VALUE-FREE POLICY OPTIMIZATION VIA REWARD PARTITIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Single-trajectory preference optimization methods learn from datasets of ((prompt, response, reward)) tuples, offering a practical alternative to pairwise preference learning by directly leveraging scalar feedback. Existing approaches such as Direct Reward Optimization (DRO) have demonstrated promising results but rely on value function estimation, introducing additional variance, optimization complexity, and sensitivity to off-policy data. We introduce *Reward Partition Optimization* (RPO), a simple and scalable reward-driven objective that eliminates the need for value function learning. RPO normalizes rewards through a partition-based formulation estimated directly from prompt-level reward distributions, yielding a stable supervised optimization objective without auxiliary models or reinforcement learning loops. We evaluate RPO across multiple encoder-decoder and decoder-only language models using automatic metrics, LLM-as-a-judge evaluations, and optimization stability analyses. Experimental results show that RPO consistently outperforms strong baselines, including SFT, KTO, and DRO, while producing more aligned, diverse, and less toxic generations.

1 INTRODUCTION

Aligning large language models (LLMs) typically relies on preference or reward data, either through reinforcement learning from human feedback (RLHF) Yuan et al. (2023a); Dong et al. (2024); Ahmadian et al. (2024) or direct preference optimization Rafailov et al. (2023); Tang et al. (2024); Richemond et al. (2024); Ethayarajh et al. (2024). RLHF generally combines supervised fine-tuning (SFT) Zhang et al. (2023), reward modeling from human-labeled preference pairs, and reinforcement learning such as PPO Schulman et al. (2017) or TRPO Schulman et al. (2015), as used in systems such as InstructGPT Ouyang et al. (2022), GPT-4 Achiam et al. (2023), and Claude 3 Anthropic (2024). Despite its effectiveness, this pipeline is costly and brittle due to preference collection, reward modeling, and reinforcement learning. RLAIF Xu et al. (2025); Du et al. (2023); Yuan et al. (2023b); Wang et al. (2025); Yuan et al. (2024) reduces human annotation by using AI-generated feedback, but still relies on large evaluators, adding inference and computational costs.

An alternative is to directly optimize policies from preference data using supervised objectives, as in DPO Rafailov et al. (2023), GPO Tang et al. (2024), and SLiC Zhao et al. (2023). These methods avoid reward models and reinforcement learning, enabling simpler and more stable training, and have been used in models such as LLaMA 3 Grattafiori et al. (2024), Qwen 2 Yang et al. (2025), and Nemotron-4 Adler et al. (2024). However, they require pairwise preference annotations, which become increasingly difficult and costly as model quality improves and differences between responses become more subtle.

This motivates *single-trajectory* learning, where each example consists of a prompt, a response, and a scalar reward. This simpler feedback format is exploited by methods such as KTO Ethayarajh et al. (2024) and DRO Richemond et al. (2024). DRO achieves strong results by regressing the optimality condition of KL-regularized reinforcement learning, but requires learning a value function from off-policy data. This introduces additional optimization instability and makes the policy supervision relative to a reference model rather than directly determined by the observed reward.

We introduce *Reward Partition Optimization* (RPO), a simple objective for single-trajectory policy optimization that replaces value estimation with empirical reward normalization. Given prompt-response-reward triplets, RPO estimates a prompt-level reward distribution and uses it to directly supervise the policy, while retaining a reference model for regularization. This removes the auxil-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

iary value function and reinforcement learning loop, yielding a stable, fully offline objective with direct reward supervision.

We make the following contributions. We introduce RPO and its reward-partition formulation for value-free single-trajectory optimization. We provide a theoretical analysis connecting the proposed objective to the optimal policy structure and analyze the limitations of value-based single-trajectory optimization. Finally, we evaluate RPO across encoder-decoder and decoder-only language models, showing consistent improvements over SFT, KTO, and DRO on automatic metrics, LLM-based evaluations, and optimization stability.

2 BACKGROUND

We briefly review the alignment objectives most closely related to RPO, from reward-based reinforcement learning to direct optimization from preferences and scalar feedback. The key distinction is whether supervision is provided through pairwise preferences or scalar rewards, and whether value estimation is required.

2.1 REINFORCEMENT LEARNING FROM HUMAN FEEDBACK (RLHF)

RLHF Yuan et al. (2023a); Dong et al. (2024); Ahmadian et al. (2024) typically consists of supervised fine-tuning (SFT) Zhang et al. (2023), reward-model learning, and policy optimization. Given preference pairs $(x, y^{\text{win}}, y^{\text{lose}})$, a reward model r_ϕ is commonly trained with the Bradley–Terry model Bradley & Terry (1952):

$$\mathcal{L}_{\text{RM}} = -\mathbb{E}\big[\log \sigma\left(r_\phi(x, y^{\text{win}}) - r_\phi(x, y^{\text{lose}})\right)\big]\,. \tag{1}$$

The resulting reward is then maximized while regularizing the policy toward a reference policy:

$$\max_{\pi}\ \mathbb{E}_{x,y\sim\pi}\left[r_\phi(x,y) - \tau\,\mathrm{KL}\left(\pi(\cdot|x)\,\|\,\pi_{\text{ref}}(\cdot|x)\right)\right], \tag{2}$$

typically using RL algorithms such as PPO Schulman et al. (2017). Although effective, this pipeline requires reward-model training and an additional reinforcement-learning stage, introducing computational cost, sampling requirements, and optimization sensitivity. RLAIF Xu et al. (2025); Du et al. (2023); Yuan et al. (2023b); Wang et al. (2025); Yuan et al. (2024) replaces human feedback with AI-generated feedback, but still relies on learned or large-model evaluators.

