ROBUST PREFERENCE OPTIMIZATION UNDER SHIFTING PREFERENCE DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Current preference-based alignment techniques for large language models typically learn from fixed preference labels that order candidate responses for a given prompt. Different but valid evaluation priorities may reorder the same responses, reflecting the multidimensional and context-dependent nature of human judgment. Standard objectives treat the observed order as a fixed target and therefore do not account for this variation. Consequently, when deployment priorities differ from those represented during training, a policy fitted to the training labels may not remain aligned. Existing robust methods mainly interpret deviations from observed labels as unreliable feedback at sample-level. We instead study robustness to conditional ranking shifts and propose Kendall Robust Preference Optimization (KRPO). KRPO captures the empirical structure of these shifts through a utility-gap-weighted Kendall ambiguity set, making reversals between similarly scored responses less costly than those between clearly separated responses. An entropy-regularized adversary then emphasizes difficult but plausible alternative rankings. Because KRPO changes only the ranking supplied to the base loss, it applies to both pairwise and listwise objectives. Experiments with seven baseline objectives on Llama-3.1-8B and Qwen3-1.7B show that KRPO consistently improves cross-benchmark generalization and surpasses sample-level robust baselines on identical, unperturbed data. KRPO thus shows that robustness to shifting evaluation priorities is achieved by modeling where a ranking is likely to change and it offers a general mechanism for extending existing preference objectives to criteria absent from their supervision.

1 INTRODUCTION

Preference optimization aligns Large Language Models (LLMs) with human expectations by learning the relative desirability of candidate responses from preference feedback (Rafailov et al., 2023). However, the human judgment underlying response desirability is multidimensional: it combines potentially competing qualities—such as helpfulness, accuracy, safety, and contextual appropriateness—whose relative importance varies across tasks and deployment settings. Thus, preference datasets that record only binary judgments reflect a balance of priorities shaped by annotators' backgrounds and biases, as well as by the prompts and evaluation criteria. However, changes in users, tasks, or contexts can shift this balance even when every training annotation is reliable (Xu et al., 2026). A policy may therefore fit the observed labels well yet fail to remain aligned when evaluation priorities change. For listwise feedback, this failure takes a concrete form: different priorities can reorder the same candidate responses, changing the conditional ranking distribution for a fixed prompt and candidate set (Figure 1). Robustness to this shift is important because a single aligned policy is often expected to serve users and tasks whose priorities may not all be represented in one training corpus. This motivates our research question: *how can a listwise policy trained on observed rankings remain aligned under plausible shifts in this conditional ranking distribution?*

Standard preference optimization methods leave this question unresolved because they treat the supplied labels as fixed targets. Pairwise methods such as Direct Preference Optimization (DPO) learn from binary chosen–rejected feedback (Rafailov et al., 2023), whereas listwise methods retain the joint order among several responses and provide richer supervision (Song et al., 2024; Liu et al., 2025). This richer label does not by itself address shift: both families optimize the supplied preference relations without representing how the joint order may change under another valid criterion.

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.

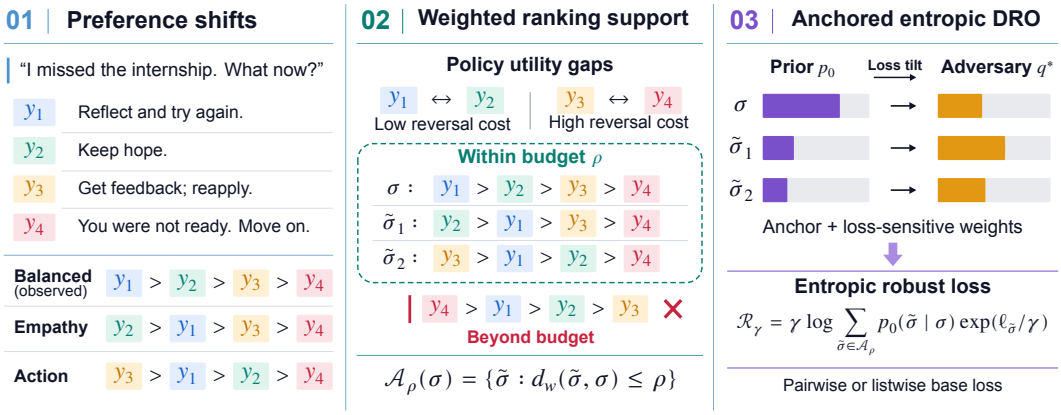


Figure 1: Overview of KRPO. Criterion changes can reorder comparable responses while preserving clear quality relations. KRPO constructs a utility-weighted Kendall neighborhood, anchors a local prior at the observed ranking, and tilts it toward plausible high-loss orders before updating the policy.

Robust preference methods instead relax label reliability across training records, downweighting or adversarially reweighting examples deemed unreliable (Wu et al., 2025; Xu et al., 2026). Such sample-level protection is appropriate for corrupted feedback, but criterion-dependent shift can reorder responses within a record whose original annotation remains valid. Concurrent listwise distributionally robust optimization (DRO) places uncertainty directly on ranking labels, yet motivates alternative orders primarily through inconsistency, lossy feedback, or noise (Wu et al., 2026). Existing approaches therefore either leave within-record ranking variation unmodeled or treat it as evidence of unreliable feedback, rather than as structured variation among valid preferences. Section 2 provides a detailed discussion of related work.

We address this gap by treating an observed order as valid for its generating criterion but incomplete under other deployment priorities. This calls for structured ranking uncertainty: criterion-sensitive relations may change, while relations supported across criteria remain stable. Our multi-attribute analysis shows that disagreement concentrates among similarly valued responses (Figure 2). Kendall Robust Preference Optimization (KRPO) encodes this signal through pair-dependent costs in a weighted Kendall ambiguity set and an entropy-regularized adversary over admitted rankings. The geometry protects clear relations, while the adversary emphasizes difficult but plausible alternatives without relying on one worst permutation. Because only the ranking supplied to the loss changes, KRPO remains independent of the base objective. Our contributions are:

- We formalize shifts in evaluation priorities as conditional ranking uncertainty, distinguishing valid preference variation from label corruption and identifying a failure mode that sample-level robustness does not capture.
- We introduce KRPO, whose utility-weighted Kendall geometry concentrates robustness on plausible small-gap reversals while protecting clear preference relations. Coverage and deployment-risk bounds state when its objective controls deployment risk.
- Across seven objectives and two model families, KRPO improves macro-average cross-benchmark win rates in all 14 combinations without altering their base losses, supporting its use as a broadly compatible extension to pairwise and listwise preference optimization.

2 Related Work

Preference objectives for ranked feedback. Preference objectives differ in both the granularity of supervision and the ranking model they optimize. DPO fits a Bradley–Terry likelihood to a binary chosen–rejected label, which means that using it with a K-way ranking requires pairwise decomposition (Rafailov et al., 2023). To retain the joint order instead, PRO uses sequential contrasts, LiPO introduces rank- and score-sensitive pair weighting, RCPO derives objectives from utility- and rank-based choice models, and PPA optimizes a differentiable NDCG surrogate (Song et al., 2024; Liu et al., 2025; Tang & Feng, 2026; Zhao et al., 2025). Related listwise formulations also enhance preference alignment through list-level reward shaping, recommendation-oriented listwise optimization, or subjective ranking constraints (Zhu et al., 2024; Li et al., 2025; Naini et al., 2026).

Although these methods extract different signals from ranked feedback, they share the assumption that the supplied order is the nominal target for its candidate set. KRPO complements this line of work by placing uncertainty over that target while leaving the underlying preference loss intact.

Robustness over preference data and ranking labels. The shared fixed-target assumption makes robustness to preference labels directly relevant. Most robust alignment methods address this issue by placing uncertainty over the distribution or reliability of training records. For example, Dr. DPO reduces the influence of likely mismatched chosen–rejected pairs, while WDPO and KLDPO construct Wasserstein and KL ambiguity sets around pairwise DPO records (Wu et al., 2025; Xu et al., 2026). Related approaches modify which samples dominate training to account for application-specific pairwise uncertainty, group-level worst-case performance, prompt-distribution shift, sample representativeness, or reward-model adaptation to policy-induced shifts (Kim et al., 2026; Ramesh et al., 2024; Mandal et al., 2025; Zhu et al., 2026; Dou et al., 2025). Because their uncertainty acts across records, prompts, or groups, these methods do not describe how a change in preference criteria can reorder responses within one record. Concurrent work by Wu et al. (2026) addresses a more closely aligned setting by applying DRO to conditional listwise labels. Its total-variation geometry, however, makes every alternative permutation equally distant from the observed label, and its adversary reduces to a mixture of the nominal Plackett–Luce loss and one globally worst ranking. KRPO instead uses pair-dependent reversal costs to define a selective ranking geometry and an entropic adversary over multiple alternatives.

Preference variation in alignment supervision. Repeated evaluations of the same candidates motivate modeling uncertainty within a record. Human disagreement often reflects task underspecification or response style rather than annotation noise (Zhang et al., 2024). Scalar aggregation can obscure quality trade-offs, underweight factuality and faithfulness, and favor assertive responses (Hosking et al., 2024). Pluralistic reward modeling likewise cautions against collapsing heterogeneous user or group priorities into one preference (Chen et al., 2024). Synthetic supervision is also sensitive to the judge, rubric, and aggregation rule: UltraFeedback averages GPT-4 scores across four dimensions while reporting GPT-4–human pairwise agreement comparable to inter-annotator agreement (Cui et al., 2023). These findings show that the preference-generating process can systematically change a ranking, with such changes propagating through Bradley–Terry and Plackett–Luce models (Xu & Kankanhalli, 2025). Prior evidence does not, however, identify which pairwise relations are most likely to change. Our analysis provides this structure (Figure 2) and uses it to define a nonuniform ranking neighborhood.

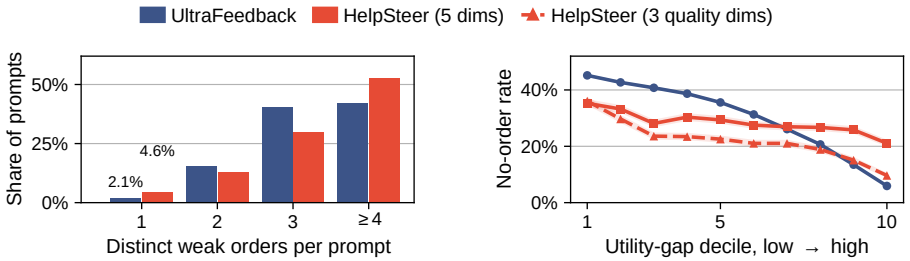
3 Conditional Preference Shifts

Problem formulation. We hold the prompt x and candidate set $\mathcal{Y} = \{y_1, \dots, y_K\}$ fixed while allowing the evaluation criterion to change. Let $\mathfrak{S}_K$ contain all rankings of these responses from most to least preferred. A training record provides one observed ranking $\sigma \in \mathfrak{S}_K$. For policy π_θ and base preference loss ℓ, write $\ell_{\tilde{\sigma}} := \ell(\theta; x, \mathcal{Y}, \tilde{\sigma})$ for any ranking $\tilde{\sigma} \in \mathfrak{S}_K$. Standard training uses the nominal loss ℓ_σ. Let $P_{\text{train}}(\cdot \mid x, \mathcal{Y})$ and $P_{\text{deploy}}(\cdot \mid x, \mathcal{Y})$ denote the ranking distributions under annotation and deployment preferences, with σ drawn from the former. A conditional ranking shift occurs when these distributions differ. We define expected conditional deployment risk as:

$$\mathcal{R}_{\text{deploy}}(\theta; x, \mathcal{Y}) = \mathbb{E}_{\tilde{\sigma} \sim P_{\text{deploy}}(\cdot \mid x, \mathcal{Y})} \left[\ell_{\tilde{\sigma}} \right] \quad . \tag{1}$$

Because P_{deploy} is unavailable and cannot be identified from one observed ranking, we seek a tractable objective over plausible alternatives that preserves stable preference relations.

Empirical structure. We compare rankings of the same responses across UltraFeedback and HelpSteer annotation dimensions (Cui et al., 2023; Wang et al., 2024). These dimensions often induce distinct weak orders, showing inconsistency in human preference (Figure 2a). Using an external reward model's utility gap as a proxy for pairwise preference strength, we find that disagreement decreases as the gap increases (Figure 2b). Relations between similarly valued responses thus appear more sensitive to the evaluation criterion, whereas larger utility gaps are associated with more consistent pairwise orderings across dimensions. This pattern motivates placing more variability on the small-gap reversals in our approach (Section 4). These diagnostic scores are separate from the utilities used by KRPO during training. KRPO computes reversal costs from policy-derived utility gaps, avoiding an additional reward model during optimization. Appendix C details the protocol.



(a) Multiple orders are the rule (b) Disagreement vs. gap

Figure 2: Criterion-induced ranking variation in UltraFeedback and HelpSteer. **(a)** Distribution of distinct weak orders across annotation dimensions (ties retained; tail pooled into ≥ 4). **(b)** No-order rate (dimension conflict or ties across all dimensions) by external utility-gap decile, with bands showing 95% Wilson intervals.

4 Kendall Robust Preference Optimization

We now construct KRPO to account for the ranking uncertainty defined in Section 3. Concurrent work by Wu et al. (2026) also applies DRO to conditional ranking labels, but its total-variation geometry treats all alternative permutations equally relative to the observed order. KRPO uses a utility-weighted Kendall neighborhood to distinguish plausible reversals and an entropy-regularized adversary to distribute weight across admissible rankings.

