text{non-it}}$. In practice, we can use inference acceleration methods to increase $C_{\text{backward}}/C_{\text{forward}}$, thereby further reducing the additional cost of ROPO. Compared with the computational cost of the entire chain of real-world LLM training (including continual pre-training and SFT), the additional cost is almost negligible.

B. Related Work

Preference Alignment of LLMs. The most representative paradigm of preference alignment is RLHF (Ziegler et al., 2019; Ouyang et al., 2022), which involves training a reward model to capture human preferences and then steering LLMs towards producing high-reward responses through RL algorithms (Schulman et al., 2017). However, in real applications, RL-based methods are complex and prone to instability during training (Rafailov et al., 2023; Wu et al., 2023; Yuan et al., 2023). Therefore, many recent studies have explored more straightforward and stable alternatives for RLHF (Yuan et al., 2023;

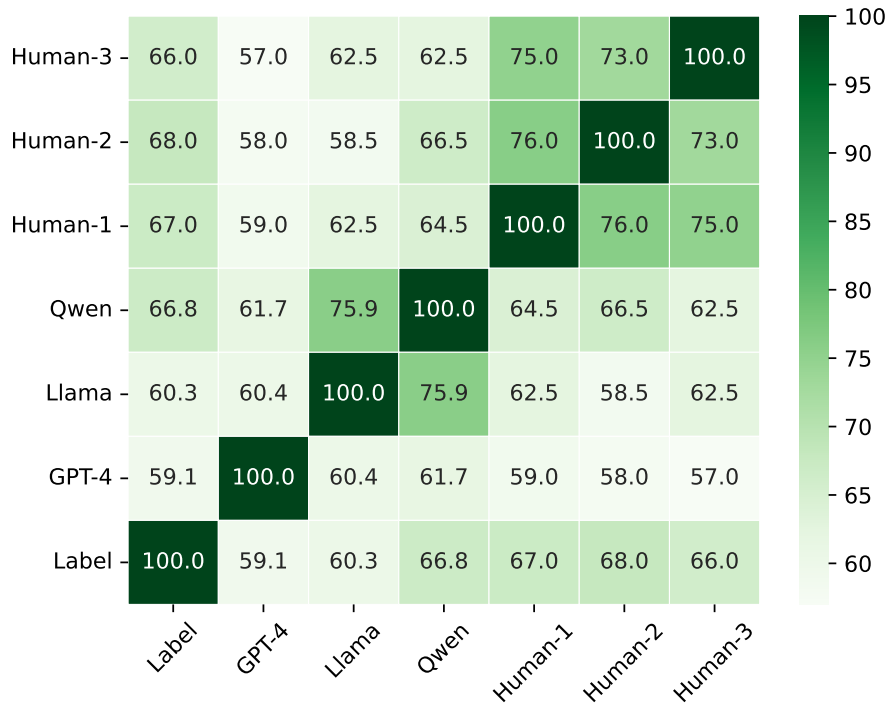


Figure 5. The inter-annotator agreement heap map on the TL;DR dataset. The "Label", "Llama", and "Qwen" refer to the original labels in the dataset, Llama-2-70B, and Qwen-Max, respectively. We assess the preferences of human annotators across 200 randomly selected samples and extend the evaluation to 1,000 samples for LLMs, which include the initial 200 samples.

Rafailov et al., 2023; Song et al., 2023; Wang et al., 2023b; Lin et al., 2023; Li et al., 2023b; Wang et al., 2023c; Zhao et al., 2022). Among these studies, the most promising direction is to use a contrastive or ranking loss to calibrate the likelihood of the output sequence. Specifically, RRHF (Yuan et al., 2023) introduces a ranking loss to encourage larger likelihoods for better responses and smaller likelihoods for worse responses. Besides, another important work is DPO (Rafailov et al., 2023), which implicitly optimizes the same objective as existing RLHF-based methods and enables human preference alignment directly with a simple cross-entropy loss. In addition to the aforementioned methods using data in the form of $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c)$, where c is the preference label, some recent studies (Duan et al., 2024; Ethayarajh et al., 2024; Chen et al., 2024) have also used data in the form of $(\mathbf{x}, \mathbf{y}, \mathbf{c})$, where c is an annotation of the response $\mathbf{y}$, for preference alignment.

Learning from Noisy Data. In the era of deep learning, there is an urgent demand for large-scale training samples, and the cost of manually annotating or filtering data is prohibitively expensive in most circumstances (Song et al., 2022). Therefore, learning from noisy data has become increasingly important, which primarily falls into three categories. The first category is sample-selection based methods (Swayamdipta et al., 2020; Pleiss et al., 2020; Paul et al., 2021; Sorscher et al., 2022), which identify high-quality samples before training and filter out noisy samples. For example, (Swayamdipta et al., 2020) uses the training dynamics to identify valuable samples. The second category is weighting-based methods, which assign greater weights for important samples and lesser weights for noisy samples (Ren et al., 2018; Han et al., 2022; Shu et al., 2019). Besides, another important area of research is dedicated to the design of loss functions that are robust to noise (Ghosh et al., 2017; Wang et al., 2019; Zhang & Sabuncu, 2018). The findings in (Ghosh et al., 2017) indicate that the traditional cross-entropy loss is sensitive to the label noise, while symmetric loss functions are robust to such noise. Furthermore, recent advances in LLMs have also underscored the essential role of data quality in both pre-training and supervised fine-tuning (SFT) phases of LLMs (Marion et al., 2023; Zhou et al., 2023; Korbak et al., 2023).

Rejection Sampling. The rejection sampling is a popular approach of data augmentation to improve the data quality and performance in existing preference alignment methods (Dong et al., 2023; Liu et al., 2023b; Xiong et al., 2023; Yang et al., 2024b; Wang et al., 2024b). Specifically, (Dong et al., 2023) ranks newly-collected responses based on their rewards and selects the highest ranked one to add to the dataset. To address the issue of the excessively high rejection rate and thus improve the effectiveness of rejection sampling, (Xiong et al., 2023) proposes a multi-step sampling technique, which also

requires an external reward model. Besides, (Wang et al., 2024b) and (Yang et al., 2024b) consider rejection sampling for the multi-objective preference alignment, where (Wang et al., 2024b) projects multi-objective reward vectors onto one dimension and then selects samples based on the scalar rewards, while (Yang et al., 2024b) augments samples near the Pareto front of multi-dimensional rewards, leading to a strong multi-objective alignment performance. Compared to the aforementioned methods, which all rely on rewards provided by external models, our robustness-guided rejection sampling technique selects new samples based on loss values that reflect the quality of the samples. Moreover, our technique benefits from being independent of external LLMs, thus leading to computational and memory efficiency.

C. Discussion on Preference Noise

Due to the inherent differences in annotators' preferences, the preference noise is usually unavoidable. In this section, we discuss the definition and identification of preference noise.

Before giving the definition of preference noise, we invite our readers to pay attention to the following two points.