2.2 DIRECT PREFERENCE OPTIMIZATION

DPO Rafailov et al. (2023), IPO Azar et al. (2024), and SLiC Zhao et al. (2023) avoid explicit reward modeling and reinforcement learning by directly optimizing the policy from pairwise preferences. For a preference pair $(x, y^{\text{win}}, y^{\text{lose}})$, a general objective compares policy-to-reference log-ratios:

$$\mathcal{L}_{\text{pref}} = \mathbb{E}\left[f\left(\beta\left[\log\frac{\pi_\theta(y^{\text{win}}|x)}{\pi_{\text{ref}}(y^{\text{win}}|x)} - \log\frac{\pi_\theta(y^{\text{lose}}|x)}{\pi_{\text{ref}}(y^{\text{lose}}|x)}\right]\right)\right], \tag{3}$$

where f is respectively the negative log-sigmoid, hinge, or squared loss for DPO, SLiC, and IPO. These methods provide simple supervised objectives, but require pairwise preference annotations. As model quality increases, obtaining reliable pairwise judgments can become increasingly costly, motivating methods that directly exploit scalar feedback.

2.3 SINGLE-TRAJECTORY REWARD OPTIMIZATION

Single-trajectory methods operate on datasets $\mathcal{D} = \{(x_i, y_i, r_i)\}_{i=1}^N$, where each response is associated with a scalar reward. KTO Ethayarajh et al. (2024) is an early approach in this setting, using prospect-theoretic loss aversion Tversky & Kahneman (1992). With

$$\Delta_\theta(x,y) = \log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}, \qquad z_0(x) = \mathbb{E}_{y\sim\pi_\theta}[\Delta_\theta(x,y)], \tag{4}$$

KTO transforms the scalar feedback through asymmetric gain/loss terms around the reference point z_0. While effective, its formulation relies on assumptions specific to binary feedback and does not explicitly model the reward structure underlying the observed trajectories.

DRO Richemond et al. (2024) instead starts directly from the KL-regularized reward objective

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{x,y\sim\pi} \left[r(x,y) - \tau \, \mathrm{KL}\left(\pi(\cdot|x) \,\|\, \pi_{\mathrm{ref}}(\cdot|x)\right)\right]. \tag{5}$$

Its optimal policy satisfies

$$\pi^*(y|x) = \pi_{\mathrm{ref}}(y|x) \exp\left(\frac{r(x,y) - V^*(x)}{\tau}\right), \tag{6}$$

where the soft value is

$$V^*(x) = \tau \log \mathbb{E}_{y\sim\pi_{\mathrm{ref}}} \left[\exp\left(\frac{r(x,y)}{\tau}\right)\right]. \tag{7}$$

Equivalently,

$$r(x,y) = V^*(x) + \tau \log \frac{\pi^*(y|x)}{\pi_{\mathrm{ref}}(y|x)}. \tag{8}$$

DRO fits this identity by jointly estimating the policy and a value function:

$$\mathcal{L}_{\mathrm{DRO}}(\theta,\phi) = \frac{1}{2}\mathbb{E}_{(x,y,r)\sim\mathcal{D}} \left[r - V_\phi(x) - \tau \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right]^2. \tag{9}$$

This formulation provides a principled bridge between scalar rewards and KL-regularized policy optimization, but its value estimate is learned from off-policy samples and jointly optimized with the policy. Errors in V_ϕ therefore directly affect the policy objective and can introduce optimization instability.

We propose RPO, that builds on this same KL-regularized structure while removing the auxiliary value function. Instead of learning $V(x)$, it estimates the normalization induced by the observed reward distribution directly from the data. This yields a value-free supervised objective, which we derive in Section 3.

3 Proposed method: Reward Partition Optimization (RPO)

3.1 From Value Estimation to Reward Partitioning

We start from the KL-regularized policy optimization problem introduced in Section 2:

$$\pi^*(y|x) = \frac{\pi_{\mathrm{ref}}(y|x)\exp(r(x,y)/\tau)}{Z(x)}, \tag{10}$$

where

$$Z(x) = \mathbb{E}_{y\sim\pi_{\mathrm{ref}}(\cdot|x)} \left[\exp(r(x,y)/\tau)\right] \tag{11}$$

is the partition function. Equivalently, defining the soft value $V^*(x) = \tau \log Z(x)$ gives

$$r(x,y) = V^*(x) + \tau \log \frac{\pi^*(y|x)}{\pi_{\mathrm{ref}}(y|x)}. \tag{12}$$

This identity reveals the role of the value function in DRO Richemond et al. (2024): $V^*(x)$ is not an independent source of supervision, but the normalization required to turn exponentiated rewards into a valid policy. DRO approximates this normalization with a learned function $V_\phi(x)$ and jointly optimizes (π_θ, V_ϕ). RPO takes a different approach: when multiple responses are observed for the same prompt, the reward normalization can be computed directly from the empirical set of observed candidates, eliminating the auxiliary value model.

3.2 Empirical Reward Partition

Let $\mathcal{D} = \{(x_i, y_i, r_i)\}_{i=1}^N$ and $\mathcal{I}_x = \{i : x_i = x\}$ denote the set of observations associated with prompt x. We define the empirical reward partition over the observed candidates as

$$\widehat{Z}(x) = \sum_{j\in\mathcal{I}_x} w_j(x) \exp\left(\frac{r_j}{\tau}\right), \tag{13}$$

Algorithm 1: Reward Partition Optimization (RPO)

Input: Dataset $\mathcal{D} = \{(x_i, y_i, r_i)\}_{i=1}^{N}$, policy π_θ, reference policy π_{ref}, temperature τ
for *each prompt* x **do**
  Construct $\mathcal{I}_x = \{i : x_i = x\}$;
  Compute $\widehat{Z}(x)$ using Eq. equation 13;
  Compute $\widehat{V}(x) = \tau \log \widehat{Z}(x)$;
while *training* **do**
  Minimize $\mathcal{L}_{\mathrm{RPO}}(\theta)$ in Eq. equation **??** with respect to θ;
Output: Optimized policy π_θ

where $w_j(x)$ denotes the reference-policy weight associated with candidate y_j. In the simplest empirical formulation, $w_j(x) = \pi_{\mathrm{ref}}(y_j|x)$. The corresponding empirical normalization is

$$\widehat{V}(x) = \tau \log \widehat{Z}(x). \tag{14}$$

The essential difference from DRO is that $\widehat{V}(x)$ is not represented by a parameterized value network and is not jointly learned with the policy. It is a deterministic quantity computed from the observed reward distribution for each prompt. RPO therefore turns value estimation into a reward-partition computation.

