Benchmarking Membership Privacy Risks in Preference-Based LLM Post-Training

Lorenzo Rossi **Kaif Shaikh** **Franziska Boenisch** **Adam Dziedzic**
CISPA Helmholtz Center for Information Security
{lorenzo.rossi, kaif.shaikh, boenisch, adam.dziedzic}@cispa.de

Abstract

Modern language models are commonly adapted after pretraining to follow instructions, align with user preferences, and improve deployment behavior. Such post-training often relies on preference data from users, annotators, or model interactions, which may contain sensitive prompts, private responses, proprietary tasks, or confidential judgments. Understanding the privacy implications of this data is therefore essential. Membership inference attacks (MIAs), which test whether a record was used for training, are the de facto standard for empirical privacy auditing in machine learning. However, preference-based post-training changes the unit of membership: a training record contains a prompt, a preferred response, and a dispreferred response, rather than a single input-output sequence. Audits that ignore this structure can under-report leakage. We therefore introduce a benchmark protocol that adapts strong reference-model MIAs to complete preference records. Our protocol asks whether membership is exposed by either response alone, by the model's preference between them, or by the two responses jointly. We find that auditing the response pair can reveal membership evidence missed when the record is reduced to a single score. Across three preference datasets, two model families, and seven post-training objectives spanning three post-training families, we find that measured leakage depends strongly on both the audited record statistic and the training objective, with imbalanced risk between preferred and dispreferred responses. We further evaluate the impact of parameter-efficient fine-tuning (PEFT) and differential privacy (DP), finding widely varying privacy-utility trade-offs. Overall, preference-based post-training leaks in ways that standard single-response audits can miss, motivating formal privacy methods that protect complete preference records while preserving post-training utility.

1 Introduction

Large language models (LLMs) are rarely deployed directly after pretraining. They are typically adapted through post-training to improve instruction following, align behavior with human or AI feedback, and shape the responses users see [18, 32, 36, 37]. A central ingredient is preference data, *i.e.,* records that pair a prompt with a preferred response and a dispreferred response. Such data is used for the three main families of post-training methods, namely supervised instruction tuning [29, 44], direct preference-learning objectives [3, 15, 20, 30], and reinforcement-learning-based alignment [31, 32]. In all these setups, the data can be sensitive: prompts may contain personal information or proprietary intents, responses may reveal private context or model failures, and the preferred/dispreferred ordering may encode confidential judgments about safety or policy compliance.

Membership inference attacks (MIAs) are the de facto standard for empirical privacy auditing in machine learning [9, 34]. Given a trained model and a target example, a MIA tests whether the example was used for training. Prior work has used MIAs to study leakage from LLM pretraining and supervised fine-tuning [13, 19, 27, 33, 41]. However, preference-based post-training changes

40th Conference on Neural Information Processing Systems (NeurIPS 2026). Track on Evaluations and Datasets.

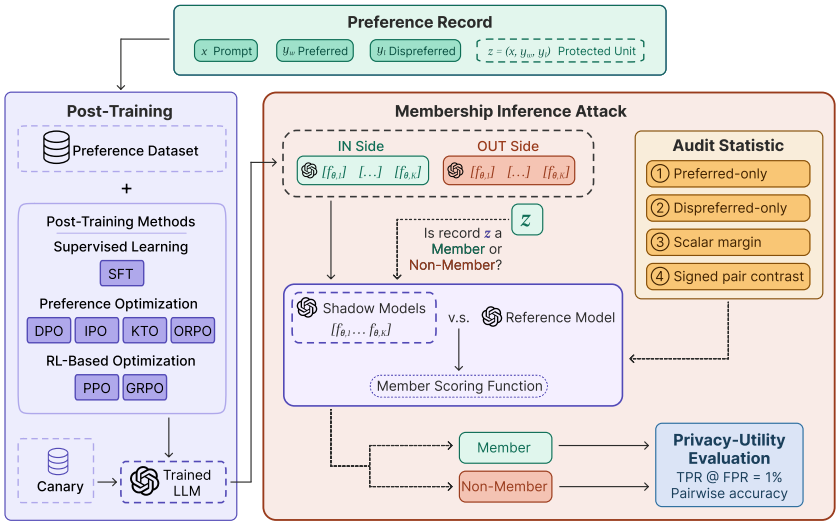


Figure 1: **Benchmark overview.** The audit treats each preference record as the protected unit, applies reference-model MIA scorers to multiple record statistics, and reports leakage alongside PEFT, DP, and canary analyses.

the *unit of membership*. A preference record consists of a prompt x, a preferred response y_w, and a dispreferred response y_l, so collapsing it to one sequence and running standard MIAs is not a neutral choice. Auditing based on the preferred-response, the dispreferred-response, or their relation asks different privacy questions and can yield widely varying risk reports, as we show in this work. This creates a gap in current privacy evaluations. As an initial solution, PREMIA [17] introduced a preference-aware signal on the model's relative behavior on preferred and dispreferred responses, but there is no systematic benchmark for membership privacy in preference-based post-training. Without a common protocol, empirical privacy conclusions can be hard to interpret. A method may appear private simply because the audit scores the wrong part of the record, or because high privacy is entangled with low utility.

We address this gap with a structure-aware benchmark for membership privacy in preference-based post-training. The benchmark evaluates membership under a common reference-model protocol [9], using three public preference datasets [4, 10, 11], two open model families [36, 37], and seven post-training objectives [3, 15, 20, 29–32].

For each target record, we audit several views of the preference tuple, namely each response in isolation, the model's relative preference between the two responses, and the two responses as a signed pair. We also test how leakage changes with adaptation capacity, formal privacy guarantees, sequence length, response-side asymmetry, objective hyperparameters, and controlled canaries. Figure 1 summarizes the benchmark flow.

Overall, our benchmark asks: ***How do the representation of a preference record, the post-training objective, training paradigm, and privacy interventions shape empirical membership leakage from preference data?*** Our results lead to several practical conclusions: First, we find that preference data should not be audited through a single response or a single scalar preference score alone, since doing so can under-report leakage. Instead, audits should preserve the structure of the preference record and report whether the detected signal comes from the preferred response, the dispreferred response, or their relation. Second, our results suggest that the post-training objective also matters: objectives that directly imitate the preferred response, or combine imitation with pairwise preference learning, expose the most membership signal. Pairwise preference optimization still leaks under structure-aware audits but a more strongly regularized pairwise objective is much less exposed. In contrast, MIAs against objectives based on binary feedback or reward-driven policy optimization perform close to random guessing in our main evaluation. Finally, we find that limiting the trainable adaptation capacity through parameter-efficient fine-tuning (PEFT) introduces a widely varying privacy-utility frontier. The formal privacy guarantees from differentially private training provides strong empirical privacy protection, however, this comes at the cost of utility.

In summary, we make the following contributions:

- We introduce a benchmark for membership privacy in preference-based post-training, where the protected unit is the full preference record rather than a single prompt-response sequence.
- We show that audits should preserve preference-record structure because testing each response and their relation reveals leakage that is missed by single-response or single-score audits.
- We identify large differences in empirical leakage across post-training objectives: imitation-heavy objectives leak most, pairwise preference objectives vary substantially, and reward-based policy optimization is near random baseline leakage in our main evaluation.
- We evaluate the impact of parameter-efficient fine-tuning and differential privacy on the empirical privacy-utility trade-offs, showing that they induce highly objective-dependent trade-offs.