4.1 Utility-Weighted Kendall Ambiguity Set

Classical Kendall distance compares rankings by counting pairwise reversals (Kendall et al., 1938). Let $\delta_{ab}(\tilde{\sigma}, \sigma) = 1$ when $\tilde{\sigma}$ reverses responses a and b relative to σ, and 0 otherwise. Then $d_K(\tilde{\sigma}, \sigma) = \sum_{a<b} \delta_{ab}(\tilde{\sigma}, \sigma)$. However, its unit costs cannot capture the differing sensitivity of pairwise relations to preference shifts. KRPO therefore weights reversals by preference strength.

Following the pattern in Section 3, we assign lower costs to small-gap reversals and higher costs to reversals across clear utility gaps. Computing these costs during training requires a utility estimate for each response. We use scores already available from the policy's base objective, avoiding a separate reward model during training. For reference-based objectives, these scores are DPO's implicit rewards (Rafailov et al., 2023),

$$u_{\theta,j} = \beta \left[\log \pi_\theta(y_j \mid x) - \log \pi_{\text{ref}}(y_j \mid x)\right] \quad , \tag{2}$$

where π_{ref} is fixed and β sets the scale. At the optimum of the KL-regularized reward objective, utility differences recover reward differences (Appendix E.2), motivating their use as a proxy for preference strength. Reference-free objectives retain their native response scores. To use a shared radius across utility scales, we normalize pairwise gaps and bound their sum over the reversals in each permutation:

$$w_{ab} = \text{Norm}(|u_{\theta,a} - u_{\theta,b}|) \quad , \qquad a \neq b \ , \tag{3}$$

$$d_w(\tilde{\sigma}, \sigma) = \sum_{a<b} w_{ab}\, \delta_{ab}(\tilde{\sigma}, \sigma) \quad , \tag{4}$$

$$\mathcal{A}_\rho(\sigma) = \{\tilde{\sigma} \in \mathfrak{S}_K : d_w(\tilde{\sigma}, \sigma) \leq \rho\} \quad , \qquad \rho \geq 0 \ . \tag{5}$$

Here $\text{Norm}(\cdot)$ scales pairwise gaps to unit mean within $\mathcal{Y}$, and $w_{aa} = 0$. The resulting weighted Kendall neighborhood can admit several small-gap reversals while preserving every pair whose cost exceeds ρ. Thus, training hedges weaker comparisons while retaining relations shared by all admitted rankings. Since changing utilities can move rankings across the radius threshold, we compute u_θ in Equation 3 with a stop-gradient exponential moving average (EMA) policy $\bar{\theta}_t$ (Tarvainen & Valpola, 2017; Morales-Brotons et al., 2024). This smooths the geometry across steps and holds it fixed within each loss update (Appendix F.3).

Because policy utilities are proxies for preference strength, coverage depends on their approximation error. The following theorem gives sufficient conditions for a weighted Kendall neighborhood to

cover local preference shifts. Its sufficient radius depends on the shift size, reward smoothness, and utility-proxy accuracy. A single observed ranking does not identify these quantities. Appendix E provides the formal assumptions, radius, and proof.

Theorem 1 (Latent-to-ranking coverage, informal)**.** *Suppose that at least $1-\delta$ of deployment conditions remain local to a nominal condition that induces σ, rewards vary smoothly with the evaluation condition, and policy utility gaps approximate nominal reward gaps. Then larger reward gaps require larger shifts to reverse a pair, and a utility-weighted Kendall neighborhood with a sufficient proxy-error correction contains at least $1-\delta$ of deployment rankings.*

4.2 Local Ranking Prior and Entropic DRO

The ambiguity set determines which rankings the adversary may consider, but not how their losses form a per-sample objective. A hard maximum, $\max_{\tilde{\sigma}\in\mathcal{A}_\rho(\sigma)} \ell_{\tilde{\sigma}}$, would let every record independently select its highest-loss admissible ranking. This can be overly conservative when a deployment shift changes the relative likelihood of plausible rankings without selecting the most damaging order for every prompt. We therefore regularize each local adversary toward a distance-aware prior, which assigns anchor mass $m_0 \in (0,1)$ to the observed ranking and decays over alternatives with distance:

$$p_0(\tilde{\sigma} \mid \sigma) = \begin{cases} m_0 \ , & \tilde{\sigma} = \sigma \ , \\ (1-m_0)\dfrac{\exp[-\eta d_w(\tilde{\sigma},\sigma)]}{Z_\sigma} \ , & \tilde{\sigma} \in \mathcal{A}_\rho(\sigma)\setminus\{\sigma\} \ , \end{cases} \tag{6}$$

Here Z_σ normalizes the alternatives and $\eta \geq 0$ controls the decay. If no alternative is admitted, $p_0(\sigma \mid \sigma) = 1$. The radius and the prior thus play distinct roles: the radius determines which rankings are considered, whereas the prior anchors the neighborhood at the observed label and grades the admitted alternatives, serving as a reference measure for robustness rather than an estimate of P_{deploy}. With p_0 as this reference measure, we robustify the base loss with the entropic, or KL-regularized, DRO formulation (Li et al., 2023a):

$$\mathcal{R}_\gamma(\theta;\sigma) = \max_{q\in\Delta(\mathcal{A}_\rho(\sigma))} \{\mathbb{E}_{\tilde{\sigma}\sim q}[\ell_{\tilde{\sigma}}] - \gamma\,\mathrm{KL}(q\,\|\,p_0)\} \ , \tag{7}$$

where $\gamma > 0$ is the temperature and $\Delta(\mathcal{A}_\rho(\sigma))$ is the probability simplex over admissible rankings. The expected-loss term moves mass toward difficult rankings, whereas the KL penalty limits how far the adversarial distribution can depart from p_0.

For fixed θ and σ, the inner maximization has the analytic solution

$$\begin{aligned} \mathcal{R}_\gamma(\theta;\sigma) &= \gamma \log \mathbb{E}_{\tilde{\sigma}\sim p_0(\cdot\mid\sigma)}[\exp(\ell_{\tilde{\sigma}}/\gamma)] \ , \\ q^\star_\theta(\tilde{\sigma} \mid \sigma) &= p_0(\tilde{\sigma} \mid \sigma)\exp\big[\big(\ell_{\tilde{\sigma}} - \mathcal{R}_\gamma(\theta;\sigma)\big)/\gamma\big] \ . \end{aligned} \tag{8}$$

The solution makes the interaction between plausibility and difficulty explicit: a ranking receives substantial mass only when it is supported by p_0 and incurs high loss under the current policy. The temperature γ controls this interaction: as $\gamma \to 0$, the risk approaches the largest admissible loss, whereas as γ increases, $q^\star_\theta(\cdot \mid \sigma)$ remains closer to the prior and spreads the update across more rankings. If the ambiguity set contains only σ, the objective reduces to the nominal loss. Averaging over N training records gives the KRPO objective $\mathcal{L}_{\text{KRPO}}(\theta) = N^{-1}\sum_{i=1}^N \mathcal{R}_\gamma(\theta;\sigma_i)$.

Because this objective is defined on a local ambiguity set, its connection to the deployment risk in Equation 1 depends on how much deployment mass that set covers. Let $\ell_{\tilde{\sigma}} \in [0, L_{\max}]$ and let $\delta = P_{\text{deploy}}(\mathcal{A}_\rho(\sigma)^c \mid x, \mathcal{Y})$ denote the deployment mass outside the ambiguity set. For $\delta < 1$, let $Q_\mathcal{A}$ be P_{deploy} conditioned on the set, while $\delta = 1$ gives the trivial bound $\mathcal{R}_{\text{deploy}} \leq L_{\max}$.

Theorem 2 (Conditional deployment-risk bound)**.** *For every $\gamma > 0$ when $\delta < 1$,*

$$\mathcal{R}_{\text{deploy}}(\theta;x,\mathcal{Y}) \ \leq \ (1-\delta)[\mathcal{R}_\gamma(\theta;\sigma) + \gamma\,\mathrm{KL}(Q_\mathcal{A}\,\|\,p_0)] \ + \ \delta L_{\max} \ , \tag{9}$$

Together, Theorems 1 and 2 characterize when KRPO can control deployment risk: coverage must be high and the in-set mismatch moderate. Although $\mathcal{R}_\gamma$ is computable, δ and $Q_\mathcal{A}$ depend on the unobserved deployment distribution. This dependence makes the result a conditional theoretical guarantee rather than a certificate verifiable at deployment time. Appendix E bounds the prior mismatch and gives the combined result. Finally, because uncertainty enters only through the ranking argument of ℓ, the construction is independent of the form of the base objective and applies to pairwise and listwise losses alike. Practical optimization and the handling of graded objectives such as LiPO-λ (Liu et al., 2025) are described in Appendix F.

5 EXPERIMENTS

Our generation-based experiments evaluate two claims: whether KRPO generalizes across benchmarks and base objectives, and whether ranking-level uncertainty offers an advantage over record-level robustness on uncorrupted rankings. These tests assess modularity and distinguish ranking-shift robustness from robustness to label noise, respectively. A complementary controlled evaluation isolates preference changes within fixed candidate sets (Section 5.4).

5.1 EXPERIMENTAL SETUP

Training data and models. We train Qwen3-1.7B and Llama-3.1-8B on the original four-response rankings of UltraFeedback (Cui et al., 2023) without synthetic corruption. The two model families test the method across different architectures and scales. Training enumerates the support exactly at $K = 4$. Appendix G shows that top-8 and sampled-8 supports closely match the exact objective and gradients. Appendix F.4 gives the full configuration and search ranges.

Objectives and baselines. To test modularity, we apply KRPO to seven objectives with different ranking assumptions: DPO-allpairs (Rafailov et al., 2023), LPO4Rec (Li et al., 2025), LiPO (Liu et al., 2025), Plackett–Luce (PL) (Plackett, 1975; Rafailov et al., 2023), IPO (Azar et al., 2024), SimPO (Meng et al., 2024), and SLiC-HF (Zhao et al., 2023), spanning pairwise decompositions and losses designed for ranked feedback. Within each base/+KRPO pair, only the distribution over ranking labels changes, with data, initialization, and training protocol held fixed. For the second question, we compare with KLDPO (Xu et al., 2026) and DRDPO (Wu et al., 2025), which robustify the influence of preference records rather than the order within a record.

Evaluation. We report win rates and their five-benchmark macro average on Arena-Hard (Arena) (Li et al., 2024a;b), AlpacaEval (Alpaca) (Li et al., 2023b), HH-RLHF (Bai et al., 2022), MT-Bench (MT-B) (Zheng et al., 2023), and Reddit TL;DR (TL;DR) (Stiennon et al., 2020). AlpacaEval stays close to UltraFeedback's single-turn instruction format, whereas the other four shift toward difficult requests, helpfulness–harmlessness trade-offs, multi-turn constraints, and faithful summarization. GPT-4.1-mini judges each policy response against a fixed reference under blinded evaluation in both presentation orders to reduce positional bias. Wins, ties, and losses receive scores of 1, 0.5, and 0, respectively. The macro average weights all five benchmarks equally, preventing larger datasets from dominating the aggregate score. We estimate uncertainty using paired prompt bootstrap intervals. The same prompts are resampled for each baseline and its KRPO counterpart, preserving their pairing when estimating uncertainty in the performance differences. Appendix B details the benchmarks, their shifts, and the reference responses.

5.2 GENERALIZATION ACROSS OBJECTIVES AND MODELS

Table 1 tests modularity by varying the objective and model family while keeping the training rankings unperturbed. In this paired design, base rows report absolute win rates and shaded rows the win-rate changes produced by KRPO.

Generalization under preference distribution shift. KRPO raises the macro average in all 14 objective–model combinations, yielding an aggregate gain of $+1.36$ points on average. Dataset-level gains track the shift profile: MT-Bench and TL;DR show the largest changes ($+2.6$/$+2.4$ and $+1.4$/$+2.0$ points for Qwen/Llama), while the near-domain AlpacaEval gains are modest ($+0.4$/$+0.2$). Although descriptive, this structure is mechanistically informative. When evaluation remains close to UltraFeedback, the observed ranking already emphasizes largely compatible comparisons, leaving limited benefit from hedging alternatives. As task requirements change, criterion-sensitive relations become less transferable. Optimizing over plausible local permutations discourages the policy from overcommitting to these fragile comparisons. The concentration of gains on MT-Bench and TL;DR is therefore more consistent with ranking-shift robustness than with a uniform regularization effect.

Mechanistic complementarity with LiPO. The strongest gains occur with LiPO ($+2.3$ on Qwen and $+2.5$ on Llama), consistent with complementary pairwise weighting. KRPO's utility-gap geometry estimates which reversals are plausible, whereas LiPO's factor $|G_i - G_j||D_i - D_j|$ measures the ranking consequence of reversing a pair. Their composition concentrates optimization pressure on relations that are both uncertain and consequential, rather than on harmless tail swaps or stable

Table 1: Cross-benchmark win rates (%, ↑) after training on unperturbed UltraFeedback. Base rows show absolute scores, while shaded rows show KRPO changes. Mean is the five-benchmark macro average. Means and changes are rounded independently. **Bold** marks mean changes. The final row averages changes across objectives.