3.3 RPO Objective

Substituting the empirical normalization into the optimality condition equation 12 yields the target

$$t_i = \frac{r_i - \widehat{V}(x_i)}{\tau}. \tag{15}$$

RPO then directly regresses the policy log-ratio $\log \frac{\pi_\theta(y_i|x_i)}{\pi_{\mathrm{ref}}(y_i|x_i)}$ toward this reward-derived target:

$$\mathcal{L}_{\mathrm{RPO}}(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \left[\log \frac{\pi_\theta(y_i|x_i)}{\pi_{\mathrm{ref}}(y_i|x_i)} - \frac{r_i - \widehat{V}(x_i)}{\tau} \right]^2. \tag{16}$$

Thus, RPO has only one trainable object, π_θ. The rewards determine the target, the reference policy provides the KL anchor, and the empirical partition provides the prompt-dependent normalization. No reward model, value network, or reinforcement-learning loop is required.

3.4 Optimization and Interpretation

For a sample i, define the residual

$$\delta_i = \log \frac{\pi_\theta(y_i|x_i)}{\pi_{\mathrm{ref}}(y_i|x_i)} - \frac{r_i - \widehat{V}(x_i)}{\tau}. \tag{17}$$

The policy is optimized by minimizing δ_i^2. Since $\widehat{V}(x_i)$ depends only on the observed rewards and the reference policy, it is treated as a fixed target during policy optimization. The resulting update therefore has the standard supervised-learning form: samples whose current log-ratio is below their reward-derived target are upweighted, whereas samples whose log-ratio exceeds the target are downweighted.

The resulting procedure can be summarized as follows (see Algorithm 1): for each prompt, RPO first constructs an empirical reward partition, converts each reward into a normalized target, and then fits the policy to these targets. Unlike pairwise preference objectives, the supervision is absolute at the sample level; unlike DRO, the normalization is obtained without learning a separate value function. RPO consequently preserves the KL-regularized optimal-policy structure while replacing parametric value estimation with direct reward partitioning.

4 EXPERIMENTS

We evaluate RPO along three complementary dimensions: in-domain alignment, out-of-distribution (OOD) generalization, and sensitivity to its design choices. We compare RPO with SFT, KTO, and DRO using automatic metrics and LLM-based preference judgments across encoder-decoder and decoder-only language models.

4.1 DATASETS AND BASE MODELS

Following prior offline RLHF work Richemond et al. (2024), we train all methods on UltraFeedback Cui et al. (2023). UltraFeedback provides offline $(\text{prompt}, \text{response}, \text{reward})$ tuples, with multiple candidate responses per prompt and scalar rewards ranging from 0 to 10. This structure directly matches RPO's reward-partition formulation, which uses responses associated with the same prompt to construct an empirical normalization. Rewards are standardized to zero mean and unit variance on the training set.

For in-domain evaluation, we use a held-out set of 10,000 UltraFeedback prompts and their highest-quality responses (reward $= 10$). For OOD evaluation, we use AlpacaEval Li et al. (2023), MT-Bench Zheng et al. (2023), IFEval Zhou et al. (2023), and GSM8K Cobbe et al. (2021), covering open-ended instruction following, multi-turn dialogue, instruction constraint following, and mathematical reasoning.

We evaluate both encoder-decoder and decoder-only architectures. The encoder-decoder models are Flan-T5 Small (77M), Large (783M), and XL (2.85B) Roberts et al. (2023). The decoder-only models are instruction-tuned Mistral-7B Jiang et al. (2023), LLaMA-3-8B Touvron et al. (2023), and Qwen2.5-7B Yang et al. (2024). Each model is initialized from its corresponding instruction-tuned checkpoint, which is also used as the reference policy π_{ref}.

4.2 EVALUATION PROTOCOL

We evaluate generation quality using complementary automatic and LLM-based metrics. On the UltraFeedback validation set, generated responses are compared with the corresponding preferred responses using BERTScore Zhang et al. (2019) and ROUGE-L Lin (2004). We additionally report Toxicity, where lower values indicate safer generations, and Distinct-2 Li et al. (2016) to measure response diversity.

For broader alignment evaluation, we use LLM-as-a-judge pairwise comparisons Gu et al. (2024). GPT-4o Hurst et al. (2024) and Claude-3.5-Sonnet Anthropic (2024) independently compare RPO-generated responses with those of each baseline, selecting the more helpful and relevant response or returning `Neutral`. We report the resulting Win Rate (WR). Unlike reference-based metrics, this evaluation can be applied consistently to both in-domain and OOD benchmarks.

4.3 TRAINING SETUP AND BASELINES

All methods are trained under the same data and computational budget using AdamW Loshchilov & Hutter (2019), a learning rate of 10^{-4}, and a linear warmup–decay schedule with 150 warmup steps. Experiments use 4 NVIDIA A100 80GB GPUs with distributed training. Unless otherwise specified, the temperature is fixed to $\tau = 1.0$.

We compare RPO with SFT, KTO Ethayarajh et al. (2024), and DRO Richemond et al. (2024). SFT provides a supervised reference baseline, while KTO and DRO represent direct optimization from scalar feedback. All methods start from the same instruction-tuned checkpoints, ensuring that differences in performance arise from the alignment objective rather than from model initialization.

Figure 1 reports training time across model families. RPO requires less training time than KTO and DRO across the evaluated architectures, consistent with its value-free formulation and single-policy optimization.

