Rank-Aware Differentially Private Release of Listwise Preferences for LLM Alignment

Junwei Chen
Institute of Science Tokyo
214wadechen@gmail.com

Manjiang Yu
University of Queensland
manjiang.yu@uq.edu.au

Pengpeng Qiao
Institute of Science Tokyo
peng2qiao@gmail.com

Yang Cao
Institute of Science Tokyo
cao@c.titech.ac.jp

Abstract

Listwise preference feedback offers richer supervision for large language model (LLM) alignment than pairwise comparisons, but a full ranking also reveals more sensitive user preferences. Existing privacy-preserving alignment methods focus mainly on pairwise feedback, while listwise alignment under local differential privacy (LDP) remains unexplored. Extending LDP to this listwise domain raises a critical challenge: the high sensitivity of the full ranking space demands excessive noise, neutralizing the benefits of listwise supervision. To address this challenge, we analyze the interaction between supervision granularity and privacy noise in downstream training, and introduce a rank-aware exponential mechanism that privatizes listwise preference data into a low-sensitivity granularity suitable for downstream alignment. The mechanism leverages ranking information to sample a fixed-size binary partition, concentrating the noise near borderline items instead of perturbing all items uniformly as randomized response (RR) does. Empirical evaluations on downstream LLM alignment tasks show that our mechanism consistently outperforms existing LDP baselines at matched privacy budgets. The code is available at `https://github.com/tdsai-lab/RAEM`.

1 Introduction

Aligning large language models (LLMs) with human preferences is a central step in post-training. Existing alignment methods often rely on pairwise feedback, such as RLHF [18] and DPO [19], where one response is preferred over another under the same prompt. While effective, pairwise supervision only captures local comparisons and discards the richer relative structure among multiple candidate responses. This motivates listwise preference alignment, such as PPA [29], LiPO [15], where several responses are evaluated jointly to provide a more informative supervision signal.

Preference signals are inherently privacy-sensitive, as they may expose a user's values, beliefs, tastes, and behavioral patterns [4; 13]. Listwise preference data further amplifies these risks by capturing richer relative judgments over multiple candidate responses than pairwise feedback. As shown in Figure 1, if the model trainer is not fully trusted, directly releasing raw listwise feedback can cause sensitive preference leakage or user profiling. Even when the trainer is trusted, preference information may still leak through the trained model. Recent work on membership inference attacks (MIA) in alignment settings suggests that preference data can be indirectly exposed after training [10; 16]. As preference alignment is increasingly used in personalized scenarios such as recommendation systems [14; 23], these privacy risks become more salient.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

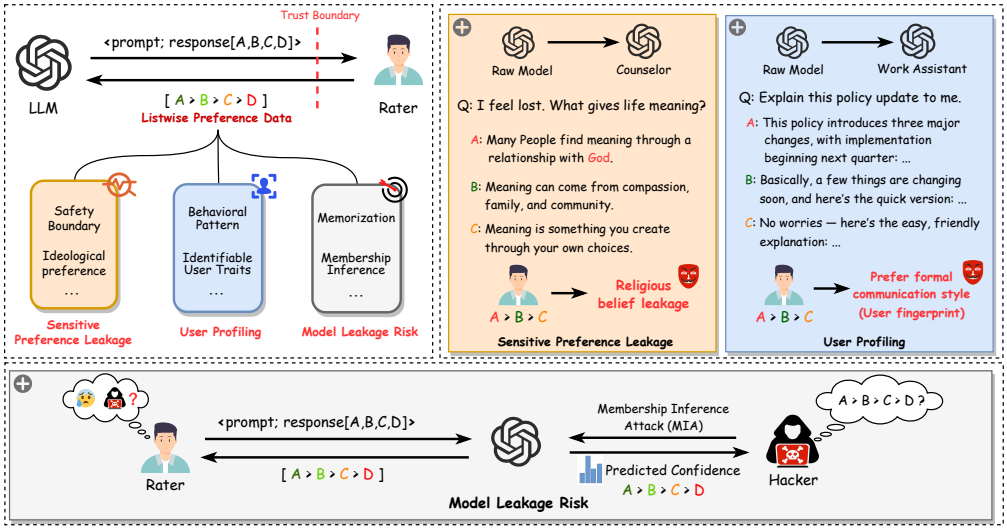


Figure 1: Threat model for privacy-preserving listwise preference alignment.

Existing privacy-preserving alignment methods mainly focus on pairwise preference data, with LDP approaches mainly using randomized response (RR) [3], where privacy is defined over binary comparisons. But listwise preference data is different: it contains scores, rankings, or group-level preferences over multiple candidates under the same prompt. This requires privacy to be formulated over the whole listwise feedback object, with sensitivity and noise calibrated to list-level changes. As a result, its sensitivity under local differential privacy can be much larger than that of a binary pairwise label. Directly privatizing the full score or ranking pattern requires substantial noise, which can destroy the ordering information that makes listwise supervision useful in the first place. Thus, a key challenge for private listwise alignment is how to inject privacy noise while preserving enough ranking information for downstream training.

In this paper, we study listwise preference alignment under LDP. Our key idea is to use the rater's local ranking only before privatization, inside a client-side sampling mechanism, and release only a low-sensitivity binary partition to the server. This allows the mechanism to exploit ordinal information locally while exposing only the signal granularity required for downstream training after privatization. Our contributions are as follows:

- We initiate the study of private alignment under listwise feedback in the LDP setting, and introduce an item-level adjacency notion suited to ranking data.
- We identify release granularity as a key factor in the privacy–utility trade-off for listwise feedback, motivating a design that exploits ordinal information locally while releasing only low-sensitivity supervision for downstream training.
- We propose a rank-aware exponential mechanism that releases a private binary partition, assigning rank-dependent privacy noise instead of perturbing all binary labels uniformly as in RR.
- We prove the privacy guarantee and a sensitivity-optimality result for the proposed rank utility, and empirically show consistent gains over RR under matched privacy budgets.

2 Related Works

2.1 Preference Alignment under Listwise Feedback

Recent work has widely studied preference alignment with multiple responses for the same prompt, using scores or rankings instead of a single pairwise comparison. RRHF [28], SLiC [30], and PRO [20] mainly exploit ordinal rankings by encouraging the model to assign higher likelihoods to more preferred responses. LiPO-λ [15] and PPA [29] use finer-grained listwise signals, such as LambdaRank-style weighting [2] and NDCG-based listwise utilities [22], to reflect ranking quality and preference strength. In contrast, KTO [9] only requires unary desirable or undesirable labels, making it convenient for multi-response data without constructing explicit preference pairs, but

it discards ordering information within the positive and negative groups. Our work uses listwise rankings locally before privatization, while releasing only unary labels for downstream KTO training.

2.2 Differentially Private Preference Alignment

Differential privacy has been used to reduce leakage of individual preference during alignment [25]. In the central-DP setting, Yu et al. [27] generate differentially private synthetic instructions to reduce exposure of real user feedback, while Gao et al. [11] propose DPPrefSyn, which privately learns a preference model from sensitive pairwise feedback and uses it with public prompts to synthesize differentially private preference data for alignment. Wu et al. [24] apply DP mechanisms such as DP-SGD across alignment stages to mitigate training-data leakage and membership inference risks. In the local-DP setting, Teku et al. [21] propose PROPS, which protects users' preference labels under local privacy and uses progressively aligned models to generate additional feedback for later training stages. Chowdhury et al. [5] study robust objectives for learning from randomized-response-perturbed preferences. However, these works focus on pairwise feedback and do not address locally private listwise release, where the mechanism must account for the structure and sensitivity of the entire ranking while exposing only low-sensitivity supervision for training.

3 Preliminaries

3.1 Exponential Mechanism

Definition 3.1 (Exponential mechanism)**.** Let $u : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ be a utility function with sensitivity

$$\Delta_u = \sup_{x,x' \in \mathcal{X}} \sup_{z \in \mathcal{Z}} |u(x,z) - u(x',z)|.$$

The exponential mechanism samples $z \in \mathcal{Z}$ with probability

$$\Pr[\mathcal{M}_u(x) = z] = \frac{\exp\left(\frac{\varepsilon u(x,z)}{2\Delta_u}\right)}{\sum_{z' \in \mathcal{Z}} \exp\left(\frac{\varepsilon u(x,z')}{2\Delta_u}\right)}.$$

The exponential mechanism is known to satisfy ε- differential privacy.

3.2 Kahneman–Tversky Optimization

Kahneman–Tversky Optimization (KTO) [9] learns from *unary* desirability labels rather than pairwise preferences. For a prompt–response pair (x, y), let $z \in \{0, 1\}$ indicate whether y is desirable (z=1) or undesirable (z=0). Given a policy π_θ and a fixed reference policy π_{ref}, KTO is built on two quantities: the policy–reference log-ratio r_θ and a detached KL baseline c_θ,

$$r_\theta(x,y) \;:=\; \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}, \qquad c_\theta \;:=\; \text{sg}\Big[\, \mathbb{E}_{(x',y') \sim \mathcal{D}}\, r_\theta(x',y') \,\Big], \tag{1}$$

where $\text{sg}[\cdot]$ is the stop-gradient operator. The per-example loss interpolates between two utility terms gated by z:

$$\ell_{\text{KTO}}(\theta; x, y, z) \;=\; \underbrace{z\, \lambda_D \big[1 - \sigma(\beta(r_\theta - c_\theta))\big]}_{\text{desirable } (z=1)} \;+\; \underbrace{(1-z)\, \lambda_U \big[1 - \sigma(\beta(c_\theta - r_\theta))\big]}_{\text{undesirable } (z=0)}, \tag{2}$$

where $\beta > 0$ sets the log-ratio scale and $\lambda_D, \lambda_U > 0$ weight the two regimes. Given a unary dataset $\mathcal{D} = \{(x_n, y_n, z_n)\}_{n=1}^N$, KTO minimizes the empirical risk

$$\mathcal{L}_{\text{KTO}}(\theta) \;=\; \frac{1}{N} \sum_{n=1}^{N} \ell_{\text{KTO}}(\theta; x_n, y_n, z_n). \tag{3}$$

In our setting, the clean label z is *not* directly observed. It is derived from a locally privatized signal.