2 Background and Related Work

Post-training objectives. Modern LLMs are commonly refined after pretraining to improve instruction following, alignment, and preference adherence. We consider three post-training families. **Supervised fine-tuning** (SFT) is an imitation objective that maximizes the likelihood of curated instruction-response pairs [29]. **Direct preference-optimization** methods use preference feedback: DPO [30] and IPO [3] optimize reference-relative likelihood gaps between preferred and dispreferred responses; ORPO [20] combines preferred-response likelihood maximization with an odds-ratio preference term, and KTO [15] converts preference data into desirable and undesirable examples. **Reinforcement learning**-based objectives, including PPO [31] and GRPO [32], update a policy using reward signals derived from preference data. These objectives differ in which parts of the preference record they imitate, contrast, or reward, and can therefore affect where membership signal appears. Appendix A.1 gives the mathematical forms of the objectives used in this work.

Privacy of RL-based post-training. We hypothesize that PPO and GRPO are inherently more privacy-preserving than the other post-training objectives. Both interact with the full preference record only indirectly, namely via the reward model while updating their policy. The path from a specific training record to the audited policy therefore runs through a low-dimensional, heavily smoothed intermediary, a scalar reward, rather than through a per-record likelihood term as in the other post-training objectives. Furthermore, both PPO and GRPO include a KL-divergence regularization term that further reduces and smooths out the impact of individual records [43].

Membership Inference Attacks (MIAs). MIAs estimate whether a target example was part of a model's training set [8, 34]. For LLMs, reference-free attacks such as Min-K% and Min-K%++ are inexpensive and widely used for auditing pretraining privacy [33, 41]. Stronger empirical audits often rely on reference models. For example LiRA [9] models the MIA decision as a likelihood-ratio test between IN and OUT reference distributions, RMIA [40] adds population normalization, and InfoRMIA [35] uses a continuous information-theoretic score. Prior work has shown that such MIA scorers can reveal leakage in pretraining and supervised fine-tuning [13, 19, 27]. However, they are usually applied to a single sequence or prompt-response pair, whereas preference-based post-training uses structured records. Our benchmark adapts reference model-based MIAs to this setting by varying the record statistic passed to each MIA scorer.

Parameter-Efficient Fine-Tuning (PEFT) methods. PEFT adapts a pretrained model by updating a restricted parameterization rather than all weights. LoRA introduces trainable low-rank updates to frozen weights [21]. Some of its variants modify the scaling rule, rank allocation, sparsity pattern, random basis, or magnitude-direction decomposition of this update [2, 5, 6, 22, 23, 25, 42]. Prior work found that LoRA can leak less than full fine-tuning in some adaptation settings [26]. In our work, we analyze PEFT as a capacity modification that can move a model to a different point on the empirical privacy-utility frontier, based on the capacity chosen. Appendix A.3 summarizes the eight adapter variants used in our benchmark.

Differential Privacy (DP). DP [14] formalizes privacy as a stability guarantee for randomized algorithms, such that changing one training example can change the distribution of the output model only by a bounded amount. DP-SGD [1] is the standard training mechanism for deep networks. It clips

per-example gradients and adds calibrated Gaussian noise before each update. Prior work has studied private adaptation of LLMs through private fine-tuning [24, 39] and prompt-based adaptation [12, 38]. In this paper, we apply DP-SGD to measure empirical privacy risks under formal privacy guarantees. Appendix A.2 provides further details on the mathematical implementation we use in this paper.

Privacy evaluation for preference and adaptation data. Recent work has begun to study privacy leakage from adapted or aligned LLMs. PREMIA [17] introduced a membership signal based on the relative behavior of preferred and dispreferred responses. We include its scalar preference-margin form as a baseline in our benchmark. Other empirical privacy benchmarks study leakage from LLM adaptation [45] and DP LLM adaptation [26]. These works establish that adapted LLMs can leak sensitive data and that empirical privacy depends on the training setup. In contrast, our work focuses on preference-based post-training: we treat the complete preference record as the protected unit and systematically vary the audited statistic, post-training objective, and privacy-utility intervention under one benchmark protocol.

Canaries for privacy analysis. Synthetic canaries are controlled, carefully crafted training examples whose content and provenance are known exactly. They are commonly used to test whether a training procedure memorizes rare or unusual patterns more strongly than ordinary data [7, 8, 28]. In this work, we use canaries to separate ordinary membership leakage from leakage caused by controlled perturbations of the prompt or responses. This is important because different post-training objectives expose different parts of a preference record, and different attack instances ask different privacy questions. By relying on canaries, we can targetly analyze leakage.

3 Benchmark Protocol for Preference-Record Privacy

To faithfully analyze the privacy leakage in preference-based post-training, we adapt scores from existing strong MIAs, to the unit of membership in this paradigm, namely a complete preference record. Throughout the paper, a *record statistic* is the quantity extracted from a preference record, a *MIA scorer* is the membership-scoring rule, and an *attack instance* is a MIA scorer applied to a record statistic. This section first introduces the framework behind these MIA scorers and then presents our adaptations to preference-based post-training.

3.1 Reference-Model Membership Auditing

MIAs can be used to audit whether a target record z was used to train an audited model $f_{\theta^\star}$. Usually, the auditor has access to the log-probability-based scores on the given prompt-response sequences but does not use generated samples, hidden states, or full-vocabulary distributions. We call a record z *IN* if it was included in the model's training data and *OUT* otherwise. Reference-model MIA scorers compare the audited model's behavior on z to matched reference models trained with and without z. For a record statistic $T_\theta(z)$, the audit obtains the value $T_{\theta^\star}(z)$ for the audited model, together with corresponding IN/OUT values from reference models that include or exclude z.

We adapt LiRA [9], RMIA [40], and InfoRMIA [35] as existing strong reference-model MIA scorers. LiRA compares estimated IN and OUT reference distributions to produce a membership score for each target record, while RMIA and InfoRMIA use population-normalized variants of the same reference-model principle. We do not modify these MIA scorers but instead, our **benchmark varies the record statistic $T_\theta(z)$** in four variants, as detailed below.

3.2 Representing Preference Records for Membership Auditing

In contrast to SFT where each record consists of a prompt-response pair, preference-based post-training induces two prompt-response pairs, (x, y_w) and (x, y_l), for the preferred and dispreferred responses. Hence, MIAs must audit the complete preference record to avoid under-reporting the risks. For a model f_θ, let

$$V(f_\theta; x, y) \in \mathbb{R} \tag{1}$$

denote the score extracted from a given prompt-response pair. We write

$$V_w := V(f_\theta; x, y_w), \qquad V_l := V(f_\theta; x, y_l) \tag{2}$$

for the preferred- and dispreferred-response scores. This score is typically the loss or a calibrated variant, such as hinge or stable scores. We compare four statistics:

1) Preferred-response statistic. The preferred-only statistic uses

$$T_\theta^{\mathrm{pref}}(z) := V_w. \tag{3}$$

This is the closest analogue of a standard SFT-style MIA: it tests whether membership is exposed through the rewarded, imitated, or reinforced response. Because it ignores y_l, it can miss leakage visible only in the paired response structure.

2) Dispreferred-response statistic. The dispreferred-only statistic uses

$$T_\theta^{\mathrm{dispref}}(z) := V_l. \tag{4}$$

This statistic asks whether the model exposes membership through the response that is contrasted against during preference learning. It captures privacy evidence that is not visible when the audit only scores the preferred response.

3) Scalar preference-margin statistic. The scalar-margin statistic compresses the preference record into a single number that measures how much the model favors the preferred response over the dispreferred response. Following PREMIA [17], we define the record statistic as

$$T_\theta^{\mathrm{margin}}(z) := u_\theta(z) = \big(\log p_\theta(y_w \mid x) - \log p_0(y_w \mid x)\big) - \big(\log p_\theta(y_l \mid x) - \log p_0(y_l \mid x)\big), \tag{5}$$

where $p_0(\cdot \mid x)$ is a fixed anchor policy. Intuitively, this statistic is large when the audited model assigns a stronger anchor-relative advantage to the preferred response than to the dispreferred response. We then pass this scalar statistic to the same LiRA, RMIA, and InfoRMIA MIA scorers as the other record statistics. Since the anchor contributes only an example-specific constant, the same reference-model membership scores are obtained from the unanchored statistic

$$\tilde{T}_\theta^{\mathrm{margin}}(z) := \log p_\theta(y_w \mid x) - \log p_\theta(y_l \mid x). \tag{6}$$

Appendix B proves that this anchor subtraction leaves the reference-model membership scores unchanged, since it shifts all target and reference statistics for the same record by the same constant.