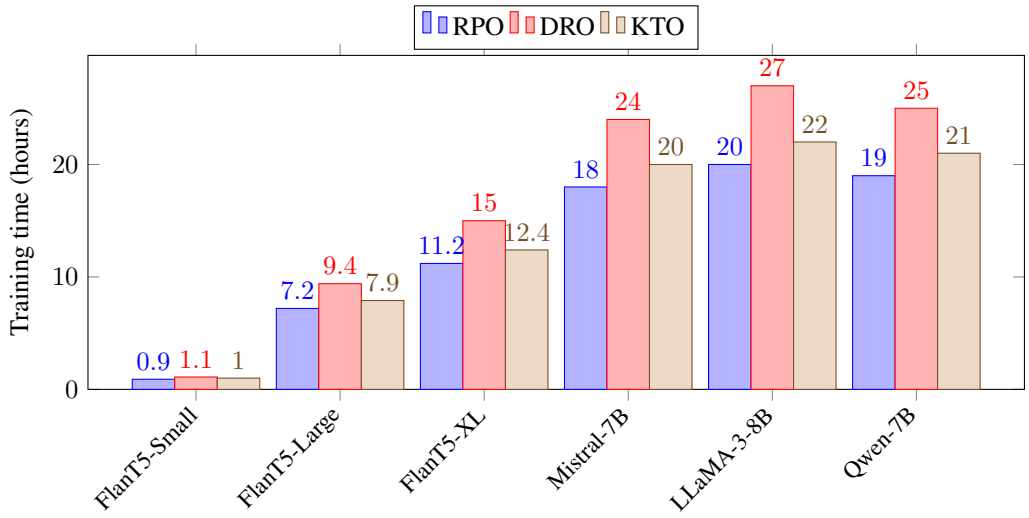

Figure 1: Training time across model families using 4×A100 80GB GPUs.

4.4 In-Domain Evaluation on UltraFeedback Validation Set

All methods are trained on UltraFeedback under identical data, optimization, and compute settings. For in-domain evaluation, we use a held-out set of 10,000 prompts with their highest-quality completions (reward $= 10$). Each model generates a response $\hat{y}$ for every prompt, which is evaluated against the corresponding preferred completion using both automatic metrics and LLM-based pairwise judgments.

Table 1 reports the results across all six architectures. RPO improves semantic and lexical alignment over the three baselines, with higher BERTScore and ROUGE-L in every model family, while also achieving the highest Distinct-2 and the lowest toxicity in all reported cases. The gains are not restricted to a single architecture: BERTScore improves by 0.010–0.016 over DRO, while ROUGE-L gains reach 0.013–0.037. RPO also produces longer responses, suggesting that its improvements in similarity and diversity are not obtained by simply generating shorter outputs. The LLM-as-a-judge results provide a complementary preference-based evaluation: GPT-4o and Claude-3.5 assign substantial pairwise preference to RPO over the corresponding baselines, including the larger FlanT5-XL model where KTO obtains particularly high preference scores despite weaker automatic metrics. Overall, these results show that the behavior induced by RPO is reflected consistently across reference-based quality, diversity, safety, and model-based preference evaluation.

We next examine whether these gains are accompanied by controlled policy updates. Figure 2 tracks the KL divergence from π_{ref} during training on four representative architectures. RPO exhibits a smooth decrease followed by a stable plateau across all four models, whereas KTO shows more pronounced oscillations and DRO exhibits sustained divergence growth, particularly on LLaMA-3-8B and Qwen2.5-7B. This behavior is consistent with the value-free formulation of RPO: the policy is optimized against a reward-derived, prompt-level normalization rather than jointly fitting an auxiliary value function. Thus, the in-domain results combine improved output quality with more controlled policy drift.

4.5 Out-of-Distribution Generalization Benchmarks

To assess whether RPO generalizes beyond its training distribution, we evaluate all trained models on four unseen benchmarks: AlpacaEval, MT-Bench, IFEval, and GSM8K. These tasks cover open-ended instruction following, multi-turn dialogue, constraint following, and mathematical reasoning, respectively. Since these benchmarks do not provide reference responses suitable for overlap-based evaluation, we use pairwise LLM-as-a-judge comparisons with GPT-4o and Claude-3.5-Sonnet. For each benchmark, we report the win rate (WR) of RPO against each baseline; WR > 50 indicates a preference for RPO.

Model	Method	BERTScore ↑	ROUGE-L ↑	Toxicity ↓	Distinct-2 ↑	Avg Len	GPT-4o WR ↑	Claude-3.5 WR ↑
FlanT5-Small	SFT	0.842	0.228	0.018	0.71	54	81.4	79.2
	KTO	0.850	0.237	0.013	0.74	59	83.1	80.8
	DRO	0.856	0.252	0.015	0.76	63	84.5	82.1
	RPO	**0.866**	**0.289**	**0.012**	**0.83**	**72**	-	-
FlanT5-Large	SFT	0.846	0.241	0.017	0.73	60	85.8	83.2
	KTO	0.840	0.249	0.024	0.75	65	87.0	84.6
	DRO	0.859	0.303	0.013	0.79	71	85.9	83.8
	RPO	**0.871**	**0.316**	**0.009**	**0.86**	**84**	-	-
FlanT5-XL	SFT	0.851	0.244	0.016	0.74	66	86.2	84.1
	KTO	0.788	0.248	0.024	0.70	57	92.7	90.1
	DRO	0.854	0.250	0.012	0.78	73	85.1	83.0
	RPO	**0.870**	**0.297**	**0.008**	**0.88**	**91**	-	-
Mistral-7B	SFT	0.861	0.312	0.013	0.81	88	88.6	86.2
	KTO	0.864	0.319	0.012	0.82	91	89.8	87.3
	DRO	0.869	0.326	0.011	0.84	94	88.9	86.5
	RPO	**0.881**	**0.348**	**0.007**	**0.91**	**108**	-	-
LLaMA-3-8B	SFT	0.858	0.305	0.014	0.80	85	89.1	86.8
	KTO	0.862	0.311	0.013	0.81	88	90.2	87.9
	DRO	0.867	0.320	0.012	0.83	93	89.5	86.9
	RPO	**0.879**	**0.344**	**0.007**	**0.90**	**106**	-	-
Qwen2.5-7B	SFT	0.860	0.309	0.013	0.80	86	89.4	87.1
	KTO	0.863	0.315	0.012	0.82	89	90.7	88.2
	DRO	0.868	0.323	0.011	0.84	95	89.8	87.4
	RPO	**0.880**	**0.346**	**0.007**	**0.91**	**107**	-	-

Table 1: In-domain evaluation results on the UltraFeedback validation set. In addition to automatic metrics, we report GPT-4o and Claude-3.5 pairwise win rates (WR) comparing RPO against each corresponding baseline method. Higher WR indicates stronger preference for RPO generations.