4 Methodology

4.1 Item-Level Local Differential Privacy on Rankings

Before introducing our methodology, we define local differential privacy directly over the ranking space. The atomic neighboring relation is an item-level relocation.

Definition 4.1 (Item-level relocation adjacency)**.** Two rankings $\phi, \phi' \in S_m$ are item-level adjacent, denoted by

$$\phi \sim_{\text{item}} \phi',$$

if ϕ' can be obtained from ϕ by removing one item from its current position and inserting it into another position, while preserving the relative order of all remaining items.

Definition 4.2 (Item-level local differential privacy on rankings)**.** Let $Q : S_m \to \mathcal{O}$ be a randomized mechanism over rankings. We say that Q satisfies ε-item-level local differential privacy if for all item-level adjacent rankings $\phi, \phi' \in S_m$ and all outputs $o \in \mathcal{O}$,

$$\frac{Q(o \mid \phi)}{Q(o \mid \phi')} \leq e^{\varepsilon}.$$

The guarantee in Definition 4.2 is a per-list item-level LDP. Intuitively, it guarantees that the exact position of any individual item in the ranking cannot be inferred by an adversary. The rank-level and user-level extension are discussed in Appendix A.

4.2 Release Granularity and Privacy Noise

In this subsection, we study the granularity of listwise preference data in alignment scenarios and how different granularity levels are affected by differential privacy noise, with the goal of choosing a suitable release granularity and privacy mechanism under local privacy constraints.

Listwise preference data can be released at several granularities: cardinal scores provide the finest feedback, ordinal rankings keep only the relative order, and binary top-B labels retain only whether each candidate belongs to the preferred set. These choices preserve different amounts of preference information and induce different local sensitivities under private release.

Supervision granularity	Representative method	Sensitivity	Noise scale
Cardinal score vector	PPA [29]	m^2	$O(m^2/\varepsilon)$
Rank vector	SLiC [30]	$2(m-1)$	$O(m/\varepsilon)$
Binary labels	KTO [9]	2	$O(1/\varepsilon)$

Table 1: Data release granularity, representative alignment methods, and noise scale. Sensitivity bounds and noise scale are proved in Appendix B.1.

Table 1 shows that finer-grained releases incur higher privacy cost. While cardinal scores and full rankings preserve more pre-privatization information, their larger sensitivities require stronger perturbation, especially for large m or small ε. This affects downstream training: PPA and SLiC rely on reliable relative preferences across the candidate list, whereas KTO only needs unary desirable/undesirable labels. Thus, DP noise in fine-grained releases can more severely degrade supervision quality and optimization stability.

However, simply operating on coarse-grained information also risks losing high-quality preference signals: binary top-B labels collapse the ordering within positive and negative groups, making it difficult to distinguish the best responses from merely acceptable ones. This motivates our design: we use local rankings only for client-side sampling and release a low-sensitivity binary top-B partition, providing the server with unary labels while retaining ordinal information before privatization.

4.3 Rank-Aware Exponential Mechanism for Binary Release

Motivated by Section 4.2, we use the exponential mechanism in Definition 3.1 to privately sample low-sensitivity binary top-B partitions while favoring outputs that preserve the local ranking.

Problem Setup For each prompt x, the client observes candidate responses $Y(x) = \{y_1, \ldots, y_m\}$ and local supervision scores $\mathbf{s} = (s_1, \ldots, s_m)$. The scores induce a clean ranking $\phi = \mathrm{Sort}(\mathbf{s})$, where $r_i(\phi)$ denotes the rank of response y_i, with smaller ranks indicating stronger preference.

The client releases a fixed-size binary vector from

$$\mathcal{Z}_B := \{z \in \{0,1\}^m : \|z\|_0 = B\}.$$

Given ϕ, the release mechanism samples

$$Z \mid \phi \sim Q(\cdot \mid \phi), \qquad Z \in \mathcal{Z}_B,$$

and outputs its realization $\widetilde{z}$. The server observes only $(x, Y(x), \widetilde{z})$, where $\widetilde{z}_i = 1$ marks y_i as desirable and $\widetilde{z}_i = 0$ marks it as undesirable.

Rank-aware utility. We instantiate the release utility with a linear boundary score. Given a clean ranking ϕ, let $r_i(\phi)$ be the rank of response y_i, with smaller rank indicating stronger preference. The linear score assigns each item a signed distance to the top-B boundary:

$$w_i(\phi) = B + \frac{1}{2} - r_i(\phi), \qquad U(\phi, z) = \sum_{i=1}^{m} z_i w_i(\phi) = \sum_{i=1}^{m} z_i \left(B + \frac{1}{2} - r_i(\phi) \right). \tag{4}$$

Thus, items above the boundary receive positive weights and items below it receive negative weights. Maximizing $U(\phi, z)$ favors partitions that select higher-ranked responses while penalizing selections from lower-ranked positions. Under item-level relocation adjacency, this utility has sensitivity

$$\Delta := \max_{\phi \sim \phi'} \max_{z \in \mathcal{Z}_B} |U(\phi, z) - U(\phi', z)| = \max\{B, m - B\},$$

This sensitivity is proved in Theorem B.1. Section 5.2 shows that this linear boundary score is sensitivity-optimal among monotone rank scores under a normalized rank-span criterion.

Private release. The client samples a binary partition by exponential mechanism

$$Q(z \mid \phi) = \frac{\exp(\eta U(\phi, z))}{\sum_{z' \in \mathcal{Z}_B} \exp(\eta U(\phi, z'))}, \qquad \eta = \frac{\varepsilon}{\Delta} = \frac{\varepsilon}{\max\{B, m - B\}},$$

and releases the sampled $\tilde{z} \sim Q(\cdot \mid \phi)$ as the private top-B selection. By Theorem B.2 in Appendix B.3, this exponential mechanismwith utility U and sensitivity Δ satisfies ε-local differential privacy under the item-level ranking adjacency.

Server-side training. After release, the server observes only $(x, Y(x), \widetilde{z})$ and expands each privatized list into unary preference examples

$$\mathcal{D}_{\mathrm{priv}}(x) = \{(x, y_i, \widetilde{z}_i)\}_{i=1}^{m}, \qquad \widetilde{z}_i = 1 \text{ desirable}, \quad \widetilde{z}_i = 0 \text{ undesirable}.$$

The policy is then optimized with the KTO objective in Section 3.2 using $\mathcal{D}_{\mathrm{priv}}$. The server never observes the local scores $\mathbf{s}$, the clean ranking ϕ, or the clean top-B labels $z(\phi)$.

5 Flip Rate Analysis

The flip rate directly affects downstream KTO training because the released binary partition is used as unary supervision. A flipped label turns a positive KTO update into a negative one, or vice versa, thereby reversing the training direction of that sample. Hence, the location of privacy-induced flips matters: top-ranked positives and bottom-ranked negatives provide the most reliable unary labels, while samples near the top-B boundary are more ambiguous. We therefore prefer a release mechanism with low flip rates at extreme ranks and higher flip rates near the boundary.

5.1 Per-Rank Flip Rates of Rank-Aware Exponential Mechanism

Let $q_i(\phi) := \Pr[\tilde{z}_i = 1 \mid \phi]$ denote the marginal probability that y_i is released as positive. Under the rank-aware exponential mechanism,

$$q_i(\phi) = \frac{e^{\eta w_i(\phi)}\, \sigma_{B-1}\big(e^{\eta w_{-i}(\phi)}\big)}{\sigma_B\big(e^{\eta w(\phi)}\big)}, \qquad \rho_i(\phi) := \Pr[\tilde{z}_i \neq z_i(\phi) \mid \phi] = \begin{cases} 1 - q_i(\phi), & r_i(\phi) \leq B, \\ q_i(\phi), & r_i(\phi) > B, \end{cases}$$

where σ_k is the k-th elementary symmetric polynomial, $e^{\eta w(\phi)}$ is applied coordinatewise, and $w_{-i}(\phi)$ drops coordinate i. The flip rate is rank-dependent: extreme ranks remain stable while flips concentrate near the top-B boundary, with derivation and monotonicity deferred to Appendix B.4.

5.2 Sensitivity-Optimality of the Linear Score

We evaluate a monotone rank score $w_1 \geq \cdots \geq w_m$ by its *calibrated rank span* S_w/Δ_w, where $S_w = w_1 - w_m$ and Δ_w is the item-level relocation sensitivity of the induced rank-based utility U_w. Since the exponential mechanism sets $\eta = \varepsilon/\Delta_w$, the top-to-bottom logit gap is

$$\eta S_w = \varepsilon \cdot \frac{S_w}{\Delta_w},$$

so maximizing S_w/Δ_w maximizes extreme-rank separation at any fixed privacy budget.

Theorem 5.1 (Optimal calibrated rank span)**.** *For any monotone rank score $w_1 \geq \cdots \geq w_m$,*

$$\frac{S_w}{\Delta_w} \;\leq\; \frac{m-1}{\max\{B, m-B\}},$$

with equality iff $w_s = \alpha - \beta s$ for some $\beta > 0$.

Thus the bound is tight only for linear scores, which therefore yield the most stable extreme releases under item-level relocation adjacency. The proof is deferred to Appendix B.5.