	Qwen3-1.7B						Llama-3.1-8B					
Loss	Arena	Alpaca	HH	MT-B	TL;DR	**Mean**	Arena	Alpaca	HH	MT-B	TL;DR	**Mean**
DPO-allpairs	18.9	22.5	78.0	59.4	41.3	44.0	34.2	37.2	89.9	75.6	21.4	51.7
KRPO (ours) Δ	+0.0	+1.4	+0.6	+1.2	+1.3	**+0.9**	-0.2	-1.1	+1.0	+2.2	+3.7	**+1.1**
LPO4Rec	18.7	23.0	75.7	55.6	43.2	43.2	33.5	33.9	89.1	80.6	23.3	52.1
KRPO (ours) Δ	+0.5	+0.5	+0.3	+6.3	-0.2	**+1.5**	+0.6	-0.2	+0.0	-1.2	+2.4	**+0.3**
LiPO	18.5	24.0	75.4	56.9	42.8	43.5	33.7	31.9	89.5	80.9	21.8	51.6
KRPO (ours) Δ	+1.1	-0.8	+2.2	+7.5	+1.4	**+2.3**	+1.1	+3.8	+2.1	+1.6	+4.1	**+2.5**
PL	16.7	23.4	76.0	56.9	40.6	42.7	34.5	34.3	89.4	79.1	26.3	52.7
KRPO (ours) Δ	+2.0	-1.0	-0.8	+1.2	+1.3	**+0.5**	-0.3	+1.6	-0.5	+4.0	-2.7	**+0.4**
IPO	15.8	20.6	73.8	53.8	40.0	40.8	33.3	32.6	80.8	77.5	15.3	47.9
KRPO (ours) Δ	+0.3	+0.1	+0.0	+2.5	+2.2	**+1.0**	+0.9	+1.0	-0.1	+0.0	+0.8	**+0.5**
SimPO	22.4	29.2	73.4	52.5	38.5	43.2	34.1	22.1	80.8	56.9	22.5	43.3
KRPO (ours) Δ	+0.0	+0.9	+0.7	+0.0	+3.0	**+0.9**	+1.8	-2.6	+3.2	+7.2	+1.8	**+2.3**
SLiC-HF	18.6	22.5	74.1	60.6	41.8	43.5	33.7	35.2	86.4	80.6	21.7	51.5
KRPO (ours) Δ	-0.2	+1.4	+0.8	-0.6	+0.5	**+0.4**	+0.5	-0.9	+1.9	+2.8	+4.1	**+1.7**
Avg. KRPO (ours) Δ	+0.5	+0.4	+0.5	+2.6	+1.4	**+1.1**	+0.6	+0.2	+1.1	+2.4	+2.0	**+1.3**

Table 2: Comparison with sample-level robust objectives on unperturbed UltraFeedback rankings. Base rows report absolute win rates (%, ↑), while other rows report changes from the corresponding base objective. Shading marks KRPO (ours), and **bold** marks the largest five-benchmark macro change in each objective–model block.

	Qwen3-1.7B						Llama-3.1-8B					
Loss	Arena	Alpaca	HH	MT-B	TL;DR	**Mean**	Arena	Alpaca	HH	MT-B	TL;DR	**Mean**
DPO-allpairs	18.9	22.5	78.0	59.4	41.3	44.0	34.2	37.2	89.9	75.6	21.4	51.7
KLDPO Δ	+0.4	+0.7	-2.6	+0.6	-1.6	-0.5	+0.2	-2.3	+0.8	+1.3	+4.0	+0.8
DRDPO Δ	-0.7	-0.4	-2.8	+0.0	-2.0	-1.2	-0.3	-2.2	+0.9	+1.3	+2.7	+0.5
KRPO (ours) Δ	+0.0	+1.4	+0.6	+1.2	+1.3	**+0.9**	-0.2	-1.1	+1.0	+2.2	+3.7	**+1.1**
LiPO	18.5	24.0	75.4	56.9	42.8	43.5	33.7	31.9	89.5	80.9	21.8	51.6
KLDPO Δ	+1.2	-1.0	+1.0	-5.6	-2.7	-1.4	-1.2	+1.0	+2.3	-1.2	+3.1	+0.8
DRDPO Δ	+0.5	-2.5	+0.1	-3.5	-2.6	-1.6	-1.2	+1.6	+3.7	+1.0	+5.9	+2.2
KRPO (ours) Δ	+1.1	-0.8	+2.2	+7.5	+1.4	**+2.3**	+1.1	+3.8	+2.1	+1.6	+4.1	**+2.5**

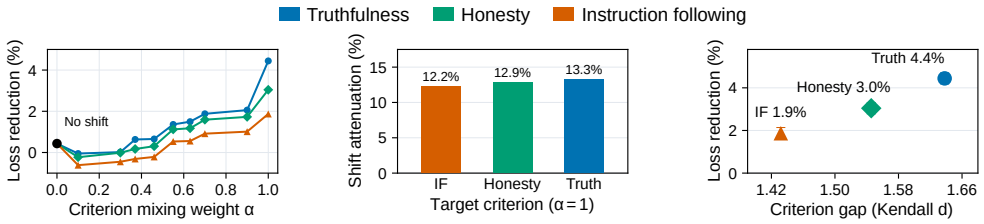
high-margin relations. Because the LiPO loss also enters $q^\star$, poorly handled alternatives receive additional mass, yielding a focused ranking-level robust-regret objective.

Statistical evidence for aggregate improvement. Our statistical analysis follows the aggregate generalization claim across objectives, models, and benchmarks. A shared-prompt bootstrap applies the same prompt resample to all 14 objective–model pairs, preserving correlations induced by shared prompts, and gives a grand-mean improvement of $+1.36$ points (bootstrap SE 0.25; 95% percentile CI $[+0.88, +1.85]$). Because the interval excludes zero, we reject the null of no aggregate improvement at the 5% level. Individual benchmark deltas target narrower and more heterogeneous effects and provide limited evidence after multiplicity correction. We interpret these deltas descriptively rather than as benchmark-specific claims.

5.3 Comparison with Sample-Level Robustness

Table 2 compares KRPO with KLDPO and DRDPO under DPO-allpairs and LiPO, holding the unperturbed UltraFeedback data, initialization, base objective, and evaluation protocol fixed within each block. This tests whether the gains reflect robustness generally or ranking-level uncertainty. KRPO leads the macro average in all four blocks. KLDPO and DRDPO reduce both Qwen averages, with Llama gains concentrated in fewer benchmarks. KRPO transfers most consistently.

The Qwen–Llama contrast suggests that model capacity changes what a high-loss record represents. KLDPO and DRDPO lower the Qwen macro averages but improve on Llama. Under LiPO,



(a) Along each shift path (b) Less loss from the shift (c) Full shift (α = 1)

Figure 3: Preference shifts from helpfulness to three attributes (Qwen3-1.7B). **(a)** Deployment-loss reduction versus DPO as the mixing weight α increases. **(b)** Reduction in the shift-induced loss increase relative to DPO. **(c)** Full-shift endpoints ($\alpha = 1$) versus criterion disagreement. Higher values indicate greater improvement. Whiskers in (c) show ± 1 paired-bootstrap SE of the loss difference, scaled by DPO's mean loss (Appendix D).

DRDPO nearly matches KRPO, although its gain is concentrated in HH and TL;DR, whereas KRPO improves all five benchmarks. On the smaller Qwen model, valid but difficult prompts can remain high-loss because they exceed its current capacity. Absolute loss then conflates capacity-induced difficulty with the distributional importance targeted by KLDPO or the potential unreliability targeted by DRDPO, causing the former to overemphasize poorly learnable records and the latter to suppress useful supervision. Llama can fit more ordinary hard examples, reducing the contribution of underfitting to its high-loss tail and making loss a more discriminative record-level signal. KRPO is less exposed to this capacity dependence because it compares alternative rankings of the same responses: difficulty shared by the record does not determine which ranking receives weight. This helps explain its leading macro gain across all four blocks. Since model family and size vary together, the pattern supports but does not causally establish this interpretation. Appendix I further compares KRPO with the concurrent Robust PL method (Wu et al., 2026).

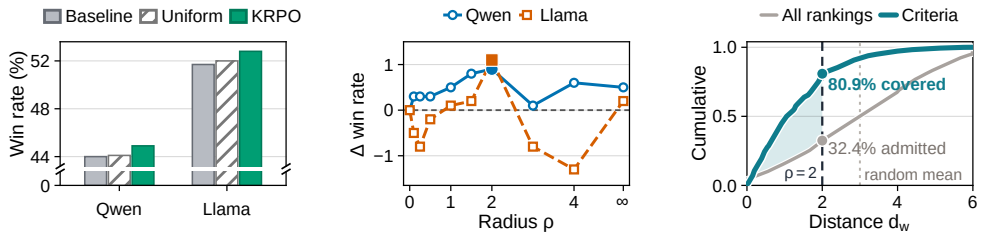
5.4 Controlled Preference Shift

Our generation benchmarks cannot isolate the fixed-candidate shift formalized in Section 3 because prompts, responses, and evaluation priorities vary together. We therefore add a controlled complement in which 2,802 held-out prompts and their four candidate responses stay fixed and only the ranking criterion changes. Training rankings follow UltraFeedback helpfulness ratings, while deployment rankings use the mixed score $(1-\alpha)\, r_{\text{help}} + \alpha\, r_t$ for a target attribute $t \in \{$instruction following, honesty, truthfulness$\}$ and ten values of $\alpha \in [0, 1]$. Increasing α moves the criterion from the training attribute to the target. Because each criterion induces one ranking per prompt, the base DPO-allpairs loss under that ranking, averaged over prompts, is exactly the conditional deployment risk of Equation 1. We evaluate helpfulness-trained DPO-allpairs and KRPO policies with this common loss and measure the size of each shift by the mean Kendall distance between training and deployment rankings (Appendix D).

Figure 3a shows an initial dip followed by increasing gains along all three attribute paths as the target weight grows. At small α, the target criterion may act as a noisy perturbation of tied or closely rated responses, leading to the initial drop. At every tested $\alpha \geq 0.55$, all three paths incur lower risk under KRPO than under DPO. At full shift the reductions reach 1.9%, 3.0%, and 4.4% for instruction following, honesty, and truthfulness, increasing with how strongly each criterion disagrees with helpfulness about the order of the same responses (Figure 3c). Measured against each policy's own no-shift loss, KRPO reduces the additional loss that a full criterion change imposes on DPO by 12.2%–13.3% (Figure 3b). A single KRPO policy therefore reduces conditional deployment risk across these target criteria at the larger tested mixing weights, without retraining for each target.

5.5 Ablation Studies

Figure 4 tests two central choices in KRPO: whether the gain requires utility-weighted ranking geometry, and how it depends on the relative budget that bounds the resulting neighborhood. Panels (a)–(b) use DPO-allpairs on Qwen3-1.7B and Llama-3.1-8B, while panel (c) provides a separate diagnostic of the weighted geometry.



(a) Geometry (b) Radius (c) Coverage

Figure 4: Geometry and radius ablations. **(a)** Macro-average win rate for DPO-allpairs, its uniform-cost robust variant, and KRPO. The axis break omits the shared middle range. **(b)** Sensitivity to the relative radius. Filled markers denote the fixed DPO-allpairs setting $\rho = 2$ used in the main experiments. **(c)** Cumulative weighted-distance distributions for all rankings and criterion-induced reorderings. Performance ablations use one run per setting.

Selective ranking geometry. Figure 4a compares weighted and uniform reversal costs within the same robust objective. At the same normalized budget, uniform costs improve the macro average by only 0.1 (Qwen) and 0.3 (Llama) points, versus 0.9 and 1.1 for utility-weighted costs. These small gains suggest a tradeoff: hedging weakly separated pairs can help, but unit costs also permit reversals across clear utility gaps, potentially weakening reliable supervision. Utility weighting instead directs the budget toward ambiguous relations. Panel (c) corroborates this selectivity independently of win rates: on 996 held-out UltraFeedback prompts with a Qwen3-1.7B LiPO diagnostic checkpoint, the radius-2 neighborhood admits only 32.4% of the 24 possible rankings yet covers 80.9% of the nontrivial criterion-induced orders. Thus, the weighted neighborhood captures most criterion-sensitive reorderings while excluding most permutations.

Interpretable robustness budget. The cost normalization gives ρ a relative meaning: since the mean pairwise reversal cost within each record is one, $\rho = 2$ admits rankings whose reversed relations cost at most two average reversals. The budget limits how much preference evidence a permutation may contradict, rather than fixing its inversion count. Several near-tie swaps may fit within the budget, while one large-gap reversal may exceed it. Guided by this scale, we use $\rho = 2$ or $\rho = 1.5$ for nearly all objectives without benchmark- or model-specific tuning. Both backbones peak at $\rho = 2$, supporting a shared operating point. This pattern is consistent with the separate coverage diagnostic in panel (c): smaller radii exclude substantial criterion-induced variation, whereas larger radii mainly add generic permutations once criterion-induced coverage saturates. Beyond $\rho = 2$, performance declines, especially for Llama. These results identify both the benefit and the boundary of KRPO: gains depend on a selective local neighborhood, while approaching unrestricted permutation robustness can weaken supervision from reliable relations. Appendix J discusses further methodological and evaluation limitations.

6 Conclusion

Valid changes in evaluation priorities can reorder the same responses even when the training rankings are reliable. KRPO accounts for this uncertainty through a utility-weighted Kendall neighborhood and an entropy-regularized adversary, emphasizing difficult but plausible alternatives while retaining the base preference loss. It improves macro-average win rates in all 14 objective–model combinations and leads all four comparisons with sample-level robust baselines. Controlled shifts further show lower conditional deployment risk along all three tested attribute paths at larger mixing weights. These reductions come from a single policy without retraining for each target criterion. Holding prompts and candidate responses fixed makes this evidence complementary to the broader generalization gains on generation benchmarks. The geometry and radius ablations support a central design principle: robustness benefits from admitting plausible ranking changes while preserving supervision from stable relations. Our evaluation covers two model families and four-response rankings, with controlled shifts limited to helpfulness-trained Qwen3-1.7B and three target attributes. Because model family and size vary together, these results do not isolate the effect of capacity. Evaluating longer lists, broader preference criteria, and multiple model sizes within a family would help establish the scope of these benefits.

AI USE STATEMENT

We used generative AI tools to assist with literature search, manuscript drafting, translation and editing, refinement of experimental designs, and preparation of analysis code and figures. We also used GPT-4.1-mini as an automated judge to compare policy responses against fixed reference responses for benchmark win-rate evaluation, with blinded comparisons in both presentation orders 5. These tools also assisted with clarifying the presentation of theoretical statements. The authors retain responsibility for the final manuscript, methodology, theoretical claims, implementation, and reported results.

REFERENCES

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.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models, 2021.