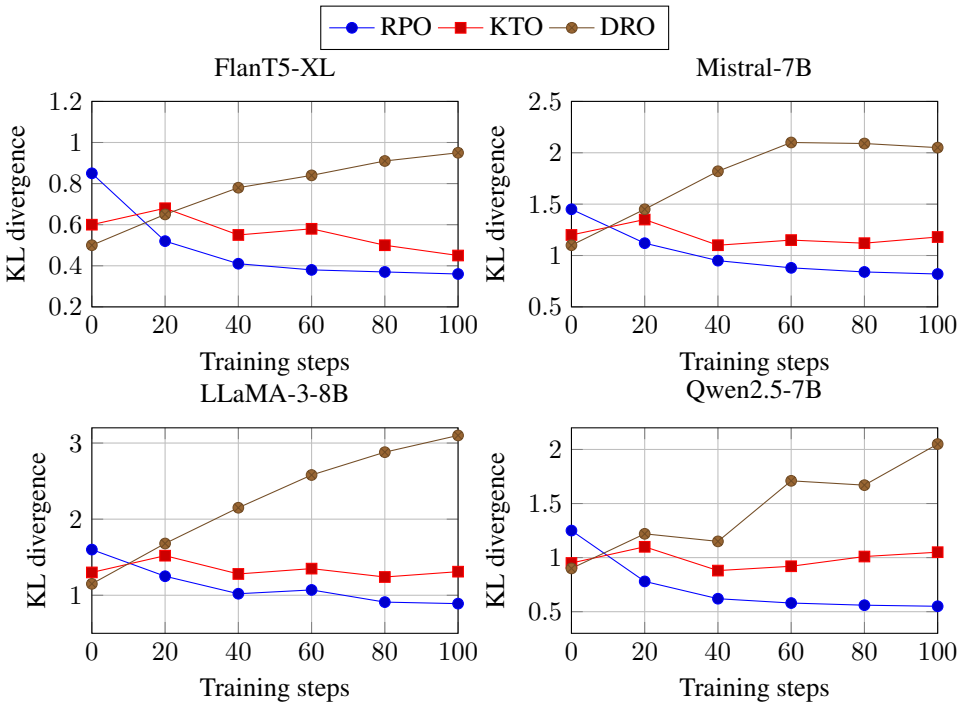

Figure 2: KL divergence evolution across architectures. Each method exhibits distinct optimization dynamics depending on model type, with RPO showing consistently controlled policy drift, KTO displaying architecture-dependent stability with occasional oscillations, and DRO producing progressively increasing divergence, especially in decoder-only models.

Model	Baseline	GPT-4o				Claude-3.5			
		AlpacaEval	MT-Bench	IFEval	GSM8K	AlpacaEval	MT-Bench	IFEval	GSM8K
FlanT5-Small	SFT	61.4	58.8	63.5	53.2	60.1	57.9	62.8	52.4
	KTO	55.9	53.7	57.1	51.8	54.6	52.9	56.4	50.9
	DRO	52.8	51.5	54.0	49.7	51.9	50.8	53.2	49.1
FlanT5-Large	SFT	66.8	64.3	68.9	58.7	65.9	63.8	67.5	57.9
	KTO	58.4	56.0	59.3	53.2	57.6	55.4	58.1	52.6
	DRO	54.7	52.9	55.8	50.6	53.8	52.1	54.6	49.9
FlanT5-XL	SFT	70.9	68.2	73.0	63.5	69.6	67.4	71.8	62.9
	KTO	59.8	57.4	61.0	54.8	58.7	56.9	60.2	53.9
	DRO	56.1	53.5	57.2	51.7	55.0	52.8	56.4	50.8
Mistral-7B	SFT	77.6	75.1	79.4	71.2	76.9	74.6	78.3	70.5
	KTO	61.5	60.2	62.1	56.7	60.8	59.1	61.3	55.9
	DRO	57.4	55.9	58.6	52.4	56.7	55.2	57.8	51.8
LLaMA-3-8B	SFT	78.8	76.4	80.5	73.0	78.1	75.9	79.6	72.1
	KTO	62.4	61.0	63.1	57.8	61.7	60.3	62.5	56.9
	DRO	58.1	56.2	59.0	52.1	57.3	55.8	58.2	51.5
Qwen2.5-7B	SFT	79.5	77.1	81.2	73.8	78.9	76.8	80.4	72.9
	KTO	63.0	61.5	63.9	58.2	62.4	60.9	63.1	57.6
	DRO	58.7	56.8	59.5	52.6	57.9	56.1	58.7	51.9

Table 2: Out-of-distribution evaluation using LLM-as-a-judge with GPT-4o and Claude-3.5. Values represent pairwise win rates against RPO. Scores above 50 indicate preference for RPO over the corresponding baseline.

Table 2 shows that RPO retains a measurable advantage under distribution shift, but the magnitude is task- and baseline-dependent. The strongest gains occur against DRO, where RPO is preferred on most instruction-following and conversational benchmarks across model families. Against KTO, the advantage is smaller but remains positive across the evaluated tasks, while comparisons with SFT show larger margins, particularly for the larger models. The agreement between GPT-4o and Claude-3.5 follows the same overall pattern, supporting the robustness of the observed preference to the choice of evaluator.