5.3 Comparison with RR

We compare the rank-aware exponential mechanism with RR, the canonical LDP baseline for binary labels. RR privatizes the clean top-B label vector

$$z_i(\phi) = \mathbf{1}[r_i(\phi) \leq B], \qquad z(\phi) = (z_1(\phi), \ldots, z_m(\phi)).$$

RR is originally designed for privatizing single binary labels. In our setting, however, each prompt produces a vector of binary labels over multiple candidate responses. Therefore, we calibrate RR to the sensitivity of the full clean top-B vector. Under item-level relocation adjacency in Definition 4.1, this vector has Hamming sensitivity

$$\Delta_H := \sup_{\phi \sim_{\text{item}} \phi'} \|z(\phi) - z(\phi')\|_0 = 2.$$

Thus, applying coordinate-wise RR with per-coordinate privacy parameter $\varepsilon_0 = \varepsilon/\Delta_H = \varepsilon/2$ yields ε-LDP for the full released vector. The calibration proof is deferred to Appendix B.6.

Let $\tilde{z}_{\mathsf{RR}}$ denote the RR release. By definition, each coordinate flips independently with probability

$$\Pr[(\tilde{z}_{\mathsf{RR}})_i = z_i(\phi)] = \frac{e^{\varepsilon/2}}{1 + e^{\varepsilon/2}}, \qquad \rho_i^{\mathsf{RR}}(\phi) := \Pr[(\tilde{z}_{\mathsf{RR}})_i \neq z_i(\phi)] = \frac{1}{1 + e^{\varepsilon/2}}.$$

Thus, RR has a rank-independent flip rate, perturbing all coordinates equally. In contrast, our method induces rank-dependent noise, preserving high-quality ranks while concentrating flips near the top-B boundary. Figure 2 further illustrates how privacy calibration changes the noise profile.

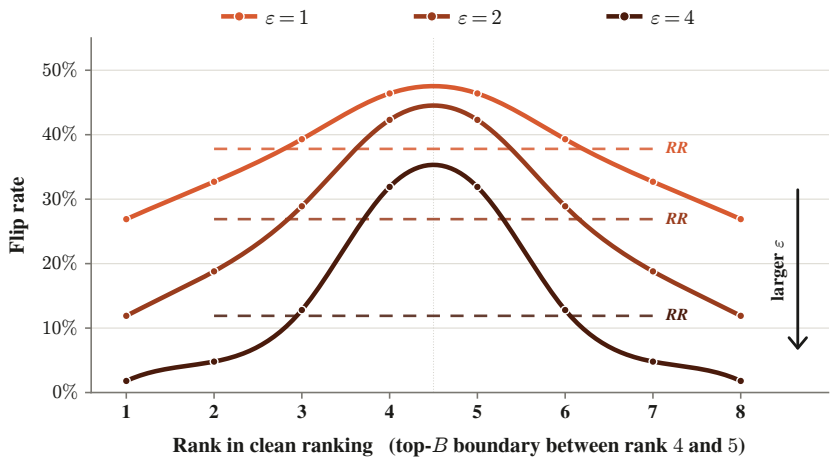


Figure 2: Simulation of per-rank flip rates for our method (solid curves) and RR (dashed lines) with $m = 8$ and $B = 4$, for different values of ε.

6 Experiments

6.1 Setting

Dataset. We use *ListUltraFeedback*, an offline listwise preference dataset constructed in PPA [29]. It is built from *UltraFeedback* [6] by combining *UltraFeedback* responses with additional SimPO-Llama3-8B[1] responses and scoring them with ArmoRM[2] to obtain listwise preferences. The dataset contains 59.9K training examples and 1,968 held-out examples.

Baselines. We compare with non-private alignment methods as utility references, including PPA, SLiC [30], and KTO [9]. These methods use clean preference supervision without privacy constraints and therefore quantify the utility loss introduced by private release. As the main private baseline, we use RR followed by KTO. We select PPA over LiPO [15] based on its stronger performance in the PPA paper's evaluation, and omit PROPS [21] because its comparison-level privacy unit, binary-comparison supervision, and additional training stages prevent a direct comparison under our ranking-level adjacency and unary-label KTO setting.

Training Details. We use Qwen2-0.5B [26] and Mistral-7B [12] as foundation models. For each model, we first perform supervised fine-tuning on UltraChat-200K [7] to obtain the SFT checkpoint, and then initialize all alignment methods from this same checkpoint on *ListUltraFeedback*. For both Qwen2-0.5B and Mistral-7B, we train all alignment methods using the AdamW optimizer [17] on 4 NVIDIA H100 80GB GPUs. Further training details are provided in Appendix C.1.

Evaluation. We report win rates of aligned models against the corresponding SFT model. Since ArmoRM is used to construct *ListUltraFeedback*, evaluation by only ArmoRM may overestimate performance due to reward-model coupling. To mitigate this, we use two complementary evaluation pipelines: an independent pair-preference reward model and general conversational benchmarks. In the proxy-model pipeline, we compute win rates on the held-out *ListUltraFeedback* split using ArmoRM as a scoring reward model and the RLHFlow pair-preference reward model[3] as a pairwise judge. In the general-benchmark pipeline, we evaluate conversational quality on AlpacaEval [8] and MT-Bench [31], using GPT-4 Turbo [1] as the judge following the original setup. To reduce A/B order bias, we randomly shuffle the two candidate responses before judging and map the decision back to the original model identity. Further evaluation details are provided in Appendix C.2.

[1] `https://huggingface.co/princeton-nlp/Llama-3-Instruct-8B-SimPO`
[2] `https://huggingface.co/RLHFlow/ArmoRM-Llama3-8B-v0.1`
[3] `https://huggingface.co/RLHFlow/pair-preference-model-LLaMA3-8B`

Method	Proxy Model		General Benchmark		Avg.
	Pair-Preference	Scoring	AlpacaEval	MT-Bench	
Qwen2-0.5B					
Non-private (ε=∞)					
PPA	66.3	61.3	61.6	56.4	61.4
SLiC	65.7	60.7	61.0	55.8	60.8
KTO	67.8	55.9	59.2	60.1	60.8
Private, $\varepsilon = 0.5$					
RR+KTO	52.9	38.6	48.0	52.5	48.0
RAEM+KTO	59.2	43.6	51.8	50.1	**51.2**
Private, $\varepsilon = 1$					
RR+KTO	57.9	43.7	50.7	52.0	51.1
RAEM+KTO	61.2	49.1	50.6	50.6	**52.9**
Private, $\varepsilon = 3$					
RR+KTO	64.4	51.5	56.5	56.0	57.1
RAEM+KTO	66.4	54.6	57.8	59.7	**59.6**
Mistral-7B					
Non-private (ε=∞)					
PPA	91.5	93.1	90.2	73.8	87.2
SLiC	92.4	92.9	89.8	72.8	87.0
KTO	92.5	92.7	90.8	72.3	87.1
Private, $\varepsilon = 0.5$					
RR+KTO	88.8	88.4	85.9	73.0	84.0
RAEM+KTO	90.4	90.6	87.0	72.8	**85.2**
Private, $\varepsilon = 1$					
RR+KTO	89.7	90.9	86.4	71.8	84.7
RAEM+KTO	90.5	91.0	87.5	72.1	**85.3**
Private, $\varepsilon = 3$					
RR+KTO	91.4	92.1	87.4	72.1	85.8
RAEM+KTO	91.5	92.4	90.4	72.4	**86.7**

Table 2: **Main results** with listwise preference alignment (m=8, B=4). **Avg.** denotes the simple mean of the four reported metrics. Shaded rows (RAEM+KTO) denote our method.

6.2 Main results

Table 2 compares our method with non-private alignment references and the private RR baseline. Our method achieves the best average performance among private methods for every tested privacy budget and both backbone models. The improvement is more pronounced for Qwen2-0.5B, while on Mistral-7B the same trend holds despite the stronger SFT and non-private alignment baselines.

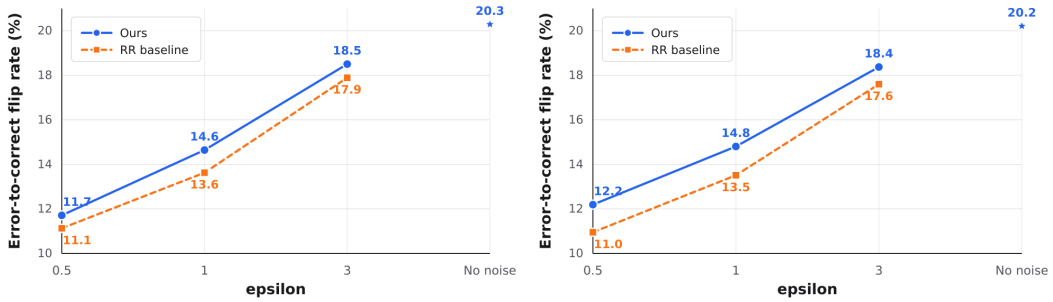
Both our method and RR train KTO on noisy unary labels, but RR assigns a rank-independent flip probability, whereas our mechanism uses the local ranking to concentrate noise near the top-B boundary and preserve extreme ranks with lower flip rates. The consistent gains show that, for private KTO training, where the noise is placed is important under the same privacy budget. Allocating more flips to boundary-near examples is more effective than perturbing all binary labels uniformly. Additional AlpacaEval results are reported in Appendix E.

6.3 Error-to-correct Flip Rate

We evaluate error-to-correct flip rate (E2C), the fraction of SFT-induced wrong pairwise preferences corrected after alignment. Pairwise preferences are induced from response log-probabilities:

$$y_i \succ_\theta y_j \quad \Longleftrightarrow \quad \log p_\theta(y_i \mid x) > \log p_\theta(y_j \mid x).$$

Higher E2C indicates more corrected initial preference errors. Figure 3 shows that our method consistently achieves higher E2C than RR and stays closer to the no-noise reference, suggesting that the rank-aware exponential mechanism preserves stronger correction signals.