4) Signed response-pair statistic. The structure-aware statistic keeps the signed contrast between the preferred and dispreferred responses:

$$T_\theta^{\mathrm{pair}}(z) := V_w - V_l. \tag{7}$$

This statistic preserves the sign of the preferred-minus-dispreferred contrast. It asks whether the two responses jointly contain membership signal that is lost when the record is reduced to a single response or scalar margin. The same LiRA, RMIA, and InfoRMIA MIA scorers are applied unchanged to $T_\theta^{\mathrm{pair}}(z)$. Combining the preferred-response and dispreferred-response information before scoring can therefore yield more effective attack instances than scoring either response in isolation.

3.3 Canaries as Controlled Targets

We also evaluate carefully crafted *canaries* as diagnostic target records. Canaries are controlled records whose content and perturbation are known, allowing us to analyze which parts of a preference record create detectable membership evidence. We insert them into the candidate training universe before the reference-model split and audit them with the same IN/OUT protocol as ordinary records.

Our canary design reflects that preference-based post-training combines aspects of both classification and language modeling. To capture classifier-like effects, we create *preference-order corruption* canaries (MisPref), which keep a real prompt and real responses but swap the preferred and dispreferred labels. These canaries test whether the assignment of a response to the preferred or dispreferred side, rather than only the response text itself, can create membership signal. To capture language-model-like memorization, we create *high-entropy string* canaries (RandStr), which replace the prompt (RandP + RealR), both the responses (RealP + RandR), or only the preferred response (RealP + RandPref) with rare random strings that should be hard to predict unless the model has fit the inserted record. Finally, we include *uncorrupted* canaries (Normal), which are ordinary-looking inserted preference records and serve as a baseline for canary insertion without deliberate corruption. Appendix K describes the full construction and gives examples for each canary type.

4 Experimental Setup

Benchmark scope. We evaluate membership privacy for preference-based post-training across three preference datasets, two open model families, and seven post-training objectives. The main evaluation covers four model sizes, and a larger-model subset checks whether the attack-instance comparison persists for Qwen3-4B and Qwen3-8B (see Appendix E). If not indicated otherwise, we implement the post-training through full fine-tuning.

Membership audit configuration. Membership is evaluated at the complete-record level: the protected unit is (x, y_w, y_l), even when the training objective uses only part of the record, as in SFT. We train four reference models for each setting following Tao and Shokri [35]. We then score the record statistics from Section 3 with LiRA, RMIA, and InfoRMIA. Following LiRA's emphasis on high-confidence membership decisions [9], our primary privacy metric is TPR at FPR $= 1\%$; utility is measured by pairwise accuracy on a held-out validation split. We nevertheless report AUC, TPR at FPR $= 0.1\%$, and TPR at FPR $= 0.01\%$ in Appendices N and O.

PEFT and DP training. In addition to the full fine-tuning, we also compare eight adapter variants on Qwen3-0.6B, in combination with each post-training method. We also evaluate DP post-training based on the DP-SGD algorithm [1] for $\varepsilon \in \{0.1, 1, 10, 100\}$ on Qwen3-0.6B. KTO requires an additional private release for its batch-dependent reference quantity, which we detail in Appendix I.1. The remaining objectives are privatized by per-sample clipping and Gaussian noise addition, see Appendix I.

5 Benchmark Design and Experiments

To address our benchmark's central question: *How do record statistic, training objective, and privacy interventions shape empirical membership leakage from preference data?*, we break it down into five research questions. The first two ask what practitioners should audit and which post-training objectives are most exposed. The third asks where leakage appears inside a preference record. The final two ask how PEFT changes empirical privacy-utility trade-offs and how DP changes the formal protection-utility trade-off.

5.1 RQ1: Which record statistic should be used to audit preference data?

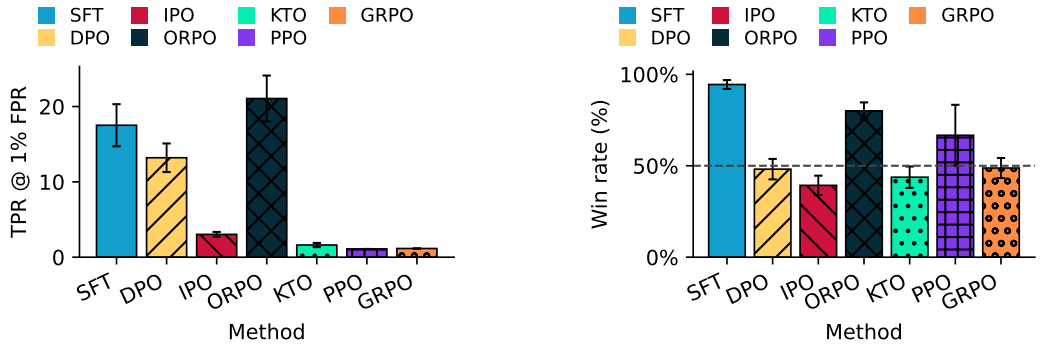
Motivation. Preference records are not ordinary prompt-response examples: they contain a prompt, a preferred response, and a dispreferred response. A privacy audit can therefore score only the preferred response, compress the record into a scalar preference margin, or preserve the signed contrast between the two responses. These choices answer different privacy questions and may expose different leakage.

Summary of findings. When both responses are used, signed response-pair is generally the most effective statistic. The main exception is SFT, where preferred-response is strongest because the objective trains only on the preferred response. Signed response-pair is strongest for DPO, ORPO, and KTO, and is tied with the baseline for PPO and GRPO. Scalar preference-margin is strongest only for IPO, and only by a small margin. However, for larger models the signed response-pair is strongest for all objectives except SFT, where preferred-response remains strongest.

Detailed results. Table 1 compares four record statistics under the same reference-model audit. Each cell reports the strongest attack instance for that statistic and objective. For SFT, preferred-response is strongest at 21.6%, ahead of signed response-pair at 17.3%, consistent with SFT using only the preferred response during training. For DPO, signed response-pair reaches 13.2%, while preferred-response and dispreferred-response are nearly balanced at 7.9% and 8.0%. ORPO follows the same paired-statistic pattern: signed response-pair is strongest at 21.1%, with preferred-response also high at 19.0%. IPO is the main exception, where scalar preference-margin is slightly strongest at 2.3% compared with 2.0% for signed response-pair. KTO, PPO, and GRPO remain close to the baseline across statistics, although signed response-pair is the largest KTO statistic at 1.6%. The larger-model subset in Appendix E shows that this pattern transfers to Qwen3-4B and Qwen3-8B for all objectives except SFT, where preferred-response remains the most effective statistic. The

Table 1: **Signed response-pair is generally strongest.** Average TPR at FPR$= 1\%$ for each record statistic and post-training objective. Each cell uses the strongest MIA scorer for that statistic and objective; bold marks the strongest statistic per objective.

