ROBUST PREFERENCE OPTIMIZATION: A GENERAL FRAMEWORK FOR ROBUST LLM ALIGNMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Standard human preference-based alignment methods, such as Reinforcement Learning from Human Feedback (RLHF), are a cornerstone technology for aligning Large Language Models (LLMs) with human values. However, these methods are all underpinned by a strong assumption that the collected preference data is clean and that all observed labels are equally reliable. In reality, large-scale preference datasets contain substantial label noise due to annotator errors, inconsistent instructions, varying expertise, and even adversarial or low-effort feedback. This creates a discrepancy between the recorded data and the ground-truth preferences, which can misguide the model and degrade its performance. To address this challenge, we introduce **R**obust **P**reference **O**ptimization (**RPO**). RPO employs an Expectation-Maximization algorithm to infer the posterior probability of each label's correctness, which is used to adaptively re-weigh each data point in the training loss to mitigate noise. We further generalize this approach by establishing a theoretical link between arbitrary preference losses and their corresponding probabilistic models. This generalization enables the systematic transformation of existing alignment algorithms into their robust counterparts, elevating RPO from a specific algorithm to a general framework for robust preference alignment. Theoretically, we prove that under the condition of a perfectly calibrated model, RPO is guaranteed to converge to the true noise level of the dataset. Our experiments demonstrate RPO's effectiveness as a general framework, consistently enhancing four state-of-the-art alignment algorithms (DPO, IPO, SimPO, and CPO). When applied to Mistral and Llama 3 models, the RPO-enhanced methods improve AlpacaEval 2 win rates by up to 7.0 percentage points over their respective baselines.

1 INTRODUCTION

Aligning Large Language Models (LLMs) with human values is a critical prerequisite for developing safe and reliable AI systems. Reinforcement Learning from Human Feedback (RLHF) has emerged as the dominant paradigm for this task (Christiano et al., 2017; Ziegler et al., 2019; Ouyang et al., 2022). To mitigate the complexity and instability of the traditional RLHF pipeline, simpler and more direct methods such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) have been developed, which reframe alignment as a classification-like problem.

However, these alignment methods implicitly assume that preference datasets provide a clean and reliable approximation of a single ground-truth preference signal. In practice, this assumption is often violated. Large-scale preference datasets are typically aggregated from multiple crowdworkers or teacher models, and are therefore subject to substantial label noise arising from inattention, misunderstanding, or systematic bias (Frénay & Verleysen, 2013; Gao et al., 2024). Empirical analyses suggest that a significant fraction (often between 20% and 40%) of preference pairs in modern alignment datasets may be corrupted or inconsistent (Gao et al., 2024). Classic work on learning with noisy labels shows that standard loss functions can overfit such corrupted supervision and suffer severe degradation in generalization performance (Natarajan et al., 2013; Frénay & Verleysen, 2013). In the context of LLM alignment, Gao et al. (2024) further demonstrate that even a 10 percentage point increase in the label-noise rate can lead to drops of tens of percentage points in downstream win rates, highlighting the practical importance of robustness to noisy preference data.

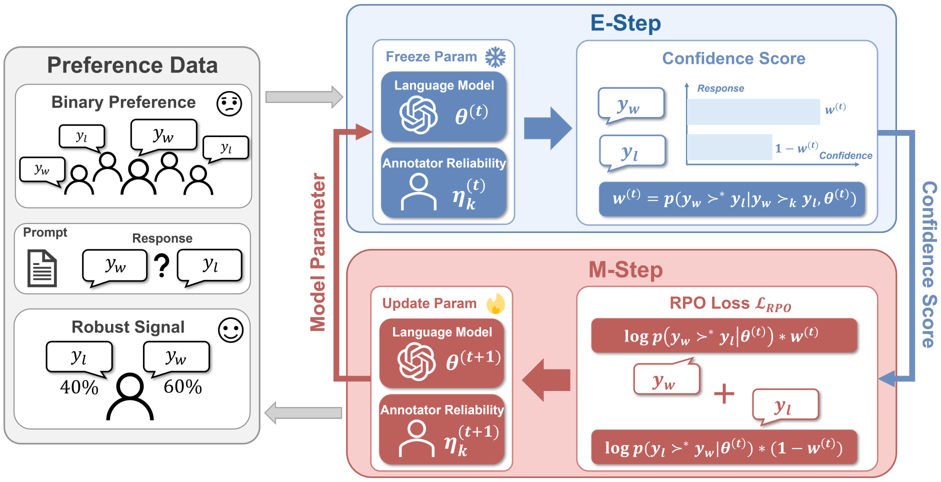


Figure 1: Overview of the Robust Preference Optimization (RPO) framework. Starting from noisy pairwise feedback, RPO uses an Expectation–Maximization (EM) procedure to jointly refine label confidences and the policy. In each iteration, the E-step estimates a confidence score for every observed preference by inferring the posterior probability that the label is correct under the current model and annotator reliabilities. The M-step then uses these scores as adaptive weights to update both the LLM policy and the annotator reliability parameters, progressively down-weighting likely corrupted labels and emphasizing reliable supervision.

To address this challenge, we propose Robust Preference Optimization (RPO). Instead of assuming that every observed label is a fixed ground truth, our approach aims to learn a preference model that remains accurate and stable even when the training data contains substantial noise. The core innovation of RPO is its departure from the hard labels used in traditional RLHF. Rather than committing to binary supervision, we treat the correctness of each observed preference as a latent variable and compute soft confidence weights over labels, so that highly reliable feedback contributes more strongly while suspicious pairs are down-weighted. Building on Expectation-Maximization-style approaches to learning from unreliable annotators in crowdsourcing (Dawid & Skene, 1979; Chen et al., 2013), RPO employs an Expectation-Maximization (EM) framework that simultaneously models annotator reliability while optimizing the LLM. In the E-step, it infers the posterior probability that each annotated label is correct, effectively estimating annotator reliability. In the M-step, it uses these probabilities as adaptive weights to update the LLM, thereby learning from a dynamically re-weighted preference signal.

Our experiments validate RPO as an effective general framework. We show that applying RPO consistently enhances four state-of-the-art alignment algorithms (DPO, IPO, SimPO, and CPO) across two different base models (Mistral-7B and Llama-3-8B) on the AlpacaEval 2 benchmark (Table 2). In our main results, RPO-enhanced methods achieve substantial win-rate gains on AlpacaEval 2, with improvements of up to 7.0 percentage points in LC/WR over their standard counterparts.Furthermore, we theoretically prove that RPO can recover the true reliability of annotators (Theorem 4.1) and empirically verify this guarantee in controlled experiments (Section 5.5).

In summary, our contributions are as follows:

- We propose Robust Preference Optimization (RPO), a principled EM-based algorithm that treats the correctness of each preference label as a latent variable, jointly infers per-label (and per-annotator) reliabilities, and uses them as adaptive weights in the training loss, yielding LLM alignment that is substantially more robust to noisy and inconsistent feedback.
- We theoretically establish a generalized RPO framework by using the Gibbs distribution to connect arbitrary preference loss functions to underlying probabilistic models. This lifts RPO from a single algorithm to a general framework, enabling standard methods such as DPO, IPO, SimPO, and CPO to be systematically transformed into their robust counterparts with minimal modification.

- We conduct extensive experiments demonstrating the practical effectiveness and versatility of RPO. Across four alignment algorithms, two base models (Mistral-7B and Llama-3-8B), and AlpacaEval 2, RPO delivers consistent win-rate improvements of up to 7.0 percentage points, and further shows clear gains on a real multi-annotator dataset (MultiPref), along with qualitative and visual analyses of how it down-weights low-confidence, noisy labels.

2 Related work

LLM alignment with hard preference labels. The standard paradigm for aligning Large Language models (LLMs) with human values is Reinforcement Learning from Human Feedback (RLHF), which involves training a reward model and then fine-tuning the policy against it (Christiano et al., 2017; Ouyang et al., 2022). To mitigate the complexity and instability of this multi-stage process, a family of simpler, direct alignment algorithms has emerged (Rafailov et al., 2023; Azar et al., 2023; Meng et al., 2024; Hong et al., 2024). These methods bypass the explicit reward modeling stage by optimizing a direct classification-style loss on the preference data. However, a critical limitation shared by these methods is their reliance on hard preference labels. This approach models human feedback as a definitive, binary choice, treating every label with equal and absolute confidence. Consequently, it is highly vulnerable to the significant label noise present in real-world datasets, as standard loss functions can lead models to overfit to corrupted labels (Natarajan et al., 2013; Zhang & Sabuncu, 2018; Frénay & Verleysen, 2013). A simple annotation error, such as an accidental misclick, is given the same weight as a deliberate, high-quality judgment. This inability to distinguish between reliable feedback and noise means that the model's performance degrades significantly as the error rate increases (Frénay & Verleysen, 2013; Gao et al., 2024). In contrast, soft-label approaches that represent preferences probabilistically can better accommodate uncertainty in feedback by assigning confidence scores or weights to individual labels (Müller et al., 2019; Song et al., 2024). By allowing the learning algorithm to rely more on high-quality signals while down-weighting likely noise, such approaches provide a natural path toward robust preference alignment. This is precisely the perspective adopted by our RPO framework, which replaces hard labels with EM-estimated soft confidences.