1. This paper focuses on **noisy preferences** rather than the more general noisy preference data. The former refers specifically to the noise in preference labels, while the noise corresponding to the latter may come from multiple factors such as preference labels, text quality, and the matching degree between queries and responses. It is interesting and meaningful to study a wider range of noisy data, but it is beyond the scope of our paper and related work (Mitchell, 2023; Chowdhury et al., 2024; Gao et al., 2024).
2. Like related work (Mitchell, 2023; Chowdhury et al., 2024; Gao et al., 2024), this paper is based on the **Bradley-Terry (BT) model**. The BT model assumes the existence of a "gold", latent, and inaccessible reward model r^*. Then, we can express the BT preference probability $P^*(\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$ for a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ using the reward model r^*. Intuitively, the BT model assumes that there are mainstream preferences in human society that reflect values such as peace, friendliness, honesty, etc. Differently, there are also studies on multifaceted or multidimensional preferences (Lou et al., 2024; Lee et al., 2024), but defining noise for them is challenging because it is difficult to have a "ground truth" label. Therefore, our following discussion is based on the assumption of the BT model.

Definition of preference noise. For a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \hat{c})$, if $P^*(\hat{c}) > 0.5$, then the sample is clean; otherwise, the sample is noisy. Because the BT model usually represents preferences that are consistent with mainstream values of human society, so the formation of such noise is usually caused by the personal preferences or cognitive biases of annotators. Please note that the annotators can be humans or LLMs.

Identification and detection of preference noise. As mentioned above, the definition of preference noise is based on the inaccessible reward model, so we can never identify preference noise accurately. However, we can estimate the noise rate by *using advanced LLMs as the proxy for the BT model* or *computing the inter-annotator agreement*.

- Using advanced LLMs as the proxy for the BT model (Gao et al., 2024). Given a dataset, we can prompt advanced LLMs (e.g., GPT-4, Llama-3-70B-Instruct (Dubey et al., 2024), and Qwen-2-72B-Instruct (Yang et al., 2024a)) to identify the noise. For example, we can provide them with rules and ask them to rate or rank the responses in the dataset. If a sample's new label is different from its original label, it is identified as noisy. The stronger the proxy LLM, the more reliable the noise identification.
- Computing the inter-annotator agreement (Ouyang et al., 2022; Wang et al., 2024a; Bai et al., 2022). We can employ different annotators (humans or LLMs) to relabel the dataset and calculate the agreement between them. For this approach, we should try to ensure that all annotators have the same criteria, and similar cognition and ability. Suppose that we have n annotators and the agreement between annotators i and j is $0 \leq a_{ij} \leq 1$, then the estimated noise rate is $\frac{1}{n(n-1)} \sum_{i,j=1, i \neq j}^{n} (1 - a_{ij})$. Take the TL;DR dataset as an example. We employ GPT-4, Llama-2-70B (Touvron et al., 2023), Qwen-Max (Bai et al., 2023), and three human annotators to relabel the TL;DR dataset. The human annotators are three of the four volunteers mentioned in Appendix E.6. The inter-annotator agreement heat map is shown in Figure 5, which indicates an estimated noise rate of 17.6%.

Besides, Table 5 from (Gao et al., 2024) summarizes the estimated noise rate in some commonly-seen datasets. As can be seen, the existence of preference noise is ubiquitous and cannot be ignored, which highlights the importance of studying robust preference optimization approaches.

D. More Details about Experiments

D.1. Tasks and Datasets

We run experiments on two dialogue datasets (i.e., UltraFeedback Binarized and Alpaca Comparison) and one post summarization dataset (i.e., TL;DR).

- The UltraFeedback Binarized dataset[4] is a pre-processed version of the UltraFeedback dataset (Cui et al., 2023), which contains 64,000 prompts and each prompt has four model responses from various LLMs. Based on the score assigned by GPT-4, (Tunstall et al., 2023) selects two responses for each prompt and construct UltraFeedback Binarized for the preference alignment.
- The Alpaca Comparison dataset contains 52,000 queries from the widely-used Stanford Alpaca dataset (Taori et al., 2023). (Peng et al., 2023) generates several responses using GPT-4 and other LLMs including text-davinci-003 to each query and employs GPT-4 to assign a score for each response.
- In the TL;DR dataset, each prompt is a forum from Reddit, and the model is required to summarize the given forum. Following (Rafailov et al., 2023), we use the Reddit TL;DR summarization dataset (Völske et al., 2017) along with human preferences collected by (Stiennon et al., 2020).

D.2. Baselines, Models, and Hyperparameters

Baselines. Our baselines are DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), and two approaches that use the label smoothing technique to alleviate the impact of noise, i.e., rDPO (Chowdhury et al., 2024) and cDPO (Mitchell, 2023).

Specifically, given a preference data $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2)$ with the ranking label $\mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x}$, the objectives of our baselines are

$$\ell_{\text{dpo}} = -\log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})}\right), \tag{9}$$

$$\ell_{\text{ipo}} = \left(\log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})} - \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})} - \frac{1}{2\beta}\right)^2, \tag{10}$$

$$\begin{aligned}\ell_{\text{rdpo}} = &-\frac{1-\varepsilon}{1-2\varepsilon} \log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})}\right) \\ &+ \frac{\varepsilon}{1-2\varepsilon} \log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})}\right),\end{aligned} \tag{11}$$

$$\begin{aligned}\ell_{\text{cdpo}} = &-(1-\varepsilon) \log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})}\right) \\ &- \varepsilon \log \sigma\left(\beta \log \frac{\pi_\theta(\mathbf{y}_2 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2 \mid \mathbf{x})} - \beta \log \frac{\pi_\theta(\mathbf{y}_1 \mid \mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1 \mid \mathbf{x})}\right),\end{aligned} \tag{12}$$

where $\varepsilon \in (0, \frac{1}{2})$ and $\beta \in (0, 1)$ are hyperparameters.

Models. We use Mistral-7B (Jiang et al., 2023) and Llama-2-7B (Touvron et al., 2023) as base models for all baselines and datasets. On UFB, we use Zephyr-7B-SFT-β (Tunstall et al., 2023) as the SFT model for experiments with Mistral-7B, and adopt the result of Zephyr-7B-β (Tunstall et al., 2023) on AlpacaEval (90.60) as the performance of DPO under no artificial noise. In other cases, we fine-tune base models on the preferred responses (SFT targets) to form the SFT models.

Hyperparameters. We run all experiments on 16 NVIDIA A100 GPUs (80 GB). Unless otherwise noted, we use a global batch size of 512 to train all models. For all hyperparameters **except for ε of label smoothing**, **we search for the best one on each dataset without artificial noise and use the same setting for 20% and 40% artificial noise**.

For all methods, we search the best learning rate in {1e-5, 5e-6, 1e-6, 5e-7, 1e-7} and the best β in {0.1, 0.5}. We find that the best performing learning rate is 1e-6, and the best β for dialogue and post summarization are 0.1 and 0.5, respectively. This conclusion is consistent with that in (Rafailov et al., 2023).

[4] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

For ROPO, we use $\alpha = 14$ and $\rho = 0.2$ in the main experiments. In ablations (Section 4.2), we tune α in $\{6, 14, 30\}$, which makes $\frac{4\alpha}{(1+\alpha)^2}$ be around $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}$, respectively, and tune ρ in $\{0.1, 0.2, 0.3\}$. We set $K = 3$ for the rejection sampling. For rDPO and cDPO, we search the best ε in $\{0{,}05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$ for each dataset and each proportion of artificial noise.