Daiwei Chen, Yi Chen, Aniket Rege, and Ramya Korlakai Vinayak. Pal: Pluralistic alignment framework for learning from heterogeneous preferences. *arXiv preprint arXiv:2406.08469*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. *arXiv preprint arXiv:2310.01377*, 2023.

Shihan Dou, Yan Liu, Enyu Zhou, Songyang Gao, Tianlong Li, Limao Xiong, Xin Zhao, Haoxiang Jia, Junjie Ye, Rui Zheng, et al. Alleviating shifted distribution in human preference alignment through meta-learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23805–23813, 2025.

Tom Hosking, Phil Blunsom, and Max Bartolo. Human feedback is not gold standard. In *International Conference on Learning Representations*, volume 2024, pp. 55864–55883, 2024.

Maurice G Kendall et al. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 1938.

Cheol Woo Kim, Shresth Verma, Mauricio Tec, and Milind Tambe. Preference robustness for dpo with applications to public health. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 38717–38725, 2026.

Tian Li, Ahmad Beirami, Maziar Sanjabi, and Virginia Smith. On tilted losses in machine learning: Theory and applications. *Journal of Machine Learning Research*, 24(142):1–79, 2023a.

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*, 2024a.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. From live data to high-quality benchmarks: The arena-hard pipeline, April 2024b. URL https://lmsys.org/blog/2024-04-19-arena-hard/.

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. https://github.com/tatsu-lab/alpaca_eval, 5 2023b.

Zihao Li, Chao Yang, Tong Zhang, Yakun Chen, Xianzhi Wang, Guandong Xu, and Daoyi Dong. Listwise preference alignment optimization for tail item recommendation. *IEEE Transactions on Computational Social Systems*, 2025.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, et al. Lipo: Listwise preference optimization through learning-to-rank. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2404–2420, 2025.

Debmalya Mandal, Paulius Sasnauskas, and Goran Radanovic. Distributionally robust reinforcement learning with human feedback. *arXiv preprint arXiv:2503.00539*, 2025.

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

Daniel Morales-Brotons, Thijs Vogels, and Hadrien Hendrikx. Exponential moving average of weights in deep learning: Dynamics and benefits. *arXiv preprint arXiv:2411.18704*, 2024.

Abinay Reddy Naini, Fernando Diaz, and Carlos Busso. Ranklist–a listwise preference learning framework for predicting subjective preferences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24486–24494, 2026.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models, 2024.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

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.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free rlhf. *Advances in Neural Information Processing Systems*, 37:37100–37137, 2024.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2024.

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.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback, 2020.

Yuxuan Tang and Yifan Feng. Beyond pairwise: Empowering llm alignment with (ranked) choice modeling. In *International Conference on Learning Representations*, volume 2026, pp. 21044–21074, 2026.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Scowcroft, Neel Kant, Aidan Swope, et al. Helpsteer: Multi-attribute helpfulness dataset for steerlm. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3371–3384, 2024.

Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jiawei Chen, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. Towards robust alignment of language models: Distributionally robustifying direct preference optimization. In *International Conference on Learning Representations*, volume 2025, pp. 16123–16150, 2025.

Xudong Wu, Jian Qian, Pangpang Liu, Vaneet Aggarwal, and Jiayu Chen. Distributionally robust listwise preference optimization. *arXiv preprint arXiv:2607.01715*, 2026.

Zaiyan Xu, Sushil Vemuri, Kishan Panaganti, Dileep Kalathil, Rahul Jain, and Deepak Ramachandran. Robust llm alignment via distributionally robust direct preference optimization. *Advances in Neural Information Processing Systems*, 38:27039–27076, 2026.

Ziwei Xu and Mohan Kankanhalli. Strong preferences affect the robustness of preference models and value alignment. In *International Conference on Learning Representations*, volume 2025, pp. 69079–69099, 2025.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting, 2018.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *International Conference on Learning Representations*, volume 2025, pp. 64845–64862, 2025.

Michael JQ Zhang, Zhilin Wang, Jena D Hwang, Yi Dong, Olivier Delalleau, Yejin Choi, Eunsol Choi, Xiang Ren, and Valentina Pyatkin. Diverging preferences: When do annotators disagree and do models know? *arXiv preprint arXiv:2410.14632*, 2024.

Yang Zhao, Yixin Wang, and Mingzhang Yin. Permutative preference alignment from listwise ranking of human judgments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 310–334, 2025.

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

Mingye Zhu, Yi Liu, Lei Zhang, Junbo Guo, and Zhendong Mao. Lire: listwise reward enhancement for preference alignment. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3377–3394, 2024.

Mingye Zhu, Yi Liu, Zheren Fu, Yongdong Zhang, and Zhendong Mao. Leveraging robust optimization for llm alignment under distribution shifts. *Advances in Neural Information Processing Systems*, 38:171182–171208, 2026.

A NOTATION

Table 3 summarizes the core notation for KRPO. Conditioning on the prompt x and candidate set $\mathcal{Y}$ is suppressed.

Table 3: Core notation for KRPO.

Symbol	Definition
Ranking uncertainty	
σ, $\widetilde{\sigma}$	Observed ranking; ranking under consideration (including σ).
$\ell_{\widetilde{\sigma}}$	Base preference loss evaluated under ranking $\widetilde{\sigma}$.
T	Temperature of the reference-based logistic loss, distinct from the DRO temperature γ.
P_{train}, P_{deploy}	Ranking distributions under annotation and deployment preferences.
$\mathcal{R}_{\mathrm{deploy}}$	Expected base preference loss under deployment rankings.
Utility-weighted neighborhood	
$u_{\theta,j}$	Policy-derived utility of response y_j.
w_{ab}	Normalized utility-gap cost for reversing (a, b).
$d_w(\widetilde{\sigma}, \sigma)$	Utility-weighted Kendall distance.
ρ	Normalized reversal-cost budget.
$\mathcal{A}_\rho(\sigma)$	Admissible rankings within weighted distance ρ of σ.
M_s	Maximum number of retained rankings in a restricted support, including σ.
Robust objective and coverage	
$p_0(\widetilde{\sigma} \mid \sigma)$	Distance-aware local prior over admissible rankings.
m_0, η	Prior mass on σ; distance-decay rate.
q, $q^\star_\theta$	Local adversarial ranking distribution; its optimizer.
γ	Temperature of the KL-regularized robust objective.
$\mathcal{R}_\gamma$, $\mathcal{L}_{\mathrm{KRPO}}$	Per-record robust risk; its average over training records.
δ	Deployment probability mass outside $\mathcal{A}_\rho(\sigma)$.

B Evaluation Datasets and Distribution Shifts

UltraFeedback is the training reference distribution: each record contains a single instruction, four model responses, and multi-aspect AI feedback used to derive the observed ranking (Cui et al., 2023). Table 4 summarizes how each evaluation set differs from this reference. We use *near-domain* and *OOD* descriptively, according to the dominant shift in task, source, interaction format, or evaluation priority; these labels are not the result of a formal density-ratio test. The evaluation suite comprises Arena-Hard (Li et al., 2024a;b), AlpacaEval (Li et al., 2023b), HH-RLHF (Bai et al., 2022), MT-Bench (Zheng et al., 2023), and Reddit TL;DR (Stiennon et al., 2020).

Table 4: Evaluation datasets and their dominant shifts relative to UltraFeedback (UF), the training set. No external benchmark is strictly ID: even the format-near sets use different prompts and response/reference distributions.

Dataset	N	Input and task	Relation to UF	Property tested
UltraFeedback	Train	Single-turn, broad instruction following; four ranked responses	Training reference	Fit to listwise, multi-aspect preference supervision
Arena-Hard v2.0	750	Hard real-user prompts: 253 coding, 247 math, and 250 creative-writing queries	Format-near OOD: single-turn, but shifted source, difficulty, and topic mix	Transfer to difficult technical and open-ended requests
AlpacaEval	805	Single-turn general instruction following	Near-domain in task format; OOD in prompt source and response/reference distribution	Broad instruction-following transfer under the smallest format change
HH-RLHF	500	Helpful–harmless human–assistant dialogue, including conversational context	OOD in interaction format and preference priorities	Transfer to dialogue-level helpfulness and safety trade-offs
MT-Bench	80	Flattened two-turn instructions spanning eight capability categories	OOD in compound-instruction structure and balanced capability mix	Revision, constraint retention, reasoning, and knowledge
Reddit TL;DR	500	Long-document summarization; the used `axis` sample contains news articles	Strong task, domain, and input-length OOD	Transfer from open-ended response ranking to constrained summarization

Reference responses. We retain the official reference responses provided with Arena-Hard and AlpacaEval, generated by Gemini-2.0-Flash and GPT-4, respectively. For the other benchmarks, we use fixed references generated by instruction-tuned Mistral models: Mistral-Nemo for HH-RLHF, Mistral-7B for Reddit TL;DR, and Mistral-7B with light Mistral-Nemo adaptation for MT-Bench.

The reported macro average therefore aggregates performance across distinct shift axes rather than five interchangeable samples from one common test distribution. Section 5.1 summarizes how these shifts change the attributes that define a preferred response.

B.1 Pretraining-contamination audit

Our preference-training pipeline reads only UltraFeedback; no prompt, response, reference, or label from the five evaluation benchmarks is supplied to the training objective. This rules out direct leakage through the training pipeline, but is not an exact string-overlap proof between UltraFeedback and every benchmark. We additionally audit a separate concern: whether the public benchmark inputs may have appeared during pretraining of the preference-untuned Llama-3.1-8B-Instruct and Qwen3-1.7B checkpoints. On the original benchmark inputs, we compute Min-K%++ ($K = 20\%$) (Zhang

Table 5: Complete contamination diagnostics on the original evaluation inputs.

Model	Dataset	$N_{\text{MIA}}/N_{\text{rank}}$	Min-K%++ ↑	Avg. NLL ↓	Zlib ratio ↓	Rank p (z)
Llama-3.1-8B-Instruct	Arena-Hard	750/750	3.56 ± 1.59 (3.24)	3.92 ± 1.48 (4.05)	1.37 ± 0.28 (1.40)	0.9998 (-3.54)
	AlpacaEval	805/805	2.32 ± 0.74 (2.12)	5.09 ± 0.82 (5.25)	1.42 ± 0.14 (1.43)	**0.0106** (2.30)
	HH	500/500	2.34 ± 0.63 (2.22)	5.34 ± 0.66 (5.41)	1.69 ± 0.25 (1.67)	0.945 (-1.60)
	MT-Bench	80/80	3.25 ± 1.00 (2.94)	4.31 ± 0.99 (4.38)	1.43 ± 0.20 (1.46)	0.198 (0.85)
	TL;DR	328/500	5.11 ± 0.40 (5.14)	2.49 ± 0.27 (2.47)	1.15 ± 0.10 (1.14)	0.847 (-1.02)
Qwen3-1.7B	Arena-Hard	750/750	3.97 ± 1.44 (3.88)	3.38 ± 1.33 (3.33)	1.19 ± 0.33 (1.15)	0.9999 (-3.78)
	AlpacaEval	805/805	2.86 ± 1.01 (2.84)	4.20 ± 1.11 (4.08)	0.97 ± 0.20 (0.96)	1.36×10^{-6} (4.69)
	HH	500/500	2.42 ± 0.82 (2.46)	5.03 ± 1.07 (4.96)	1.47 ± 0.47 (1.47)	0.355 (0.37)
	MT-Bench	80/80	3.71 ± 0.92 (3.69)	3.63 ± 0.90 (3.63)	1.18 ± 0.23 (1.14)	0.239 (0.71)
	TL;DR	328/500	4.34 ± 0.30 (4.33)	3.02 ± 0.31 (3.03)	1.43 ± 0.14 (1.43)	0.662 (-0.42)

Notes. Min-K%++, average NLL, and zlib ratio are mean $\pm$ SD (median); arrows indicate the direction associated with greater model familiarity. Their absolute levels are not comparable across datasets because genre and input length are confounders. Rank reports the one-sided p-value and z-statistic; bold denotes $p < 0.05$. TL;DR has 328 unique input articles for the MIA summaries and retains all 500 records for the order-based rank test.

et al., 2025; Shi et al., 2024), average negative log-likelihood (Yeom et al., 2018), zlib-normalized negative log-likelihood (Carlini et al., 2021), and the sharded rank test (Oren et al., 2024). The rank test uses shards of 25 examples and 50 random permutations per shard; it is the only diagnostic in this audit with a calibrated significance test. Table 5 reports all four diagnostics; Min-K%++, loss, and zlib ratio are treated as uncalibrated familiarity scores rather than hypothesis tests.

The audit finds no significant exposure signal for Arena-Hard, HH, MT-Bench, or TL;DR, but it does flag AlpacaEval on both backbones. This AlpacaEval result does not by itself establish verbatim leakage. Under the paraphrase control, the signal remains significant ($p = 0.00277$ for Llama and $p = 3.63 \times 10^{-5}$ for Qwen). Its persistence after surface-form changes indicates that prompt content or genre familiarity may contribute to the result, rather than exact wording alone. We therefore conclude that there is no evidence of pervasive benchmark contamination, while retaining an explicit caveat for AlpacaEval rather than claiming that all pretraining corpora are clean. All base/KRPO pairs share the same pretrained checkpoint and evaluation prompts, so this familiarity does not give KRPO differential access to the benchmark, although it may affect absolute AlpacaEval performance.

C Disagreement Analysis and Protocol

This appendix details the analysis behind Figure 2. The purpose of the analysis is not to estimate a deployment ranking distribution, but to test the structural premise used by KRPO: changing the evaluation criterion can change the ranking of a fixed candidate set, and these changes should concentrate among pairs whose preference strength is weak.