Record Statistic	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
Preferred-response	**21.6 ± 3.6**	7.9 ± 1.0	1.5 ± 0.1	19.0 ± 3.4	1.3 ± 0.1	**1.1 ± 0.0**	**1.1 ± 0.0**	8.8 ± 1.1
Dispreferred-response	1.7 ± 0.1	8.0 ± 1.2	1.6 ± 0.1	5.5 ± 1.3	1.2 ± 0.1	**1.1 ± 0.0**	**1.1 ± 0.0**	3.1 ± 0.3
Scalar preference-margin	8.7 ± 2.1	2.1 ± 0.3	**2.3 ± 0.3**	8.7 ± 1.9	1.1 ± 0.0	1.0 ± 0.0	**1.1 ± 0.0**	4.0 ± 0.5
Signed response-pair	17.3 ± 2.8	**13.2 ± 1.9**	2.0 ± 0.3	**21.1 ± 3.1**	**1.6 ± 0.3**	**1.1 ± 0.0**	**1.1 ± 0.0**	**9.3 ± 1.0**



(a) Objective-level leakage.

(b) Preferred-response win rate.

Figure 2: **Post-training objectives differ in both leakage and response-side dominance.** Left: average TPR at FPR$= 1\%$ on the main evaluation set after selecting the strongest attack instance for each run. Right: reports how often the preferred response carries the larger signal.

hyperparameter ablations in Appendix M show that these results remain broadly consistent across the tested objective hyperparameters. We also find that the score feature affects MIA success in our benchmark. In Appendix H, Table 7 shows that cross-entropy is strongest overall.

5.2 RQ2: Which post-training objectives expose the most membership signal?

Motivation. Post-training objectives use preference records in different ways. Some directly imitate the preferred response, some optimize a pairwise preference signal, and others update a policy through reward-based learning. These differences may change how much record-specific information is encoded in the final model.

Summary of findings. The choice of post-training objective is one of the strongest determinants of empirical leakage. In our main evaluation, SFT and ORPO expose the most membership signal, DPO still leaks substantially under structure-aware record statistics, IPO is much less exposed, and MIAs against KTO, PPO, and GRPO perform close to random guessing.

Detailed results. Figure 2a gives average TPR at FPR$= 1\%$ by post-training objective, using the strongest attack instance per run. ORPO and SFT are the most exposed objectives, with average TPRs of 21.1% and 17.5%. DPO reaches 13.2%, showing that pairwise preference optimization can still leak substantially when audited with a structure-aware statistic. IPO is much lower at 3.0%. KTO, PPO, and GRPO remain close to random guessing at the same false-positive rate, with TPRs of 1.6%, 1.1%, and 1.2%. The privacy-utility view in Figure 3 shows that IPO yields the best privacy-utility trade-off, namely high utility with low leakage.

Finally, we also report a separate sequence-length sensitivity analysis in Appendix G. It shows that, across all objectives, we see the common pattern that leakage often rises with available response length and then plateaus. These results suggest that leakage is both a function of the preference data and the objective and depends on how the objective uses the preferred response, the dispreferred response, and any reference or reward model.

Table 2: **Canary-type leakage.** TPR at FPR$= 1\%$ for controlled canary types. The types are uncorrupted inserted records (Normal), high-entropy strings (RandStr), preference-order corruption (MisPref), random prompt with real responses (RandP + RealR), real prompt with random responses (RealP + RandR), and real prompt with only the preferred response randomized (RealP + RandPref).

Method	Normal	RandStr	MisPref	RandP + RealR	RealP + RandR	RealP + RandPref
SFT	2.0 ± 0.0	1.3 ± 0.0	3.5 ± 0.0	$\mathbf{3.7 \pm 0.0}$	2.6 ± 0.0	3.1 ± 0.0
DPO	3.2 ± 0.0	9.3 ± 0.0	$\mathbf{12.8 \pm 0.0}$	6.1 ± 0.0	7.9 ± 0.0	3.1 ± 0.0
IPO	4.5 ± 0.0	$\mathbf{10.6 \pm 0.0}$	9.5 ± 0.0	2.1 ± 0.0	4.4 ± 0.0	2.6 ± 0.0
ORPO	3.2 ± 0.0	$\mathbf{3.4 \pm 0.0}$	1.5 ± 0.0	1.9 ± 0.0	2.8 ± 0.0	1.5 ± 0.0
All	3.2 ± 0.5	6.2 ± 2.2	$\mathbf{6.8 \pm 2.6}$	3.5 ± 1.0	4.4 ± 1.2	2.6 ± 0.4

5.3 RQ3: Where does membership signal appear within a preference record?

Motivation. A preference record contains a prompt, a preferred response, a dispreferred response, and the assignment of each response to its side of the comparison. Leakage may therefore arise from the text of one response, from the relation between the two responses, or from the preference ordering itself. Identifying these sources helps explain why single-response audits can under-report risk and why post-training objectives differ in leakage.

Summary of findings. On naturally occurring (*i.e.,* non-canary) records, SFT and ORPO leak mainly through the preferred response, while DPO shows a more balanced signal across the preferred and dispreferred responses. Our controlled *canary* analysis further suggests that, for several objectives, response-side perturbations are more detectable than prompt-side perturbations, and perturbing both responses can create more signal than perturbing only the preferred response. Together, these results suggest that membership evidence in preference-based post-training is often carried by the response components and their role in the preference objective, not only by the preferred response in isolation.

Detailed results. Figure 2b shows how often the preferred response carries the stronger single-response membership signal. Preferred responses dominate in 94.4% of SFT comparisons and 80.0% of ORPO comparisons. This is consistent with how these objectives use the data: SFT trains only on the preferred response, while ORPO combines preferred-response imitation with a pairwise preference term. In contrast, DPO, IPO, KTO, PPO, and GRPO are much closer to parity between preferred and dispreferred responses. The full preferred-versus-dispreferred comparison in Appendix F shows the same pattern: SFT reaches 21.6% TPR on preferred responses but only 1.7% on dispreferred responses, while ORPO reaches 19.0% and 5.5%, respectively. DPO is nearly balanced, with about 7.9% TPR on both response sides. Thus, for objectives that directly imitate the preferred response, leakage is concentrated on that side, while for pairwise preference objectives, membership evidence can be distributed across both responses.

To complement this analysis on naturally occurring records, we use our canaries as controlled probes. Table 2 reports the canary results. For DPO, IPO, and ORPO, canaries with a real prompt and random responses yield higher leakage than canaries with a random prompt and real responses: 7.9% vs. 6.1% for DPO, 4.4% vs. 2.1% for IPO, and 2.8% vs. 1.9% for ORPO. This suggests that, for these objectives, response-side perturbations are more detectable than prompt-side perturbations. Moreover, corrupting both responses is stronger than corrupting only the preferred response for the same objectives: 7.9% vs. 3.1% for DPO, 4.4% vs. 2.6% for IPO, and 2.8% vs. 1.5% for ORPO. These controlled perturbations support the natural-record finding that membership evidence in preference-based post-training is not fully explained by the preferred response alone; it can also arise from the response pair and from how the objective uses the two responses. Additional canary diagnostics, including the full preference-order and high-entropy memorization results, are reported in Appendix K.1.

5.4 RQ4: To what degree do PEFT methods reduce empirical privacy leakage?

Motivation. PEFT constrains adaptation capacity. This may reduce the ability to fit individual records and thereby reduce empirical privacy leakage. Practitioners therefore need to know whether adapter choice reliably improves privacy or simply changes the privacy-utility trade-off.

Table 3: **PEFT privacy-utility**.

Adapter	$\mathrm{TPR}_{1\%} \downarrow$	Acc. $\uparrow$	# Pareto $\uparrow$
Full FT	14.1	54.6	5/15
LoRA	11.5	**55.8**	4/15
RSLoRA	13.7	55.3	5/15
DoRA	11.5	55.6	6/15
VeRA	**1.3**	54.4	**11/15**
AdaLoRA	1.6	54.5	**11/15**
RandLoRA	10.1	54.7	8/15
ShiRA	10.0	55.3	5/15
DeLoRA	17.3	54.9	9/15

Summary of findings. The choice of the PEFT method creates a broad privacy-utility frontier: some configurations reduce measured leakage, while others expose substantial membership signal, especially for objectives that already leak strongly. VeRA and AdaLoRA provide the strongest empirical protection and the best Pareto frontier among the evaluated adapters. The LoRA-rank sweep suggests that trainable capacity is an important factor, with lower-rank configurations often reducing leakage.