D.3. Evaluation

For models trained on UFB and Alpaca Comparison, we evaluate them on the AlpacaEval benchmark (Li et al., 2023a)—a widely used dialogue benchmark—by comparing their outputs with those of text-davinci-003 (recommended by the benchmark for comparison). AlpacaEval contains 805 queries in various domains and exhibit a strong concordance with ground truth human annotators. For TL;DR, we randomly select 500 queries from the test split of it and evaluate ROPO and baselines by comparing their outputs with the chosen responses (SFT targets) for the queries.

Following existing studies (Rafailov et al., 2023; Tunstall et al., 2023), we employ GPT-4 as the referee to conduct head-to-head comparisons, using the win rate as the metric. On AlpacaEval, we conduct evaluations using the API provided by AlpacaEval. On TL;DR, we use the following prompt, which is similar to that used by AlpacaEval, to conduct GPT-4 evaluation.

```
You are a helpful assistant that ranks models by the quality of their
summaries of given forum posts.

I want you to create a leaderboard of different of large-language models.
To do so, I will give you the instructions (forum posts) given to the
models, and the responses of two models.  Please rank the models based on
which responses would be preferred by humans.

Here is the post:
<Forum Post>

Here are the outputs of the models:
Model 1:  <Summary 1>
Model 2:  <Summary 2>

Now please rank the models by the quality of their answers, so that the
model with rank 1 has the best output.  Please provide the ranking that
the majority of humans would give.  Your response should use the format:
Better:  <Model 1 or Model 2>
```

E. More Experiments

E.1. Experiments on Llama-2-13B-Base and Llama-3-70B-Base

To evaluate ROPO and baselines on models larger than 7B, we supplement experiments on Llama-2-13B-Base and Llama-3-70B-Base.

Experiments on Llama-2-13B-Base. We run SFT on UltraChat-200k for one epoch with the learning rate of 1e-5, the global batch size of 128, the weight decay of 0.1, and a cosine-type learning rate scheduler. Then, we fine-tune the SFT model with ROPO and baselines for two epochs on UFB (under artificial noise ratio of 0 and 20%) with the learning rate of 1e-6 and the global batch size of 512. In the experiments, we fix $\alpha = 14$ and $\rho = 0.2$ for ROPO without tuning them, and tune β in $[0.1, 0.5, 1.0]$ for IPO and tune ε in $[0.1, 0.2, 0.3, 0.4]$ for cDPO and rDPO. The results are shown in Table 6.

Experiments on Llama-3-70B-Base. We run SFT on UltraChat-200k for one epoch with the learning rate of 1e-5, the global batch size of 128, the weight decay of 0.1, and a cosine-type learning rate scheduler. Then, we fine-tune the SFT model with ROPO and DPO for two epochs on UFB (under artificial noise ratio of 0 and 20%) with the learning rate of 5e-7 and the global batch size of 512. We fix $\alpha = 14$ and $\rho = 0.2$ for ROPO without tuning them. The results are shown in Table 7. From the results we can conclude that: (1) 70B models outperform 7B/13B models in terms of win rate. However, the

Table 6. Win rates (%) of **different methods vs SFT targets** under different proportions (i.e., 0 and 20%) of artificial noise, evaluated by GPT-4 on AlpacaEval.

Dataset		UFB	
Model	Method	0%	20%
Llama-2-13B	DPO	<u>82.98</u>	80.50
	IPO	81.99	79.75
	rDPO	81.37	<u>80.87</u>
	cDPO	82.36	80.50
	ROPO	**83.23**	**82.98**

Table 7. Win rates (%) of **ROPO/DPO vs SFT targets** under different proportions (i.e., 0 and 20%) of artificial noise, evaluated by GPT-4 on AlpacaEval.

Dataset		UFB	
Model	Method	0%	20%
Llama-2-70B	DPO	94.29	88.70
	ROPO	**95.53**	**94.04**

Table 8. Win rates (%) of **different methods vs SFT targets** under different proportions (i.e., 0 and 20%) of artificial noise, evaluated by GPT-4 on AlpacaEval.

Dataset		UFB	
Model	Method	0%	20%
Mistral-7B	DPO-RM	69.69	68.32
	cPPO-RM	68.45	67.95
	rPPO-RM	68.70	67.33
	ROPO-RM	**69.94**	**70.43**

performance of the models trained with DPO still has a non-negligible drop under 20% artificial noise. (2) Our ROPO still significantly exceeds DPO on the scale of 70B.

E.2. Experiments on reward modeling

In the main text of our paper, the baselines are reward-free. Considering the reward modeling (RM) still plays an important role in many real-world LLM applications, although RM is not our focus, we supplement experiments on RM with Mistral-7B-Base to test the potential of ROPO in scenarios including reward modeling. Given a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c = 0)$, if we denote $P = \sigma(r(\mathbf{x}, \mathbf{y}_1) - r(\mathbf{x}, \mathbf{y}_2))$, then the RM-training losses of ROPO and our baselines are as follows.

- DPO-RM: $-\log P$
- cPPO-RM: $-(1-\varepsilon)\log P - \varepsilon \log(1-P)$
- rPPO-RM: $-\frac{1-\varepsilon}{1-2\varepsilon}\log P + \frac{\varepsilon}{1-2\varepsilon}\log(1-P)$
- ROPO-RM (Ours): $-(4\alpha/(1+\alpha)^2)\cdot P + (4\alpha^2/(1+\alpha)^2)\cdot(1-P)$

We train Mistral-7B-v0.1 on UFB for two epochs with the aforementioned losses to obtain reward models. Then, we use Best of N Sampling ($N = 16$) to generate responses based on RMs and Mistral-7B-SFT-Beta (SFT model). We use the learning rate of 5e-6, the batch size of 512, and a cosine-type learning rate scheduler. The results are shown in Table 8

Table 9. Win rates (%) of **different methods vs SFT targets** under noise coming from the annotators' trust in larger models over smaller ones, evaluated by GPT-4 on AlpacaEval.

Dataset		UFB
Model	Method	
Mistral-7B	DPO	75.16
	IPO	72.55
	cDPO	76.27
	rDPO	78.26
	ROPO	**80.50**

Table 10. Win rates (%) of **different methods vs SFT targets** under noise coming from LLM preference comparisons, evaluated by GPT-4 on AlpacaEval.

Dataset		UFB
Model	Method	
Mistral-7B	DPO	84.22
	IPO	84.84
	cDPO	85.22
	rDPO	86.21
	ROPO	**88.07**

E.3. More practical noise settings

The experiments in the main text cover two types of practical noise as follows.

1. **Practical noise coming from human comparisons.** In the original TL;DR dataset, the preferences are labeled by human annotators who compare the post-summaries generated by different models in pairs. This leads to unavoidable noise due to the diversity of human preferences.

2. **Practical noise coming from LLM (GPT-4) rating.** Each query (instruction) in the original UltraFeedback dataset has four responses coming from different models. GPT-4 scores them based on criteria such as instruction-following, honesty, helpfulness, etc. Then, for each query, the highest ranked response is selected as "preferred", and one of the remaining responses is randomly selected as "dis-preferred". This leads to unavoidable noise due to the bias of GPT-4.

In this section, we explore another two practical noise settings in Appendices E.3.1 and E.3.2.

E.3.1. Experiments under noise coming from annotators' trust in larger models over smaller ones

It is common practice to treat the response from a larger model as the chosen (preferred) one and the response from a smaller model as the rejected (dis-preferred) one. Therefore, we obtain new noisy preferences from UFB (each of query has four LLM responses) based on the sizes of models that generate the responses. As shown in Table 9, under this practical noise setting, ROPO still significantly outperforms DPO and other baselines.