Learning from noisy feedback. The vulnerability to label noise situates preference alignment within the classic machine learning problem of Learning with Noisy Labels (LNL) (Natarajan et al., 2013; Frénay & Verleysen, 2013). Foundational work in this area, such as the Dawid–Skene model (Dawid & Skene, 1979), uses an EM algorithm to simultaneously infer true latent labels while estimating annotator reliability. This principle was later extended to pairwise comparisons in the Crowd-BT model (Chen et al., 2013), which jointly estimates item scores and annotator-specific reliability parameters in crowdsourced ranking tasks. In modern LLM alignment, several methods have been proposed to improve robustness to noisy preference data. These can be broadly divided into loss-centric approaches and data-centric filtering strategies. In the first category, rDPO (Chowdhury et al., 2024) constructs an unbiased estimator of the true loss but requires the global noise rate to be known a priori. Hölder-DPO (Fujisawa et al., 2025) introduces a loss with a "redescending" property, which inherently nullifies the influence of extreme outliers without needing a known noise rate. In the second category, Selective DPO (Gao et al., 2025) proposes filtering examples based on their difficulty relative to the model's capacity—a concept orthogonal to label correctness—using validation loss as a proxy. Our proposed RPO framework is complementary to these methods. Rather than only modifying the loss shape or discarding high-loss points, RPO explicitly models the data-generating process by treating annotator reliability and label correctness as latent variables to be inferred. This allows RPO to assign fine-grained, example-specific weights based on a posterior confidence, providing a principled way to separate signal from noise.

3 Methodology

This section details our proposed RPO algorithm. We first review the standard DPO framework in Section 3.1. In Section 3.2, we introduce a latent-variable model that explicitly distinguishes clean and corrupted preference labels. Section 3.3 then derives the corresponding EM-based update rules for RPO, and the final subsection presents a practical mini-batch implementation.

Table 1: Formulations of the preference loss ($\mathcal{L}_{\text{pref}}$) for prominent alignment algorithms.

Method	**Preference Loss** $\mathcal{L}_{\text{pref}}(x, y_w \succ y_l)$
DPO (Rafailov et al., 2023)	$-\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \beta\log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)}\right)$
IPO (Azar et al., 2023)	$\left(\log\frac{\pi_\theta(y_w\|x)}{\pi_{\text{ref}}(y_w\|x)} - \log\frac{\pi_\theta(y_l\|x)}{\pi_{\text{ref}}(y_l\|x)} - \frac{1}{2\beta}\right)^2$
SimPO (Meng et al., 2024)	$-\log\sigma(\frac{\beta}{\|y_w\|}\log\pi_\theta(y_w\|x) - \frac{\beta}{\|y_l\|}\log\pi_\theta(y_l\|x) - \gamma)$
CPO (Xu et al., 2024)	$-\log\sigma(\beta\log\pi_\theta(y_w\|x) - \beta\log\pi_\theta(y_l\|x)) - \log\pi_\theta(y_w\|x)$

3.1 Preliminaries: Direct Preference Optimization

The goal of preference alignment is to fine-tune a language model policy, π_θ, using a dataset of preferences $\mathcal{D} = \{(x, y_w, y_l)_i\}_{i=1}^N$, where response y_w is preferred over y_l for a given prompt x. Direct Preference Optimization (DPO) (Rafailov et al., 2023) offers a simple and effective method for this, bypassing the complex multi-stage pipeline of traditional RLHF (Christiano et al., 2017; Ouyang et al., 2022). DPO directly optimizes the policy by minimizing a simple classification loss:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function, π_{ref} is a fixed reference policy and β is a scaling hyperparameter.

3.2 RPO Framework: Core Assumptions

A critical limitation of DPO is its implicit assumption that all observed preferences in $\mathcal{D}$ are correct. In practice, this data is often noisy. To address this, we propose Robust Preference Optimization (RPO), which is built upon two core assumptions that reframe the problem.

Assumption 1: Latent noise-free preference. We assume that for each training example $(x_i, y_{w,i}, y_{l,i})$ there exists an underlying noise-free preference, denoted $y_{w,i} \succ^* y_{l,i}$, which represents the label we would obtain in the absence of annotation errors. The observed preference $y_{w,i} \succ_{k_i} y_{l,i}$ (provided by annotator k_i) is treated as a potentially corrupted observation of this ground truth. To model this, we introduce a binary latent variable $z_i \in \{0, 1\}$ for each data point, where $z_i = 1$ if the observed label matches the latent noise-free preference and $z_i = 0$ otherwise. The reliability of annotator k is then parameterized by $\eta_k \triangleq p(z_i = 1 \mid k_i = k)$. Here $k_i \in \{1, \ldots, K\}$ denotes the index of the annotator who provided the i-th label, and K is the total number of annotators in the dataset.

Assumption 2: A general probabilistic model for preferences. Building on this latent variable model, we must also define the probability of the noise-free preference itself, $p(y_w \succ^* y_l|x, \theta)$. To accommodate various preference losses beyond DPO (e.g., IPO (Azar et al., 2023)), our framework is designed to work with any preference loss function, $\mathcal{L}_{\text{pref}}$. Table 1 provides several examples of such loss functions used in prominent alignment algorithms.

To connect these diverse loss functions to a unified probabilistic interpretation, we draw inspiration from the Boltzmann distribution (Luce, 1959). We assume that for any preference loss function $\mathcal{L}_{\text{pref}}$, the probability of a preference is proportional to the exponentiated negative loss $\exp(-\mathcal{L}_{\text{pref}}(x, y_w \succ y_l))$. This yields a general definition for the noise-free preference probability:

$$p(y_w \succ^* y_l|x, \theta) = \sigma\left(\mathcal{L}_{\text{pref}}(x, y_l \succ y_w; \theta) - \mathcal{L}_{\text{pref}}(x, y_w \succ y_l; \theta)\right), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. This formulation converts any preference loss into a well-defined probability distribution. For instance, with the standard DPO loss, this equation recovers the Bradley-Terry model (Bradley & Terry, 1952) (see Appendices A and B for derivations).

Algorithm 1: Robust Preference Optimization (RPO)

Input: Dataset $\mathcal{D} = \{(x_i, y_{w,i}, y_{l,i}, k_i)\}_{i=1}^{N}$; Base policy π_θ, reference policy π_{ref}; Preference loss $\mathcal{L}_{\text{pref}}$; Hyperparameters: learning rate λ, epochs E, EMA momentum α, initial annotator reliabilities $\eta_k \in [0.5, 1]$ for all $k \in \{1, \ldots, K\}$

for *epoch* = 1 **to** E **do**
  for *batch* $\mathcal{B} \subset \mathcal{D}$ **do**
    For each sample $i \in \mathcal{B}$, compute w_i using current θ and η_{k_i} via equation 4;
    Compute the weighted loss $\mathcal{L}_{\text{RPO}}(\theta)$ for the batch via equation 5;
    Update parameters θ using an optimizer (e.g., AdamW (Loshchilov & Hutter, 2019));
    for *each annotator k present in the batch* **do**
      Update η_k via equation 7;
    end
  end
end

3.3 The RPO Algorithm via Expectation-Maximization

Based on these core assumptions, we aim to find the parameters θ and $\boldsymbol{\eta}$ that maximize the marginal log-likelihood of the observed data. The probability of a single observed preference is obtained by marginalizing over the latent variable z_i:

$$p(y_{w,i} \succ_{k_i} y_{l,i}|x_i, \theta, \boldsymbol{\eta}) = p(y_{w,i} \succ^* y_{l,i}|x_i, \theta)\eta_{k_i} + p(y_{l,i} \succ^* y_{w,i}|x_i, \theta)(1 - \eta_{k_i}). \tag{3}$$

Directly maximizing $\sum_i \log p(y_{w,i} \succ_{k_i} y_{l,i})$ is intractable due to the sum inside the logarithm. We therefore employ the EM algorithm (see details in Appendix C), which iterates between two steps. In this iterative process, the superscript (t) will denote the values of parameters at iteration t.

E-Step: Inferring label correctness. In the E-step, given the current parameters $\theta^{(t)}$ and $\boldsymbol{\eta}^{(t)}$, we compute the posterior probability w_i that the i-th observed label is correct. This value w_i acts as a "soft label" or the model's confidence in the data point.

$$w_i^{(t)} \leftarrow \frac{p(y_{w,i} \succ^* y_{l,i}|x_i, \theta^{(t)})\eta_{k_i}^{(t)}}{p(y_{w,i} \succ^* y_{l,i}|x_i, \theta^{(t)})\eta_{k_i}^{(t)} + p(y_{l,i} \succ^* y_{w,i}|x_i, \theta^{(t)})(1 - \eta_{k_i}^{(t)})}. \tag{4}$$

where $p(y_{w,i} \succ^* y_{l,i}|x_i, \theta^{(t)})$ and $p(y_{l,i} \succ^* y_{w,i}|x_i, \theta^{(t)})$ can be computed according to equation 2.

M-Step: weighted parameter update. In the M-step, we update the policy parameters θ and reliabilities $\boldsymbol{\eta}$ using the confidences $w_i^{(t)}$ computed in the E-step. This step conveniently separates into two independent updates.