(a) Qwen2-0.5B **(b)** Mistral-7B

Figure 3: Error-to-correct flip rate (E2C) of our method and the RR baseline on both models. The starred point marks the no-noise reference.

6.4 Effect of List Size

We analyze whether full-list private release preserves high-quality training signals better than selecting a small candidate subset before release, using Qwen2-0.5B as the backbone model.

Figure 4 illustrates the role of list size under private release. In the ideal oracle setting, the learner keeps the clean top-ranked and bottom-ranked responses, which provide the strongest KTO supervision (Figure 4(a)). Under this selection rule, increasing the candidate list size is not always beneficial: additional boundary-near, low-confidence responses introduce more released labels and hence more privacy noise, without necessarily improving the best positive or worst negative examples. However, this oracle selection is not a valid private procedure, because identifying the clean extremes already reveals information from the user's ranking.

In a strict privacy-preserving setting, a valid small-subset method must select candidates before observing the clean ranking. Without prior quality information, this reduces to random sampling, which may miss high-quality extremes. As shown by the random-subset curve in Figure 4(a), larger subsets increase the chance of including strong positives and negatives, but the sampling risk remains. Our method avoids this risk by applying private release to the full list.

Larger listwise preference sets are more likely to contain high-quality extreme responses. Meanwhile, Figure 4(b) shows that our method keeps the flip rates of the top and bottom ranks stable as m increases, with additional privacy noise absorbed by boundary-near responses. Thus, in the private setting, increasing m improves the available listwise supervision while our mechanism prevents the most valuable examples from becoming less reliable under LDP.

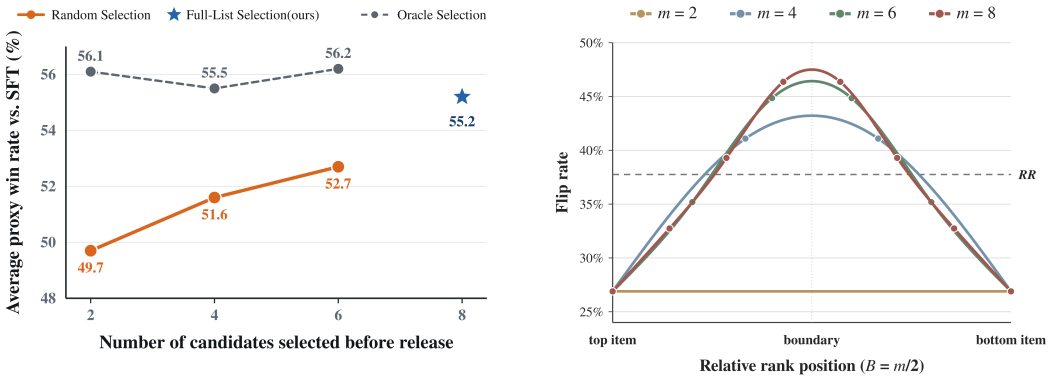
Additional experiments on DPO compatibility, training-seed variability, and mechanism ablations are reported in Appendix D.

7 Limitation

Our analysis focuses on item-level local differential privacy in Definition 4.2. Under item-level relocation adjacency, the guarantee protects the position of any individual response: two rankings that differ only by relocating one item cannot be easily distinguished from the released signal. This captures the main leakage risk in a listwise feedback setting. However, optimizing for these stricter notions is outside the scope of this work. Appendix A defines stricter rank-level and user-level variants, which protect an entire ranking or all rankings from the same user. Optimizing for these stronger notions is left for future work, since direct extensions require calibration to linear composition over multiple released lists, causing a large privacy-budget cost, especially at the user level.

8 Conclusion

We studied listwise preference alignment under local differential privacy. We showed that releasing scores or full rankings incurs high sensitivity, and proposed a rank-aware exponential mechanism



(a) Candidate selection before private release

(b) Flip-rate profile as m varies

Figure 4: Candidate selection and flip-rate allocation at $\varepsilon = 1$. (a) Average proxy win rate against the corresponding SFT model after applying our rank-aware private release and KTO training to Qwen2-0.5B under different candidate-selection rules. Oracle top-bottom selection retains the clean best and worst responses and randomly fills the remaining slots. Random subset selection samples all k candidates before release, whereas our default method retains all candidates. (b) Flip-rate profiles of the rank-aware exponential mechanism concentrate near the top-B boundary as m increases.

that uses rankings locally while releasing only binary top-B labels. Theoretical and empirical results show that our method better preserves preference signals than RR under matched privacy budgets, highlighting that private alignment depends not only on noise scale but also on where errors occur.

Acknowledgments and Disclosure of Funding

This work was supported by JST PRESTO (Grant No. JPMJPR23P5), JST CREST (Grant No. JPMJCR21M2), JSPS KAKENHI (Grant No. 26K21227) and JST NEXUS (Grant No. JPMJNX25C4).

References

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

[2] Christopher Burges, Robert Ragno, and Quoc Le. Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19. MIT Press, 2006. URL `https://proceedings.neurips.cc/paper_files/paper/2006/file/af44c4c56f385c43f2529f9b1b018f6a-Paper.pdf`.

[3] Arijit Chaudhuri and Rahul Mukerjee. *Randomized response: Theory and techniques*. Routledge, 2020.

[4] Junwei Chen, Pengpeng Qiao, Wei Yang Bryan Lim, Manjiang Yu, and Yang Cao. Privacy-preserving llm preference alignment: A systematic survey. *TechRxiv*, 2026(0130), 2026. doi: 10.36227/techrxiv.176978434.43544529/v1. URL `https://www.techrxiv.org/doi/abs/10.36227/techrxiv.176978434.43544529/v1`.

[5] Sayak Ray Chowdhury, Xingyu Zhou, and Nagarajan Natarajan. Differentially private reward estimation with preference feedback. In *International Conference on Artificial Intelligence and Statistics*, pages 4843–4851. PMLR, 2024.

[6] 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.

[7] Ning Ding, Bokai Chen, Yulin andsong2024pro Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. Enhancing chat language models by scaling high-quality instructional conversations. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore, December 2023. Association for

Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.183. URL https://aclanthology.org/2023.emnlp-main.183/.

[8] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators, 2025. URL https://arxiv.org/abs/2404.04475.

[9] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[10] Qizhang Feng, Siva Rajesh Kasa, Santhosh Kumar Kasa, Hyokun Yun, Choon Hui Teo, and Sravan Babu Bodapati. Exposing privacy gaps: Membership inference attack on preference data for llm alignment. *arXiv preprint arXiv:2407.06443*, 2024.

[11] Fengyu Gao and Jing Yang. Differentially private preference data synthesis for large language model alignment. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=Ctk4XLaUpl.

[12] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[13] Chengxin Li, Yujie Gu, Pengpeng Qiao, Shengli Pan, Kouichi Sakurai, and Zhetao Li. Privacy-preserving yet vulnerable: Data poisoning attacks against differential privacy sparse mobile crowdsensing system. *IEEE Transactions on Mobile Computing*, 2026.

[14] 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.

[15] Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, Peter J Liu, and Xuanhui Wang. LiPO: Listwise preference optimization through learning-to-rank. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2404–2420, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.121. URL https://aclanthology.org/2025.naacl-long.121/.

[16] Yule Liu, Heyi Zhang, Jinyi Zheng, Zhen Sun, Zifan Peng, Tianshuo Cong, Yilong Yang, Xinlei He, and Zhuo Ma. Grpo privacy is at risk: A membership inference attack against reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.14045*, 2025.

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[18] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

[19] 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. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/a85b405ed65c6477a4fe8302b5e06ce7-Paper-Conference.pdf.

[20] 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 Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i17.29865. URL https://doi.org/10.1609/aaai.v38i17.29865.

[21] Noel Teku, Fengwei Tian, Payel Bhattacharjee, Souradip Chakraborty, Amrit Singh Bedi, and Ravi Tandon. Props: Progressively private self-alignment of large language models. *arXiv preprint arXiv:2508.06783*, 2025.

[22] Daniel Valcarce, Alejandro Bellogín, Javier Parapar, and Pablo Castells. Assessing ranking metrics in top-n recommendation. *Inf. Retr.*, 23(4):411–448, June 2020. ISSN 1386-4564. doi: 10.1007/s10791-020-09377-x. URL https://doi.org/10.1007/s10791-020-09377-x.

[23] Jianling Wang, Yifan Liu, Yinghao Sun, Xuejian Ma, Yueqi Wang, He Ma, Zhengyang Su, Minmin Chen, Mingyan Gao, Onkar Dalal, Ed H. Chi, Lichan Hong, Ningren Han, and Haokai Lu. User feedback alignment for LLM-powered exploration in large-scale recommendation systems. In Georg Rehm and Yunyao Li, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 996–1003, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-288-6. doi: 10.18653/v1/2025.acl-industry.70. URL https://aclanthology.org/2025.acl-industry.70/.

[24] Fan Wu, Huseyin A Inan, Arturs Backurs, Varun Chandrasekaran, Janardhan Kulkarni, and Robert Sim. Privately aligning language models with reinforcement learning. *arXiv preprint arXiv:2310.16960*, 2023.

[25] Wen Xu, Pengpeng Qiao, Shang Liu, Zhirun Zheng, Yang Cao, and Zhetao Li. Continuous publication of weighted graphs with local differential privacy. *Proceedings of the VLDB Endowment*, 18(11):4214–4226.

[26] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *eprint arXiv: 2407.10671*, 2024.

[27] Da Yu, Peter Kairouz, Sewoong Oh, and Zheng Xu. Privacy-preserving instructions for aligning large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57480–57506. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yu24e.html.