Detailed results. Table 3 reports the PEFT privacy-utility summary, and Figure 13 gives the corresponding privacy-utility frontier. The TPR column averages the strongest measured membership leakage for each run, while the accuracy column averages held-out pairwise accuracy. The Pareto column counts how often an adapter is non-dominated within a fixed experimental setting: an adapter is Pareto-optimal if no other adapter in the same setting achieves both lower or equal leakage and higher or equal utility, with at least one strict improvement. Thus, a higher Pareto count means that the adapter more often lies on the best empirical privacy-utility frontier, not that it uniformly dominates all alternatives. Our results show that PEFT configurations vary widely in both pairwise accuracy and membership leakage. VeRA and AdaLoRA have the lowest aggregate leakage and highest Pareto counts in this grid, while some objective-adapter combinations still reach high privacy risk, especially for SFT and ORPO. PEFT can help navigate the privacy-utility frontier, but should be audited rather than assumed private. The LoRA-rank sweep in Figure 15 further suggests that adaptation capacity matters: lower-rank configurations often reduce measured leakage relative to higher-capacity configurations, though the effect is objective-dependent. VeRA and AdaLoRA are also consistent with this capacity-based interpretation, since both constrain the trainable update more strongly than standard LoRA. We provide the adapter-specific details and extended analysis in Appendices L and M.

5.5 RQ5: What are the privacy-utility trade-offs of formal privacy guarantees?

Motivation. DP is the standard formal framework for privacy protection, but empirical leakage and utility can still vary with the objective, privacy budget, and data distribution. For preference-based post-training, the relevant trade-off can differ from pretraining or ordinary fine-tuning because the objectives use the preferred and dispreferred responses asymmetrically.

Summary of findings. DP-SGD creates different privacy-utility trade-offs across privacy budgets and objectives. In our sweep, all evaluated attacks remain close to random guessing, while utility varies widely.

Detailed results. Figure 11 plots pairwise accuracy against TPR at FPR$= 1\%$ for DP training with $\varepsilon \in \{0.1, 1, 10, 100\}$. Across SFT, DPO, IPO, ORPO, and KTO, the TPR stays below 3% while pairwise accuracy ranges from roughly 51% to 58%, so DP settings should be compared as privacy-utility trade-offs rather than by leakage alone. The low measured leakage may partly reflect the small clipping norm used in these runs, not only the noise level. SFT is the clearest exception: lowering ε visibly reduces leakage but also lowers utility. KTO, IPO, and DPO provide the clearest Pareto-relevant operating points in the plotted range.

6 Discussion and Conclusion

We benchmark membership privacy in preference-based post-training, treating the full preference record as the protected unit rather than reducing it to a single prompt-response sequence. Our results show that this distinction is essential. Audits based only on the preferred response, the dispreferred response, or a single scalar preference score can give incomplete risk estimates. In contrast, structure-aware audits that test both responses and their relation reveal membership signal that would otherwise be missed. Practical privacy evaluations should therefore report which part of the preference record is audited and should preserve the preferred/dispreferred structure of the data.

We also find that empirical leakage depends strongly on the post-training objective. Objectives that directly imitate the preferred response, or combine imitation with pairwise preference learning, expose the most membership signal in our main evaluation. Pairwise preference objectives vary substantially, with more regularized variants showing much lower leakage. MIAs against objectives based on binary feedback or reward-driven policy optimization are close to random guessing. These differences show that privacy risk is not only a property of the dataset, but also of how the training objective uses the preferred and dispreferred responses.

Finally, we study how training choices and privacy interventions shape the privacy-utility trade-off. Parameter-efficient fine-tuning changes both adaptation capacity and leakage, producing a wide range of outcomes across objectives and adapters. Differentially private training provides formal protection for complete preference records, but its utility cost highly depends on the objective and privacy budget. Controlled canaries further show that both mislabeled preference examples and high-entropy strings can create detectable membership evidence. Overall, our findings support privacy assessments that preserve preference-record structure, evaluate leakage together with utility, and account for the specific post-training objective and training paradigm.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[2] Paul Albert, Frederic Z. Zhang, Hemanth Saratchandran, Cristian Rodriguez-Opazo, Anton van den Hengel, and Ehsan Abbasnejad. RandloRA: Full rank parameter-efficient fine-tuning of large models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=Hn5eoTunHN`.

[3] 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*, pages 4447–4455. PMLR, 2024.

[4] 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. *arXiv preprint arXiv:2204.05862*, 2022.

[5] Kartikeya Bhardwaj, Nilesh Pandey, Sweta Priyadarshi, Viswanath Ganapathy, Shreya Kadambi, Rafael Esteves, Shubhankar Borse, Paul Whatmough, Risheek Garrepalli, Mart van Baalen, et al. Sparse high rank adapters. *Advances in Neural Information Processing Systems*, 37: 13685–13715, 2024.

[6] Massimo Bini, Leander Girrbach, and Zeynep Akata. Decoupling angles and strength in low-rank adaptation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=X1U74IwuxG`.

[7] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 267–284, Santa Clara, CA, August 2019. USENIX Association. ISBN 978-1-939133-06-9. URL `https://www.usenix.org/conference/usenixsecurity19/presentation/carlini`.

[8] 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. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[9] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

[10] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. Chatbot arena: An open platform for evaluating llms by human preference, 2024.

[11] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL `https://openreview.net/forum?id=BOorDpKHiJ`.

[12] Haonan Duan, Adam Dziedzic, Nicolas Papernot, and Franziska Boenisch. Flocks of stochastic parrots: Differentially private prompt learning for large language models. *Advances in Neural Information Processing Systems*, 36:76852–76871, 2023.

[13] Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*, 2024.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

[15] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*. PMLR, 2024.

[16] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.

[17] 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. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan, editors, *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pages 5221–5229. PMLR, 03–05 May 2025. URL `https://proceedings.mlr.press/v258/feng25a.html`.

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

[19] Jamie Hayes, Ilia Shumailov, Christopher A Choquette-Choo, Matthew Jagielski, Georgios Kaissis, Milad Nasr, Meenatchi Sundaram Muthu Selva Annamalai, Niloofar Mireshghallah, Igor Shilov, Matthieu Meeus, et al. Exploring the limits of strong membership inference attacks on large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[20] Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11170–11189, 2024.

[21] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[22] Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv preprint arXiv:2312.03732*, 2023.

[23] Dawid Jan Kopiczko, Tijmen Blankevoort, and Yuki M Asano. VeRA: Vector-based random matrix adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNfLdxr3A.

[24] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations*, 2022.

[25] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. 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 32100–32121. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/liu24bn.html.

[26] Bartłomiej Marek, Lorenzo Rossi, Vincent Hanke, Xun Wang, Michael Backes, Franziska Boenisch, and Adam Dziedzic. Benchmarking empirical privacy protection for adaptations of large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=jY7fAo9rfK.

[27] Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343, 2023.

[28] Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre De Montjoye. Copyright traps for large language models. In *International Conference on Machine Learning*, pages 35296–35309. PMLR, 2024.

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

[30] 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.

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

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[33] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zWqr3MQuNs.

[34] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.

[35] Jiashu Tao and Reza Shokri. (token-level) \textbf{InfoRMIA}: Stronger membership inference and privacy assessment for LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4KVeb0Vv13.

[36] Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

[37] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[38] Tong Wu, Ashwinee Panda, Jiachen T Wang, and Prateek Mittal. Privacy-preserving in-context learning for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[39] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. In *International Conference on Learning Representations*, 2022.