The task breakdown is also informative. AlpacaEval and IFEval, which are closest to the instruction-following nature of the training data, show the most consistent preference for RPO. MT-Bench yields smaller margins, suggesting that multi-turn interaction introduces additional factors beyond the single-turn reward structure used during training. GSM8K is the most challenging transfer setting: improvements are generally smaller, although the preference for RPO becomes more consistent as model scale increases. These results indicate that RPO does not rely exclusively on memorizing the UltraFeedback distribution; its reward-partition objective transfers to several unseen tasks, with the extent of transfer depending on the capabilities required by each benchmark.

Overall, the OOD evaluation complements the in-domain results by showing that RPO's gains persist beyond the training distribution, while also revealing clear task-dependent variation rather than uniform improvements across all benchmarks.

4.6 Ablation Studies on RPO Design Components

We perform four targeted ablations to test the components that define RPO: temperature scaling, prompt-level empirical normalization, the density of candidate responses used for partition estimation, and robustness to reward noise. Full results and per-model breakdowns are provided in Appendix A.

Temperature scaling. We evaluate $\tau \in \{0.1, 1.0, 5.0\}$, with $\tau = 1.0$ as the default configuration. The results show a clear sensitivity to excessively small temperatures: $\tau = 0.1$ consistently reduces semantic alignment and diversity, increases toxicity, and lowers LLM-based preference scores. In contrast, $\tau = 5.0$ preserves most of the performance of the default setting, with only moderate degradation. This indicates that RPO is tolerant to smoother reward scaling but becomes less stable when reward differences are amplified too aggressively.

Empirical normalization. To isolate the contribution of the partition-based normalization, we remove $\widehat{V}(x)$ while keeping all other components unchanged. Across all architectures, this modification consistently degrades BERTScore, ROUGE-L, Distinct-2, toxicity, and LLM-based win rates. The result directly supports the role of prompt-level normalization in converting raw rewards into stable learning targets rather than treating $\widehat{V}(x)$ as an auxiliary modeling component.

Partition density. We progressively retain 25%, 50%, and 75% of the candidate completions available for each prompt, compared with the full-density configuration. Performance improves monotonically with partition density across architectures and metrics, while the 75% setting remains close to the full configuration. This trend supports the central premise of RPO: a denser set of same-prompt candidates provides a more informative empirical reward distribution and therefore a better normalization target.

Reward noise. Finally, we perturb training rewards with Gaussian noise of standard deviation $\sigma \in \{0.1, 0.5, 1.0\}$. RPO remains close to the clean-reward setting under mild noise and degrades progressively as the noise increases, without exhibiting collapse even at $\sigma = 1.0$. This suggests that the partition-based objective can tolerate moderate imperfections in scalar reward annotations.

Taken together, these ablations support the main design choices of RPO: reward scaling should avoid excessively sharp updates, prompt-level normalization is essential, partition quality benefits from multiple same-prompt candidates, and the resulting objective remains reasonably robust to noisy rewards. Detailed tables and additional analyses are deferred to Appendix A to preserve space in the main paper.

5 Limitations

RPO relies on multiple candidate responses per prompt to construct its empirical reward partition; consequently, limited within-prompt candidate diversity can reduce the quality of the normalization, as confirmed by our density ablation (Appendix A.3). It also remains dependent on the underlying reward signal: while moderate noise is tolerated, systematic or severe reward corruption can still affect optimization. Finally, our evaluation is centered on instruction-following and alignment tasks, leaving broader domains such as multilingual generation, specialized knowledge, and long-horizon reasoning for future study.

6 Conclusion

We introduced Reward Partitioning Optimization (RPO), a value-free objective that derives prompt-level reward normalization directly from the empirical reward distribution of same-prompt candidate responses. This replaces learned normalization with a data-dependent partition, without requiring an auxiliary value model or a separate reinforcement-learning stage.

Across encoder-decoder and decoder-only architectures, RPO achieves strong alignment performance across reference-based metrics, diversity, toxicity, and LLM-based preferences, while exhibiting controlled policy deviation. OOD experiments further show transfer beyond UltraFeedback, with task-dependent gains. Ablations confirm the importance of candidate density, temperature scaling, and reward quality.

Overall, RPO demonstrates that the observed reward distribution of grouped candidate responses can provide a simple and effective normalization mechanism for scalar-feedback preference optimization, motivating further study of more efficient partition estimators and larger-scale alignment settings.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 12248–12267. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.662. URL https://doi.org/10.18653/v1/2024.acl-long.662.

Anthropic. Claude 3.5 sonnet. https://www.anthropic.com/news/claude-3-5-sonnet, 2024.

AI Anthropic. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1:1, 2024.

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

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.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. *arXiv preprint 10.48550/ARXIV.2310.01377*, 2023.

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*, 2024.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pp. 8657–8677. PMLR, 2023.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=iUwHnoENnl.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint arXiv:2411.15594*, 2024.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and William B Dolan. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 110–119, 2016.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models, 2023.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*, 2024.

Adam Roberts, Hyung Won Chung, Gaurav Mishra, Anselm Levskaya, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. Scaling up models and data with t5x and seqio. *Journal of Machine Learning Research*, 24(377):1–8, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=gu3nacA9AH.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. CREAM: consistency regularized self-rewarding language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Vf6RDObyEF.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=Pnk7vMbznK.

An Yang, Baosong Yang, Binyuan Hui, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 10935–10950. Curran Associates, Inc., 2023a. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/23e6f78bdec844a9f7b6c957de2aae91-Paper-Conference.pdf.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=0NphYCmgua.

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

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*, 2019.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

A Additional Ablation Results

This appendix provides the complete results of the ablation studies summarized in Section 4.6. We analyze four aspects of RPO: temperature scaling, prompt-level empirical normalization, partition density, and robustness to reward noise. Unless otherwise stated, all experiments use the default RPO configuration with $\tau = 1.0$, the full set of candidate responses, and the original UltraFeedback rewards.