E.3.2. Experiments under noise coming from LLM preference comparisons

We use Llama-3-70B-Instruct (Dubey et al., 2024), which is one of the most advanced open source LLM, to relabel the preferences in UFB. To make the labels as reliable as possible, we instruct the model to list the advantages of each response. The prompt we use is as follows.

```
For the given instruction and two responses (A and B), please answer:
(1) which response is better overall, (2) the aspects in which A is
superior to B, and (3) the aspects in which B is superior to A.

Strictly adhere to the following rules:
1.  Answer in bullet points, with each point starting with a gerund or
adjective, excluding the words ``response A'' and ``response B''.
2.  If a response has no superior aspects over another, output NONE.

Instruction:
{instruction}

Response A
{responseA}

Response B
{responseB}

Your answer MUST STRICTLY follow the format as follows: **Better**
<Choose A or B>

**Why A is better than B**
```

```
- <First aspect for which A is superior to B>
- <Continue with other points if any>

**Why B is better than A**
- <First aspect for which B is superior to A>
- <Continue with other points if any>
```

However, we observe that **about 30% of the labels are different from those in the original UFB dataset**. This shows that noise are unavoidable due to the diversity in LLM preferences. Then, we train Mistral-7B with different methods on the new noisy dataset. As shown in Table 10, under this practical noise setting, ROPO still significantly outperforms DPO and other baselines.

Table 11. Performance of difference methods on Arena-Hard and MT-Bench. The bold font indicates the best result and an underline indicates the second-best result.

Benchmark		Arena-Hard			MT-Bench		
Model	Method	0%	20%	40%	0%	20%	40%
Mistral-7B	DPO	<u>10.7</u>	8.5	6.3	**7.3**	5.7	4.3
	IPO	9.2	7.9	7.3	<u>7.2</u>	5.9	4.9
	rDPO	9.8	<u>9.2</u>	<u>8.9</u>	7.1	<u>6.4</u>	<u>5.8</u>
	cDPO	10.3	9.0	8.4	<u>7.2</u>	6.2	5.2
	ROPO	**13.1**	**12.6**	**11.8**	**7.3**	**6.9**	**6.5**
Llama-3-8B	DPO	17.9	15.3	14.1	**7.8**	6.1	4.6
	IPO	<u>18.6</u>	16.8	16.0	7.4	6.3	5.0
	rDPO	18.3	<u>17.5</u>	<u>17.1</u>	7.5	<u>6.9</u>	<u>6.1</u>
	cDPO	17.5	16.4	15.3	<u>7.7</u>	6.7	5.8
	ROPO	**20.5**	**19.6**	**18.5**	<u>7.7</u>	**7.0**	**6.7**

Table 12. Win rates (%) of **different variants of DPO vs SFT targets** under 0% and 20% artificial noise, evaluated by GPT-4 on AlpacaEval. The base model is Mistral-7B and the training dataset is UFB.

	0%	20%
DPO	**90.60**	**86.21**
DPO + NSF	90.06	85.09
DPO + NSF + RS	90.31	84.97

E.4. Experiments on more benchmarks

To comprehensively explore the performance of ROPO and baseline methods, we evaluate them on another two widely-used benchmarks, i.e., Arena-Hard (Li et al., 2024) and MT-Bench (Zheng et al., 2023). The details of the benchmarks are as follows.

- MT-Bench (Zheng et al., 2023) contains 80 two-turn conversations, each of which has an open-ended instruction and a corresponding follow-up question. Due to the well-designed questions and the wide coverage of topics, MT-Bench has become a widely-used benchmark to evaluate the multi-turn conversational and instruction-following abilities of AI models.
- Arena-Hard (Li et al., 2024) is a challenging benchmark containing 500 single-turn conversations. Compared to AlpacaEval and MT-Bench, Arena-Hard features better model separability, tighter confidence intervals, and achieves a correlation of 98.6% with Chatbot Arena rankings (Chiang et al., 2024).

We evaluate ROPO and baseline methods using Mistral-7B and Llama-3-8B (Dubey et al., 2024). For Mistral-7B, we use the same models as evaluated on AlpacaEval in the main experiments. For Llama-3-8B, we first train a Llama-3-8B-Base[5] on UltraChat-200k[6] to obtain an SFT model (one epoch with the learning rate of 1e-5, global batch size of 128, weight decay of 0.1, and a consine-type learning rate scheduler), and then continue training with ROPO and baseline methods. The results are shown in Table 11. As observed, under various artificial noise levels, ROPO consistently outperforms baseline methods in most cases and demonstrates superior robustness in noisy scenarios.

E.5. Experiments of combining DPO with noisy samples filtering and rejection sampling

As shown in Figure 2, the distributions of the DPO loss on clean and noisy samples are very similar, and the difference gradually decreases as the training proceeds. This shows that the DPO loss is prone to overfitting to noise, hence cannot serve as a reliable measure of model uncertainty in noisy scenarios. In this section, to further support our claim, we conduct experiments of combining DPO with noisy samples filtering (NSF) and rejection sampling (RS) using Mistral-7B as the base model and UFB as the training dataset. Please note that our proposed robustness-guided RS only works on the filtered samples, so we do not conduct experiments combining DPO and RS alone. The results are shown in Table 12. As can be seen, the incorporation of noisy samples filtering and rejection sampling degrades the performance of DPO, especially at 20% artificial noise.

E.6. Human evaluation

We invite four lab members with no conflicts of interest to this paper to serve as volunteers to conduct human evaluations. Two of them are PhDs and the other two are doctoral students, so we believe that they have the ability to understand the evaluation rules and make reliable judgments.

We randomly select 200 queries from the AlpacaEval benchmark. Then, we pair the corresponding responses of ROPO, DPO, and rDPO under 0% and 20% artificial noise to form four groups: (1) ROPO vs DPO under 0% artificial noise, (2) ROPO vs rDPO under 0% artificial noise, (3) ROPO vs DPO under 20% artificial noise, and (4) ROPO vs rDPO under 20% artificial noise.

[5] https://huggingface.co/meta-llama/Meta-Llama-3-8B
[6] https://huggingface.co/datasets/HuggingFaceH4/ultrachat_200k

Table 13. Human evaluation of ROPO vs DPO and ROPO vs rDPO on AlpacaEval. The base model is Mistral-7B and the training dataset is UFB. The #(Win), #(Tie), and #(Lose) are the numbers of ROPO's wins, ties, and ROPO's losses.

Artificial Noise Ratio	0%				20%			
	#(Win)	#(Tie)	#(Lose)	WR (%)	#(Win)	#(Tie)	#(Lose)	WR (%)
ROPO vs DPO	77	69	54	55.8	103	59	38	66.5
ROPO vs rDPO	84	63	53	57.8	89	64	47	60.5

Table 14. Win rates (%) of **DPO with confidence penalty vs SFT targets** under 20% and 40% artificial noise, evaluated by GPT-4 on AlpacaEval. The base model is Mistral-7B and the training dataset is UFB.