C.1 Data and weak-order construction

We use the dimension-level ratings released with UltraFeedback and HelpSteer (Cui et al., 2023; Wang et al., 2024). UltraFeedback contains $55{,}684$ prompts with four responses and four evaluation dimensions. HelpSteer contributes $7{,}485$ prompts with four responses and five attributes. For HelpSteer, we report both all five attributes and the three quality-oriented attributes. Complexity and verbosity describe response properties without consistently indicating which response is preferable, and need not align with a quality margin.

Because the ratings are discrete, ties are common and must be retained. For each prompt and dimension, we therefore construct a weak order in which responses with the same rating share a rank. Arbitrarily breaking a tie would introduce a preference relation that the annotation does not express. Figure 2a counts the number of distinct weak orders induced by the available dimensions for each prompt. Only 2.1% of UltraFeedback prompts and 4.6% of HelpSteer prompts receive one common weak order across all dimensions, showing that alternative criterion-dependent orders are the typical rather than exceptional case in these corpora.

C.2 Pairwise disagreement statistic

For every unordered response pair, we compare the directions expressed by the dimensions. We label a pair *consensus* when at least one dimension expresses a strict preference and all dimensions that do so agree, *conflict* when both strict directions occur, and *all-tie* when no dimension separates the pair. The reported *no-order* event is the union of conflict and all-tie: in either case, the dimensions do not determine one common pairwise direction.

We calculate the no-order rate over all nonidentical response pairs, yielding 332,337 UltraFeedback pairs and 41,645 HelpSteer pairs. Byte-identical responses are excluded because both the reward model and every valid evaluation criterion must assign them the same content-based comparison. Such duplicates account for 1,767 pairs (0.5%) in UltraFeedback and 3,265 pairs (7.3%) in HelpSteer. Including them would mechanically concentrate ties in the smallest-gap bin. We also do not condition on a minimum number of dimensions expressing a strict preference: that filter disproportionately removes small-gap pairs and would induce selection bias in the trend.

C.3 Utility-gap stratification and uncertainty

We score each response with Skywork-Reward-V2-Llama-3.1-8B and define a pair's external utility gap as the absolute difference between its two reward-model scores. Because this model is separate from the dimension annotations used to define disagreement, the gap does not reproduce those labels by construction. Within each corpus, we divide nonidentical pairs into equal-frequency deciles of the utility gap and compute the no-order proportion in each decile. The bands in Figure 2b are 95% Wilson score intervals for these binomial proportions. They quantify sampling uncertainty within a decile and are not training-run confidence intervals.

On UltraFeedback, the no-order rate decreases from 45.2% in the smallest-gap decile to 5.9% in the largest-gap decile. On the three quality-oriented HelpSteer attributes, it decreases from 36.1% to 9.7%. When all five HelpSteer attributes are included, the decline is weaker (35.2% to 21.1%), as expected because complexity and verbosity can disagree with quality without indicating an uncertain quality comparison. This distinction prevents descriptive attributes from being interpreted as noisy quality labels.

C.4 Robustness checks

The qualitative decline is unchanged when the raw reward gap is replaced by a within-prompt standardized gap or by its corpus percentile. Randomly shuffling gaps across response pairs while retaining the disagreement labels destroys the monotone relationship, providing a negative control. Together, these checks support the use of utility gaps as a relative proxy for pairwise preference strength. They do not imply that the external reward model recovers a unique human utility or that every small-gap pair must reverse under deployment shift.

D Controlled Preference Shift Protocol

Data and criteria. We use the 2,802 held-out UltraFeedback prompts whose four responses all carry ratings for helpfulness, honesty, instruction following, and truthfulness. Ratings are used on their original scale. The target criterion t is instruction following, honesty, or truthfulness, and the deployment score is

$$r_\alpha = (1-\alpha)\, r_h + \alpha\, r_t, \qquad \alpha \in \{0, 0.1, 0.3, 0.37, 0.46, 0.55, 0.63, 0.7, 0.9, 1\}, \tag{10}$$

where h denotes helpfulness. Candidates are ranked by decreasing score, with equal scores keeping the original candidate order. Figure 3a uses the mixing weight α as its horizontal axis. The Kendall distance in Figure 3c is the mean number of candidate pairs, out of six, that the training and deployment rankings order differently; this unweighted distance is distinct from the utility-weighted geometry of Section 4.1.

Deployment risk. Both policies are evaluated with the same base DPO-allpairs loss, reference model, utility scale $\beta = 0.1$, and temperature $T = 1.5$, and remain fixed throughout. Let $\sigma_{\alpha,t,i}$ be

Table 6: Full-shift endpoints of the controlled preference shift (Figure 3c). Kendall distance is the mean number of candidate pairs, out of six, ordered differently by the training and deployment criteria; losses are mean base DPO-allpairs losses under the deployment ranking; reduction is $G_t(1)$ and attenuation is $100 - I_t$.

Deployment criterion	Kendall distance	DPO loss	KRPO loss	Reduction (%)	Attenuation (%)
No shift (helpfulness)	0.000	0.498	0.495	0.4	–
Instruction following	1.432	0.566	0.556	1.9	12.2
Honesty	1.546	0.630	0.610	3.0	12.9
Truthfulness	1.638	0.724	0.692	4.4	13.3

the ranking that criterion (α, t) induces on prompt i. The empirical risk of policy θ is

$$\widehat{R}_{\theta,t}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} \ell(\theta; x_i, \mathcal{Y}_i, \sigma_{\alpha,t,i}). \tag{11}$$

Because each criterion induces a single ranking per prompt, this quantity is the deployment risk of Section 3 under a point-mass deployment distribution, averaged over held-out records.

Reported quantities. Writing $R_{D,t}(\alpha)$ and $R_{K,t}(\alpha)$ for the empirical risks of DPO and KRPO, panels (a) and (c) of Figure 3 report the relative reduction $G_t(\alpha) = 100\,[R_{D,t}(\alpha) - R_{K,t}(\alpha)]/R_{D,t}(\alpha)$, and panel (b) reports the reduction in the shift-induced loss increase, $100 - I_t$, where $I_t = 100\,[R_{K,t}(1) - R_K(0)]/[R_{D,t}(1) - R_D(0)]$ is the percentage of DPO's increase retained by KRPO. This is the attenuation quoted in the text. The no-shift losses $R_D(0)$ and $R_K(0)$ are shared by all three paths. Table 6 lists the full-shift endpoints. For helpfulness to truthfulness, for example, DPO loss rises from 0.498 to 0.724 and KRPO loss from 0.495 to 0.692; the increments (0.226 and 0.196 before rounding) give 13.3% attenuation, whereas the endpoint losses give a 4.4% reduction. The two percentages measure different quantities.

Uncertainty. Uncertainty is estimated with a paired prompt bootstrap of 1,000 resamples. Panel (a) shows point estimates. Panel (c) shows ± 1 SE of the paired loss difference divided by the observed DPO mean, which is not a confidence interval for the ratio. Panel (b) shows point estimates because joint bootstrap draws of the endpoint and no-shift losses are unavailable. The 95% intervals of the endpoint loss differences are $[0.008, 0.014]$ nats for instruction following, $[0.016, 0.022]$ for honesty, and $[0.029, 0.036]$ for truthfulness; they reflect evaluation-prompt uncertainty for fixed policies, not variation across training seeds.

E Supporting Theory and Proofs

Theorem 1 connects a local shift in evaluation conditions to the policy-weighted ambiguity set used by KRPO. This appendix states the formal coverage result, relates policy utility gaps to its oracle reward margins, bounds the prior-mismatch term in Theorem 2, and combines these results into a conditional deployment-risk bound. Complete proofs follow in Section E.5.

E.1 Formal latent-to-ranking coverage

Let $\mathcal{Z}$ be the space of latent evaluation conditions, with distance $d_{\mathcal{Z}}$. For fixed $(x, \mathcal{Y})$, let Q_{deploy} denote the conditional deployment distribution over conditions $z \in \mathcal{Z}$. Each condition assigns rewards $r_z(x, y)$, and $T_x(z)$ ranks the candidates by these rewards with a fixed rule for ties. The conditional ranking distribution $P_{\text{deploy}}(\cdot \mid x, \mathcal{Y})$ from Section 3 is thus the distribution of $T_x(z)$ when $z \sim Q_{\text{deploy}}$. The following proposition relates locality in the condition space to coverage by the utility-weighted Kendall neighborhood.

Proposition 1 (Formal version of Theorem 1)**.** *Fix $(x, \mathcal{Y}, \sigma)$ and a nominal condition z_0 such that $\sigma = T_x(z_0)$, and write $r_0 = r_{z_0}$. All distributions below are conditional on $(x, \mathcal{Y})$. Suppose that, for some $L > 0$, every deployment condition satisfies $\max_j |r_z(x, y_j) - r_0(x, y_j)| \le L d_{\mathcal{Z}}(z, z_0)$ and, for some $\epsilon \ge 0$ and $\delta \in [0, 1]$, $Q_{\text{deploy}}(d_{\mathcal{Z}}(z, z_0) \le \epsilon) \ge 1 - \delta$.*

Let $n_K = \binom{K}{2}$, $g^\star_{ab} = |r_0(x, y_a) - r_0(x, y_b)|$, *and* $\bar{g}^\star = n_K^{-1} \sum_{a<b} g^\star_{ab} > 0$. *Suppose the policy gaps* $\widehat{g}_{ab} = |u_{\theta,a} - u_{\theta,b}|$ *satisfy* $\max_{a<b} |\widehat{g}_{ab} - g^\star_{ab}| \le e < \bar{g}^\star$, *and define*

$$\rho^\star_\epsilon = \sum_{a<b:\, g^\star_{ab} \le 2L\epsilon} \frac{g^\star_{ab}}{\bar{g}^\star} \ , \qquad \Delta_e = \frac{n_K e}{\bar{g}^\star - e} \ . \tag{12}$$

Then

$$\begin{aligned} \Pr_{z \sim Q_{\text{deploy}}} [\delta_{ab}(T_x(z), \sigma) = 1] \le Q_{\text{deploy}} \left(d_{\mathcal{Z}}(z, z_0) \ge \frac{g^\star_{ab}}{2L} \right) \ , \\ P_{\text{deploy}} \left(\mathcal{A}_{\rho^\star_\epsilon + \Delta_e}(\sigma) \right) \ge 1 - \delta \ . \end{aligned} \tag{13}$$

E.2 Policy utility gaps identify reward margins

Fix a prompt x and the nominal reward r_0 used in Proposition 1. For a reference policy with $\pi_{\text{ref}}(y \mid x) > 0$, consider the KL-regularized reward objective

$$\max_\pi \left\{ \mathbb{E}_{y \sim \pi(\cdot|x)}[r_0(x, y)] - \beta \, \mathrm{KL}(\pi(\cdot \mid x) \,\|\, \pi_{\text{ref}}(\cdot \mid x)) \right\} \ . \tag{14}$$

Proposition 2 (Policy-gap identification). *Assume* $Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp(r_0(x, y)/\beta) < \infty$, *and let* $\pi^\star$ *maximize Equation 14. Then*

$$\beta \log \frac{\pi^\star(y \mid x)}{\pi_{\text{ref}}(y \mid x)} = r_0(x, y) - \beta \log Z(x) \ . \tag{15}$$

Consequently, the policy utility difference of every response pair equals its reward difference. If a policy π_θ *satisfies*

$$\max_{1 \le j \le K} |\log \pi_\theta(y_j \mid x) - \log \pi^\star(y_j \mid x)| \le \varepsilon_\pi \ , \tag{16}$$

then the gap error in Proposition 1 obeys $e \le 2\beta\varepsilon_\pi$.

Under a Bradley–Terry preference model, the reward difference in Equation 15 is also the pairwise preference log-odds. The coverage result therefore uses the policy gap as a proxy for preference strength, not merely as an arbitrary model score. Proposition 2 is an identity at the KL-regularized optimum followed by a conditional approximation bound; it does not assume that every finite training run reaches $\pi^\star$. Because normalized pair costs are invariant to positive scaling and additive shifts, the same argument extends to scores that approximate a positive affine transformation of r_0.

E.3 Prior-mismatch bound

Proposition 3 (Finite prior-mismatch penalty). *If* $\mathcal{A}_\rho(\sigma)$ *contains alternatives, then every distribution* $Q_{\mathcal{A}}$ *supported on the set satisfies*

$$\mathrm{KL}(Q_{\mathcal{A}} \| p_0) \le \varepsilon_{\text{KL}} := \max\{-\log m_0, \ \eta\rho + \log Z_\sigma - \log(1 - m_0)\} \ . \tag{17}$$

For a singleton ambiguity set, the divergence is zero.

E.4 End-to-end implication

Proposition 1 guarantees coverage at the sufficient radius $\rho^\star_\epsilon + \Delta_e$. Any larger radius preserves this coverage. The next result combines that fact with the conditional deployment-risk bound.

Corollary 1 (Conditional end-to-end deployment bound). *Suppose the conditions of Proposition 1 hold and KRPO uses* $\rho \ge \rho^\star_\epsilon + \Delta_e$. *Let* ε_{KL} *be the bound in Proposition 3, and define*

$$B_\gamma = \min\{L_{\max}, \ \mathcal{R}_\gamma(\theta; \sigma) + \gamma\varepsilon_{\text{KL}}\} \ . \tag{18}$$

Then

$$\mathcal{R}_{\text{deploy}}(\theta; x, \mathcal{Y}) \le (1 - \delta) B_\gamma + \delta L_{\max} \ . \tag{19}$$

The bound separates the two roles of KRPO. The ambiguity-set construction controls the omitted deployment mass through δ, while entropic DRO controls the covered distribution through B_γ. Neither result requires the ranking distribution itself to follow a parametric model.