A.1 Temperature Sensitivity

RPO directly depends on the temperature parameter τ through the reward normalization term in Equation equation 16. In all main experiments reported in Table 1 , we use $\tau = 1.0$. To evaluate the sensitivity of RPO to this parameter, we additionally consider two extreme settings corresponding

Model	τ	BERTScore ↑	ROUGE-L ↑	Toxicity ↓	Distinct-2 ↑	GPT-4o WR ↑	Claude-3.5 WR ↑
FlanT5-Small	0.1	0.841	0.241	0.021	0.72	77.8	75.9
	5.0	0.860	0.278	0.014	0.80	83.2	81.0
FlanT5-Large	0.1	0.848	0.266	0.018	0.75	81.5	79.6
	5.0	0.866	0.304	0.011	0.84	86.1	84.2
FlanT5-XL	0.1	0.846	0.251	0.017	0.73	82.9	80.7
	5.0	0.865	0.289	0.012	0.86	90.1	88.0
Mistral-7B	0.1	0.863	0.319	0.015	0.82	84.6	82.5
	5.0	0.876	0.340	0.009	0.89	88.7	86.6
LLaMA-3-8B	0.1	0.861	0.314	0.016	0.81	85.0	83.1
	5.0	0.874	0.337	0.009	0.88	89.2	87.1
Qwen2.5-7B	0.1	0.862	0.317	0.015	0.82	85.3	83.5
	5.0	0.875	0.339	0.009	0.89	89.5	87.4

Table 3: Ablation study on the temperature parameter τ. Results are compared against the reference RPO configuration with $\tau = 1.0$ reported in Table 1. GPT-4o WR and Claude-3.5 WR scores correspond to pairwise win rates against the corresponding SFT baseline models.

to a sharp policy regime ($\tau = 0.1$) and a smoother regime ($\tau = 5.0$).

Table 3 reports the results across all architectures using the same automatic metrics and LLM-based evaluation protocol described previously. For LLM-as-a-judge evaluation, we report pairwise win rates against the corresponding SFT baseline models.
Overall, the results show that very small temperature values significantly degrade performance across nearly all metrics. In particular, $\tau = 0.1$ produces lower semantic similarity, reduced diversity, higher toxicity, and weaker preference scores from both GPT-4o and Claude-3.5. This behavior suggests that excessively sharp reward scaling leads to unstable optimization and overly aggressive policy updates.
In contrast, larger temperature values ($\tau = 5.0$) remain substantially more stable and preserve most of the gains obtained with the default configuration $\tau = 1.0$, although performance consistently remains slightly below the reference setting. These observations indicate that RPO is relatively robust to moderate smoothing, while excessively low temperatures negatively affect optimization stability and alignment quality.

A.2 Effect of Prompt-Level Empirical Normalization

RPO relies on empirical value normalization $\widehat{V}(x)$ (see Equation equation 16), which standardizes rewards at the prompt level before computing the partition-based objective. In the main results reported in Table 1, this normalization is applied in all RPO experiments with $\tau = 1.0$. To quantify its impact, we remove $\widehat{V}(x)$ and train RPO under identical settings, keeping all other components unchanged.

Table 4 reports results without value normalization. Across all architectures, removing $\widehat{V}(x)$ leads to a consistent degradation in both automatic metrics and LLM-based preference judgments. We observe lower semantic similarity (BERTScore, ROUGE-L), reduced diversity (Distinct-2), and increased toxicity, indicating less stable optimization and poorer alignment quality compared to the normalized RPO configuration in Table 1.
The effect is even more pronounced in LLM-as-a-judge evaluations. GPT-4o and Claude-3.5 Win Rates (WR), computed against the corresponding SFT baselines, drop substantially compared to the normalized setting. This confirms that value normalization plays a critical role in stabilizing reward scales across prompts and preventing dominance of high-reward variance samples during optimization.

Model	Setting	BERTScore ↑	ROUGE-L ↑	Toxicity ↓	Distinct-2 ↑	GPT-4o WR ↑	Claude-3.5 WR ↑
FlanT5-Small	w/o $\widehat{V}(x)$	0.842	0.232	0.022	0.71	76.9	75.1
	RPO (ref.)	0.866	0.289	0.012	0.83	81.4	79.2
FlanT5-Large	w/o $\widehat{V}(x)$	0.845	0.251	0.019	0.73	83.9	82.0
	RPO (ref.)	0.871	0.316	0.009	0.86	85.8	83.2
FlanT5-XL	w/o $\widehat{V}(x)$	0.846	0.258	0.018	0.74	85.1	82.9
	RPO (ref.)	0.870	0.297	0.008	0.88	86.2	84.1
Mistral-7B	w/o $\widehat{V}(x)$	0.862	0.318	0.014	0.82	86.7	84.5
	RPO (ref.)	0.881	0.348	0.007	0.91	88.6	86.2
LLaMA-3-8B	w/o $\widehat{V}(x)$	0.860	0.312	0.015	0.81	87.0	84.9
	RPO (ref.)	0.879	0.344	0.007	0.90	89.1	86.8
Qwen2.5-7B	w/o $\widehat{V}(x)$	0.861	0.314	0.014	0.81	87.3	85.1
	RPO (ref.)	0.880	0.346	0.007	0.91	89.4	87.1

Table 4: Ablation study on empirical value normalization $\widehat{V}(x)$. "w/o $\widehat{V}(x)$" denotes RPO trained without prompt-level reward normalization, while the reference corresponds to the full RPO configuration with $\tau = 1.0$ (Table 1). Removing value normalization leads to consistent degradation in both automatic metrics and LLM-as-a-judge Win Rates against SFT baselines, highlighting its importance for stable reward scaling and alignment quality.

A.3 Effect of Partition Density

RPO relies on estimating the partition function from multiple candidate completions associated with the same prompt. Consequently, the density of prompt-completion groups directly affects the quality of reward normalization and policy optimization. To evaluate this dependency, we progressively reduce the number of completions available per prompt in UltraFeedback while keeping the overall training configuration unchanged. We consider three reduced-density settings corresponding to 75%, 50%, and 25% of the original completions per prompt. The full-density setting (100%) corresponds to the standard RPO configuration reported in Table 1.