	20%	40%
DPO	86.21	**82.67**
DPO + CP	**86.96**	81.86

For each group, we randomly shuffle the order of the queries and the order of responses in each pair. Each volunteer is in charge of one group. None of the volunteers know which method corresponds to each response. They are asked to compare the responses in 200 pairs and choose the better one. If they are unsure about which response is better, they can choose "Tie". During the evaluation process, we allow the volunteers to use translation tools and search engines.

We count the number of ROPO's wins, ties, and losses, and compute the win rate of ROPO by $\Omega = \frac{\#(\text{Win})+\#(\text{Tie})/2}{200}$. The results are shown in Table 13. We have the following interesting observations from the table: (1) The win rate of ROPO against DPO and rDPO is consistently over 55%, demonstrating ROPO's advantages over the baselines. (2) As the artificial noise rate increases, the win rate of ROPO increases to more than 60%, which shows the superiority of ROPO in noisy scenarios. (3) All four volunteers give at least 29% tie judgments, indicating the limitations of human evaluation: it is challenging for most human evaluators to make reliable evaluations on difficult tasks such as long-context reasoning, coding, mathematics, etc. This highlights the importance of developing automated LLM evaluation tools.

E.7. Experiments of applying regularization strategies to DPO

In experiments in the main text, we have evaluate the performance of label smoothing (i.e., cDPO and rDPO) under noisy scenarios. The label smoothing techniques can be seen as regularization strategies applied to DPO. As shown in the experiments, they bring performance improvements over DPO under 20% and 40% artificial noise, but underperform ROPO. Their limited effectiveness might be attributed to the fact that rDPO and cDPO are noise-tolerant only under specific conditions: when the hyperparameter ε exactly matches the noise proportion for rDPO, and when $\varepsilon = 0.5$ for cDPO. Achieving these conditions in practice is challenging due to the lack of prior knowledge about the exact noise proportion.

In this section, we explore another two widely-used types of regularization strategies in noisy scenarios, i.e., the *normalized negative loss* and *confidence penalty*.

- Normalized negative loss (NNL) (Ye et al., 2023), such as normalized negative cross entropy (NNCE) and normalized negative focal loss (NNFL), are shown to be effective when combined with the cross-entropy loss (i.e., the DPO loss in preference optimization). However, when the problem is binary classification like preference comparison, NNCE and NNFL degenerate into constant terms. Specifically, for a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, if we denote $P = \sigma(r(\mathbf{x}, \mathbf{y}_1) - r(\mathbf{x}, \mathbf{y}_2))$, then we have

$$\begin{aligned}\ell_{\text{nnce}} &= 1 - \frac{-\log\min(P, 1-P) + \log P}{-2\log\min(P, 1-P) + \log P + \log(1-P)} \\ &= \begin{cases} 1, & \text{if } P \le 0.5, \\ 0, & \text{if } P > 0.5, \end{cases}\end{aligned}$$

and

$$\ell_{\text{nnfl}} = 1 - \frac{-(1-\min(P,1-P))^\gamma \log\min(P,1-P) + (1-P)^\gamma \log P}{-2(1-\min(P,1-P))^\gamma \log\min(P,1-P) + (1-P)^\gamma \log P + P^\gamma \log(1-P)}$$
$$= \begin{cases} 1, & \text{if } P \le 0.5, \\ 0, & \text{if } P > 0.5. \end{cases}$$

Therefore, NNL does not work for DPO.

- Confidence penalty (CP) (Pereyra et al., 2017) is an entropy-aware regularizer for the cross-entropy loss, which prevents the model from making overconfident inferences. Specifically, for a sample $(\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_1 \succ \mathbf{y}_2 \mid \mathbf{x})$, if we denote $P_\theta = \sigma(r_\theta(\mathbf{x}, \mathbf{y}_1) - r_\theta(\mathbf{x}, \mathbf{y}_2))$, CP computes the entropy by

$$H_\theta = -P_\theta \log P_\theta - (1-P_\theta)\log(1-P_\theta).$$

Then, the CP regularizer is

$$\ell_{\text{cp}} = -\lambda \max(0, \gamma - H_\theta).$$

We combine DPO with CP and tune the hyperparameters λ and γ in the range of $\lambda \in \{0.01, 0.1\}$ and $\gamma \in \{0.1, 0.25, 0.5\}$. As shown in Table 14, we do not observe a significant improvement over DPO in noisy scenarios. We speculate that the limited effectiveness of CP is because CP has no guaranteed noise-tolerance.

F. Mathematical Derivations and Theoretical Analysis

F.1. Proof of Theorem 3.1

Proof. As $\sum_{i=1}^N w_i = N_\rho$ is a hyperplane and $w_i \in [0,1]$ for $i = 1, \dots, N$, $S \triangleq \{\mathbf{w} : w_i \in [0,1], \sum_{i=1}^N w_i = N_\rho\}$ is compact. Because Θ is compact, $\Theta \times S$ is compact. Therefore, the continuous $\frac{1}{N}\sum_{i=1}^N w_i \ell\left(\theta; \mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)}, \pi_\theta\right)$ admits an optimal solution $(\theta^*, \mathbf{w}^*)$ on $\Theta \times S$.

Assume that $\ell\left(\theta^*; \mathbf{x}^{(i_1)}, \mathbf{y}_1^{(i_1)}, \mathbf{y}_2^{(i_1)}, \pi_{\theta^*}\right) < \cdots < \ell\left(\theta^*; \mathbf{x}^{(i_N)}, \mathbf{y}_1^{(i_N)}, \mathbf{y}_2^{(i_N)}, \pi_{\theta^*}\right)$ but with $w_{i_j}^* < 1$ for some $1 \le j \le N_\rho$. Then, we have

$$\sum_{k=1}^{N_\rho} w_{i_k}^* < 1 + (N_\rho - 1) = N_\rho, \tag{13}$$

hence there exists $w_{i_l}^* > 0$ for some $N_\rho < l \le N$. By letting $w'_{i_j} = 1$, $w'_{i_l} = w_{i_j}^* + w_{i_l}^* - 1$, and $w'_{i_k} = w_{i_k}^*$ for $k \ne j, l$, we have $\sum_{k=1}^N w'_{i_k} = 1$ and

$$\begin{aligned}
\frac{1}{N}\sum_{i=1}^N w_i' \ell\left(\theta^*; \mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)}, \pi_{\theta^*}\right) &= \frac{1}{N}\sum_{k\ne j,l} w'_{i_k} \ell\left(\theta^*; \mathbf{x}^{(i_k)}, \mathbf{y}_1^{(i_k)}, \mathbf{y}_2^{(i_k)}, \hat{c}^{(i_k)}, \pi_{\theta^*}\right) \\
&\quad + w'_{i_j} \ell\left(\theta^*; \mathbf{x}^{(i_j)}, \mathbf{y}_1^{(i_j)}, \mathbf{y}_2^{(i_j)}, \hat{c}^{(i_j)}, \pi_{\theta^*}\right) \\
&\quad + w'_{i_l} \ell\left(\theta^*; \mathbf{x}^{(i_l)}, \mathbf{y}_1^{(i_l)}, \mathbf{y}_2^{(i_l)}, \hat{c}^{(i_l)}, \pi_{\theta^*}\right) \\
&< \frac{1}{N}\sum_{k\ne j,l} w_{i_k}^* \ell\left(\theta^*; \mathbf{x}^{(i_k)}, \mathbf{y}_1^{(i_k)}, \mathbf{y}_2^{(i_k)}, \hat{c}^{(i_k)}, \pi_{\theta^*}\right) \\
&\quad + w_{i_j}^* \ell\left(\theta^*; \mathbf{x}^{(i_j)}, \mathbf{y}_1^{(i_j)}, \mathbf{y}_2^{(i_j)}, \hat{c}^{(i_j)}, \pi_{\theta^*}\right) \\
&\quad + w_{i_l}^* \ell\left(\theta^*; \mathbf{x}^{(i_l)}, \mathbf{y}_1^{(i_l)}, \mathbf{y}_2^{(i_l)}, \hat{c}^{(i_l)}, \pi_{\theta^*}\right) \\
&= \frac{1}{N}\sum_{i=1}^N w_i^* \ell\left(\theta^*; \mathbf{x}^{(i)}, \mathbf{y}_1^{(i)}, \mathbf{y}_2^{(i)}, \hat{c}^{(i)}, \pi_{\theta^*}\right),
\end{aligned} \tag{14}$$