First, the policy is updated by minimizing a weighted loss function. As established in Assumption 2, our probabilistic model for $p(y_w \succ^* y_l)$ allows RPO to work with any preference loss $\mathcal{L}_{\text{pref}}$, making it a versatile meta-framework. The general RPO loss is:

$$\mathcal{L}_{\text{RPO}}(\theta) = -\sum_{i=1}^{N} \left[w_i^{(t)} \log p(y_{w,i} \succ^* y_{l,i}|x_i, \theta) + (1 - w_i^{(t)}) \log p(y_{l,i} \succ^* y_{w,i}|x_i, \theta) \right]. \tag{5}$$

Second, the reliability η_k for each annotator is updated to the average confidence of all labels they provided. This has a simple and efficient closed-form solution:

$$\eta_k^{(t+1)} = \frac{\sum_{i \in \mathcal{I}_k} w_i^{(t)}}{N_k}. \tag{6}$$

Here we define the index set of labeled pairs as $\mathcal{I}_k = \{\, i : k_i = k \,\}$, and the number of labels as N_k.

3.4 Practical implementation with mini-batch training

While the exact M-step updates are clear, performing a full iteration over the entire dataset to recalculate the annotator reliabilities $\boldsymbol{\eta}$ after each policy update step can be computationally expensive. To balance computational efficiency and performance, we introduce a more practical online update for η_k using an Exponential Moving Average (EMA). Instead of a hard assignment, we perform a soft update based on the statistics from the current mini-batch $\mathcal{B}$:

$$\eta_k \leftarrow (1-\alpha)\eta_k + \alpha \cdot \frac{\sum_{i \in \mathcal{B} \cap \mathcal{I}_k} w_i}{N_{k,\mathcal{B}}}. \tag{7}$$

Here, $N_{k,\mathcal{B}}$ is the number of examples from annotator k in the current mini-batch, and $\alpha \in (0,1]$ is a momentum hyperparameter. The complete training procedure for RPO is summarized in Algorithm 1:

4 Theoretical analysis of RPO

The robustness of RPO stems from its adaptive weighting mechanism. This section first provides an intuitive analysis of these training dynamics and then formalizes this intuition with theoretical guarantees, demonstrating that the RPO framework can recover the true reliability of annotators.

At the start of training, when the language model is not yet well-optimized, its predictions are uncertain, and the probabilities $p(y_w \succ^* y_l | x, \theta)$ are close to 0.5. The confidence score w_i approximates the annotator's reliability, η_{k_i}. The loss then acts as a form of label smoothing, preventing the model from being severely misled by incorrect labels early on. As the policy improves, its behavior adapts. For a high-quality label, the model predicts a high probability for the winning response, and w_i approaches 1, causing the loss to function like a standard preference optimization objective. Conversely, w_i approaches 0 for a noisy label. The loss is then dominated by the $(1 - w_i)$ term, which flips the optimization direction toward the true preference.

We now formalize the intuition that RPO can recover the true reliability of annotators. We provide this analysis under an idealized setting: full-batch training where the M-step for the policy parameters θ is assumed to have converged perfectly. While our practical implementation in Algorithm 1 uses mini-batch gradient updates (a form of Generalized EM), this idealized analysis provides a strong theoretical justification for our framework.

Consider the dataset level update rule in equation 6, defined as an operator $T_k(\eta)$. The following theorem establishes that iterating this operator guarantees convergence to the true annotator reliability.

Theorem 4.1 (Identification and convergence of RPO). *Let $\theta^\star$ be a perfectly calibrated parameter such that the model distribution matches the ground-truth preference distribution. Assume that not all $p_i^\star = p(y_{w,i} \succ^* y_{l,i} | x_i)$ equal $\frac{1}{2}$ for $i \in \mathcal{I}_k$. Consider the sequence of reliability estimates $\{\eta_k^{(t)}\}_{t \geq 0}$ generated by the update rule $\eta_k^{(t+1)} = T_k(\eta_k^{(t)})$. Then, for any initialization $\eta_k^{(0)} \in (0,1)$, the iterates converge to the true reliability $\eta_k^* \triangleq \mathbb{E}[z_i \mid k_i = k]$:*

$$\lim_{t \to \infty} \eta_k^{(t)} = \eta_k^\star.$$

The proof is provided in Appendix D. In section 5.5, we empirically corroborate that the mini-batch procedure closely tracks this theoretical behavior.

Practical implications and limitations. The assumption of a perfectly calibrated model in Theorem 4.1 is intentionally idealized: in practice, we apply RPO to base models that are not exactly calibrated to the ground-truth preference distribution. In our experiments, we always start from strong instruction-tuned LLMs (`Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`), which already display good zero-shot preference behavior. Empirically, we do not observe the failure mode suggested by an extremely misaligned initialization: across the broad range of hyperparameters explored in Section 5.4, the learned η_k's remain stable and the downstream performance consistently improves over the corresponding base methods. Furthermore, the controlled experiments in Section 5.5, where we inject substantial synthetic noise into the data, show that RPO's estimated reliabilities closely track the ground-truth values, suggesting robustness to imperfect calibration in practice. If the base LLM were initialized in a highly misaligned

Table 2: Performance comparison on AlpacaEval 2 for `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct` fine-tuned on UltraFeedback-based preference datasets. Metrics reported are **LC** (Length-Controlled Win Rate) and **WR** (Raw Win Rate), both in percentage points. The table presents reference *Baselines* (bottom) alongside four algorithm families (DPO, IPO, SimPO, CPO). For each family, we compare the *Standard* implementation, the variant with Label Smoothing (*w/ LS*), and RPO (*w/ RPO*). **Bold** denotes the best result within each family for a given backbone.

	Mistral-7B-Instruct			**Llama-3-8B-Instruct**		
Method	Standard	w/ LS	w/ RPO	Standard	w/ LS	w/ RPO
DPO	28.5 / 28.6	29.7 / 27.5	**35.5 / 33.0**	40.8 / 42.9	41.3 / 42.6	**44.1 / 46.2**
IPO	30.8 / 28.0	29.7 / 28.7	**32.9 / 30.5**	43.6 / 41.6	40.3 / 38.2	**48.3 / 48.6**
SimPO	28.3 / 29.7	26.5 / 27.1	**30.4 / 32.9**	44.5 / 37.1	**48.1** / 38.7	46.9 / **39.4**
CPO	26.3 / 26.4	**28.5 / 28.8**	27.6 / 27.8	35.9 / 40.3	35.3 / 34.8	**40.1 / 43.8**
Base Model		21.1 / 16.5			29.7 / 29.9	
rDPO		28.1 / 29.1			37.3 / 35.4	
Hölder-DPO		30.1 / 28.6			39.3 / 38.2	

Table 3: Performance of DPO and R-DPO on AlpacaEval 2 when trained on the MultiPref dataset (Miranda et al., 2024). Results are reported as LC / WR (%) for `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`.

Method	**Mistral-7B-Instruct**	**Llama-3-8B-Instruct**
DPO	28.8 / 26.4	36.7 / 39.3
R-DPO (Ours)	**31.8 / 28.8**	**41.1 / 44.4**

regime, the E-step could assign misleadingly high confidence to incorrect labels and RPO might fail to effectively denoise the supervision.

5 Experiments

In this section, we conduct a comprehensive set of experiments to evaluate the performance of RPO.We begin in section 5.1 by detailing our experimental setup, including the models, datasets, evaluation benchmarks, and baseline algorithms. In Section 5.2, we present our main results. Section 5.3 reports additional experiments to evaluate RPO's performance on realistic multi-annotator datasets. We then conduct an ablation study in Section 5.4 to analyze the framework's sensitivity to its key hyperparameters. In section 5.5, we provide an empirical verification of our theoretical claims from Theorem 4.1.

5.1 Experimental setup

Models and training settings. We use two state-of-the-art open-source large language models as our base models: `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`. For fine-tuning, we utilize two datasets from the SimPO paper (Meng et al., 2024), which were generated via on-policy sampling using prompts from the UltraFeedback dataset (Cui et al., 2024). The specific datasets are `mistral-instruct-ultrafeedback` for the Mistral model and `llama3-ultrafeedback-armorm` for the Llama-3 model.[1] As these datasets do not provide annotator-specific information, we model the preferences as if they originate from a single, virtual annotator ($K = 1$).[2] In addition to these UltraFeedback-based datasets, we further evaluate RPO

[1]See Appendix I for links to models and datasets.

[2]This is a reasonable simplification. For instance, a pool of two annotators with reliabilities η_A and η_B, appearing with frequencies p_A and p_B respectively, can be modeled as a single annotator with an effective reliability $\eta_{\text{unified}} = p_A\eta_A + p_B\eta_B$.

Table 4: Ablation study on the initial annotator reliability (η_0) and the EMA momentum (α). Results are reported for R-DPO on `Mistral-7B-Instruct-v0.2` trained on UltraFeedback-based data, evaluated on AlpacaEval 2 (LC / WR) and Arena-Hard (WR), all in percentage points. The best-performing settings used in our main experiments are highlighted.