Model	Density	BERTScore ↑	ROUGE-L ↑	Toxicity ↓	Distinct-2 ↑	GPT-4o WR ↑	Claude-3.5 WR ↑
FlanT5-Small	25%	0.845	0.246	0.020	0.74	77.9	75.8
	50%	0.853	0.262	0.016	0.78	80.6	78.5
	75%	0.862	0.281	0.013	0.81	83.8	81.7
FlanT5-Large	25%	0.849	0.271	0.018	0.76	82.0	80.1
	50%	0.858	0.289	0.014	0.81	84.2	82.3
	75%	0.867	0.309	0.011	0.85	86.9	84.8
FlanT5-XL	25%	0.850	0.259	0.018	0.75	83.4	81.3
	50%	0.859	0.278	0.014	0.82	85.7	83.6
	75%	0.867	0.291	0.010	0.86	89.3	87.2
Mistral-7B	25%	0.866	0.327	0.013	0.84	85.1	83.0
	50%	0.872	0.336	0.010	0.87	87.0	84.9
	75%	0.879	0.344	0.008	0.90	89.0	86.9
LLaMA-3-8B	25%	0.864	0.322	0.013	0.83	85.4	83.3
	50%	0.871	0.333	0.010	0.86	87.4	85.2
	75%	0.877	0.341	0.008	0.89	89.5	87.3
Qwen2.5-7B	25%	0.865	0.324	0.013	0.83	85.8	83.6
	50%	0.872	0.335	0.010	0.87	87.7	85.5
	75%	0.878	0.343	0.008	0.90	89.8	87.6

Table 5: Ablation study on dataset density for partition estimation. Density indicates the percentage of candidate completions retained per prompt from UltraFeedback. The 100% density reference corresponds to the standard RPO configuration reported in Table 1. GPT-4o WR and Claude-3.5 WR correspond to pairwise win rates against the corresponding SFT baseline models. Lower-density settings consistently degrade performance, highlighting the importance of diverse prompt-level reward distributions for reliable partition estimation.

Table 5 shows that reducing the number of completions per prompt consistently degrades performance across all architectures and evaluation metrics. The degradation is particularly pronounced

at 25% density, where semantic alignment, lexical diversity, and LLM preference scores decrease substantially while toxicity increases. In contrast, the 75% setting remains relatively close to the full-density reference, suggesting that RPO remains robust under moderate reduction of prompt-level diversity. Overall, these results confirm that accurate partition estimation benefits from sufficiently dense reward distributions over candidate completions.

A.4 Robustness to Reward Noise

Real-world preference datasets often contain imperfect or inconsistent reward annotations. To evaluate the robustness of RPO under noisy supervision, we inject controlled Gaussian noise into the reward values during training. More precisely, each reward is perturbed as $r' = r + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$, where $\sigma \in \{0.1, 0.5, 1.0\}$. The reference clean-reward configuration corresponds to the standard RPO results reported in Table 1. The objective of this experiment is to assess whether RPO remains stable when reward annotations become progressively corrupted.

Model	σ	BERTScore ↑	ROUGE-L ↑	Toxicity ↓	Distinct-2 ↑	GPT-4o WR ↑	Claude-3.5 WR ↑
FlanT5-Small	0.1	0.862	0.282	0.013	0.81	83.9	81.7
	0.5	0.853	0.263	0.016	0.77	80.8	78.6
	1.0	0.844	0.245	0.020	0.73	77.2	75.0
FlanT5-Large	0.1	0.867	0.309	0.010	0.84	87.0	84.9
	0.5	0.859	0.291	0.013	0.80	84.1	82.0
	1.0	0.850	0.270	0.017	0.75	80.4	78.3
FlanT5-XL	0.1	0.866	0.292	0.009	0.86	89.5	87.4
	0.5	0.858	0.278	0.012	0.82	86.3	84.1
	1.0	0.849	0.259	0.017	0.76	82.1	80.0
Mistral-7B	0.1	0.878	0.343	0.008	0.90	89.2	87.0
	0.5	0.872	0.334	0.010	0.87	87.0	84.8
	1.0	0.865	0.320	0.014	0.83	83.4	81.2
LLaMA-3-8B	0.1	0.876	0.339	0.008	0.89	89.6	87.4
	0.5	0.870	0.331	0.010	0.86	87.3	85.0
	1.0	0.863	0.317	0.014	0.82	83.7	81.5
Qwen2.5-7B	0.1	0.877	0.341	0.008	0.90	89.9	87.7
	0.5	0.871	0.333	0.010	0.87	87.6	85.3
	1.0	0.864	0.319	0.014	0.83	84.0	81.8

Table 6: Robustness of RPO to noisy reward annotations. Gaussian noise with standard deviation σ is added to reward values during training. The clean reward reference corresponds to the standard RPO configuration reported in Table 1. GPT-4o WR and Claude-3.5 WR correspond to pairwise win rates against the corresponding SFT baseline models. While increasing reward noise progressively degrades performance, RPO remains relatively stable under moderate perturbations, indicating robustness to imperfect reward annotations.

Table 6 shows that RPO remains relatively robust under moderate reward perturbations. Small noise levels ($\sigma = 0.1$) produce only marginal degradation compared to the clean-reward reference, indicating that the optimization process tolerates mild annotation inconsistencies. As noise intensity increases, performance progressively decreases across all metrics, particularly for semantic alignment and LLM preference scores. Nevertheless, even under strong corruption ($\sigma = 1.0$), RPO maintains competitive behavior and does not collapse, suggesting that the reward normalization mechanism provides additional stability against noisy supervision.

A.5 Summary of Ablation Findings

Across the four ablations, the results support the main design choices of RPO. The temperature controls the sharpness of reward weighting, with excessively small values being detrimental. Removing the prompt-level empirical normalization consistently reduces performance, confirming its importance to the objective. Increasing the number of same-prompt candidates improves the quality of the empirical partition, while moderate reward noise causes only gradual performance degrada-

tion. Together, these findings provide empirical support for the central components of the value-free reward-partitioning formulation.