[40] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Proceedings of the 41st International Conference on Machine Learning*, pages 58244–58282, 2024.

[41] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for pre-training data detection from large language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=ZGkfoufDaU.

[42] Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

[43] Yizhou Zhang, Kishan Panaganti, Laixi Shi, Juba Ziani, and Adam Wierman. KL-regularization itself is differentially private in bandits and RLHF, 2025. URL https://arxiv.org/abs/2505.18407.

[44] Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36:55006–55021, 2023.

[45] Derui Zhu, Dingfan Chen, Xiongfei Wu, Jiahui Geng, Zhuo Li, Jens Grossklags, and Lei Ma. Privauditor: Benchmarking data protection vulnerabilities in LLM adaptation techniques. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=VpkfxuVXwx.

A Extended Related Work

A.1 Post-Training Objectives

Supervised fine-tuning. SFT [29] adapts a pretrained language model to follow instructions by minimizing the negative log-likelihood on supervised input–output pairs. Given a dataset $\mathcal{D}$ of prompts x and reference responses y, the objective is

$$L_{\mathrm{SFT}} = -\mathbb{E}_{(x,y)\sim\mathcal{D}} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}). \tag{8}$$

In our setting, SFT is the single-response baseline for the preference-based comparisons.

Direct preference optimization. DPO [30] optimizes a logistic loss on the reference-relative log-likelihood gap between a preferred response y_w and a dispreferred response y_l:

$$L_{\mathrm{DPO}} = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \log \sigma\Big(\beta\Big[\log \tfrac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log \tfrac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)}\Big]\Big). \tag{9}$$

DPO is central here because its training signal depends directly on the relation between the preferred and dispreferred responses.

Identity preference optimization. IPO [3] uses the same reference-relative preference gap as DPO but replaces the logistic loss with a squared loss:

$$L_{\mathrm{IPO}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left(\log \tfrac{\pi_\theta(y_w|x)}{\pi_{\mathrm{ref}}(y_w|x)} - \log \tfrac{\pi_\theta(y_l|x)}{\pi_{\mathrm{ref}}(y_l|x)} - \tfrac{1}{\tau}\right)^2. \tag{10}$$

This gives a preference-based objective with different curvature but the same pairwise structure.

Odds ratio preference optimization. ORPO [20] combines supervised imitation on the preferred response with an odds-ratio term that contrasts the preferred and dispreferred responses:

$$L_{\mathrm{ORPO}} = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[-\log \pi_\theta(y_w \mid x) - \lambda \log \sigma\Big(\log \tfrac{\mathrm{odds}_\theta(y_w|x)}{\mathrm{odds}_\theta(y_l|x)}\Big)\right]. \tag{11}$$

Because ORPO mixes imitation and pairwise preference learning, it is useful for comparing single-response and preference-aware membership signals.

Kahneman–Tversky optimization. KTO [15] converts preference data into binary desirable and undesirable examples and optimizes a prospect-theoretic value function. Writing $s \in \{+1, -1\}$ for the label and $r_\theta(x, y) = \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}$, the objective is

$$L_{\mathrm{KTO}} = -\mathbb{E}_{(x,y,s)\sim\mathcal{D}} v\big(s \cdot (r_\theta(x, y) - r_0(x))\big), \tag{12}$$

where $v(\cdot)$ is the Kahneman–Tversky value function and $r_0(x)$ is a batch-estimated reference point. KTO uses the same membership-evaluation protocol as the other objectives, but its batch-dependent reference point requires the KTO-specific DP construction in Appendix I.1.

Proximal policy optimization. PPO [31] is a policy-gradient method used in reinforcement learning from human feedback. With policy ratio $r_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$ and advantage estimate $\hat{A}_t$, PPO optimizes the clipped surrogate objective

$$L_{\mathrm{PPO}} = \mathbb{E}_t\Big[\min\big(r_t(\theta)\hat{A}_t,\ \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\big)\Big]. \tag{13}$$

In our evaluation, a learned reward model trained on the preference data supplies the reward signal.

Group relative policy optimization. GRPO [32] normalizes rewards across multiple sampled responses for the same prompt and avoids a separate value function. With grouped advantage estimate $\hat{A}_t^{\mathrm{group}}$, a generic form of the objective is

$$L_{\mathrm{GRPO}} = \mathbb{E}\big[r_t(\theta)\hat{A}_t^{\mathrm{group}} - \beta\,\mathrm{KL}(\pi_\theta \,\|\, \pi_{\mathrm{ref}})\big]. \tag{14}$$

GRPO is relevant because it remains comparative while changing how the reward signal is aggregated.

A.2 DP and DP-SGD

DP [14] is the standard formal framework for releasing training artifacts with a worst-case privacy guarantee. It bounds how much the output distribution of a randomized algorithm can change when one protected record is added or removed. In neural language modeling, the canonical training mechanism is DP-SGD [1], which clips per-example gradients and adds calibrated Gaussian noise to each update. Prior work applies this idea to private LLM fine-tuning and prompt-based adaptation [12, 24, 38, 39]. These works highlight the same basic trade-off we study empirically: DP gives a formal guarantee over complete preference records, while clipping, noise, and per-example gradient computation can change both utility and training cost.

In our setting, the protected record is the complete preference tuple $z = (x, y_w, y_l)$. Let $\mathcal{Z}$ denote the universe of such records, and let $D \in \mathcal{Z}^N$ be a post-training dataset. We use add/remove adjacency over complete preference records: $D \simeq D'$ if one dataset can be obtained from the other by adding or removing one complete preference record. A randomized training algorithm $M : \mathcal{Z}^N \to \Theta$ is (ε, δ)-DP if, for every adjacent pair $D \simeq D'$ and every measurable set of model parameters $E \subseteq \Theta$,

$$\Pr[M(D) \in E] \leq e^{\varepsilon} \Pr[M(D') \in E] + \delta. \tag{15}$$

Thus the protected unit in this work is the full preference record, even when a particular objective, such as SFT, reads only the preferred response during optimization. The guarantee is closed under post-processing: once the training algorithm has released a private model, any deterministic or randomized computation on that model remains protected by the same (ε, δ) guarantee.

We instantiate this guarantee with Poisson-subsampled DP-SGD. At step t, each record is sampled independently with probability q, giving a Poisson batch S_t and expected batch size $\bar{m} = qN$. For a per-record post-training loss $\ell_i(\theta) = \ell(\theta; z_i)$, the unclipped gradient is

$$g_i = \nabla_\theta \ell_i(\theta). \tag{16}$$

The mechanism clips each record contribution to norm C_g,

$$\bar{g}_i = g_i \min\left(1, \frac{C_g}{\|g_i\|_2}\right), \tag{17}$$

and releases the noisy mean

$$\tilde{g}_t = \frac{1}{\bar{m}} \sum_{i \in S_t} \bar{g}_i + \mathcal{N}\left(0, \sigma_g^2 \left(\frac{C_g}{\bar{m}}\right)^2 I\right). \tag{18}$$

Under add/remove adjacency, the fixed-batch sensitivity of the mean clipped gradient is $C_g/\bar{m}$. The noise multiplier σ_g, sampling rate q, and number of update steps determine the final ε for a fixed δ through an accountant for the subsampled Gaussian mechanism. The model update is post-processing of $\tilde{g}_t$, so privacy composes over training steps through the accountant.

SFT, DPO, IPO, and ORPO fit this mechanism once the complete preference record is fixed: each sampled record contributes an objective-specific per-record loss and gradient. KTO is handled separately because its loss uses a batch-dependent reference point. Appendix I.1 gives the private cyclic release used for that reference quantity before the gradient step is computed. PPO and GRPO are not included in the DP sweep in this paper. We use DP-SGD as the formal privacy mechanism and test whether strong MIAs can still detect residual membership.