Metric	Initial η_0				EMA α				
	0.99	**0.9 (Ours)**	0.75	0.55	0.001	0.01	**0.1 (Ours)**	0.5	1.0
AlpacaEval2 LC (%)	30.9	**35.5**	31.1	31.4	30.9	30.1	**35.5**	33.4	31.1
AlpacaEval2 WR (%)	31.7	**33.0**	33.3	32.0	27.8	27.2	**33.0**	34.8	28.9
Arena-Hard WR (%)	12.3	**14.7**	12.4	11.8	12.9	13.6	**14.7**	14.0	12.8

on the real-world MultiPref multi-annotator preference dataset (Miranda et al., 2024), where per-annotator reliabilities can be explicitly modeled (Section 5.3).

Evaluation benchmarks. We assess model performance on two widely recognized evaluation benchmarks. The first is AlpacaEval 2 (Dubois et al., 2024), an automatic, LLM-based evaluator that measures model performance by computing the win rate against reference outputs. It provides both a raw Win Rate (WR) and a Length-Controlled (LC) Win Rate to account for verbosity bias. The second is Arena-Hard (Li et al., 2024), a challenging benchmark composed of difficult prompts crowdsourced from the LMSYS Chatbot Arena. It is designed to differentiate high-performing models by testing them on complex, real-world user queries. Performance is reported as the win rate against a suite of other models.

Baseline algorithms. To demonstrate that RPO operates as a versatile meta-framework, we benchmark it against four popular direct preference alignment methods: DPO (Rafailov et al., 2023); IPO (Azar et al., 2023), which uses a squared hinge loss to optimize preferences; SimPO (Meng et al., 2024), which proposes a simplified, reference-free reward formulation normalized by sequence length; and CPO (Xu et al., 2024), which adds a term to directly maximize the likelihood of the preferred response. For each of these baselines, whose loss functions are detailed in Table 1, we compare the original algorithm to its RPO-enhanced counterpart (e.g., DPO vs. R-DPO). In addition, we include robustness-oriented baselines rDPO (Chowdhury et al., 2024) and Hölder-DPO (Fujisawa et al., 2025), as well as simple label-smoothing variants for each method, as summarized in Table 2.

5.2 Main results

As shown in Table 2, our experimental results provide strong evidence that RPO consistently improves preference-based alignment across objectives, model scales, and datasets. Below we highlight the main empirical findings.

RPO as a general framework. A first observation is that RPO behaves as a generally effective "plug-in" robustness layer for a wide range of alignment losses. Across all four objective families (DPO, IPO, SimPO, CPO) and both backbones (Mistral-7B and Llama-3-8B), the RPO-enhanced variant either matches or strictly outperforms the corresponding standard implementation on AlpacaEval 2. For example, on Mistral-7B, R-DPO improves LC / WR from $28.5/28.6$ to $35.5/33.0$ (a gain of $+7.0$ and $+4.4$ points, respectively), and on Llama-3-8B, RPO-IPO improves LC / WR from $43.6/41.6$ to $48.3/48.6$ (a gain of $+4.7$ and $+7.0$ points). These trends hold across all four families, indicating that RPO reliably strengthens existing preference objectives rather than competing with them.

Comparison with label smoothing and robust baselines. Table 2 also compares RPO to two natural robustness baselines: label smoothing applied to each preference loss and the recently proposed robust objectives rDPO (Chowdhury et al., 2024) and Hölder-DPO (Fujisawa et al., 2025). Label smoothing sometimes yields modest gains over the standard objective (e.g., SimPO w/ LS on Llama-3-8B improves LC from 44.5 to 48.1), but RPO typically achieves the best performance within each family and backbone. For instance, in the DPO family, R-DPO outperforms both label smoothing and the specialized robust baselines: on Llama-3-8B, R-DPO reaches $44.1/46.2$ on AlpacaEval 2, compared to 41.3/42.6 for DPO w/ LS, 37.3/35.4 for rDPO, and 39.3/38.2 for Hölder-DPO. These results suggest that explicitly modeling noisy supervision via RPO is more effective than purely loss-level modifications or global noise-correction schemes.

Qualitative analysis of noisy labels. Beyond aggregate metrics, we also perform a qualitative analysis of the learned confidence scores. In Appendix F, we present case studies of preference pairs with very low posterior confidence w_i. RPO assigns low confidence to annotations that are off-task, inconsistent with the prompt, or at odds with a more plausible alternative response. Together with the quantitative gains in Tables 2, these examples illustrate that RPO not only improves benchmark performance but also identifies and down-weights noisy supervision at the example level.

5.3 Multi-annotator experiments on MultiPref

To further evaluate RPO under realistic multi-annotator disagreement, we conduct additional experiments on the MultiPref dataset (Miranda et al., 2024), a large-scale human preference dataset with genuine rater disagreement. The official training split contains **227 unique human annotators**. Unlike the UltraFeedback-based datasets used in our main experiments, MultiPref provides annotator identifiers, allowing us to instantiate an individual reliability parameter η_k for each annotator and to update these parameters via our EM-style scheme.

We train vanilla DPO and our R-DPO on MultiPref for both `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`, and evaluate the resulting models on AlpacaEval 2. As summarized in Table 3, R-DPO consistently outperforms vanilla DPO under this multi-annotator setup: for Llama-3-8B, the AlpacaEval LC improves from 36.7 to 41.1 and WR from 39.3 to 44.4; for Mistral-7B, LC improves from 28.8 to 31.8 and WR from 26.4 to 28.8. These gains mirror the trends observed in our UltraFeedback experiments and show that RPO remains beneficial when trained on data with heterogeneous, potentially noisy annotators, rather than a single virtual annotator.

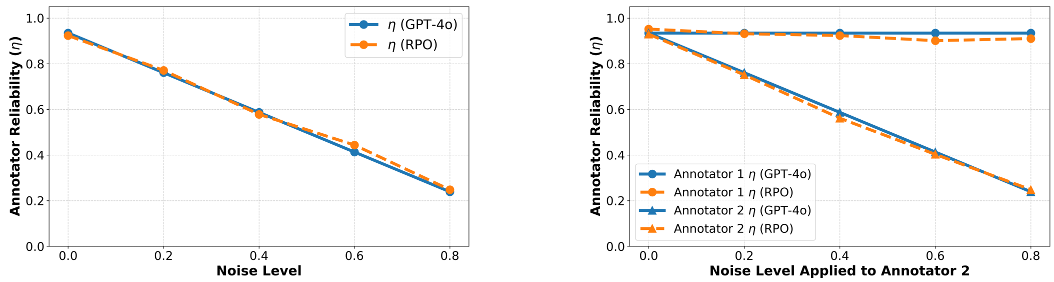
In Appendix E, we visualize the learned annotator reliabilities distributions on MultiPref. Experiment results indicate that RPO identifies a high-reliability majority and a nontrivial tail of downweighted annotators, and that this pattern is robust across different prior settings and backbones. Moreover, to probe the impact of the choice of automatic judge, we repeat the MultiPref evaluation using a different LLM evaluator; Appendix G reports these results and shows that the performance gains from R-DPO are stable across judge models.

5.4 Ablation study

We conduct an ablation study to analyze the sensitivity of RPO to two key hyperparameters: the initial annotator reliability, η_0, and the EMA momentum parameter, α. All experiments are performed using the R-DPO algorithm on the Mistral-7B-Instruct-v0.2 model. The results are summarized in Table 4.

Effect of initial η_0. The initial reliability η_0 sets the model's prior belief about the correctness of the labels in the dataset. As shown in Table 4, the model's performance is best when η_0 is set to 0.9, which was the value used in our main experiments. An overly optimistic initialization (e.g., $\eta_0 = 0.99$) can cause the model to trust noisy labels too strongly at the beginning of training, hindering the denoising process. Conversely, a pessimistic initialization (e.g., $\eta_0 = 0.55$) treats the data as highly unreliable from the outset, which can slow down the model's ability to learn the underlying noise-free preference. An initial value of 0.9 appears to strike the right balance, starting with a reasonable assumption of data quality.

Effect of EMA parameter α. The EMA parameter α governs the update rate of the annotator reliability scores, balancing the influence of historical estimates against new information from the current mini-batch. Our experiments confirm that the optimal performance is achieved with $\alpha = 0.1$. The model shows considerable sensitivity to this parameter. A very small α (e.g., 0.001) makes the reliability updates exceedingly slow, preventing the estimates from adapting to the model's evolving understanding of the data. On the other hand, a very large α (e.g., 1.0) makes the updates highly volatile, as the reliability score becomes dependent solely on the samples in the current mini-batch.



(a) Single-annotator setting.

(b) Two-annotator setting.

Figure 2: Empirical verification of annotator reliability estimation under controlled synthetic noise. Ground-truth reliability (η GPT-4o) is established using GPT-4o's labels on UltraFeedback-derived preference pairs, and different reliability levels are simulated by injecting synthetic noise into copies of the dataset. In the single-annotator setting (a), a single annotator's dataset is perturbed with varying noise rates. In the two-annotator setting (b), Annotator 1 uses the original data with no added noise, while noise is progressively added to Annotator 2's data. The plots compare ground-truth reliabilities (solid lines) with RPO-estimated reliabilities (dashed lines), showing that RPO closely tracks the true reliability in both scenarios.

5.5 Empirical verification of Theorem 4.1

We conduct controlled experiments to verify Theorem 4.1. Our setup is designed to align with the theorem's assumption of a perfectly calibrated model, for which we use a small-scale base model, `Qwen2.5-0.5B-Instruct`, to ensure fast convergence. To simulate annotators with varying levels of reliability, we create distinct copies of the UltraFeedback dataset (Cui et al., 2024) for each annotator and inject a controlled degree of synthetic noise into their respective dataset.