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

[29] Yang Zhao, Yixin Wang, and Mingzhang Yin. Permutative preference alignment from listwise ranking of human judgments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 310–334, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.17. URL https://aclanthology.org/2025.emnlp-main.17/.

[30] Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J. Liu. Slic-hf: Sequence likelihood calibration with human feedback, 2023. URL https://arxiv.org/abs/2305.10425.

[31] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

A Additional Privacy Granularities

This appendix records two stronger privacy granularities over listwise preference data. These definitions are provided for completeness only. The main analysis in the paper is based on item-level relocation adjacency in Definition 4.1.

A.1 Rank-Level Local Differential Privacy

Definition A.1 (Rank-level replacement adjacency)**.** Two rankings $\phi, \phi' \in S_m$ are rank-level adjacent, denoted by

$$\phi \sim_{\text{rank}} \phi',$$

if they are two possible complete rankings over the same candidate set. Equivalently, ϕ and ϕ' may differ arbitrarily as permutations in S_m.

Definition A.2 (Rank-level local differential privacy on rankings)**.** Let $Q : S_m \to \mathcal{O}$ be a randomized mechanism over rankings. We say that Q satisfies ε-rank-level local differential privacy if for all rank-level adjacent rankings $\phi, \phi' \in S_m$ and all outputs $o \in \mathcal{O}$,

$$\frac{Q(o \mid \phi)}{Q(o \mid \phi')} \leq e^{\varepsilon}.$$

Rank-level LDP protects the entire ordering for a single prompt. Unlike item-level LDP, which only requires indistinguishability under one item relocation, rank-level LDP requires indistinguishability between any two complete rankings over the same candidate set.

A.2 User-Level Local Differential Privacy in Ranking

For user-level privacy, the private input is not a single ranking but the user's full local preference profile. Let

$$\Phi_u = \left(\phi_{u,1}, \ldots, \phi_{u,T_u}\right)$$

denote the collection of listwise rankings provided by user u, where $\phi_{u,t} \in S_{m_t}$ is the ranking for the t-th prompt and candidate set.

Definition A.3 (User-level replacement adjacency)**.** Two user preference profiles Φ_u and Φ'_u are user-level adjacent, denoted by

$$\Phi_u \sim_{\text{user}} \Phi'_u,$$

if they are two possible complete preference profiles of the same user. Equivalently, Φ_u and Φ'_u may differ arbitrarily in any or all of their component rankings.

Definition A.4 (User-level local differential privacy)**.** Let $M : \mathcal{X}_{\text{user}} \to \mathcal{O}$ be a randomized mechanism that takes a user's full preference profile as input. We say that M satisfies ε-user-level local differential privacy if for all user-level adjacent profiles $\Phi_u, \Phi'_u \in \mathcal{X}_{\text{user}}$ and all outputs $o \in \mathcal{O}$,

$$\frac{M(o \mid \Phi_u)}{M(o \mid \Phi'_u)} \leq e^{\varepsilon}.$$

User-level LDP protects the complete preference profile of a user, including all rankings submitted across prompts. This is stronger than protecting a single listwise ranking and generally requires privacy calibration at the profile level rather than at the single-ranking level.

A.3 Relation to item-level LDP

Definition 4.2 gives a per-list guarantee: for a single prompt, two rankings that differ by relocating one item are indistinguishable up to e^{ε}. If the same user contributes N listwise feedbacks and the mechanism is applied independently to each list, sequential composition gives

$$\prod_{t=1}^{N} \frac{Q_t(o_t \mid \phi_{u,t})}{Q_t(o_t \mid \phi'_{u,t})} \leq e^{N\varepsilon},$$

whenever each pair $(\phi_{u,t}, \phi'_{u,t})$ is item-level adjacent. Thus, the per-list item-level guarantee induces a user-level privacy loss of $\varepsilon_{\text{user}} = N\varepsilon$ for N released lists under this adjacency. For example, $N = 10$ prompts with per-list $\varepsilon = 1$ gives $\varepsilon_{\text{user}} = 10$, which is already a loose worst-case bound.

For the stronger user-level replacement adjacency in Definition A.3, where each ranking may change arbitrarily, the direct implication from item-level LDP is even more conservative. Since any two rankings in S_m can be connected by at most $m - 1$ item relocations, group privacy and sequential composition give the worst-case bound

$$\varepsilon_{\text{user}} \leq N(m-1)\varepsilon$$

for N lists of size m. This illustrates that item-level LDP protects the main per-item position leakage considered in the paper, while direct extensions to rank-level or user-level privacy can incur large privacy-budget costs. We leave a tight user-level analysis as future work.

B Proofs

B.1 Sensitivity and Noise Scale Calculation

To compare different release granularities on a unified scale, we report their L_1 sensitivities and the corresponding Laplace noise scales. This analysis is intended as a common-scale comparison rather than an optimal mechanism choice: ranking vectors and binary partitions may admit better discrete DP mechanisms, so their Laplace noise scales serve as conservative upper bounds.

We provide the sensitivity and noise scale calculation for the three representation levels used in Section 4.2. Throughout this section, sensitivity is measured in L_1 distance:

$$\Delta_R = \sup_{x \sim x'} \|R(x) - R(x')\|_1,$$

where $x \sim x'$ denotes the item-level relocation adjacency in 4.1.

Score representation. For the score representation, the released object is the cardinal score vector

$$R_0(\mathbf{s}) = \mathbf{s}.$$

We normalize the score range to $[0, m]$, so that

$$\mathbf{s} \in [0, m]^m.$$

For two adjacent inputs $\mathbf{s}$ and $\mathbf{s}'$, the L_1 change is

$$\|R_0(\mathbf{s}) - R_0(\mathbf{s}')\|_1 = \|\mathbf{s} - \mathbf{s}'\|_1 = \sum_{i=1}^{m} |s_i - s'_i|.$$

Since each coordinate lies in $[0, m]$, each coordinate can change by at most m:

$$|s_i - s'_i| \leq m.$$

There are m coordinates, so

$$\sum_{i=1}^{m} |s_i - s'_i| \leq \sum_{i=1}^{m} m = m^2.$$

This bound is tight: for example, one may take

$$\mathbf{s} = (m, m, \ldots, m), \qquad \mathbf{s}' = (0, 0, \ldots, 0),$$

which gives

$$\|\mathbf{s} - \mathbf{s}'\|_1 = m^2.$$

Therefore,

$$\Delta_{R_0} = m^2.$$

Ranking representation. For the ranking representation, the released object is the full ranking encoded by fixed positional scores. Let $r_i(\phi) \in \{1, \dots, m\}$ denote the rank position of item i, where smaller rank means higher preference. We encode the ranking as

$$\rho_i(\phi) = m - r_i(\phi),$$

so that the top-ranked item has positional score $m-1$, the bottom-ranked item has positional score 0, and adjacent rank positions differ by one. Thus,

$$R_1(\phi) = \boldsymbol{\rho}(\phi) = (\rho_1(\phi), \dots, \rho_m(\phi)).$$

Under the local adjacency relation for rankings, two adjacent rankings differ by relocating one response from one rank position to another, while preserving the relative order of all other responses. Suppose item i moves from rank position p to rank position q. Let

$$d = |p - q|$$

be the number of positions it moves.

The moved item changes its positional score from

$$m - p$$

to

$$m - q.$$

Therefore, its absolute change is

$$|(m-p) - (m-q)| = |q - p| = d.$$

The only other affected items are the d responses between positions p and q. Each of these responses shifts by exactly one rank position, and since adjacent positional scores differ by one, each contributes 1 to the L_1 change. Hence, the total contribution of the shifted intervening responses is

$$d.$$

All remaining responses keep the same rank position and contribute 0 to the L_1 change. Therefore, the total L_1 change is

$$d + d = 2d.$$

Since the largest possible relocation distance is

$$d_{\max} = m - 1,$$

we obtain

$$\|R_1(\phi) - R_1(\phi')\|_1 \le 2(m-1).$$

This bound is tight. For example, moving the top-ranked item to the bottom position gives $d = m-1$, and the total L_1 change is

$$2(m-1).$$

Therefore,

$$\Delta_{R_1} = 2(m-1).$$

Binary partition representation. For the binary top-B representation, the released object is

$$R_2(\phi) = Y(\phi) \in \{0,1\}^m, \qquad Y_i(\phi) = \mathbf{1}[r_i(\phi) \le B].$$

Thus, exactly B coordinates are equal to 1, and the remaining $m - B$ coordinates are equal to 0.

Under the local adjacency relation for binary top-B labels, two adjacent labels differ by one top-B replacement: one response leaves the top-B group and one response enters it. Equivalently, one coordinate changes from 1 to 0, and another coordinate changes from 0 to 1. Therefore, the L_1 change is

$$|1-0| + |0-1| = 2.$$

All other coordinates remain unchanged. Hence,

$$\|R_2(\phi) - R_2(\phi')\|_1 \le 2.$$

The bound is tight whenever one item is swapped across the top-B boundary. Therefore,

$$\Delta_{R_2} = 2.$$

Combining the three cases gives

$$\Delta_{R_0} = m^2, \qquad \Delta_{R_1} = 2(m-1), \qquad \Delta_{R_2} = 2.$$

Consequently, under the Laplace mechanism, the coordinate-wise noise scales are

$$\frac{\Delta_{R_0}}{\varepsilon} = \frac{m^2}{\varepsilon}, \qquad \frac{\Delta_{R_1}}{\varepsilon} = \frac{2(m-1)}{\varepsilon}, \qquad \frac{\Delta_{R_2}}{\varepsilon} = \frac{2}{\varepsilon}.$$

B.2 Sensitivity of the rank-aware utility

Lemma B.1 (Sensitivity of the linear rank-aware utility)**.** *Let*

$$U(\phi, z) = \sum_{i=1}^{m} z_i \left(B + \frac{1}{2} - r_i(\phi) \right), \qquad z \in \mathcal{Z}_B,$$

where $\mathcal{Z}_B = \{z \in \{0,1\}^m : \sum_i z_i = B\}$*. Under item-level relocation adjacency,*

$$\Delta := \max_{\phi \sim \phi'} \max_{z \in \mathcal{Z}_B} |U(\phi, z) - U(\phi', z)| = \max\{B, m - B\}.$$

Proof. Consider two adjacent rankings $\phi \sim \phi'$, where ϕ' is obtained from ϕ by relocating one item x from rank a to rank b. It suffices to consider the case $a < b$, since the case $a > b$ is symmetric. Let $L = b - a$. Then item x moves down by L positions, while the L items between ranks $a+1$ and b each move up by one position.