which leads to a contradiction. Therefore, we must have $w_{i_k}^* = 1$ for $1 \le k \le N_\rho$ and $w_{i_k}^* = 0$ for $N_\rho < k \le N$. □

F.2. Proof of Theorem 3.2

Proof. For $\ell = \ell_{\text{dpo}}$, we have

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c})\sim\mathcal{D}_\eta}[\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{c|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2}\mathbb{E}_{\hat{c}|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c}[\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\Big[(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})(1-\eta)+(1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))\eta)\cdot\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,0,\pi_\theta) \\
&\qquad + (P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta+(1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))(1-\eta))\cdot\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,1,\pi_\theta)\Big] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\Big[-(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})+\eta-2P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta)\log P_\theta(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) \\
&\qquad - (2P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta+1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})-\eta)\log(1-P_\theta(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))\Big]. \qquad (15)
\end{aligned}
$$

Consider

$$
f(p) = -(p^*+\eta-2p^*\eta)\log p - (2p^*\eta+1-p^*-\eta)\log(1-p), \qquad (16)
$$

we have

$$
f'(p) = -\frac{p^*+\eta-2p^*\eta}{p} + \frac{2p^*\eta+1-p^*-\eta}{1-p}. \qquad (17)
$$

From $f'(p)$ we know that f decrease when $p \le p^*+\eta-2p^*\eta$ and increases when $p \ge p^*+\eta-2p^*\eta$, which means that f reaches its minimum at $p_0 = p^*+(1-2p^*)\eta$.

Therefore, Eq. (15) reaches its minimum when

$$
P_{\theta^*_\eta}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) = P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) + (1-2P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))\eta \qquad (18)
$$

for any $(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)$. Specifically, for $\eta=0$, we have $P_{\theta^*}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) = P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})$, which leas to

$$
\big|P_{\theta^*_\eta}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) - P_{\theta^*}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\big| = 2\eta\big|P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})-1/2\big|. \qquad (19)
$$

□

F.3. Proof of Theorem 3.3

Proof. For samples $(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}=c^{(1)})$ and $(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}=1-c^{(2)})$, according to Eq. (18), we have

$$
\begin{aligned}
P_{\theta^*_\eta}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}\right) &= P_{\theta^*_\eta}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},c^{(1)}\right) \\
&= P^*(c^{(1)}) + (1-2P^*(c^{(1)}))\eta \qquad (20)
\end{aligned}
$$

and

$$
\begin{aligned}
P_{\theta^*_\eta}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}\right) &= P_{\theta^*_\eta}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},1-c^{(2)}\right) \qquad (21) \\
&= P^*(1-c^{(2)}) + (1-2P^*(1-c^{(2)}))\eta \\
&= 1-P^*(c^{(2)}) + (2P^*(c^{(2)})-1)\eta. \qquad (22)
\end{aligned}
$$

Therefore, to ensure that

$$
\ell_{\text{dpo}}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}\right) - \ell_{\text{dpo}}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}\right) < 0, \qquad (23)
$$

we must have

$$
-\log\left(P^*(c^{(1)}) + (1-2P^*(c^{(1)}))\eta - \varepsilon\right) < -\log\left(1-P^*(c^{(2)}) + (2P^*(c^{(2)})-1)\eta + \varepsilon\right), \qquad (24)
$$

which is equivalent to

$$\varepsilon < \frac{1-2\eta}{2}\left(P^*(c^{(1)}) + P^*(c^{(2)}) - 1\right). \tag{25}$$

□

F.4. Detailed Derivation of Eq. (6)

From the definition of w_{ropo} we have

$$w_{\text{ropo}} = \frac{4\alpha}{(1+\alpha)^2}\sigma(\Delta(\mathbf{y}_2,\mathbf{y}_1,\mathbf{x})) + \frac{4\alpha^2}{(1+\alpha)^2}\sigma(\Delta(\mathbf{y}_2,\mathbf{y}_1,\mathbf{x}))\sigma(\Delta(\mathbf{y}_1,\mathbf{y}_2,\mathbf{x})). \tag{26}$$

According to Eq. (4) we know that

$$-\int \beta\frac{4\alpha}{(1+\alpha)^2}\sigma(\Delta(\mathbf{y}_2,\mathbf{y}_1,\mathbf{x}))\nabla\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})}\,\mathrm{d}\theta = \frac{4\alpha}{(1+\alpha)^2}\ell_{\text{dpo}}. \tag{27}$$

Beside, note that for $\sigma(x) = \frac{e^x}{1+e^x}$, we have

$$\sigma'(x) = \left(\frac{e^x}{1+e^x}\right)' = \frac{e^x(1+e^x) - e^x\cdot e^x}{(1+e^x)^2} = \frac{e^x}{(1+e^x)^2} = \frac{e^x}{1+e^x}\cdot\frac{1}{1+e^x} = \sigma(x)\sigma(-x) \tag{28}$$

and

$$\sigma'(-x) = -\sigma(x)\sigma(-x). \tag{29}$$

Letting

$$t(\theta) = \beta\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1\mid\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2\mid\mathbf{x})}, \tag{30}$$

we have

$$\nabla_\theta t(\theta) = \beta\nabla\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})} \tag{31}$$

Hence,

$$\begin{aligned}
&-\frac{4\alpha^2}{(1+\alpha)^2}\int\beta\sigma(\Delta(\mathbf{y}_2,\mathbf{y}_1,\mathbf{x}))\sigma(\Delta(\mathbf{y}_1,\mathbf{y}_2,\mathbf{x}))\nabla\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})}\,\mathrm{d}\theta\\
&= \frac{4\alpha^2}{(1+\alpha)^2}\int\left(-\sigma(t(\theta))\sigma(-t(\theta))\right)\cdot\left(\beta\nabla\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})}\right)\mathrm{d}\theta\\
&= \frac{4\alpha^2}{(1+\alpha)^2}\int\nabla_{t(\theta)}\sigma(-t(\theta))\cdot\nabla_\theta t(\theta)\,\mathrm{d}\theta\\
&= \frac{4\alpha^2}{(1+\alpha)^2}\int\nabla_\theta\sigma(-t(\theta))\,\mathrm{d}\theta\\
&= \frac{4\alpha^2}{(1+\alpha)^2}\sigma(-t(\theta))\\
&= \frac{4\alpha^2}{(1+\alpha)^2}\cdot\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_2\mid\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_2\mid\mathbf{x})} - \beta\log\frac{\pi_\theta(\mathbf{y}_1\mid\mathbf{x})}{\pi_{\text{ref}}(\mathbf{y}_1\mid\mathbf{x})}\right),
\end{aligned} \tag{32}$$

where we omit the constant term of the primitive function.