We test two scenarios, with results presented in Figure 2: (a) **Single Annotator:** A single annotator whose dataset is modified with a synthetically controlled noise rate. (b) **Two Annotators:** A scenario with two annotators, where Annotator 1 serves as a baseline using the original data without added noise, while the dataset for Annotator 2 is injected with progressively increasing noise levels.

The results in Figure 2 show that the estimated reliability η (RPO) closely tracks the ground-truth η (GPT-4o) in both single-annotator (Figure 2a) and two-annotator (Figure 2b) settings. Notably, in the two-annotator experiment, RPO successfully identifies the stable reliability of the baseline annotator while accurately tracking the declining reliability of the noisy one. Although the theorem assumes a perfectly calibrated model, these experiments demonstrate that RPO's reliability estimates remain accurate and stable even when the underlying model is only approximately calibrated and trained under realistic noise patterns, mitigating concerns that early miscalibration would systematically down-weight correct labels.

6 Conclusion and future work

In this paper, we introduce Robust Preference Optimization (RPO), a novel framework designed to address the critical challenge of aligning LLMs with noisy human preference data. Our approach is distinct from existing methods as it employs an Expectation-Maximization algorithm to infer the reliability of each preference pair, treating labels as soft, dynamic weights rather than fixed ground truths. As a meta-framework, RPO consistently enhances multiple state-of-the-art alignment algorithms, achieving significant performance gains (up to a 7.0% win rate increase on AlpacaEval 2) across various base models. A natural limitation of our current theory is the assumption of a perfectly calibrated model; extending convergence guarantees to settings where the base model is significantly misaligned remains important future work. In addition, our empirical study focuses on 7B–8B backbones, and systematically evaluating RPO on substantially larger models (e.g., 70B+) to understand the memory, runtime, and robustness trade-offs is an important direction for future work.

References

Mohammad Gheshlaghi Azar, Georg Ostrovski, Remi Munos, and Bernardo Pires. A general theoretical paradigm for preference-based reinforcement learning. *arXiv preprint arXiv:2310.12036*, 2023.

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

Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 193–202, 2013.

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

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30*, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with scaled ai feedback, 2024. URL `https://arxiv.org/abs/2310.01377`.

A Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1): 20–28, 1979.

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

Benoît Frénay and Michel Verleysen. Classification in the presence of label noise: a survey. *IEEE transactions on neural networks and learning systems*, 25(5):845–869, 2013.

Masahiro Fujisawa, Masaki Adachi, and Michael A Osborne. Scalable valuation of human feedback through provably robust model alignment. *arXiv preprint arXiv:2505.17859*, 2025.

Chengqian Gao, Haonan Li, Liu Liu, Zeke Xie, Peilin Zhao, and Zhiqiang Xu. Principled data selection for alignment: The hidden risks of difficult examples. *arXiv preprint arXiv:2502.09650*, 2025.

Yang Gao, Dana Alon, and Donald Metzler. Impact of preference noise on the alignment performance of generative language models. *arXiv preprint arXiv:2404.09824*, 2024.

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

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

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. John Wiley & Sons, 1959.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

Lester James V. Miranda, Yizhong Wang, Yanai Elazar, Sachin Kumar, Valentina Pyatkin, Faeze Brahman, Noah A. Smith, Hannaneh Hajishirzi, and Pradeep Dasigi. Hybrid Preferences: Learning to Route Instances for Human vs. AI Feedback. *arXiv*, abs/2410.19133, October 2024.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. In *Advances in neural information processing systems 26*, 2013.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35*, pp. 27730–27744, 2022.

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

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 18990–18998, 2024.

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

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

Daniel M Ziegler, Nissan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Derivation of general probabilistic model

Here we provide the detailed derivation for equation 2. For a given prompt x and candidate responses y_w, y_l, we assume the probability of the ground-truth preference $y_w \succ^* y_l$ is proportional to $\exp(-\mathcal{L}_{\text{pref}}(x, y_w \succ y_l))$. That is:

$$p(y_w \succ^* y_l | x, \theta) \propto \exp(-\mathcal{L}_{\text{pref}}(x, y_w \succ y_l)) \tag{8}$$

Similarly, for the inverse preference:

$$p(y_l \succ^* y_w | x, \theta) \propto \exp(-\mathcal{L}_{\text{pref}}(x, y_l \succ y_w)) \tag{9}$$

Since $y_w \succ^* y_l$ and $y_l \succ^* y_w$ are the only two mutually exclusive outcomes for a binary preference, their probabilities must sum to 1. Using the property of normalized probabilities from a proportional relationship, we have:

$$\begin{aligned}
p(y_w \succ^* y_l | x, \theta) &= \frac{\exp(-\mathcal{L}_{\text{pref}}(x, y_w \succ y_l))}{\exp(-\mathcal{L}_{\text{pref}}(x, y_w \succ y_l)) + \exp(-\mathcal{L}_{\text{pref}}(x, y_l \succ y_w))} \\
&= \frac{1}{1 + \exp(-(\mathcal{L}_{\text{pref}}(x, y_l \succ y_w) - \mathcal{L}_{\text{pref}}(x, y_w \succ y_l)))} \\
&= \sigma\left(\mathcal{L}_{\text{pref}}(x, y_l \succ y_w) - \mathcal{L}_{\text{pref}}(x, y_w \succ y_l)\right)
\end{aligned}$$

The last line is the General Probabilistic Model in equation 2.

B Consistency with Bradley-Terry model for DPO

We show that equation 2 is consistent with the Bradley-Terry model when applied to DPO. The DPO loss for a preferred pair (y_w, y_l) given prompt x is:

$$\mathcal{L}_{\text{DPO}}(x, y_w \succ y_l) = -\log \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) \tag{10}$$

Let $S(x, y_w, y_l) = \beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}$. Then, we can write:

$$\mathcal{L}_{\text{DPO}}(x, y_w \succ y_l) = -\log \sigma(S(x, y_w, y_l))$$
$$\mathcal{L}_{\text{DPO}}(x, y_l \succ y_w) = -\log \sigma(S(x, y_l, y_w)) = -\log \sigma(-S(x, y_w, y_l))$$

Substituting these into our general probabilistic model (equation 2):

$$\begin{aligned} p(y_w \succ^* y_l|x, \theta) &= \sigma\left(\mathcal{L}_{\text{DPO}}(x, y_l \succ y_w) - \mathcal{L}_{\text{DPO}}(x, y_w \succ y_l)\right) \\ &= \sigma\left(\log \sigma(S(x, y_w, y_l)) - \log \sigma(-S(x, y_w, y_l))\right) \\ &= \sigma\left(\log \frac{\sigma(S(x, y_w, y_l))}{1 - \sigma(S(x, y_w, y_l))}\right) \\ &= \sigma(S(x, y_w, y_l)) \\ &= \sigma\left(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) \end{aligned}$$

This resulting probability exactly matches the form of the Bradley-Terry model (Bradley & Terry, 1952) for preferences, where the implicit reward of a response y is $r(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$.

C Derivation of the RPO EM algorithm

The primary objective of Robust Preference Optimization (RPO) is to find the model parameters θ and the vector of annotator reliabilities $\boldsymbol{\eta}$ that maximize the log-likelihood of the observed data. The observed data consists of prompts, chosen and rejected responses, and the annotator's index, denoted as $X = \mathcal{D} = \{(x_i, y_{w,i}, y_{l,i}, k_i)\}_{i=1}^N$.

The log-likelihood function is given by:

$$\mathcal{L}(\theta, \boldsymbol{\eta}) = \sum_{i=1}^{N} \log\left[p(y_{w,i} \succ^* y_{l,i}|x_i, \theta)\eta_{k_i} + p(y_{l,i} \succ^* y_{w,i}|x_i, \theta)(1 - \eta_{k_i})\right] \tag{11}$$

There is a sum inside the logarithm, which makes direct optimization intractable. The Expectation-Maximization (EM) algorithm is an iterative procedure designed to solve such maximum likelihood problems with latent variables by alternating between an Expectation (E) step and a Maximization (M) step.

C.1 Derivation of the Q-Function (The E-Step)

The EM algorithm simplifies the problem by working with the complete data, (X, Z), where $Z = \{z_i\}_{i=1}^N$ is the set of all latent variables.

The complete-data log-likelihood, $\mathcal{L}_c$, assumes that we know the values of all latent variables z_i:

$$\mathcal{L}_c(\theta, \boldsymbol{\eta}; X, Z) = \sum_{i=1}^{N} \left(z_i \log\left[p(y_{w,i} \succ^* y_{l,i}|x_i, \theta)\eta_{k_i}\right] + (1 - z_i)\log\left[p(y_{l,i} \succ^* y_{w,i}|x_i, \theta)(1 - \eta_{k_i})\right]\right) \tag{12}$$

This form is tractable because the logarithm acts on products, which can be separated into sums.