For any fixed $z \in \mathcal{Z}_B$, the constant term $B + \frac{1}{2}$ cancels, and therefore

$$U(\phi, z) - U(\phi', z) = \sum_{i=1}^{m} z_i \big(r_i(\phi') - r_i(\phi) \big).$$

Let I be the set of items shifted up by one position, and define $c = \sum_{i \in I} z_i$. Then

$$U(\phi, z) - U(\phi', z) = z_x L - c.$$

If $z_x = 0$, then

$$|U(\phi, z) - U(\phi', z)| = c \le \min\{B, L\} \le B.$$

If $z_x = 1$, then $B - 1$ selected items other than x must be placed among the $m - 1$ remaining items. Since only $m - L - 1$ of these items lie outside the shifted interval I, at least

$$\max\{0,\ B - 1 - (m - L - 1)\} = \max\{0,\ B + L - m\}$$

selected items must lie in I. Hence

$$|U(\phi, z) - U(\phi', z)| = L - c \le L - \max\{0,\ B + L - m\} = \min\{L, m - B\} \le m - B.$$

Combining the two cases gives

$$|U(\phi, z) - U(\phi', z)| \le \max\{B, m - B\}.$$

It remains to show that this bound is tight. Relocate the item at rank 1 to rank m, so $L = m - 1$. If $z_x = 0$ and z selects B of the shifted items, then the absolute utility change is B. If $z_x = 1$ and z selects x together with $B - 1$ shifted items, then the absolute utility change is

$$(m - 1) - (B - 1) = m - B.$$

Therefore the maximum possible change is exactly

$$\Delta = \max\{B, m - B\}.$$

□

B.3 Privacy of the rank-aware release mechanism

Theorem B.2 (Item-level privacy and tightness)**.** *Fix $m \geq 2$, $1 \leq B \leq m-1$, and $\eta \geq 0$. Let*

$$\mathcal{Z}_B = \left\{ z \in \{0,1\}^m : \sum_{i=1}^m z_i = B \right\}, \qquad U(\phi, z) = \sum_{i=1}^m z_i \left(B + \frac{1}{2} - r_i(\phi) \right),$$

and define

$$Q_\eta(z \mid \phi) = \frac{\exp(\eta U(\phi, z))}{Z_\phi(\eta)}, \qquad Z_\phi(\eta) = \sum_{z' \in \mathcal{Z}_B} \exp(\eta U(\phi, z')).$$

Under item-level relocation adjacency,

$$\sup_{\phi \sim_{\text{item}} \phi'} \max_{z \in \mathcal{Z}_B} \left| \log \frac{Q_\eta(z \mid \phi)}{Q_\eta(z \mid \phi')} \right| = \eta \max\{B, m-B\}.$$

Consequently, choosing $\eta = \varepsilon / \max\{B, m-B\}$ yields ε-item-level local differential privacy, and the privacy bound is tight for $\varepsilon > 0$.

Proof. First, the normalizing constant is independent of the ranking. Write $a_r = B + \frac{1}{2} - r$. For any fixed ranking ϕ, the map

$$z \longmapsto R_\phi(z) := \{r_i(\phi) : z_i = 1\}$$

is a bijection between $\mathcal{Z}_B$ and the size-B subsets of $\{1, \ldots, m\}$. Therefore,

$$\begin{aligned} Z_\phi(\eta) &= \sum_{z \in \mathcal{Z}_B} \exp\left(\eta \sum_{i=1}^m z_i a_{r_i(\phi)} \right) \\ &= \sum_{\substack{R \subseteq \{1,\ldots,m\} \\ |R| = B}} \exp\left(\eta \sum_{r \in R} a_r \right) =: Z_{m,B}(\eta), \end{aligned}$$

which depends only on m, B, η, not on ϕ.

Next, consider adjacent rankings ϕ, ϕ'. By symmetry, it suffices to consider moving an item x from rank a to rank $b > a$. Put $L = b - a$, and let I be the set of the L items that move up by one rank. For a fixed output $z \in \mathcal{Z}_B$, set $c = \sum_{i \in I} z_i$. Then

$$U(\phi, z) - U(\phi', z) = z_x L - c.$$

If $z_x = 0$, the absolute difference is $c \leq B$. If $z_x = 1$, it is $L - c$, the number of unselected items in I, which is at most $m - B$. Hence,

$$|U(\phi, z) - U(\phi', z)| \leq \max\{B, m-B\} =: \Delta.$$

Since all probabilities on $\mathcal{Z}_B$ are positive and $Z_\phi(\eta) = Z_{\phi'}(\eta)$, we have

$$\begin{aligned} \left| \log \frac{Q_\eta(z \mid \phi)}{Q_\eta(z \mid \phi')} \right| &= \eta \, |U(\phi, z) - U(\phi', z)| \\ &\leq \eta \Delta. \end{aligned}$$

Thus $\eta = \varepsilon / \Delta$ implies $Q_\eta(z \mid \phi) \leq e^\varepsilon Q_\eta(z \mid \phi')$ for every adjacent pair and every output, establishing the privacy guarantee.

To prove tightness, take

$$\phi = (1, 2, \ldots, m), \qquad \phi' = (2, 3, \ldots, m, 1).$$

These rankings are item-level adjacent. If $z_1 = 0$ and z selects B of the remaining items, then

$$U(\phi, z) - U(\phi', z) = -B.$$

If $z_1 = 1$ and z selects $B - 1$ of the remaining items, then

$$U(\phi, z) - U(\phi', z) = m - B.$$

Both outputs are feasible because $1 \leq B \leq m-1$. The largest absolute utility difference is therefore Δ, so the largest absolute log-likelihood ratio is exactly $\eta\Delta$. This proves the claimed equality and tightness. □

B.4 Derivation of the Flip Rate Formula

For $v \in \mathbb{R}^d$, define

$$\sigma_k(v) := \sum_{\substack{S \subseteq [d] \\ |S|=k}} \prod_{j \in S} v_j, \qquad \sigma_0(v) = 1, \qquad \sigma_k(v) = 0 \text{ if } k < 0 \text{ or } k > d.$$

Under the rank-aware exponential mechanism,

$$\Pr(\widetilde{z} = z \mid \phi) = \frac{\exp(\eta U(\phi, z))}{\sum_{z' \in \mathcal{Z}_B} \exp(\eta U(\phi, z'))}, \qquad U(\phi, z) = \sum_{j=1}^{m} z_j w_j(\phi).$$

Since

$$\exp(\eta U(\phi, z)) = \prod_{j : z_j = 1} e^{\eta w_j(\phi)}, \qquad \sum_{z' \in \mathcal{Z}_B} \exp(\eta U(\phi, z')) = \sigma_B(e^{\eta w(\phi)}),$$

the marginal probability that y_i is released as positive is

$$q_i(\phi) := \Pr(\widetilde{z}_i = 1 \mid \phi) = \frac{e^{\eta w_i(\phi)} \sigma_{B-1}(e^{\eta w_{-i}(\phi)})}{\sigma_B(e^{\eta w(\phi)})}.$$

Using

$$\sigma_B(e^{\eta w(\phi)}) = e^{\eta w_i(\phi)} \sigma_{B-1}(e^{\eta w_{-i}(\phi)}) + \sigma_B(e^{\eta w_{-i}(\phi)}),$$

we obtain

$$\frac{q_i(\phi)}{1 - q_i(\phi)} = e^{\eta w_i(\phi)} \frac{\sigma_{B-1}(e^{\eta w_{-i}(\phi)})}{\sigma_B(e^{\eta w_{-i}(\phi)})}.$$

Holding $w_{-i}(\phi)$ fixed, this odds ratio is increasing in $w_i(\phi)$. Thus $q_i(\phi)$ is monotone increasing in the rank-position weight.

Finally, with $z_i(\phi) = \mathbf{1}[r_i(\phi) \le B]$, the item-wise flip rate is

$$\rho_i(\phi) := \Pr(\widetilde{z}_i \neq z_i(\phi) \mid \phi) = \begin{cases} 1 - q_i(\phi), & r_i(\phi) \le B, \\ q_i(\phi), & r_i(\phi) > B. \end{cases}$$

B.5 Proof of Theorem 5.1

We prove that for any monotone rank score $w_1 \ge \cdots \ge w_m$,

$$\frac{S_w}{\Delta_w} \le \frac{m-1}{\max\{B, m-B\}},$$

with equality iff $w_s = \alpha - \beta s$ for some $\beta > 0$.