E.5 Complete Proofs

E.5.1 Proof of Proposition 2

For fixed x, write $r_y = r_0(x, y)$, $\pi_y = \pi(y \mid x)$, and $\pi_{\mathrm{ref},y} = \pi_{\mathrm{ref}}(y \mid x)$. The Lagrangian of Equation 14 is

$$\mathcal{L}(\pi, \lambda) = \sum_y \pi_y r_y - \beta \sum_y \pi_y \log \frac{\pi_y}{\pi_{\mathrm{ref},y}} + \lambda \left(\sum_y \pi_y - 1 \right) \ .$$

Its stationary condition gives $\pi^\star(y \mid x) = Z(x)^{-1} \pi_{\mathrm{ref}}(y \mid x) \exp(r_0(x, y)/\beta)$. Taking the reference-relative log ratio proves Equation 15; subtracting two instances cancels the shared normalizer.

Let $u_{\theta,j}$ and $u_j^\star$ denote the corresponding policy utilities. From Equation 16, $|u_{\theta,j} - u_j^\star| \le \beta \varepsilon_\pi$. The reverse triangle inequality then gives

$$||u_{\theta,a} - u_{\theta,b}| - |u_a^\star - u_b^\star|| \le |u_{\theta,a} - u_a^\star| + |u_{\theta,b} - u_b^\star| \le 2\beta\varepsilon_\pi \ .$$

Equation 15 identifies the second gap with $g_{ab}^\star$, proving the result. □

E.5.2 Proof of Proposition 1

Suppose that $T_x(z)$ reverses pair (a, b) relative to $\sigma = T_x(z_0)$. Without loss of generality, let $r_0(x, y_a) - r_0(x, y_b) \ge 0$ and $r_z(x, y_a) - r_z(x, y_b) \le 0$. Then

$$\begin{aligned} g_{ab}^\star &\le |[r_0(x, y_a) - r_0(x, y_b)] - [r_z(x, y_a) - r_z(x, y_b)]| \\ &\le |r_0(x, y_a) - r_z(x, y_a)| + |r_0(x, y_b) - r_z(x, y_b)| \\ &\le 2L d_{\mathcal{Z}}(z, z_0) \ . \end{aligned}$$

Thus a reversal implies $d_{\mathcal{Z}}(z, z_0) \ge g_{ab}^\star/(2L)$; taking deployment probabilities proves the first inequality in Equation 13. It also follows that every condition inside the latent ϵ-ball reverses only pairs satisfying $g_{ab}^\star \le 2L\epsilon$. For oracle weights $w_{ab}^\star = g_{ab}^\star/\bar{g}^\star$,

$$d_{w^\star}(T_x(z), \sigma) \le \sum_{a<b:\, g_{ab}^\star \le 2L\epsilon} w_{ab}^\star = \rho_\epsilon^\star \ . \tag{20}$$

It remains to transfer this envelope to the policy weights. Regard the n_K pair gaps as vectors $g^\star$ and $\widehat{g}$, and let $S = \sum_{a<b} g_{ab}^\star = n_K \bar{g}^\star$, $\widehat{S} = \sum_{a<b} \widehat{g}_{ab}$, and $D = \|\widehat{g} - g^\star\|_1 \le n_K e$. Since $e < \bar{g}^\star$, $\widehat{S} \ge S - D > 0$. The normalized weight vectors satisfy

$$\begin{aligned} \|w - w^\star\|_1 &= n_K \left\| \frac{\widehat{g}}{\widehat{S}} - \frac{g^\star}{S} \right\|_1 && (21) \\ &\le n_K \left(\frac{D}{\widehat{S}} + \frac{|\widehat{S} - S|}{\widehat{S}} \right) && (22) \\ &\le \frac{2 n_K D}{S - D} \ . && (23) \end{aligned}$$

Moreover, both weight vectors sum to n_K, so their difference sums to zero. For the inversion set $I(\tilde{\sigma}, \sigma)$ of any ranking $\tilde{\sigma}$, this gives

$$\begin{aligned} |d_w(\tilde{\sigma}, \sigma) - d_{w^\star}(\tilde{\sigma}, \sigma)| &= \left| \sum_{(a,b) \in I(\tilde{\sigma}, \sigma)} (w_{ab} - w_{ab}^\star) \right| && (24) \\ &\le \frac{1}{2} \|w - w^\star\|_1 && (25) \\ &\le \frac{n_K D}{S - D} \le \frac{n_K e}{\bar{g}^\star - e} = \Delta_e \ . && (26) \end{aligned}$$

Combining Equations 20 and 26 shows that every condition in the latent ϵ-ball induces a ranking in $\mathcal{A}_{\rho_\epsilon^\star + \Delta_e}(\sigma)$. The latent coverage assumption then proves the second inequality in Equation 13. □

E.5.3 Proof of Theorem 2

Let $Q = P_{\text{deploy}}(\cdot \mid x, \mathcal{Y})$ and $\mathcal{A} = \mathcal{A}_\rho(\sigma)$. By the definition of δ, $Q = (1-\delta)Q_{\mathcal{A}} + \delta Q_{\mathcal{A}^c}$. Since $\ell_{\tilde{\sigma}} \in [0, L_{\max}]$,

$$\mathbb{E}_Q[\ell_{\tilde{\sigma}}] \le (1-\delta)\mathbb{E}_{Q_{\mathcal{A}}}[\ell_{\tilde{\sigma}}] + \delta L_{\max} \ . \tag{27}$$

The Donsker–Varadhan variational identity gives, for every $Q' \ll p_0$,

$$\mathbb{E}_{Q'}[\ell_{\tilde{\sigma}}] \le \mathcal{R}_\gamma(\theta;\sigma) + \gamma \mathrm{KL}(Q' \| p_0) \ .$$

The prior is positive on $\mathcal{A}$, so the inequality applies to $Q_{\mathcal{A}}$. Substitution into Equation 27 proves Equation 9. □

E.5.4 Proof of Proposition 3

For any Q' supported on $\mathcal{A}$,

$$\mathrm{KL}(Q' \| p_0) \le \log \frac{1}{\min_{\tilde{\sigma} \in \mathcal{A}} p_0(\tilde{\sigma} \mid \sigma)} \ .$$

If the set contains alternatives, Equation 6 implies

$$\min_{\tilde{\sigma} \in \mathcal{A}} p_0(\tilde{\sigma} \mid \sigma) \ge \min\left\{ m_0, \ \frac{1-m_0}{Z_\sigma} e^{-\eta\rho} \right\} \ ,$$

which yields the stated $\varepsilon_{\mathrm{KL}}$. For a singleton set, both $Q_{\mathcal{A}}$ and p_0 are point masses at σ, so the divergence is zero. □

E.5.5 Proof of Corollary 1

Let $\delta_{\mathcal{A}} = P_{\text{deploy}}(\mathcal{A}_\rho(\sigma)^c \mid x, \mathcal{Y})$. Proposition 1 and $\rho \ge \rho^\star_\epsilon + \Delta_e$ give $\delta_{\mathcal{A}} \le \delta$. Theorem 2, Proposition 3, and the trivial in-set bound $\mathbb{E}_{Q_{\mathcal{A}}}[\ell_{\tilde{\sigma}}] \le L_{\max}$ give

$$\mathcal{R}_{\text{deploy}} \le (1-\delta_{\mathcal{A}})B_\gamma + \delta_{\mathcal{A}} L_{\max} \ .$$

Because $B_\gamma \le L_{\max}$, the right-hand side is nondecreasing in $\delta_{\mathcal{A}}$. Replacing it by its upper bound δ proves Equation 19. □

F Baseline Objectives and Implementation Details

F.1 Baseline objectives

For response y_i, let $r_i = \log \pi_\theta(y_i \mid x) - \log \pi_{\text{ref}}(y_i \mid x)$, $u_i = \beta r_i$, and $\varphi(z) = \log(1 + \exp(-z))$. All records have $K = 4$ responses and therefore $M = \binom{K}{2} = 6$ implied pairs. We use $\beta = 0.1$ and temperature $T = 1.5$ for reference-based logistic losses.

DPO-allpairs. We average DPO (Rafailov et al., 2023) over all implied pairs:

$$\mathcal{L}_{\text{DPO-AP}}(\sigma) = \frac{1}{M} \sum_{a<b} \varphi\left(\frac{u_{\sigma_a} - u_{\sigma_b}}{T} \right) \ . \tag{28}$$

The adaptation is exhaustive pair extraction from the four-way ranking.

LPO4Rec. Our LPO4Rec instantiation (Li et al., 2025) is the rank-weighted all-pairs loss

$$\mathcal{L}_{\text{LPO4Rec}}(\sigma) = \frac{1}{Z_K} \sum_{a<b} \frac{1}{K-a+1} \varphi\left(\frac{u_{\sigma_a} - u_{\sigma_b}}{T} \right) \ , \quad Z_K = \sum_{a<b} \frac{1}{K-a+1} \ . \tag{29}$$

We use fixed ranked text responses and omit the source method's recommendation-specific negative sampling and tail-item reweighting.

LiPO. We use LiPO-λ (Liu et al., 2025). For graded score ψ_i, define $G_i = 2^{\psi_i} - 1$, predicted rank τ_i, and $D_i = 1/\log(1+\tau_i)$. Then

$$\mathcal{L}_{\mathrm{LiPO}} = \frac{1}{|\mathcal{P}_\psi|} \sum_{(i,j)\in\mathcal{P}_\psi} |G_i - G_j|\,|D_i - D_j|\,\varphi\!\left(\frac{u_i - u_j}{T}\right) \quad , \quad \mathcal{P}_\psi = \{(i,j) : \psi_i > \psi_j\} \quad . \tag{30}$$

The nominal baseline uses the released UltraFeedback scores directly.

Plackett–Luce. We apply the standard Plackett–Luce likelihood (Plackett, 1975) to implicit utilities:

$$\mathcal{L}_{\mathrm{PL}}(\sigma) = \frac{1}{K}\sum_{a=1}^{K}\left[\log\sum_{b=a}^{K}\exp(u_{\sigma_b}) - u_{\sigma_a}\right] \quad . \tag{31}$$

IPO. For $h_{ij} = r_i - r_j$, we average sampled IPO (Azar et al., 2024) over all implied pairs:

$$\mathcal{L}_{\mathrm{IPO}}(\sigma) = \frac{1}{M}\sum_{a<b}\left(h_{\sigma_a\sigma_b} - \frac{1}{2\tau_{\mathrm{IPO}}}\right)^2 \quad , \qquad \tau_{\mathrm{IPO}} = 0.2 \quad . \tag{32}$$

SimPO. With $s_i = |y_i|^{-1}\log\pi_\theta(y_i \mid x)$, our all-pairs SimPO (Meng et al., 2024) loss is

$$\mathcal{L}_{\mathrm{SimPO}}(\sigma) = \frac{1}{M}\sum_{a<b}\varphi(\beta_s(s_{\sigma_a} - s_{\sigma_b}) - \gamma_s) \quad , \quad \beta_s = 2.5 \quad , \quad \gamma_s = 0.55\beta_s \quad . \tag{33}$$

SLiC-HF. We use an all-pairs SLiC-style hinge (Zhao et al., 2023):

$$\mathcal{L}_{\mathrm{SLiC}}(\sigma) = \frac{1}{M}\sum_{a<b}\max\{0, 1 - (u_{\sigma_a} - u_{\sigma_b})\} \quad . \tag{34}$$

This instantiation uses policy–reference log-ratios and omits the original cross-entropy regularizer.

F.2 Sample-level robust baselines

For DPO-allpairs and LiPO comparisons, let L_n be the complete list loss for prompt n in a physical minibatch of $B = 8$. We apply Dr. DPO (Wu et al., 2025) and KLDPO (Xu et al., 2026) at the prompt level:

$$\mathcal{L}_{\mathrm{Dr.DPO}} = -\beta'\left[\log\sum_{n=1}^{B}\exp(-L_n/\beta') - \log B\right] \quad , \qquad \beta' = 1 \quad , \tag{35}$$

$$q_n = \frac{\exp((L_n - \bar{L})/\tau_{\mathrm{KL}})}{\sum_{m=1}^{B}\exp((L_m - \bar{L})/\tau_{\mathrm{KL}})} \quad , \qquad \tau_{\mathrm{KL}} = 1 \quad , \tag{36}$$

$$\mathcal{L}_{\mathrm{KLDPO}} = \sum_{n=1}^{B} q_n L_n \quad , \qquad \bar{L} = B^{-1}\sum_{n=1}^{B} L_n \quad . \tag{37}$$

The KLDPO weights q_n are detached. Dr. DPO downweights high-loss prompts; KLDPO upweights them. No synthetic corruption is added.

F.3 KRPO implementation

Let θ_t denote the trainable policy at step t. We maintain a stop-gradient geometry policy by the exponential moving average

$$\bar{\theta}_t = \alpha_{\mathrm{EMA}}\bar{\theta}_{t-1} + (1 - \alpha_{\mathrm{EMA}})\theta_t \quad , \qquad \bar{\theta}_0 = \theta_0 \quad , \tag{38}$$

with $\alpha_{\mathrm{EMA}} = 0.9$. Pair costs, the resulting support, and the local prior use $u_{\bar{\theta}_t}$, whereas every permutation loss $\ell_{\tilde{\sigma}}$ and its gradient use the current policy θ_t. This separates the discrete support update from the policy update that it supervises. In our adapter-tuned implementation, only the

trainable DoRA parameters are averaged; the frozen backbone is shared rather than duplicated. At any fixed step, the analysis in Appendix E applies with $\bar{\theta}_t$ as the utility proxy because it conditions on the proxy-gap error, not on how the proxy is updated.

For each prompt, response log-probabilities are computed once and reused for all admitted permutations. Pair costs use the normalized utility gap

$$\bar{g}_{\bar{\theta}_t} = \frac{2}{K(K-1)} \sum_{p<q} |u_{\bar{\theta}_t,p} - u_{\bar{\theta}_t,q}| + \varepsilon_g \ . \tag{39}$$

In the exact configuration, we enumerate all $K!$ permutations, retain those satisfying Equation 5, and always retain the observed ranking. The observed ranking receives prior mass m_0; remaining mass is proportional to $\exp(-\eta d_w)$ over admitted alternatives. Appendix G describes and evaluates restricted-support alternatives.