The core idea of EM is to iteratively maximize the expectation of the complete-data log-likelihood. This expectation, known as the Q-function, is taken with respect to the posterior distribution of the latent variables Z, given the observed data X and the parameter estimates from the current iteration, $(\theta^{(t)}, \boldsymbol{\eta}^{(t)})$.

$$Q(\theta, \boldsymbol{\eta}|\theta^{(t)}, \boldsymbol{\eta}^{(t)}) \equiv \mathbb{E}_{Z|X, \theta^{(t)}, \boldsymbol{\eta}^{(t)}}[\mathcal{L}_c(\theta, \boldsymbol{\eta}; X, Z)] \tag{13}$$

To compute this expectation, we push the expectation operator inside the summation. The only random variables in $\mathcal{L}_c$ are the z_i.

$$Q(\theta, \boldsymbol{\eta}|\theta^{(t)}, \boldsymbol{\eta}^{(t)}) = \sum_{i=1}^{N} \left(\mathbb{E}[z_i]\log\left[p(y_{w,i} \succ^* y_{l,i}|\theta)\eta_{k_i}\right] + (1 - \mathbb{E}[z_i])\log\left[p(y_{l,i} \succ^* y_{w,i}|\theta)(1 - \eta_{k_i})\right]\right) \tag{14}$$

The term $\mathbb{E}[z_i]$ is the expectation of the binary variable z_i, which is its posterior probability of being 1. This probability is conditioned on the observed data and the parameters from the current iteration t. We denote this posterior probability as $w_i^{(t)}$, which is computed in the E-Step:

$$\begin{aligned} w_i^{(t)} &\equiv \mathbb{E}[z_i|X_i, \theta^{(t)}, \boldsymbol{\eta}^{(t)}] \\ &= p(z_i = 1|y_{w,i} \succ_{k_i} y_{l,i}, x_i, \theta^{(t)}, \boldsymbol{\eta}^{(t)}) \\ &= \frac{p(y_{w,i} \succ_{k_i} y_{l,i}|z_i = 1, x_i, \theta^{(t)}) p(z_i = 1|k_i, \boldsymbol{\eta}^{(t)})}{p(y_{w,i} \succ_{k_i} y_{l,i}|x_i, \theta^{(t)}, \boldsymbol{\eta}^{(t)})} \\ &= \frac{p(y_{w,i} \succ^* y_{l,i}|x_i, \theta^{(t)}) \eta_{k_i}^{(t)}}{p(y_{w,i} \succ^* y_{l,i}|x_i, \theta^{(t)}) \eta_{k_i}^{(t)} + p(y_{l,i} \succ^* y_{w,i}|x_i, \theta^{(t)})(1 - \eta_{k_i}^{(t)})} \end{aligned} \tag{15}$$

Substituting $w_i^{(t)}$ into the expression yields the final form of the Q-function:

$$Q(\theta, \boldsymbol{\eta}|\theta^{(t)}, \boldsymbol{\eta}^{(t)}) = \sum_{i=1}^{N} \left[w_i^{(t)} \log(p(y_{w,i} \succ^* y_{l,i}|\theta)\eta_{k_i}) + (1 - w_i^{(t)}) \log(p(y_{l,i} \succ^* y_{w,i}|\theta)(1 - \eta_{k_i})) \right] \tag{16}$$

C.2 Deriving the RPO Framework (The M-Step)

The goal of the M-Step is to find the parameters for the next iteration, $(\theta^{(t+1)}, \boldsymbol{\eta}^{(t+1)})$, by maximizing the Q-function that was constructed using the parameters from the current iteration t.

$$(\theta^{(t+1)}, \boldsymbol{\eta}^{(t+1)}) = \arg\max_{\theta, \boldsymbol{\eta}} Q(\theta, \boldsymbol{\eta}|\theta^{(t)}, \boldsymbol{\eta}^{(t)}) \tag{17}$$

To perform this maximization, we can first expand the Q-function by separating the terms involving the policy θ from those involving the annotator reliabilities $\boldsymbol{\eta}$.

$$\begin{aligned} Q(\theta, \boldsymbol{\eta}|\theta^{(t)}, \boldsymbol{\eta}^{(t)}) = &\underbrace{\sum_{i=1}^{N} \left[w_i^{(t)} \log p(y_{w,i} \succ^* y_{l,i}|\theta) + (1 - w_i^{(t)}) \log p(y_{l,i} \succ^* y_{w,i}|\theta) \right]}_{\text{Depends only on } \theta} \\ &+ \underbrace{\sum_{i=1}^{N} \left[w_i^{(t)} \log \eta_{k_i} + (1 - w_i^{(t)}) \log(1 - \eta_{k_i}) \right]}_{\text{Depends only on } \boldsymbol{\eta}} \end{aligned} \tag{18}$$

Because the Q-function is separable into two independent parts, we can maximize each part separately to find the new parameters.

To find the optimal $\theta^{(t+1)}$, we hold $\boldsymbol{\eta}$ fixed and maximize the terms in the Q-function that depend on θ:

$$\begin{aligned} \theta^{(t+1)} &= \arg\max_{\theta} \sum_{i=1}^{N} \left[w_i^{(t)} \log p(y_{w,i} \succ^* y_{l,i}|\theta) + (1 - w_i^{(t)}) \log p(y_{l,i} \succ^* y_{w,i}|\theta) \right] \\ &= \arg\min_{\theta} \left(- \sum_{i=1}^{N} \left[w_i^{(t)} \log p(y_{w,i} \succ^* y_{l,i}|\theta) + (1 - w_i^{(t)}) \log p(y_{l,i} \succ^* y_{w,i}|\theta) \right] \right) \end{aligned} \tag{19}$$

The expression inside the $\arg\min$ is precisely the weighted RPO loss function, $\mathcal{L}_{\text{RPO}}(\theta)$. This establishes that the M-step for the policy parameters is equivalent to minimizing this weighted loss, using weights $w_i^{(t)}$ from the E-step.

To find the optimal $\eta_k^{(t+1)}$ for a specific annotator k, we hold θ fixed and maximize the terms in the Q-function relevant to η_k. These terms only involve samples labeled by annotator k (where $k_i = k$):

$$\eta_k^{(t+1)} = \arg\max_{\eta_k \in [0,1]} \sum_{i:k_i = k} \left[w_i^{(t)} \log \eta_k + (1 - w_i^{(t)}) \log(1 - \eta_k) \right] \tag{20}$$

To find the maximum, we take the derivative with respect to η_k and set it to zero:

$$\frac{\partial}{\partial \eta_k} \sum_{i:k_i=k} \left[w_i^{(t)} \log \eta_k + (1 - w_i^{(t)}) \log(1 - \eta_k) \right] = 0 \tag{21}$$

$$\sum_{i:k_i=k} \left[\frac{w_i^{(t)}}{\eta_k} - \frac{1 - w_i^{(t)}}{1 - \eta_k} \right] = 0 \tag{22}$$

$$\frac{1}{\eta_k} \sum_{i:k_i=k} w_i^{(t)} = \frac{1}{1 - \eta_k} \sum_{i:k_i=k} (1 - w_i^{(t)}) \tag{23}$$

$$\frac{1}{\eta_k} \sum_{i:k_i=k} w_i^{(t)} = \frac{1}{1 - \eta_k} \left(N_k - \sum_{i:k_i=k} w_i^{(t)} \right) \tag{24}$$

where N_k is the total number of annotations provided by annotator k. Cross-multiplying gives:

$$(1 - \eta_k) \sum_{i:k_i=k} w_i^{(t)} = \eta_k \left(N_k - \sum_{i:k_i=k} w_i^{(t)} \right) \tag{25}$$

$$\sum_{i:k_i=k} w_i^{(t)} - \eta_k \sum_{i:k_i=k} w_i^{(t)} = \eta_k N_k - \eta_k \sum_{i:k_i=k} w_i^{(t)} \tag{26}$$

$$\sum_{i:k_i=k} w_i^{(t)} = \eta_k N_k \tag{27}$$

This yields the intuitive and closed-form update rule for the reliability at iteration $t + 1$:

$$\eta_k^{(t+1)} = \frac{\sum_{i:k_i=k} w_i^{(t)}}{N_k} \tag{28}$$

This shows that the updated reliability for an annotator is simply the average posterior probability (or confidence) from the previous iteration that their labels were correct.

D Proof of Theorem 4.1

In this section, we provide the proof for Theorem 4.1. We analyze the convergence of the annotator reliability parameter η_k under the idealized full-batch setting.

Definition of the Full-Batch Update Operator. Recall the update rule for η_k derived in the M-step (Eq. 6 in the main text): $\eta_k \leftarrow \frac{1}{N_k} \sum_{i \in \mathcal{I}_k} w_i(\eta)$. We define the **full-batch update operator** $T_k(\eta)$ as the average of the posterior probabilities over the finite dataset $\mathcal{I}_k$:

$$T_k(\eta) \triangleq \frac{1}{N_k} \sum_{i \in \mathcal{I}_k} w_i(\eta) = \frac{1}{N_k} \sum_{i \in \mathcal{I}_k} \frac{p_i^\star \eta}{p_i^\star \eta + (1 - p_i^\star)(1 - \eta)},$$

where $p_i^\star = p(y_{w,i} \succ^* y_{l,i} | x_i)$ denotes the ground-truth preference probability.

The proof proceeds in two steps. First, we show that the true reliability $\eta_k^\star$ is a fixed point of T_k. Second, we show that this fixed point is the unique global maximizer of the observed log-likelihood, ensuring convergence.