Proof. Let

$$\delta_s := w_s - w_{s+1}, \quad s \in [m-1], \qquad \delta_{(1)} \ge \cdots \ge \delta_{(m-1)}$$

be the adjacent score gaps in decreasing order. Consider relocating one item from rank a to rank $b > a$; the reverse case is symmetric. Let i_s be the item originally at rank s. Then i_a is moved, while $i_{a+1}, \ldots, i_b$ shift forward by one rank. For any $z \in \mathcal{Z}_B$,

$$U_w(\phi, z) - U_w(\phi', z) = \sum_{\ell=a}^{b-1} \delta_\ell \big(z_{i_a} - z_{i_{\ell+1}}\big).$$

If $z_{i_a} = 0$, at most B selected shifted items contribute. If $z_{i_a} = 1$, at most $m - B$ unselected shifted items contribute. Hence

$$|U_w(\phi, z) - U_w(\phi', z)| \le \max\left\{ \sum_{j=1}^{B} \delta_{(j)}, \sum_{j=1}^{m-B} \delta_{(j)} \right\}.$$

Let $K := \max\{B, m-B\}$. The bound is tight by relocating an item across the full ranking and choosing z so that the contributing shifted items correspond to the largest K gaps. Therefore,

$$\Delta_w = \sum_{j=1}^{K} \delta_{(j)}.$$

Since

$$w_1 - w_m = \sum_{s=1}^{m-1} \delta_s$$

and the average of the largest K gaps is at least the average of all gaps,

$$\Delta_w = \sum_{j=1}^{K} \delta_{(j)} \geq \frac{K}{m-1} \sum_{s=1}^{m-1} \delta_s = \frac{K}{m-1}(w_1 - w_m).$$

Rearranging gives

$$\frac{w_1 - w_m}{\Delta_w} \leq \frac{m-1}{K} = \frac{m-1}{\max\{B, m-B\}}.$$

Equality holds iff all adjacent gaps are equal,

$$\delta_1 = \cdots = \delta_{m-1},$$

which is equivalent to $w_s = \alpha - \beta s$. Strict monotonicity gives $\beta > 0$. □

B.6 Calibration of Randomized Response in Listwise Preference Data

Recall the clean top-B label vector

$$z_i(\phi) = \mathbf{1}[r_i(\phi) \leq B], \qquad z(\phi) = (z_1(\phi), \ldots, z_m(\phi)).$$

RR with per-coordinate privacy parameter ε_0 releases $\tilde{z}_{\mathsf{RR}} \in \{0,1\}^m$ independently across coordinates:

$$\Pr[(\tilde{z}_{\mathsf{RR}})_i = z_i(\phi) \mid \phi] = \frac{e^{\varepsilon_0}}{1+e^{\varepsilon_0}}, \qquad \Pr[(\tilde{z}_{\mathsf{RR}})_i \neq z_i(\phi) \mid \phi] = \frac{1}{1+e^{\varepsilon_0}}.$$

Under item-level relocation adjacency, the clean label vector changes in at most two coordinates:

$$\Delta_H := \sup_{\phi \sim_{\text{item}} \phi'} \|z(\phi) - z(\phi')\|_0 = 2.$$

Indeed, if one item enters the top-B set, one item must leave it, and vice versa.

For any $u \in \{0,1\}^m$, the RR likelihood ratio factorizes as

$$\frac{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi]}{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi']} = \prod_{i=1}^{m} \frac{\Pr[(\tilde{z}_{\mathsf{RR}})_i = u_i \mid z_i(\phi)]}{\Pr[(\tilde{z}_{\mathsf{RR}})_i = u_i \mid z_i(\phi')]}.$$

Unchanged coordinates contribute ratio 1, and each changed coordinate contributes at most e^{ε_0}. Since at most two coordinates change,

$$\frac{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi]}{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi']} \leq e^{2\varepsilon_0}.$$

Choosing $\varepsilon_0 = \varepsilon/2$ yields

$$\frac{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi]}{\Pr[\tilde{z}_{\mathsf{RR}} = u \mid \phi']} \leq e^{\varepsilon},$$

so RR is ε-LDP under item-level relocation adjacency.

Substituting $\varepsilon_0 = \varepsilon/2$, the RR flip rate is

$$\rho_i^{\mathsf{RR}}(\phi) := \Pr[(\tilde{z}_{\mathsf{RR}})_i \neq z_i(\phi) \mid \phi] = \frac{1}{1+e^{\varepsilon/2}}, \qquad i \in [m].$$

C Experiments Details

C.1 Training Details

The training hyperparameters for RR+KTO and our method on Qwen2-0.5B and Mistral-7B are shown in Table 3. For non-KTO baselines, PPA and SLiC, we use the best hyperparameters reported in their original papers. Using 4 NVIDIA H100 80GB GPUs, each run takes about 2 hours for Qwen2-0.5B and about 8 hours for Mistral-7B.

Table 3: Training details for private KTO experiments.

Hyperparameter	**Value**
Training dataset	ListUltraFeedback
Number of candidates m	8
Top boundary B	4
Privacy budgets ε	0.5, 1, 3
KTO β	0.1
KTO λ_D, λ_U	1.0, 1.0
Mini batch / GPU	2
Gradient accumulation steps	8
GPUs	4
Effective batch size	64
Learning rate	5×10^{-6}
Epochs	1
Max prompt length	512
Max response length	1024
Optimizer	AdamW
LR scheduler	Cosine
Warmup ratio	0.1
Weight decay	0.01
Max grad norm	1.0
DeepSpeed	ZeRO-2
Random seed	42

C.2 Evaluation Details

Table 4 reports the datasets and judge models used in our experiments. UltraChat200k is used for SFT, ListUltraFeedback$_{\text{train}}$ for preference alignment, and ListUltraFeedback$_{\text{test}}$ for in-domain proxy evaluation with ArmoRM and the RLHFlow pair-preference model. AlpacaEval and MT-Bench provide general instruction-following benchmarks judged by GPT-4 Turbo.

Table 4: Datasets and judge models used in our experiments.

Datasets	**Examples**	**Judge Model**
UltraChat200k	207,865	–
ListUltraFeedback$_{\text{train}}$	59,880	–
ListUltraFeedback$_{\text{test}}$	1,968	RLHFlow Pair-Preference, ArmoRM
AlpacaEval	805	GPT-4 Turbo
MT-Bench	80	GPT-4 Turbo

D Additional Experiments

We report additional experiments on alternative alignment objectives, variation across training seeds, and controlled mechanism ablations. The four alignment metrics follow the evaluation protocol in the main text. All scores are reported as percentages, with higher values indicating better performance. For the alignment evaluations, Avg. denotes the unweighted mean of Pair-Preference, Scoring, AlpacaEval, and MT-Bench.

Table 5: RAEM with KTO and DPO on Qwen2-0.5B. Bold values indicate the higher average score at each privacy budget.

Method	ε	Pair-Preference	Scoring	AlpacaEval	MT-Bench	Avg.
RAEM+KTO (ours)	1	61.2	49.1	50.6	50.6	**52.9**
RAEM+DPO	1	58.1	48.5	48.9	53.5	52.3
RAEM+KTO (ours)	3	66.4	54.6	57.8	59.7	**59.6**
RAEM+DPO	3	64.8	51.1	54.2	54.5	56.1

Table 6: Mistral-7B results at $\varepsilon = 1$ across three training seeds. Summary rows report mean $\pm$ sample standard deviation ($n = 3$). Bold per-seed averages identify the better method for each seed; bold summary means identify the higher mean in each metric.

Seed	Method	Pair-Preference	Scoring	AlpacaEval	MT-Bench	Avg.
42 (original)	RR+KTO	89.70	90.90	86.40	71.80	84.70
42 (original)	RAEM+KTO	90.50	91.00	87.50	72.10	**85.28**
123	RR+KTO	90.21	91.02	86.48	70.20	84.48
123	RAEM+KTO	91.11	91.22	88.05	70.44	**85.21**
456	RR+KTO	90.77	90.15	86.17	70.89	84.50
456	RAEM+KTO	91.08	90.94	87.02	73.58	**85.66**
–	RR+KTO, mean	90.23 ± 0.54	90.69 ± 0.47	86.35 ± 0.16	70.96 ± 0.80	84.56 ± 0.12
–	RAEM+KTO, mean	$\mathbf{90.90} \pm 0.34$	$\mathbf{91.05} \pm 0.15$	$\mathbf{87.52} \pm 0.52$	$\mathbf{72.04} \pm 1.57$	$\mathbf{85.38} \pm 0.24$

D.1 Compatibility with pairwise alignment

RAEM releases a privatized top-B partition independently of the downstream alignment objective. To apply DPO, we construct preference pairs using only the released positive and negative groups: the preferred response is drawn from the released positive group and the rejected response from the released negative group. The admissible pair set is

$$\mathcal{P}(\tilde{z}) = \{(y_i, y_j) : \tilde{z}_i = 1,\ \tilde{z}_j = 0\}.$$

Because all pairs are constructed only from the released partition and the candidate responses, this conversion is post-processing and incurs no additional privacy cost. The same construction is also applicable to pairwise supervision for reward-model training in RLHF.

Table 5 compares RAEM followed by KTO or DPO on Qwen2-0.5B at $\varepsilon \in \{1, 3\}$. RAEM is compatible with both objectives, while RAEM+KTO achieves the higher average score at both privacy budgets.

We retain KTO as the primary alignment objective for two reasons. First, its unary supervision format directly matches the released top-B partition, whereas DPO requires conversion into pairwise comparisons. Second, the sigmoid utility in KTO attenuates the gradient weight of examples with extreme policy–reference log-ratios relative to the detached KL baseline. This includes strongly contradictory examples, such as an undesirable response with a very high implied reward or a desirable response with a very low implied reward. Downweighting such examples provides a motivation for using KTO with privacy-corrupted labels, and the results in Table 5 support this choice in the evaluated setting.

D.2 Variation across training seeds

We conduct additional experiments on Mistral-7B at $\varepsilon = 1$ using training seeds 123 and 456, in addition to seed 42 used in the main results. Table 6 reports the complete results for RR+KTO and RAEM+KTO. The summary rows show the mean and sample standard deviation over the three training seeds. For Avg., we first compute the average of the four metrics for each seed and then compute its mean and sample standard deviation across seeds.