A.3 PEFT Methods

PEFT methods constrain the update applied to a pretrained model. For a linear layer with frozen pretrained weight $W_0 \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, full fine-tuning learns an unconstrained update ΔW and uses $W_0 + \Delta W$. PEFT instead restricts ΔW to a lower-dimensional or structured family. This reduces memory and compute and changes the effective capacity available for fitting individual records. Unlike DP-SGD, PEFT is not a formal privacy mechanism. We use it as an empirical modification of the privacy-utility frontier.

LoRA. LoRA [21] freezes W_0 and learns

$$\Delta W = \frac{\alpha}{r} BA, \tag{19}$$

where $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $B \in \mathbb{R}^{d_{\text{out}} \times r}$, r is the rank, and α is a scaling parameter. The rank controls both trainable parameter count and the dimension of the update subspace.

RSLoRA. RSLoRA [22] changes the LoRA scaling rule, commonly replacing α/r with $\alpha/\sqrt{r}$. This stabilizes the effective update scale as rank changes.

DoRA. DoRA [25] decomposes adaptation into magnitude and direction components. The low-rank update primarily adjusts direction, while a separate magnitude term changes the weight norm.

VeRA. VeRA [23] uses frozen random projection matrices and trains compact scaling vectors. This sharply reduces the number of trainable parameters while retaining a structured random adapter basis.

AdaLoRA. AdaLoRA [42] allocates a rank budget adaptively across layers during training. The method can concentrate capacity in layers where the low-rank update appears most useful.

RandLoRA. RandLoRA [2] uses randomized low-rank structure to obtain a PEFT update with broader effective rank. It changes the adapter basis while keeping trainable parameter count low.

ShiRA. ShiRA [5] uses sparse high-rank adapters rather than dense low-rank factors. It separates rank from trainable density, making it useful for distinguishing rank from parameter count.

DeLoRA. DeLoRA [6] decouples angular and strength components in low-rank adaptation. This changes how update direction and update magnitude are represented during fine-tuning.

These distinctions matter because the privacy signal measured by MIAs is sensitive to how far and in which directions a model can move toward individual training records. Our PEFT experiments compare adapters under the same target examples, metrics, and reference-model protocol used for full fine-tuning, varying only the adaptation parameterization.

B Scalar Preference-Margin and Anchor Invariance

This appendix formalizes why the fixed anchor policy in the scalar preference-margin statistic can be removed without changing the scores of the reference-based MIAs considered in this work.

For a target example $z = (x, y_w, y_l)$, define

$$u_\theta(z) = \big(\log p_\theta(y_w \mid x) - \log p_0(y_w \mid x)\big) - \big(\log p_\theta(y_l \mid x) - \log p_0(y_l \mid x)\big) \tag{20}$$

and

$$\tilde{u}_\theta(z) = \log p_\theta(y_w \mid x) - \log p_\theta(y_l \mid x). \tag{21}$$

Let $c(z) = \log p_0(y_w \mid x) - \log p_0(y_l \mid x)$. Then $u_\theta(z) = \tilde{u}_\theta(z) - c(z)$, where $c(z)$ is fixed for the target example and does not depend on the model θ. Lemma 1 states the resulting invariance property for LiRA, RMIA, and InfoRMIA.

Lemma 1 (Anchor invariance)**.** *Fix a target example z. Let every target and reference statistic $\tilde{u}_\theta(z)$ be replaced by $u_\theta(z) = \tilde{u}_\theta(z) - c(z)$ for the same constant $c(z)$. The LiRA, RMIA, and InfoRMIA membership scores computed from these statistics are unchanged.*

Proof. For LiRA, let $\{\tilde{u}_j^{\mathrm{in}}\}$ and $\{\tilde{u}_j^{\mathrm{out}}\}$ be the IN and OUT reference statistics, and let $\tilde{u}_*$ be the target statistic. LiRA fits location-equivariant density estimates to the two reference sets. For Gaussian LiRA, the transformed IN mean is $\mu_{\mathrm{in}} - c(z)$, the transformed OUT mean is $\mu_{\mathrm{out}} - c(z)$, and the variances are unchanged. Therefore

$$\frac{p_{\mathrm{in}}(\tilde{u}_* - c(z); \mu_{\mathrm{in}} - c(z), \sigma_{\mathrm{in}}^2)}{p_{\mathrm{out}}(\tilde{u}_* - c(z); \mu_{\mathrm{out}} - c(z), \sigma_{\mathrm{out}}^2)} = \frac{p_{\mathrm{in}}(\tilde{u}_*; \mu_{\mathrm{in}}, \sigma_{\mathrm{in}}^2)}{p_{\mathrm{out}}(\tilde{u}_*; \mu_{\mathrm{out}}, \sigma_{\mathrm{out}}^2)}. \tag{22}$$

The LiRA score is unchanged.

For RMIA and InfoRMIA, the scalar statistic enters through model-relative likelihood ratios. Replacing $\tilde{u}_\theta(z)$ by $u_\theta(z)$ multiplies every exponentiated statistic by the same positive factor $\exp(-c(z))$. This factor appears in both the target term and every population-normalization term for the fixed z. All normalized ratios are therefore unchanged. RMIA is a deterministic function of these normalized ratios and InfoRMIA applies a deterministic information correction to the same normalized quantities. Their scores are unchanged. □

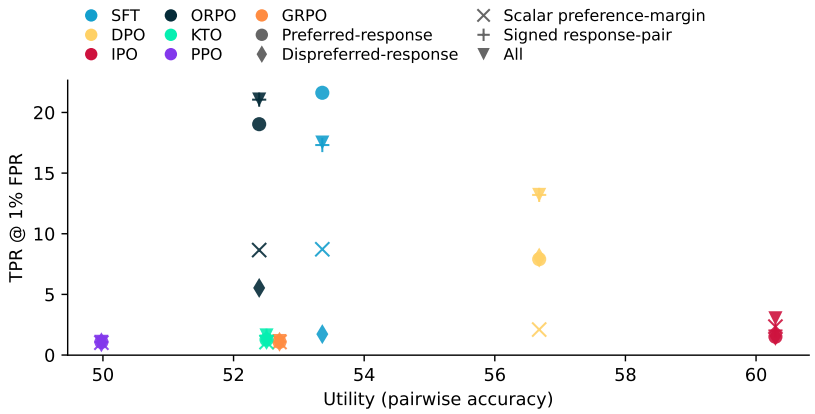


Figure 3: **Record statistic changes measured privacy at the same utility.** Average pairwise accuracy against average TPR at FPR$= 1\%$ for the main record statistics.

Table 4: **Full attack-instance breakdown.** Average TPR at FPR$= 1\%$ on the main evaluation set. Rows are attack instances, columns are post-training objectives, and entries report mean $\pm$ SE.