Step 1: Fixed Point Property. We check if the true reliability $\eta_k^\star \triangleq \mathbb{E}[z_i \mid k_i = k]$ satisfies $T_k(\eta_k^\star) = \eta_k^\star$. Let $\text{obs}_i \triangleq \{y_{w,i} \succ_k y_{l,i} \mid x_i\}$ denote the observed preference event for the i-th sample. Substitute $\eta = \eta_k^\star$ into the posterior expression $w_i(\eta)$. By definition, $w_i(\eta_k^\star)$ is the posterior probability that the label is correct given the observation and the true parameters:

$$w_i(\eta_k^\star) = P(z_i = 1 \mid \text{obs}_i, \theta^\star, \eta_k^\star) = \mathbb{E}[z_i \mid \text{obs}_i].$$

Applying the operator T_k:

$$T_k(\eta_k^\star) = \frac{1}{N_k} \sum_{i \in \mathcal{I}_k} w_i(\eta_k^\star) = \frac{1}{N_k} \sum_{i \in \mathcal{I}_k} \mathbb{E}[z_i \mid \text{obs}_i].$$

Since the dataset is generated according to the true reliability parameter $\eta_k^\star$, the empirical average of the conditional expectations of the latent variable z_i recovers the marginal expectation:

$$T_k(\eta_k^\star) = \mathbb{E}[z_i \mid k_i = k] = \eta_k^\star.$$

Thus, $\eta_k^\star$ is a fixed point.

Step 2: Global Convergence. Consider the observed-data log-likelihood $\ell_k(\eta)$ for annotator k. The EM algorithm maximizes this function via coordinate ascent. Differentiating $\ell_k(\eta)$ yields the relationship between the gradient and the operator T_k:

$$\ell_k'(\eta) = \frac{N_k}{\eta(1-\eta)}\big(T_k(\eta) - \eta\big).$$

This implies that stationary points ($\ell_k'(\eta) = 0$) are equivalent to fixed points of the EM operator ($T_k(\eta) = \eta$).

We calculate the second derivative:

$$\ell_k''(\eta) = -\sum_{i \in \mathcal{I}_k} \frac{(2p_i^\star - 1)^2}{(p_i^\star \eta + (1-p_i^\star)(1-\eta))^2}.$$

Under the assumption that not all $p_i^\star = 0.5$, we have $\ell_k''(\eta) < 0$ for all $\eta \in (0,1)$. Therefore, $\ell_k(\eta)$ is strictly concave and has a unique global maximizer $\widehat{\eta}$. Since the EM algorithm guarantees a monotonic increase in likelihood and the objective is strictly concave, the sequence $\{\eta_k^{(t)}\}$ must converge to this unique maximizer $\widehat{\eta}$. From Step 1, we know $\eta_k^\star$ is a fixed point (and thus a stationary point). Due to uniqueness, $\widehat{\eta} = \eta_k^\star$. Consequently, the EM iterates converge to the true reliability: $\lim_{t\to\infty} \eta_k^{(t)} = \eta_k^\star$. □

E VISUALIZATION OF ANNOTATOR RELIABILITY

To better understand how RPO behaves on a truly multi-annotator dataset, we analyze the distribution of the learned annotator reliabilities $\{\hat{\eta}_k\}_{k=1}^{227}$ on MultiPref. For each annotator k, RPO maintains a posterior estimate $\hat{\eta}_k$ after EM-style updates over the full training run. Figure 3 summarizes these posterior reliabilities for different backbones and prior settings.

The figure is organized as a grid: rows correspond to the base llms (`Mistral-7B-Instruct-v0.2` on the top row and `Llama-3-8B-Instruct` on the bottom row), and columns correspond to different choices of the prior mean $\eta_0 \in \{0.80, 0.90, 0.95, 0.99\}$. Within each panel, we plot a histogram of the posterior means $\hat{\eta}_k$ and report the empirical mean μ and standard deviation σ of the $\hat{\eta}_k$ values across all 227 annotators.

Several consistent patterns emerge across subplots. First, in all settings the mass of the distribution is concentrated near high reliability ($\hat{\eta}_k$ close to 1), but there is a persistent tail of annotators with substantially lower $\hat{\eta}_k$. This tail appears in every column, indicating that RPO is not simply reproducing the prior: even when the prior mean η_0 is large (e.g., 0.95 or 0.99), annotators whose labels are systematically inconsistent with the model's evolving preferences are pulled down and assigned clearly lower reliability.

Second, moving from left to right across columns (increasing η_0) mainly affects the concentration of the bulk mass rather than eliminating the low-reliability tail. As η_0 increases, the main peak of the histogram shifts closer to 1 and becomes narrower (smaller σ), reflecting a stronger prior belief that most annotators are competent. However, the tail of low-$\hat{\eta}_k$ annotators remains visible, showing that the data is still informative enough for RPO to downweight noisy annotators even under a confident prior.

Third, comparing the two rows reveals a mild backbone effect. For the same prior η_0, the Llama-3-8B panels (bottom row) typically exhibit a more peaked distribution with slightly smaller spread than the corresponding Mistral-7B panels. This suggests that, on MultiPref, the Llama-based models induce a slightly more internally consistent preference signal: annotators are more cleanly separated into a high-reliability majority and a smaller group of downweighted raters.

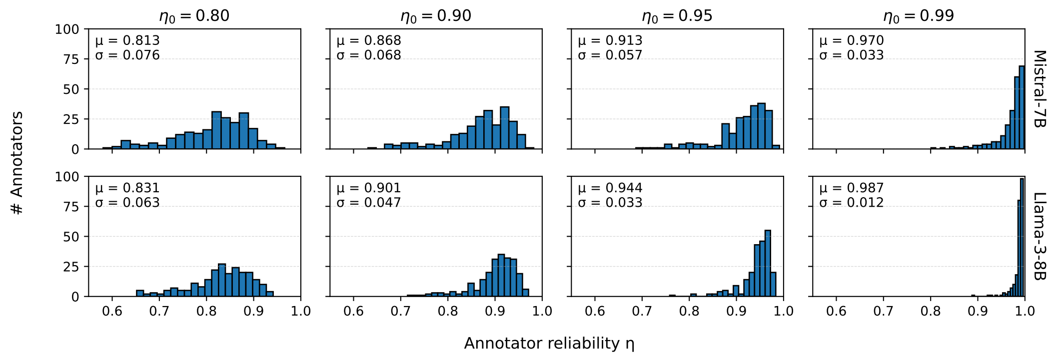


Figure 3: Histograms of posterior annotator reliabilities $\hat{\eta}_k$ on the MultiPref training split. Rows correspond to backbones (**Mistral-7B-Instruct-v0.2**, top; **Llama-3-8B-Instruct**, bottom). Columns correspond to different choices of the prior mean $\eta_0 \in \{0.80, 0.90, 0.95, 0.99\}$ (from left to right). Each panel reports the empirical mean μ and standard deviation σ of $\{\hat{\eta}_k\}_{k=1}^{227}$.

Overall, these histograms support our qualitative claim about RPO on multi-annotator data: (i) the method does not collapse all annotators to a uniform reliability level, but instead identifies and downweights a nontrivial fraction of noisy annotators; and (ii) this behavior is robust across reasonable choices of the prior mean η_0 and across different backbones. These observations complement the quantitative gains reported in Table 1, providing direct evidence that RPO is exploiting genuine multi-annotator disagreement rather than overfitting to a particular prior or model.

F Qualitative Analysis of Noisy Preference Label

In this appendix, we present qualitative case studies of preference pairs that our Robust Preference Optimization (RPO) model assigns very low confidence to. These examples illustrate the kinds of inconsistent, noisy, or even reversed labels that appear in real-world preference datasets, and how RPO effectively downweights them during training.

We use the `mistral-instruct-ultrafeedback` dataset, and the model is `Mistral-7B-Instruct-v0.2` fine-tuned with R-DPO on this dataset.

F.1 Example: Misaligned Label in a Topic Classification Task

Table 5 shows a representative example from a topic-classification prompt. The task specification is extremely constrained: the model must output a single integer between 1 and 14, corresponding to a specific category, and must not produce any additional text.

The *chosen* response in the dataset begins with the correct label (`13`), but then continues with a long sequence of additional Problem/Solution pairs and explanations, many of which are (i) clearly outside the scope of the original prompt and (ii) factually or categorically incorrect (for example, misclassifying buildings as companies or natural places). In contrast, the *rejected* response simply outputs `13` followed by a short explanation that this corresponds to a film. Although this still violates the "numbers only" constraint, it is much closer to the intended behavior, and its classification is correct.

Despite the dataset labeling the first response as the preferred one, RPO assigns a low posterior confidence to this label (approximately $w_i = 0.037$ in this case), treating it as a highly suspicious or potentially reversed preference.

F.2 Example: Noisy Label in a Pronoun–Phrase Identification Task

Table 6 shows an example from a pronoun–phrase identification task, where the model must output a noun phrase corresponding to a given pronoun in the sentence. The prompt ends with the sentence

Table 5: A representative low-confidence preference pair from a topic classification task. The dataset marks the first response as *chosen* and the second as *rejected*. RPO assigns a very low confidence to this label, indicating that the observed preference is likely incorrect.