RAEM+KTO achieves a higher average score than RR+KTO for each of the three seeds. Its mean average score is 85.38, compared with 84.56 for RR+KTO, an improvement of 0.82 percentage

Table 7: Controlled mechanism ablations on Qwen2-0.5B. All methods use KTO for downstream alignment. Bold values identify the highest average score at each privacy budget.

Method	Pair-Preference	Scoring	AlpacaEval	MT-Bench	Avg.
$\varepsilon = 0.5$					
Coordinate-wise RR + KTO	52.9	38.6	48.0	52.5	48.0
RR + uniform fixed-B + KTO	53.1	39.1	48.1	49.8	47.5
Top-membership EM + KTO	52.9	40.1	49.5	48.8	47.8
Linear rank-aware EM + KTO (ours)	59.2	43.6	51.8	49.7	**51.1**
$\varepsilon = 1$					
Coordinate-wise RR + KTO	57.9	43.7	50.7	52.0	51.1
RR + uniform fixed-B + KTO	56.9	43.0	50.8	50.5	50.3
Top-membership EM + KTO	58.1	45.1	50.1	50.2	50.9
Linear rank-aware EM + KTO (ours)	61.2	49.1	50.6	50.6	**52.9**

points. These runs support the consistency of the observed advantage under training-seed variation for Mistral-7B at this privacy budget.

D.3 Controlled mechanism ablations

We compare two additional ablations on Qwen2-0.5B to examine the roles of fixed cardinality, joint sampling, and within-group rank information. All methods use the same training and evaluation setup.

1. **RR + uniform fixed-B + KTO.** We apply coordinate-wise RR, followed by randomized post-processing that enforces exactly B positive labels. This step uses only the RR output and randomness independent of the clean ranking, and therefore incurs no additional privacy cost.
2. **Top-membership EM + KTO.** We use an exponential mechanism on the same fixed-cardinality output space $\mathcal{Z}_B$ as RAEM, with a utility that depends only on clean top-B membership. All items within the same clean group receive the same weight, so the utility does not use within-group ranks.

Table 7 reports the results at $\varepsilon = 0.5$ and $\varepsilon = 1$, together with the coordinate-wise RR baseline and our linear rank-aware mechanism.

Fixed-cardinality post-processing alone does not improve the average score over coordinate-wise RR at either budget. The top-membership exponential mechanism also remains below the linear rank-aware mechanism, despite using the same fixed-cardinality output space and joint exponential-mechanism sampling. Based on the displayed average scores, the linear rank-aware mechanism improves over top-membership EM by 3.3 percentage points at $\varepsilon = 0.5$ and 2.0 percentage points at $\varepsilon = 1$.

These controlled comparisons support the benefit of using within-group rank information in the evaluated setting. The results are consistent with the design of RAEM: concentrating noise near the top-B boundary helps preserve higher-quality preference signals for downstream alignment.

E Supplementary Results

Table 8: Qualitative Qwen2-0.5B examples from AlpacaEval where our method produces a more correct, relevant, or complete response than RR. Scores in parentheses are AlpacaEval GPT-4 Turbo preference scores against the same SFT reference; higher is better. Responses are summarized for space.

Prompt	Given an algorithm or a coding search query, provide a related code snippet. Search query: `Python loop`.
$\varepsilon = 1$	
RR	*(1.004)* Gives `for I in range(10): print(i)`, which raises a `NameError` because Python identifiers are case-sensitive, and then incorrectly describes the query as "10."
Ours	*(2.000)* Provides a valid `search_list` function that iterates over a list, returns the index of the queried element when found, and returns -1 otherwise.
Prompt	Write a thematic tweet announcing that applications for summer internships are open. Start with a Halloween tie-in.
$\varepsilon = 1$	
RR	*(1.004)* Produces a generic internship announcement with a celebration emoji, but the tweet itself contains no Halloween-related opening or theme.
Ours	*(2.000)* Opens with "Halloween is just around the corner!" and then clearly announces that summer internship applications are open, satisfying both requested elements.
Prompt	Write a JavaScript function that takes an array of five numbers, checks whether any element is prime, and prints every prime number.
$\varepsilon = 3$	
RR	*(1.002)* Prints an element only when both it and the following element are prime, and iterates only to `arr.length - 1`. It therefore misses valid primes, including a prime in the final position.
Ours	*(2.000)* Implements a correct primality test, examines every array element independently, and prints each element for which the test returns true.
Prompt	Write a response that disagrees with the post: "Technology is everything that doesn't work yet."
$\varepsilon = 3$	
RR	*(1.024)* Misreads the task as a request to interpret or rewrite the quotation and does not present a clear counterargument to it.
Ours	*(2.000)* Directly disagrees and supports the response with concrete examples of technologies already working in practice, including AI applications, e-commerce, and renewable energy.

Table 9: Qualitative Mistral-7B examples from AlpacaEval where our method produces a more correct, relevant, or complete response than RR. Scores in parentheses are AlpacaEval GPT-4 Turbo preference scores against the same SFT reference; higher is better. Responses are summarized for space.

Prompt	Does the dinosaurs really exist?
$\varepsilon = 0.5$	
RR	*(1.000)* Initially frames dinosaur existence as a matter of belief and interpretation, weakening an otherwise scientific answer.
Ours	*(2.000)* States the scientific consensus directly and supports it with fossilized bones, footprints, anatomical reconstruction, and examples of documented dinosaur species.
Prompt	Suggest product bundles that are often purchased with a MacBook Air.
$\varepsilon = 0.5$	
RR	*(1.097)* Recommends an Apple Pencil on the mistaken premise that the MacBook Air has a touchscreen and describes several adapters inaccurately as cables.
Ours	*(2.000)* Suggests compatible and practically useful accessories, including AppleCare+, an external drive, a USB-C hub, headphones, a mouse, and a protective sleeve or backpack.
Prompt	As a sports commentator, describe the winning play in the final seconds of a championship game.
$\varepsilon = 0.5$	
RR	*(1.024)* Begins a football scenario but is repetitive and terminates in the middle of the decisive play, leaving the requested winning moment unresolved.
Ours	*(2.000)* Delivers a complete and coherent basketball sequence: a tie game, ten seconds remaining, the final possession, a made three-pointer, and the resulting championship celebration.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's scope: preference alignment with listwise data under local differential privacy, using a rank-aware mechanism for binary top-B release and KTO-compatible training. The claims are limited to this setting and supported by the paper's analysis and experiments.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a limitations section discussing the scope of the privacy guarantee, the distinction between per-list item-level LDP and user-level privacy.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper states the ranking adjacency relation, the item-level local differential privacy definition, and the assumptions used for the exponential mechanism and flip-rate analysis. Formal proofs of the sensitivity bounds, flip-rate derivation, randomized response calibration, and calibrated rank-span optimality are provided in the appendix.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper describes the dataset, train/held-out split, backbone models, SFT initialization, privacy budgets, release mechanism, baselines, evaluation protocols, optimizer, and compute resources. Additional training and evaluation details are provided in the appendix to support reproduction of the main experimental comparisons.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: We will release the code following publication.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the use of ListUltraFeedback, its train/held-out split, the Qwen2-0.5B and Mistral-7B backbones, SFT initialization on UltraChat-200K, AdamW optimization, privacy budgets, and evaluation pipelines. Further hyperparameters and evaluation details are provided in the appendix.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

Justification: The main experiments are reported as single-run results because full alignment training on multiple backbones and GPT-based evaluation with AlpacaEval and MT-Bench are computationally and financially expensive. The paper uses held-out evaluation sets and complementary metrics, but it does not report multi-seed error bars or confidence intervals for all main results.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix C.1 specifies that all experiments are run on 4 NVIDIA H100 80GB GPUs, with each training run taking approximately 2 hours for Qwen2-0.5B and 8 hours for Mistral-7B. Section 6.1 additionally describes the SFT and alignment training pipeline, and Table 3 in Appendix C.1 lists the training hyperparameters used by RR+KTO and our method.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The work uses existing preference-alignment datasets and publicly available models/evaluation protocols, and it proposes a privacy-preserving mechanism intended to reduce exposure of sensitive preference data. The paper does not involve collecting new human-subject data or releasing high-risk models.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses the positive impact of reducing preference leakage in alignment pipelines, especially when listwise feedback may encode sensitive beliefs, values, or behavioral patterns. It also discusses limitations and residual risks, including user profiling across repeated interactions and possible post-training leakage.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not introduce or release a new high-risk pretrained model, image generator, or scraped dataset. The contribution is an algorithmic privacy mechanism for preference release and is evaluated using existing datasets and models.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets used in this paper are properly cited in Section 6.1 and Appendix A/B. The datasets (ListUltraFeedback, UltraFeedback, UltraChat-200K), foundation models (Qwen2-0.5B, Mistral-7B), reward models (ArmoRM-Llama3-8B, RLHFlow pair-preference-model-LLaMA3-8B), and evaluation benchmarks (AlpacaEval, MT-Bench) are released under their respective open-source licenses (MIT, Apache 2.0, or Llama Community License), and our use is consistent with the terms of these licenses.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: The paper does not introduce a new dataset, benchmark, or released model asset. The contribution is a privacy-preserving release mechanism and its evaluation on existing listwise preference data.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not conduct new crowdsourcing experiments or new research with human subjects. It uses existing datasets and benchmark evaluation protocols.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not collect new data from human subjects or conduct new human-subject experiments. Therefore, IRB approval or equivalent review is not applicable.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: LLMs are central to the experimental setting: the paper studies preference alignment for language models, trains Qwen2-0.5B and Mistral-7B backbones, and uses GPT-4 Turbo as an evaluation judge for general conversational benchmarks. These uses are described in the experimental setup and evaluation sections.