For LiPO, an alternative order $\tilde{\sigma}$ receives scores by rank position: $\psi^{\tilde{\sigma}}_{\tilde{\sigma}_r} = \psi_{\sigma_r}$. This preserves the score multiset and recovers the original labels for $\tilde{\sigma} = \sigma$.

At step t, we use $\lambda_t = \min(1, t/T_{\text{warm}})$ and optimize

$$\mathcal{L}_t(\theta; \sigma) = \ell_\sigma + \lambda_t \gamma \log \mathbb{E}_{\tilde{\sigma} \sim p_0} \left[\exp\left(\frac{\ell_{\tilde{\sigma}} - \ell_\sigma}{\gamma} \right) \right] \ . \tag{40}$$

Pair costs, support, and prior are stop-gradient quantities. The expectation is evaluated with log-sum-exp.

Time complexity. The following accounting describes the exact configuration. Let B be the number of records in a minibatch, $J_K = \binom{K}{2}$ the number of response pairs, and $c_b(K)$ the scalar cost of evaluating the base loss for one ranking. Let P_A denote the number of trainable adapter parameters and $T_{\text{EMA-fwd}}$ one no-gradient forward over the BK prompt–response sequences using the EMA adapter. Let T_{base} denote the model computation of the base objective: one batched policy forward over the BK prompt–response sequences, one backward pass, and, for a reference-based objective, one no-gradient reference forward unless its scores are cached. Updating the EMA state costs $\mathcal{O}(P_A)$, and KRPO computes the pair costs in $\Theta(BJ_K)$ time. Exact enumeration then evaluates J_K reversal indicators for each of the $K!$ rankings, requiring $\Theta(BK!J_K)$ time. The permutation losses add $\mathcal{O}(BK!c_b(K))$, and evaluating the closed-form entropic DRO objective by log-sum-exp adds $\mathcal{O}(BK!)$. The objectives used here have $c_b(K) = \mathcal{O}(K^2)$ in our implementation, so the total cost is

$$T_{\text{KRPO}} = T_{\text{base}} + T_{\text{EMA-fwd}} + \mathcal{O}(P_A + BK!K^2) \ . \tag{41}$$

All permutation losses reuse the same BK response log-probabilities; therefore, the factorial term counts only scalar ranking operations and does not multiply the number of transformer evaluations. The only additional transformer evaluation is the single no-gradient EMA-policy forward. An optimized Plackett–Luce evaluator can reduce $c_b(K)$ to $\mathcal{O}(K)$, although the exact support construction remains $\Theta(BK!J_K)$. In the preference post-training setting, K is fixed by the data-collection protocol and does not grow with dataset size, model size, or sequence length; supervision is typically pairwise or a short ranked list. All end-to-end training experiments use $K = 4$, giving only $J_K = 6$ response pairs and $K! = 24$ possible rankings. Exact KRPO therefore makes 144 weighted pair–ranking comparisons, evaluates at most 24 scalar ranking losses with at most six pair terms each, and applies log-sum-exp to 24 values, only a few hundred scalar operations per record. For substantially longer lists, retaining at most M_s rankings reduces the scalar loss-evaluation cost to $\mathcal{O}(BM_sK^2)$, in addition to support-construction costs (Appendix G).

Space complexity. Let S_{base} include the policy state, gradients, optimizer state, and activations required by the base objective. The EMA geometry policy adds $\mathcal{O}(P_A)$ persistent adapter state; its no-gradient forward is executed sequentially and does not retain activations for backpropagation. The permutation table is shared across the minibatch and stores $\mathcal{O}(K!K)$ indices. Per-record pair costs require $\mathcal{O}(K^2)$ space, while distances, masks, priors, and losses require $\mathcal{O}(K!)$ scalars. A straightforward reverse-mode implementation additionally retains $\mathcal{O}(BK!c_b(K))$ scalar intermediates. Hence,

$$S_{\text{KRPO}} = S_{\text{base}} + \mathcal{O}\big(P_A + K!K + BK^2 + BK!c_b(K)\big) \ . \tag{42}$$

Table 7: End-to-end time and peak allocated GPU memory for DPO and DPO+KRPO (ours) on one NVIDIA A100 80GB under the shared configuration in Table 8. Step time reports mean ± standard deviation in seconds per optimizer step; relative and extra columns compare against DPO, with extra reported as a percentage increase.

		Time			Memory		
Model	Method	Step (s)	Rel.	Extra (%)	Peak (GB)	Rel.	Extra (%)
Qwen3-1.7B	DPO	3.86 ± 0.25	1.000	–	13.46	1.000	–
Qwen3-1.7B	**+KRPO (ours)**	4.53 ± 0.24	1.173	+17.3	13.59	1.010	+1.0
Llama-3.1-8B	DPO	7.72 ± 1.08	1.000	–	25.12	1.000	–
Llama-3.1-8B	**+KRPO (ours)**	9.01 ± 1.32	1.167	+16.7	25.43	1.013	+1.3

Table 8: Training configuration and KRPO hyperparameter search ranges.

Model and data	
Backbones	Qwen3-1.7B; Llama-3.1-8B-Instruct
Sequence length	512 prompt + 512 response tokens (1024 total)
LoRA	DoRA, rank 32, $\alpha = 64$, dropout 0.05, on all attention and MLP projections (query, key, value, output, gate, up, down)
Optimization	
Optimizer	AdamW, bf16, linear decay, no warm-up, weight decay 0
Learning rate	5×10^{-5} default; SimPO $5 \times 10^{-6}/10^{-6}$ (Qwen/Llama)
Batching	batch size 8 prompts per update
Schedule	1 epoch (≈7.2k steps), seed 42, one NVIDIA A100 80GB
Geometry policy	Adapter-only EMA, $\alpha_{\mathrm{EMA}} = 0.9$, one stop-gradient forward per step
KRPO search ranges	
Radius ρ	$\{1.5, 2.0\}$; ablation grid 0.1–4.0 and unrestricted
Prior mass m_0	$\{0.7, 0.9, 0.95\}$
Warm-up steps T_{warm}	$\{100, 150, 200\}$
Qwen DPO-allpairs: fixed settings	
Entropic temperature γ	0.5
Prior distance-decay rate η	1.0

Only the current-policy forward retains transformer activations for backward; the EMA forward shares the frozen backbone and carries no gradient graph. For $K = 4$, the shared permutation table contains only $24 \times 4 = 96$ integer indices; each record adds six distinct pair costs and 24 entries in each of the distance, mask, prior, and loss vectors. These small tensors and their scalar differentiation graph are negligible relative to the model state and transformer activations in S_{base}. With a directly sampled support of M_s rankings, longer lists require $\mathcal{O}(M_s K)$ support storage and $\mathcal{O}(BM_s c_b(K))$ scalar working memory.

KRPO adds 17.3% and 16.7% step time but only 1.0% and 1.3% peak memory on Qwen and Llama, respectively. The latter includes the adapter-only EMA state and the small ranking tensors in Equation 42. Consistent with Equations 41 and 42, this overhead requires neither a second full backbone nor an additional backward pass and is confined to post-training: the EMA policy and local adversary are discarded at deployment, so inference cost is unchanged.

F.4 Training configuration and hyperparameter search

Table 8 summarizes the training configuration shared by all objective–model combinations and the KRPO-specific hyperparameter search ranges. Within each base/+KRPO/robust-baseline compar-

ison, every run uses this configuration; only the loss changes. The main experiments use $\rho = 2$ or $\rho = 1.5$ per objective, $m_0 = 0.9$, and $T_{\text{warm}} = 150$. The Qwen3-1.7B DPO-allpairs main-experiment checkpoint used as the exact anchor in Appendix G has entropic temperature $\gamma = 0.5$ and prior distance-decay rate $\eta = 1$; the support-approximation variants retain these values. These checkpoint-specific settings are listed separately from the search ranges in Table 8.

G Ranking Support Approximation

Exact enumeration is practical for the four-response rankings used in our training experiments. This appendix tests whether a restricted support can preserve the exact KRPO objective and its utility gradients, and examines how the required budget depends on the ambiguity neighborhood.

G.1 Support construction and evaluation protocol

The top-M_s variant retains at most M_s admissible rankings nearest to the observed order under d_w, including the observed ranking itself. It starts at the observed ranking and expands adjacent swaps that add an inversion relative to it. Each swap adds its nonnegative pair cost, and a priority queue emits rankings in nondecreasing weighted distance until the budget is reached or the ambiguity set is exhausted, pruning candidates outside radius ρ and deduplicating visited permutations. Weighted Kendall distance is additive over inversions, and every admissible ranking has an inversion-increasing adjacent-swap path from σ whose intermediate distances do not exceed its final distance. Nonnegative swap costs therefore justify both radius pruning and best-first enumeration of the nearest admissible rankings without first enumerating all $K!$ permutations. The sampled variant always includes the observed ranking and draws up to $M_s - 1$ distinct alternatives without replacement according to the distance-based prior. In this $K = 4$ comparison, sampling uses the enumerated ambiguity set. Both variants preserve anchor mass m_0 and redistribute the remaining prior mass over the retained alternatives using the original distance-decay rule; the adversary is then recomputed on the restricted support. If no alternative is retained, the observed ranking has mass one.

The offline analysis uses cached utilities and pair weights from the Qwen3-1.7B LiPO–KRPO diagnostic checkpoint used for Figure 4c, on 996 held-out UltraFeedback prompts. This diagnostic checkpoint uses current-policy geometry without EMA. We reuse its available cache because, for fixed utilities and pair weights, the DPO-allpairs robust objective and its utility gradients can be evaluated without a model forward pass. Both exact and approximate computations receive the same inputs; this is a numerical fidelity test, not a comparison of LiPO and DPO training outcomes. On these fixed inputs, we evaluate the DPO-allpairs robust objective with $K = 4$, $\rho = 2$, $\gamma = 0.5$, $\eta = 1$, $m_0 = 0.9$, and base-loss temperature $T = 1.5$. The full-training comparison below separately uses the Qwen3-1.7B DPO-allpairs configuration with EMA geometry.

We compare the approximate and exact adversaries, the robust correction $c_\sigma = \mathcal{R}_\gamma - \ell_\sigma$, and its derivatives with respect to response utilities, keeping pair weights fixed during differentiation. Captured mass is the probability that the exact adversary assigns to the retained support. Total variation (TV) compares the recomputed approximate adversary, padded with zeros outside its support, with the exact distribution. Gradient cosine measures agreement between the two utility gradients. We report absolute correction error because the exact correction can be near zero, making relative error unstable.

G.2 Offline approximation fidelity

Table 9 shows that a budget of eight closely approximates exact KRPO with either construction. Both capture 99.7% of the exact adversarial mass, with gradient cosines of 0.997 and 0.996 and absolute correction errors below 5×10^{-4}. A budget of twelve covers the complete ambiguity set on every analyzed prompt and recovers the exact objective to numerical precision. More aggressive truncation loses fidelity: the top-4 and top-2 gradient cosines fall to 0.900 and 0.620.

The exact ambiguity set contains 7.77 rankings on average, and top-8 already covers the full set on 74% of prompts. At $\rho = 2$, budgets of 7 and 8 are needed for 99% mass capture and a gradient cosine of 0.99, respectively: approximately 0.9 and 1.0 times the mean neighborhood size. The

Table 9: Approximation fidelity on 996 held-out prompts at $K = 4$ and $\rho = 2$. All quantities are averaged over prompts. Sampled results also average five support-sampling seeds, which are distinct from training seeds. M_s is a maximum budget including the observed ranking. Shading highlights the budget-eight configurations.

Support	M_s	Captured mass ↑	TV ↓	Gradient cosine ↑	Correction error ↓
Top-M_s	2	91.3%	0.090	0.620	1.1×10^{-2}
Top-M_s	4	95.2%	0.049	0.900	6.9×10^{-3}
Top-M_s	6	98.1%	0.020	0.960	3.4×10^{-3}
Top-M_s	8	99.7%	0.003	0.997	4.5×10^{-4}
Top-M_s	12	100.0%	0.000	1.000	$< 10^{-15}$
Top-M_s	24	100.0%	0.000	1.000	$< 10^{-15}$
Sampled	4	94.3%	0.060	0.870	5.6×10^{-3}
Sampled	8	99.7%	0.004	0.996	4.4×10^{-4}
Sampled	12	100.0%	0.000	1.000	$< 10^{-15}$

computational saving thus comes primarily from avoiding the full $K!$ search space while retaining most of $\mathcal{A}_\rho$. The observed ranking has anchor prior mass $m_0 = 0.9$, so a high captured mass alone can conceal errors in the alternative-ranking contribution. The gradient deterioration at small budgets shows why those alternatives must also be retained to preserve the robust update.

G.3 Training behavior

We train top-8, sampled-8, top-4, and top-2 variants with Qwen3-1.7B and DPO-allpairs, matching the exact KRPO configuration specified in Appendix F.4 ($\rho = 2$, $\gamma = 0.5$, $\eta = 1$, $m_0 = 0.9$) and using EMA geometry. The exact comparison uses the corresponding main-experiment checkpoint. Each configuration uses one training seed. We assess whether the agreement on fixed offline inputs also appears in the logged training statistics.

The evaluation-loss column is the held-out mean of each configuration's own objective in Equation 40:

$$\overline{L}^{(a)}_{\text{eval}} = \frac{1}{N_{\text{eval}}} \sum_{i=1}^{N_{\text{eval}}} \left[\ell^{(a)}_{\sigma_i} + \lambda_{\text{end}} c^{(a)}_{\sigma_i} \right] \quad . \tag{43}$$

Here a identifies the configuration, N_{eval} is the number of held-out records, and λ_{end} is the warm-up coefficient at the evaluated checkpoint. The correction $c^{(a)}$ uses that configuration's support; DPO has $c^{(a)} = 0$. Since truncation changes the correction, these entries evaluate different objectives. A lower value across support budgets therefore does not by itself indicate better performance or closer agreement with exact KRPO.