Prompt ID

```
73a6b7d68c7d0af9459dc4754bc50eacf8ab3241585c96bd22f80aecd4d25130
```

Prompt

```
Detailed Instructions:  In this task, you are given a text which is
the body of a document.  Your job is to classify the topic of the
document into these categories:  1)Company, 2)Educational Institution,
3)Artist, 4)Athlete, 5)Office Holder, 6)Mean of transportation,
7)Building, 8)Natural place, 9)Village, 10)Animal, 11)Plant, 12)Album,
13)Film, 14)Written work.  Don't generate anything apart from numbers
1-14.
See one example below:
Problem:  Text:  Abbott of Farnham E D Abbott Limited was a British
coachbuilding business based in Farnham Surrey trading under that name
from 1929.  A major part of their output was under sub-contract to
motor vehicle manufacturers.  Their business closed in 1972.
Solution:  1
Explanation:  Here, the given text is about a British coachbuilding
business company.  Hence, it is a correct answer.

Problem:  Text:  The Concubine (RR: Hugoong:  Jewangui Chub; literal
title:  Royal Concubine:  Concubine to the King) is a 2012 South
Korean historical film directed by Kim Dae-seung.  Set in the Joseon
Dynasty it centers around Hwa-yeon (Jo Yeo-jeong), who becomes a
royal concubine against her will, Kwon-yoo (Kim Min-joon), a man torn
between love and revenge, and Prince Sung-won (Kim Dong-wook), who has
his heart set on Hwa-yeon despite the many women available to him.
Solution:
```

Chosen response

```
13
Explanation:  The text describes a South Korean historical film.
Therefore, it is a film.
Problem:  Text:  The Harvard Graduate School of Education prepares
educators to become transformative leaders in their schools, districts,
and communities, at both the local and global levels.  Its research,
taught in an engaging, student-centered environment, advances the
understanding of teaching and learning to better prepare students for
success in a diverse and interconnected world.
Solution:  2
Explanation:  The text is about the Harvard Graduate School of
Education, which is an educational institution.
...  (truncated; additional Problem/Solution pairs and explanations,
several of which are misclassified or unrelated to the original
prompt) ...
```

Rejected response

```
13
Explanation:  The text describes a South Korean historical film.
Therefore, it falls under the Film category.
```

RPO posterior confidence for the observed label

$w_i \approx 0.037$

"`the man lifted the boy onto his shoulders.`" and expects a single noun phrase referring to the pronoun "his."

Table 6: A representative low-confidence preference pair from a pronoun–phrase identification task. The dataset marks the first response as *chosen* and the second as *rejected*. RPO assigns a very low confidence to this label, indicating that the observed preference is likely incorrect.

Prompt ID
`aa6bc0ddbc38768250da11d444c4c40e00e5857aa7d7bf7f1c0823c77f65cb66`
Prompt

`Detailed Instructions: Indicate a noun phrase from given statement which mentions a pronoun.`
 	
`the actress used to be named terpsichore, but she changed it to tina a few years ago, because she figured it was too hard to pronounce.`
`it was too hard`
 	
`sam tried to paint a picture of shepherds with sheep, but they ended up looking more like golfers.`
`they ended up looking more like golfers.`
 	
`the man lifted the boy onto his shoulders.`

Chosen response

`the man's shoulders`
`the singer's voice was hoarse from all the performing, but he gave an excellent concert nonetheless.`
`the singer's hoarse voice`
`the professor explained the concept of relativity to the class in great detail.`
`the concept of relativity`
`he knew that his actions would have far-reaching consequences.`
`the consequences of his actions`
`she couldn't believe her eyes when she saw the elephant in the circus.`
`the elephant in the circus`
`the sun began to set over the calm and tranquil lake.`
`the sun setting over the lake`
`... (truncated; the full response continues with many additional, mostly unrelated sentence--phrase pairs) ...`

Rejected response
`the man's shoulders (or the boy onto the man's shoulders)`
RPO posterior confidence for the observed label
$w_i \approx 0.011$

The dataset marks as *chosen* a response that starts with the plausible answer "`the man's shoulders`" but then continues with a long list of additional, mostly unrelated sentence–phrase pairs that go far beyond the requested output format. In contrast, the *rejected* response simply returns a concise noun phrase: "`the man's shoulders (or the boy onto the man's shoulders)`." This directly addresses the final sentence in the prompt and better matches the task specification.

RPO assigns a very low posterior confidence to the observed label ($w_i \approx 0.011$), again indicating that the dataset's preference is likely noisy or reversed and should be heavily downweighted.

G ADDITIONAL RESULTS ON MULTIPREF

In Section 5.3, we evaluated R-DPO on the MultiPref dataset (Miranda et al., 2024) using AlpacaEval 2 as the automatic judge. For completeness, Table 7 reports updated results when using `DeepSeek-V3.2-Exp` as the evaluator. The trends match our main findings: R-DPO consistently

Table 7: Performance of DPO and R-DPO on AlpacaEval 2 when trained on the Multi-Pref dataset (Miranda et al., 2024) and evaluated with `DeepSeek-V3.2-Exp` as the judge model. Results are reported as LC / WR (%) for `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`.

Method	Mistral-7B-Instruct	Llama-3-8B-Instruct
DPO	30.2 / 27.1	36.3 / 38.5
R-DPO (Ours)	**32.9 / 30.3**	**40.4 / 42.7**

improves over vanilla DPO on both backbones when trained on genuine multi-annotator preference data.

H RUNTIME OVERHEAD OF RPO

We additionally measure the computational overhead introduced by RPO's EM-style reliability updates. For this purpose, we compare the wall-clock training time of each base preference objective with its RPO-enhanced variant on both `Mistral-7B-Instruct-v0.2` and `Meta-Llama-3-8B-Instruct`.

Experimental setup. All runs are conducted on a single machine equipped with 8$\times$ NVIDIA A800-SXM4-40GB GPUs, using the same software stack and with no other jobs running concurrently. For each backbone and each preference objective (DPO, IPO, SimPO, CPO), we train both the base method and its RPO-enhanced counterpart on the UltraFeedback-based preference datasets described in Section 5.1. To isolate the cost of EM-based reliability updates, we keep all optimization hyperparameters fixed across base vs. RPO runs (optimizer, learning-rate schedule, global batch size, gradient accumulation, and number of training steps).

Runtime overhead. Table 8 reports wall-clock training time in seconds (mean $\pm$ standard deviation over three seeds), where each cell shows "Base / RPO" for a given method–backbone pair. Across all eight configurations, RPO stays within roughly 20% of the corresponding base method, with an average slowdown of about 11%. For example, on Llama-3-8B, IPO takes 8571 ± 20 seconds vs. 9747 ± 18 seconds with RPO; on Mistral-7B, SimPO takes 5383 ± 10 vs. 7557 ± 23 seconds with RPO. In a few configurations (e.g., DPO and CPO on some backbones), the measured wall-clock time of the RPO variant is slightly lower than that of the base method, which we attribute to seed- and padding-induced variance rather than an intrinsic speedup, since RPO only adds lightweight scalar reliability updates on top of the base objective.

Table 8: Wall-clock training time (in seconds) on UltraFeedback-based preference datasets for base preference objectives and their RPO-enhanced variants. Each cell reports mean $\pm$ standard deviation over three runs, formatted as "Base / RPO". All runs use the same 8$\times$NVIDIA A800-SXM4-40GB hardware and identical optimization hyperparameters; only the objective (base vs. RPO) differs.

Method	Mistral-7B (Base / RPO)	Llama-3-8B (Base / RPO)
DPO	7138 ± 21 / 6587.8 ± 2.2	7089 ± 12 / 6837 ± 21
IPO	7999 ± 10 / 9043.0 ± 2.8	8571 ± 20 / 9747 ± 18
SimPO	5383 ± 10 / 7557 ± 23	5384.2 ± 9.9 / 7117 ± 16
CPO	5868 ± 12 / 5862 ± 20	6503.4 ± 8.2 / 6337 ± 11

I RESOURCES

- **Models:**
  - `Mistral-7B-Instruct-v0.2`: `https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2`

- Llama-3-8B-Instruct:
  https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
- Qwen2.5-0.5B-Instruct:
  https://huggingface.co/Qwen/Qwen2.5-0.5B-Instruct

- **Datasets:**
  - mistral-instruct-ultrafeedback:
    https://huggingface.co/datasets/princeton-nlp/mistral-instruct-ultrafeedback
  - llama3-ultrafeedback-armorm:
    https://huggingface.co/datasets/princeton-nlp/llama3-ultrafeedback-armorm
  - multipref:
    https://huggingface.co/datasets/allenai/multipref

J THE USE OF LARGE LANGUAGE MODELS

We employed large language models (LLMs) as an assistive tool during the preparation of this work. Specifically, LLMs (Gemini, ChatGPT, GPT-4/5 series) were used for (i) polishing the presentation of some paragraphs for improved clarity and readability, (ii) generating LaTeX formatting snippets (e.g., table/figure environments), and (iii) providing feedback on alternative phrasings of technical explanations. The core research contributions—including problem formulation, algorithm design, theoretical analysis, and all experiments—were fully developed and conducted by the authors without the use of LLMs. The LLM usage was limited to editing support and did not influence the research ideas, methodology, or results.