F.5. Proof of Theorem 3.4

Proof. For $\ell = \ell_{\text{na}}$, we have

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c})\sim\mathcal{D}_\eta}[\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\mathbb{E}_{c|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2}\mathbb{E}_{\hat{c}|\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c}[\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\Big[(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})(1-\eta)+(1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))\eta)\cdot\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,0,\pi_\theta) \\
&\qquad\qquad +(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta+(1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}))(1-\eta))\cdot\ell(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,1,\pi_\theta)\Big] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)}\Big[\left(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})+\eta-2P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta\right)(1-P_\theta(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})) \\
&\qquad\qquad +(2P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\eta+1-P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})-\eta)\,P_\theta(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})\Big].
\end{aligned}
\tag{33}
$$

Consider

$$
\begin{aligned}
f(p) &= (p^*+\eta-2p^*\eta)(1-p)+(2p^*\eta+1-p^*-\eta)p \\
&= (1-2\eta)(1-2p^*)p+(p^*+\eta-2p^*\eta).
\end{aligned}
\tag{34}
$$

Therefore, when $p^* > 1/2$, $f(p)$ reaches its minimum at $p=1$; when $p^* < 1/2$, $f(p)$ reaches its minimum at $p=0$. This means that the optimal point of $f(p)$ is $p_0 = \mathbb{I}(p^* > 1/2)$.

Therefore, Eq. (33) reaches its minimum when

$$
P_{\theta_\eta^*}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) = \mathbb{I}\left(P^*(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x})>\frac{1}{2}\right)
\tag{35}
$$

for any $(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)$. Obviously, we have

$$
P_{\theta_\eta^*}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}) = P_{\theta^*}(\mathbf{y}_1\succ\mathbf{y}_2\mid\mathbf{x}).
\tag{36}
$$

□

F.6. Proof of Theorem 3.5

Proof. For samples $(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}=c^{(1)})$ and $(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}=1-c^{(2)})$. Without loss of generality, we only need to consider two cases: (1) $c^{(1)}=c^{(2)}=0$ and (2) $c^{(1)}=0, c^{(2)}=1$. For the first case, we have

$$
\ell_{\text{na}}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}\right) = P_\theta(\mathbf{y}_2^{(1)}\succ\mathbf{y}_1^{(1)}\mid\mathbf{x})\in[0,\varepsilon)
\tag{37}
$$

and

$$
\ell_{\text{na}}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}\right) = P_\theta(\mathbf{y}_1^{(2)}\succ\mathbf{y}_2^{(2)}\mid\mathbf{x})\in(1-\varepsilon,1].
\tag{38}
$$

For the second case, we have

$$
\ell_{\text{na}}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}\right) = P_\theta(\mathbf{y}_2^{(1)}\succ\mathbf{y}_1^{(1)}\mid\mathbf{x})\in[0,\varepsilon)
\tag{39}
$$

and

$$
\ell_{\text{na}}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}\right) = P_\theta(\mathbf{y}_2^{(2)}\succ\mathbf{y}_1^{(2)}\mid\mathbf{x})\in(1-\varepsilon,1].
\tag{40}
$$

Therefore, to ensure that

$$
\ell_{\text{na}}\left(\mathbf{x}^{(1)},\mathbf{y}_1^{(1)},\mathbf{y}_2^{(1)},\hat{c}^{(1)}\right) < \ell_{\text{na}}\left(\mathbf{x}^{(2)},\mathbf{y}_1^{(2)},\mathbf{y}_2^{(2)},\hat{c}^{(2)}\right),
\tag{41}
$$

we must have $\varepsilon < \frac{1}{2}$. □

F.7. rDPO and cDPO Are Not Noise-Tolerant In Most Cases

Proof. According to Lemma 3.2 in (Chowdhury et al., 2024), the noise-tolerance of rDPO is only guaranteed when the proportion of noise, i.e., η_0, exactly equals the hyperparameter ε.

Next we show that ℓ_{cdpo} is not noise-tolerant for $\varepsilon \in (0, \frac{1}{2})$. Let

$$\begin{aligned}\mathcal{L}_{\text{cdpo}}(\theta) &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c,\pi_\theta)],\\ \mathcal{L}^{\eta_0}_{\text{cdpo}}(\theta) &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c})\sim\mathcal{D}_{\eta_0}}[\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)],\end{aligned}$$

and assume that θ^* and $\theta^*_{\eta_0}$ are the minimizers of $\mathcal{L}_{\text{cdpo}}$ and $\mathcal{L}^{\eta_0}_{\text{cdpo}}$, respectively. For any θ in the space of parameters, we have

$$\begin{aligned}&\mathcal{L}^{\eta_0}_{\text{cdpo}}(\theta)\\ &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}\mathbb{E}_{\hat{c}|(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)}[\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c},\pi_\theta)]\\ &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[(1-\eta_0)\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c,\pi_\theta)+\eta_0\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,1-c,\pi_\theta)]\\ &= (1-\eta_0)\mathcal{L}_{\text{cdpo}}(\theta)+\eta_0\mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[\ell_{\text{cdpo}}(\theta;\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,1-c,\pi_\theta)].\end{aligned} \tag{42}$$

Next, we give a counter-example to show that ℓ_{cdpo} is not noise-tolerant. Suppose that

$$P\left((\mathbf{x},\mathbf{y}_1,\mathbf{y}_2)=(\mathbf{x}^{(0)},\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)})\right)=1 \quad \text{and} \quad \mathbf{y}_1^{(0)}\succ\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)}, \tag{43}$$

where $\mathbf{x}^{(0)}$ is a fixed input and $(\mathbf{y}_1^{(0)},\mathbf{y}_2^{(0)})$ is a fixed pair of responses. Hence Eq. (42) becomes

$$\begin{aligned}&\mathcal{L}^{\eta_0}_{\text{cdpo}}(\theta)\\ &= (2\varepsilon\eta_0-\eta_0-\varepsilon)\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}-\beta\log\frac{\pi_\theta(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}\right)\\ &\quad+(\eta_0+\varepsilon-2\varepsilon\eta_0-1)\log\sigma\left(\beta\log\frac{\pi_\theta(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}-\beta\log\frac{\pi_\theta(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}\right).\end{aligned} \tag{44}$$

Let

$$\Delta(\theta)=\beta\log\frac{\pi_\theta(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)}\mid\mathbf{x}^{(0)})}-\beta\log\frac{\pi_\theta(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)}\mid\mathbf{x}^{(0)})}, \tag{45}$$

then Eq. (44) becomes

$$\mathcal{L}^{\eta_0}_{\text{cdpo}}(\theta)=(2\varepsilon\eta_0-\eta_0-\varepsilon)\log\sigma(\Delta(\theta))+(\eta_0+\varepsilon-2\varepsilon\eta_0-1)\log\sigma(-\Delta(\theta)). \tag{46}$$