For each logged training step, observed-ranking mass is the record mean of $q^\star_t(\sigma \mid \sigma)$, and effective support is the record mean of $\exp H(q^\star_t)$. Their reported summaries average these statistics over logged steps. Trajectory correlation compares the stepwise mean corrections at matched logged steps; mean stepwise difference is the mean absolute difference between the two correction trajectories at those steps.

The mean exact robust correction is 0.023. Top-8 and sampled-8 each have a stepwise correction correlation of 0.97 with exact KRPO and a mean absolute stepwise difference of 0.005. Their mean observed-ranking mass is 0.862 versus 0.861 for exact, and their effective support is 1.92 versus 1.95 (Table 10). These statistics provide training-time evidence complementary to the fixed-input fidelity test.

More aggressive truncation reduces the mean robust correction from 0.023 for exact KRPO to 0.010 for top-4 and 0.005 for top-2. The top-2 evaluation loss reaches 0.448, the DPO value, but its mean correction remains nonzero, so it has not become exactly equivalent to DPO. Together with the offline gradient errors, this reduction shows that aggressive truncation weakens the robust correction. The results support numerical and optimization fidelity for sufficiently large local supports; they do not by themselves establish equivalence in generation quality.

Table 10: Qwen3-1.7B support-approximation training results. Effective support is $\exp H(q^\star)$, averaged over records and logged training steps. The evaluation column uses each configuration's own objective (Equation 43); its values across different supports are not evaluations under one common loss.

Configuration	Effective support	Eval objective
Exact KRPO	1.95	0.468
Top-8	1.92	0.467
Sampled-8	1.92	0.468
Top-4	1.62	0.455
Top-2	1.41	0.448
DPO-allpairs	—	0.448

Table 11: Support requirements on calibrated synthetic utilities. Set sizes are prompt means; M_{99} denotes the support budget needed for 99% adversarial-mass capture. Entries > 128 mean that the tested budget range was insufficient. These are offline comparisons, not longer-list training experiments.

		Fixed $\rho = 2$		Radius scaled by pair count		
K	$K!$	Mean $\lvert\mathcal{A}_\rho\rvert$	M_{99}	ρ	Mean $\lvert\mathcal{A}_\rho\rvert$	M_{99}
4	24	7.7	7	2.00	7.7	7
5	120	15.7	14	3.33	33.8	26
6	720	33.0	30	5.00	178.6	104
7	5,040	62.0	58	7.00	1,134	> 128
8	40,320	114.4	106	9.33	7,921	> 128

G.4 List length, radius, and computational cost

The required support budget depends on the size of the admitted neighborhood. On the real $K = 4$ utilities, increasing ρ from 0.5 to 4 raises the average ambiguity-set size from 2.29 to 16.33; the budgets needed to capture 99% of the adversarial mass rise from 3 to 14. Thus, a budget adequate at one radius need not remain adequate at another.

For longer lists, we use synthetic utilities calibrated to the $K = 4$ diagnostic. Each synthetic prompt draws its utility mean and scale from a real prompt and samples K Gaussian utilities. Noise is added to construct the observed ranking. Pairwise agreement is the fraction of candidate pairs whose order under the policy utilities agrees with the observed label ranking; the noise scale matches this statistic at $K = 4$ (0.761 synthetic versus 0.760 real). Absolute utility gaps are normalized to unit mean over the unordered pairs within each synthetic candidate set, using the same convention as Equation 3. Each setting uses 300 synthetic prompts. We compare a fixed absolute radius $\rho = 2$ with $\rho = 2\binom{K}{2}/6$, which scales with the number of candidate pairs.

At $K = 8$ and fixed $\rho = 2$, the average set contains 114.4 of the 40,320 possible rankings, and a budget of 106 captures 99% of the adversarial mass (Table 11). When the radius scales with pair count, the average set grows to 7,921 rankings and a budget of 128 is insufficient. For extending the same local-shift interpretation to longer lists, these results motivate keeping ρ as an absolute normalized reversal-cost budget and choosing M_s to cover the resulting neighborhood. Scaling ρ with pair count instead allows progressively broader ranking variation and substantially increases the required support.

Table 12 isolates the cost avoided by best-first construction. At $K = 8$, enumerating and scoring all permutations takes 179 ms per prompt, whereas constructing top-8 takes 0.120 ms, about 1,500 times less. This comparison measures construction with a fixed small budget: top-8 does not meet the 99% mass criterion at $K = 8$, for which Table 11 requires a budget of 106 at $\rho = 2$. The timing factor therefore does not describe a 99%-accurate approximation or end-to-end training speedup.

With at most M_s retained rankings, the scalar loss evaluations cost $\mathcal{O}(BM_sK^2)$ for minibatch size B, in addition to support-construction costs. The number of transformer evaluations is unchanged because all ranking losses reuse the response log-probabilities. In the $K = 4$ Qwen training com-

Table 12: Support-construction time on a single CPU thread, in median milliseconds per prompt. Exact construction enumerates all permutations and computes their weighted distances; top-M_s uses best-first expansion. These timings cover support construction only, with different retained support budgets; they are not comparisons at matched approximation error.

K	$K!$	Exact (ms)	Top-8 (ms)	Top-24 (ms)
4	24	0.040	0.018	0.016
6	720	2.060	0.037	0.071
8	40,320	179.000	0.120	0.113

parison, measured step times are 7.56 s for exact KRPO, 7.42 s for top-8, and 7.39 s for sampled-8. The end-to-end timing benefit is therefore modest in this setting; longer-list evidence here is limited to synthetic offline analysis.

H Ablating the Entropic Temperature through Its Limiting Cases

The geometry and radius ablations in Section 5.5 examine which rankings KRPO admits. Here we examine how their losses are aggregated, comparing finite-temperature KRPO with the two limiting cases of its entropic temperature on Qwen3-1.7B, using DPO-allpairs and LiPO as base objectives.

Temperature limits. The relevant hyperparameter is γ, the coefficient of the KL penalty in Equation 7. For a fixed policy, ambiguity set $\mathcal{A}_\rho(\sigma)$, and normalized prior p_0, the robust risk is

$$\mathcal{R}_\gamma(\theta;\sigma) = \gamma \log \sum_{\tilde\sigma \in \mathcal{A}_\rho(\sigma)} p_0(\tilde\sigma \mid \sigma) \exp\left(\frac{\ell_{\tilde\sigma}}{\gamma}\right). \tag{44}$$

Because the set is finite and p_0 is positive on every admitted ranking, its limits are

$$\lim_{\gamma\to 0^+} \mathcal{R}_\gamma(\theta;\sigma) = \max_{\tilde\sigma\in\mathcal{A}_\rho(\sigma)} \ell_{\tilde\sigma} \qquad \text{(hard maximum)}, \tag{45}$$

$$\lim_{\gamma\to\infty} \mathcal{R}_\gamma(\theta;\sigma) = \sum_{\tilde\sigma\in\mathcal{A}_\rho(\sigma)} p_0(\tilde\sigma \mid \sigma)\ell_{\tilde\sigma} \qquad \text{(prior average)}. \tag{46}$$

The first limit removes the KL penalty, concentrating the adversary on loss-maximizing rankings; the second returns it to p_0. Prior averaging therefore retains the original anchor mass and distance decay but removes the loss-dependent adversarial tilt. It is generally nonuniform. These are limits of γ, with ρ and the prior parameters fixed. The base objective ℓ_σ is a separate reference. With warm-up (Equation 40), each limit replaces $\mathcal{R}_\gamma$ in $(1-\lambda_t)\ell_\sigma + \lambda_t\mathcal{R}_\gamma$.

Table 13: Entropic-temperature ablation on Qwen3-1.7B. Entries are win rates (%, ↑); Mean is the five-benchmark macro average. Shading marks finite-temperature KRPO; **bold** marks each column's best value within each base-objective block, including ties.

Variant	γ	Arena	Alpaca	HH	MT-B	TL;DR	Mean
DPO-allpairs							
Base	–	**18.9**	22.5	78.0	59.4	41.3	44.0
Prior average	$\to\infty$	18.8	23.7	78.0	60.0	**42.7**	44.6
Hard maximum	$\to 0^+$	18.1	21.6	76.3	**63.1**	41.2	44.1
KRPO (ours)	$0<\gamma<\infty$	**18.9**	**23.9**	**78.6**	60.6	42.6	**44.9**
LiPO							
Base	–	18.5	**24.0**	75.4	56.9	42.8	43.5
Prior average	$\to\infty$	18.8	22.9	**77.9**	59.7	39.6	43.8
Hard maximum	$\to 0^+$	16.7	23.7	76.1	61.9	44.1	44.5
KRPO (ours)	$0<\gamma<\infty$	**19.6**	23.2	77.6	**64.4**	**44.2**	**45.8**

Results. Finite-temperature KRPO achieves the highest observed macro average for both base objectives (Table 13), while the relative performance of the two limits depends on the base loss. For DPO-allpairs, prior averaging reaches 44.6, close to KRPO's 44.9 and above the hard maximum's 44.1. Averaging over the prior therefore already improves on the base score of 44.0, with a smaller additional gain from adversarial tilting. For LiPO, the ordering of the limits reverses: the hard maximum reaches 44.5 versus 43.8 for prior averaging, while KRPO attains 45.8. Neither limit is uniformly preferable across the two objectives.

Individual benchmarks exhibit trade-offs: for example, the DPO-allpairs hard maximum outperforms KRPO on MT-Bench (63.1 versus 60.6). The comparison supports finite-temperature aggregation in the two tested settings, with a larger observed advantage over both limits for LiPO. This conclusion is descriptive and concerns macro averages; the results do not establish statistical significance or an optimal finite value of γ.

I Comparison with Concurrent Robust Listwise Optimization

We complement the sample-level comparisons in Section 5.3 with Robust PL, the concurrent ranking-level method of Wu et al. (2026). Both methods account for uncertainty in the ranking of a fixed candidate list, but use different ambiguity models and robust objectives. We compare them with the same PL base objective on Qwen3-1.7B and Llama-3.1-8B, reporting the five generation benchmarks used in the main experiments.

Robust PL. The method places a total-variation ball around the point mass at the observed ranking. Its robust objective has the exact form

$$\mathcal{L}_{\text{Robust PL}}(\sigma) = (1-\rho_{\text{TV}})\mathcal{L}_{\text{PL}}(\sigma) + \rho_{\text{TV}} \max_{\tilde{\sigma}\in\mathfrak{S}_K} \mathcal{L}_{\text{PL}}(\tilde{\sigma}), \tag{47}$$

where $\rho_{\text{TV}} \in [0,1]$ bounds the probability mass that can move away from the observed ranking. The maximizing ranking sorts the current implicit scores in ascending order. We report $\rho_{\text{TV}} \in \{0.05, 0.10\}$, the two values included in their main offline comparison. This radius differs from KRPO's ρ, which bounds the weighted Kendall distance of admitted rankings. Robust PL mixes the nominal loss with a maximum over all permutations, whereas KRPO uses an entropic adversary within a selective neighborhood.

Table 14: Comparison with Robust PL on the main generation benchmarks. PL rows report absolute win rates (%, ↑); other rows report changes from PL in percentage points. Mean is the five-benchmark macro average. PL and KRPO entries match Table 1. **Bold** marks the largest mean change in each model block.

Method	ρ_{TV}	Arena	Alpaca	HH	MT-B	TL;DR	Mean
Qwen3-1.7B							
PL	–	16.7	23.4	76.0	56.9	40.6	42.7
Robust PL Δ	0.10	+0.3	-1.5	-1.4	-0.6	+0.5	-0.6
Robust PL Δ	0.05	+0.8	-1.3	+0.5	-0.6	+1.3	+0.1
KRPO (ours) Δ	–	+2.0	-1.0	-0.8	+1.2	+1.3	**+0.5**
Llama-3.1-8B							
PL	–	34.5	34.3	89.4	79.1	26.3	52.7
Robust PL Δ	0.10	+0.8	+0.4	-2.9	-1.9	-4.7	-1.6
Robust PL Δ	0.05	+0.1	-0.9	-4.7	+3.1	-2.6	-1.0
KRPO (ours) Δ	–	-0.3	+1.6	-0.5	+4.0	-2.7	**+0.4**

Results. KRPO achieves the largest observed mean improvement over PL on both models (Table 14). On Qwen, Robust PL changes the mean by -0.6 points at $\rho_{\text{TV}} = 0.10$ and $+0.1$ at 0.05, while KRPO improves it by $+0.5$. On Llama, the corresponding Robust PL changes are -1.6 and -1.0, while KRPO yields $+0.4$. Relative to the better of the two tested Robust PL settings, KRPO improves the reported mean by 0.4 and 1.4 percentage points on Qwen and Llama, respectively.

The comparison favors KRPO at the level of the macro average, with trade-offs across individual benchmarks. Robust PL gives the largest gains on Qwen HH ($+0.5$) and Llama Arena ($+0.8$),

while KRPO leads on MT-Bench for both models. These results extend the empirical comparison to another objective that acts directly on ranking labels. They concern our generation-based evaluation setting; the original work also studies synthetic ranking-label corruption and online alignment. We interpret the differences descriptively, without claiming statistical significance or isolating the separate effects of geometry and loss aggregation.

J LIMITATIONS

Deployment preferences. Our controlled shift evaluation uses UltraFeedback's annotation dimensions. Validation in broader applications requires judgments of the same responses from representative users under relevant criteria, sometimes with domain expertise.

Longer candidate lists. Our training experiments use four-response rankings, while evidence for longer lists remains synthetic. Evaluating longer lists end to end requires more response generation and preference annotation, together with efficient approximations to the larger ranking space.