We have

$$\begin{aligned}\theta^* &= \arg\min_{\theta\in\Theta}\ \mathcal{L}_{\text{cdpo}}\\ &= \arg\min_{\theta\in\Theta}\ -\varepsilon\log\sigma(\Delta(\theta))-(1-\varepsilon)\log\sigma(\Delta(-\theta))\\ &\in \left\{\theta\in\Theta:\Delta(\theta)=\log\frac{\varepsilon}{1-\varepsilon}\right\},\end{aligned} \tag{47}$$

and

$$\begin{aligned}\theta^*_{\eta_0} &= \arg\min_{\theta\in\Theta}\ \mathcal{L}^{\eta_0}_{\text{cdpo}}\\ &= \arg\min_{\theta\in\Theta}\ (2\varepsilon\eta_0-\eta_0-\varepsilon)\log\sigma(\Delta(\theta))+(\eta_0+\varepsilon-2\varepsilon\eta_0-1)\log\sigma(-\Delta(\theta))\\ &\in \left\{\theta\in\Theta:\Delta(\theta)=\log\frac{\eta_0+\varepsilon-2\varepsilon\eta_0}{1-\eta_0-\varepsilon+2\varepsilon\eta_0}\right\}.\end{aligned} \tag{48}$$

Hence $\theta^* = \theta^*_{\eta_0}$ if and only if

$$\frac{\varepsilon}{1-\varepsilon} = \frac{\eta_0 + \varepsilon - 2\varepsilon\eta_0}{1 - \eta_0 - \varepsilon + 2\varepsilon\eta_0}, \tag{49}$$

which means that $\varepsilon = \frac{1}{2}$. However, $\varepsilon \in (0, \frac{1}{2})$. Therefore, $\theta^* \neq \theta^*_{\eta_0}$ and thus ℓ_{cdpo} is not noise-tolerant. □

F.8. IPO Is Not Noise-Tolerant

Proof. Let

$$\begin{aligned}
\mathcal{L}_{\text{ipo}}(\theta) &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c, \pi_\theta)], \\
\mathcal{L}^{\eta_0}_{\text{ipo}}(\theta) &= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,\hat{c})\sim\mathcal{D}_{\eta_0}}[\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \hat{c}, \pi_\theta)],
\end{aligned}$$

and assume that θ^* and $\theta^*_{\eta_0}$ are the minimizers of $\mathcal{L}_{\text{ipo}}$ and $\mathcal{L}^{\eta_0}_{\text{ipo}}$, respectively. For any θ in the space of parameters, we have

$$\begin{aligned}
&\mathcal{L}^{\eta_0}_{\text{ipo}}(\theta) \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}\mathbb{E}_{\hat{c}|(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)}[\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \hat{c}, \pi_\theta)] \\
&= \mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[(1-\eta_0)\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, c, \pi_\theta) + \eta_0\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, 1-c, \pi_\theta)] \\
&= (1-\eta_0)\mathcal{L}_{\text{ipo}}(\theta) + \eta_0\mathbb{E}_{(\mathbf{x},\mathbf{y}_1,\mathbf{y}_2,c)\sim\mathcal{D}}[\ell_{\text{ipo}}(\theta; \mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, 1-c, \pi_\theta)].
\end{aligned} \tag{50}$$

Next, we give a counter-example to show that ℓ_{ipo} is not noise-tolerant. Suppose that

$$P\left((\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2) = (\mathbf{x}^{(0)}, \mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})\right) = 1 \quad \text{and} \quad \mathbf{y}_1^{(0)} \succ \mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)}, \tag{51}$$

where $\mathbf{x}^{(0)}$ is a fixed input and $(\mathbf{y}_1^{(0)}, \mathbf{y}_2^{(0)})$ is a fixed pair of responses. Hence Eq. (50) becomes

$$\begin{aligned}
&\mathcal{L}^{\eta_0}_{\text{ipo}}(\theta) \\
&= (1-\eta_0)\left(\log\frac{\pi_\theta(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})} - \log\frac{\pi_\theta(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})} - \frac{1}{2\beta}\right)^2 \\
&\quad + \eta_0\left(\log\frac{\pi_\theta(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})} - \log\frac{\pi_\theta(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})} - \frac{1}{2\beta}\right)^2.
\end{aligned} \tag{52}$$

Let

$$\Delta(\theta) = \log\frac{\pi_\theta(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_1^{(0)} \mid \mathbf{x}^{(0)})} - \log\frac{\pi_\theta(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})}{\pi_{\text{ref}}(\mathbf{y}_2^{(0)} \mid \mathbf{x}^{(0)})}, \tag{53}$$

then Eq. (52) becomes

$$\begin{aligned}
\mathcal{L}^{\eta_0}_{\text{ipo}}(\theta) &= (1-\eta_0)\left(\Delta(\theta) - \frac{1}{2\beta}\right)^2 + \eta_0\left(-\Delta(\theta) - \frac{1}{2\beta}\right)^2 \\
&= (\Delta(\theta))^2 + \frac{2\eta_0 - 1}{\beta}\Delta(\theta) + \frac{1}{4\beta^2},
\end{aligned} \tag{54}$$

which is a quadratic function. Hence

$$\theta^*_{\eta_0} \in \left\{\theta \in \Theta : \Delta(\theta) = \frac{1}{2\beta} - \frac{\eta_0}{\beta}\right\}. \tag{55}$$

However,

$$\begin{aligned}
\theta^* &= \arg\min_{\theta\in\Theta} \ \mathcal{L}_{\text{ipo}} \\
&= \arg\min_{\theta\in\Theta}\left(\Delta(\theta) - \frac{1}{2\beta}\right)^2 \\
&\in \left\{\theta \in \Theta : \Delta(\theta) = \frac{1}{2\beta}\right\},
\end{aligned} \tag{56}$$

which means that $\theta^* \neq \theta^*_{\eta_0}$. Therefore, ℓ_{ipo} is not noise-tolerant. □

F.9. The normalization of w_{ropo}

In Eq. (5), we use $\frac{4\alpha}{(1+\alpha)^2}$ to scale the maximum value of w_{ropo} to 1. Here, we provide the details about it. Let

$$g(t) = \sigma(t)(1 + \alpha\sigma(-t)) = \frac{e^{2t} + (1+\alpha)e^t}{(1+e^t)^2},$$

where $\alpha > 2$, then we have

$$\begin{aligned} g'(t) &= \frac{(2e^{2t} + (\alpha+2)e^t)(e^{2t} + 2e^t + 1) - (2e^{2t} + 2e^t)(e^{2t} + (\alpha+1)e^t)}{(1+e^t)^4} \\ &= \frac{1}{e^t(1+e^t)^4} \cdot \big((2-\alpha)e^{2t} + 4e^t + (\alpha+2)\big) \\ &= \frac{1}{e^t(1+e^t)^4} \cdot (1+e^t)((2-\alpha)e^t + \alpha + 2) \\ &= \frac{1}{e^t(1+e^t)^3} \cdot ((2-\alpha)e^t + \alpha + 2). \end{aligned}$$

Hence, $g(t)$ increases if and only if $(2-\alpha)e^t + \alpha + 2 \geq 0$.

Since $\alpha > 2$, $g(t)$ increases when $t < \log\frac{\alpha+2}{\alpha-2}$ and decreases when $t > \log\frac{\alpha+2}{\alpha-2}$. Therefore, we have

$$\max_t g(t) = g\left(\log\frac{\alpha+2}{\alpha-2}\right) = \frac{(1+\alpha)^2}{4\alpha}.$$