Attack	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
LiRA Preferred-response	10.3 ± 2.4	2.6 ± 0.4	1.0 ± 0.0	9.9 ± 2.6	1.0 ± 0.0	$\mathbf{1.1 \pm 0.0}$	$\mathbf{1.0 \pm 0.0}$	4.3 ± 0.7
RMIA Preferred-response	$\mathbf{21.6 \pm 3.6}$	7.9 ± 1.0	1.5 ± 0.1	19.0 ± 3.4	1.2 ± 0.1	1.0 ± 0.1	$\mathbf{1.0 \pm 0.0}$	$\mathbf{8.8 \pm 1.1}$
InfoRMIA Preferred-response	21.1 ± 3.6	6.8 ± 1.0	1.3 ± 0.1	18.6 ± 3.4	1.2 ± 0.1	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	8.4 ± 1.1
LiRA Dispreferred-response	1.2 ± 0.0	2.9 ± 0.4	1.0 ± 0.0	2.2 ± 0.4	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	1.6 ± 0.1
RMIA Dispreferred-response	1.7 ± 0.1	8.0 ± 1.2	1.6 ± 0.1	5.4 ± 1.3	1.2 ± 0.1	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	3.1 ± 0.3
InfoRMIA Dispreferred-response	1.7 ± 0.1	6.8 ± 1.1	1.4 ± 0.1	4.6 ± 1.1	1.1 ± 0.1	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	2.7 ± 0.3
LiRA Scalar preference-margin	6.2 ± 1.8	1.2 ± 0.1	1.6 ± 0.2	6.9 ± 1.8	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	3.0 ± 0.5
RMIA Scalar preference-margin	8.6 ± 2.1	1.9 ± 0.2	$\mathbf{2.3 \pm 0.2}$	8.4 ± 1.9	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	3.9 ± 0.5
InfoRMIA Scalar preference-margin	8.6 ± 2.1	2.0 ± 0.3	2.2 ± 0.3	7.2 ± 1.4	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	3.7 ± 0.5
LiRA Signed response-pair	7.6 ± 1.6	1.2 ± 0.0	1.0 ± 0.0	3.7 ± 0.8	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	2.6 ± 0.4
RMIA Signed response-pair	14.3 ± 2.3	12.3 ± 1.7	2.0 ± 0.3	$\mathbf{20.6 \pm 3.0}$	$\mathbf{1.6 \pm 0.3}$	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	8.5 ± 0.9
InfoRMIA Signed response-pair	16.8 ± 2.7	$\mathbf{13.0 \pm 1.9}$	1.8 ± 0.2	18.3 ± 3.0	1.4 ± 0.2	1.0 ± 0.0	$\mathbf{1.0 \pm 0.0}$	8.7 ± 0.9

C Reference-Model Evaluation for PPO and GRPO

This appendix describes how PPO and GRPO are evaluated under the same membership game as the other post-training objectives. The key point is that the target example remains the original preference example even though policy optimization uses generated responses and reward-model scores.

For each setting, we train one reward model on the corresponding preference dataset and use it for the target model and all reference models in that setting. The reward model is treated as a fixed component of the training procedure, so the membership claim for PPO and GRPO is conditional on that reward model and concerns whether the target preference record was included in the policy-training data. The reference models differ through their policy-training data, which are sampled by the same inclusion rule used in the rest of the evaluation. During policy optimization, prompts are used to sample responses and the reward model supplies the scalar reward. During membership evaluation, the MIA is always evaluated on the original target example (x, y_w, y_l). This keeps the target examples, utility metric, and reference-model protocol aligned across SFT, direct preference objectives, PPO, and GRPO.

D Full MIA Comparison on the Main Evaluation Set

This appendix collects the attack-instance comparison used by the main analysis. Figure 3 plots the record-statistic privacy-utility view. Table 4 reports the corresponding attack instances.

Table 5: **The larger-model subset tests whether the attack-instance comparison persists at larger scale.** Average TPR at FPR$= 1\%$ on Qwen3-4B and Qwen3-8B. Rows are attack instances, columns are post-training objectives, and entries report mean $\pm$ SE. The standard errors are computed across models and datasets; Appendix T reports the per-model and per-dataset breakdown.

Attack	SFT	DPO	IPO	ORPO	All
LiRA Preferred-response	18.8 ± 9.9	2.9 ± 1.0	1.2 ± 0.1	1.2 ± 0.1	5.6 ± 2.6
RMIA Preferred-response	$\mathbf{25.7 \pm 11.8}$	8.1 ± 3.0	4.7 ± 1.7	1.7 ± 0.2	9.8 ± 3.2
InfoRMIA Preferred-response	25.5 ± 11.9	7.3 ± 2.6	4.3 ± 1.6	1.5 ± 0.1	9.3 ± 3.2
LiRA Dispreferred-response	1.4 ± 0.2	3.0 ± 1.3	1.1 ± 0.1	1.0 ± 0.0	1.6 ± 0.3
RMIA Dispreferred-response	3.5 ± 1.0	9.1 ± 3.3	5.4 ± 2.0	1.3 ± 0.1	5.0 ± 1.1
InfoRMIA Dispreferred-response	3.4 ± 1.0	7.6 ± 3.1	4.7 ± 1.8	1.0 ± 0.0	4.4 ± 1.0
LiRA Scalar preference-margin	17.8 ± 8.7	1.6 ± 0.4	1.7 ± 0.2	1.1 ± 0.1	5.3 ± 2.3
RMIA Scalar preference-margin	19.9 ± 8.7	2.9 ± 1.1	2.9 ± 0.4	1.4 ± 0.1	6.5 ± 2.4
InfoRMIA Scalar preference-margin	20.0 ± 8.7	2.5 ± 0.8	2.8 ± 0.3	1.7 ± 0.2	6.5 ± 2.4
LiRA Signed response-pair	17.5 ± 8.6	2.0 ± 0.5	1.3 ± 0.1	1.1 ± 0.1	5.2 ± 2.3
RMIA Signed response-pair	16.8 ± 5.8	$\mathbf{12.0 \pm 3.8}$	$\mathbf{7.5 \pm 3.1}$	$\mathbf{1.8 \pm 0.2}$	9.6 ± 2.1
InfoRMIA Signed response-pair	20.9 ± 8.8	11.6 ± 4.1	6.8 ± 2.8	1.4 ± 0.1	$\mathbf{10.1 \pm 2.7}$

Table 6: **The record-statistic and objective orderings persist at 32B.** TPR at FPR$= 1\%$ on Qwen3-32B trained on Chatbot Arena. Rows are record statistics, columns are post-training objectives, and each entry is the maximum over MIA scorers.

Record statistic	SFT	DPO	IPO	ORPO
Preferred-response	**32.3**	6.7	1.1	16.3
Dispreferred-response	1.3	6.3	1.1	1.2
Scalar preference-margin	2.2	1.3	**3.4**	2.1
Signed response-pair	27.8	**26.0**	1.1	**19.0**

E Full MIA Comparison on the Larger-Model Subset

This appendix repeats the full attack-instance breakdown on the larger-model subset, which contains Qwen3-4B, Qwen3-8B, and Qwen3-32B. PPO, GRPO, and KTO are omitted from this subset for computational reasons. The tables are used to check whether the attack-instance comparison observed in the main evaluation persists for larger Qwen3 models. Table 5 reports this larger-model comparisonfor Qwen3-4B and Qwen3-8B. To test a further increase in scale, we additionally evaluate Qwen3-32B, four times larger than Qwen3-8B, on Chatbot Arena with SFT, DPO, IPO, and ORPO, and report TPR at FPR$= 1\%$ as the maximum over MIA scorers in Table 6. Across the 16 (record statistic, objective) cells, the Qwen3-32B leakage profile agrees with the average over the models of at most 1.7B parameters on Chatbot Arena, with Spearman $\rho = 0.95$ ($p = 1.8 \times 10^{-8}$). The record-statistic and objective orderings of the main evaluation are therefore overall preserved at 32B.

F Preferred and Dispreferred Responses

This appendix provides the full response-asymmetry diagnostics. Figure 4 compares the average preferred-response and dispreferred-response leakage by objective. Figure 5 gives the pointwise view: each point compares the preferred-response and dispreferred-response score for one evaluated configuration. Figure 6 separates the same preferred-response win-rate comparison by MIA scorer. For most post-training objectives, leakage lies near the diagonal or has a small edge toward the dispreferred response; SFT and ORPO are the main exceptions, with much stronger preferred-response leakage.

G Sequence-Length Sensitivity

This appendix reports the privacy-utility trade-off for the sequence-length sensitivity analysis. Figure 7 sweeps the maximum training sequence length from 32 to 4096 tokens, keeping the model and all

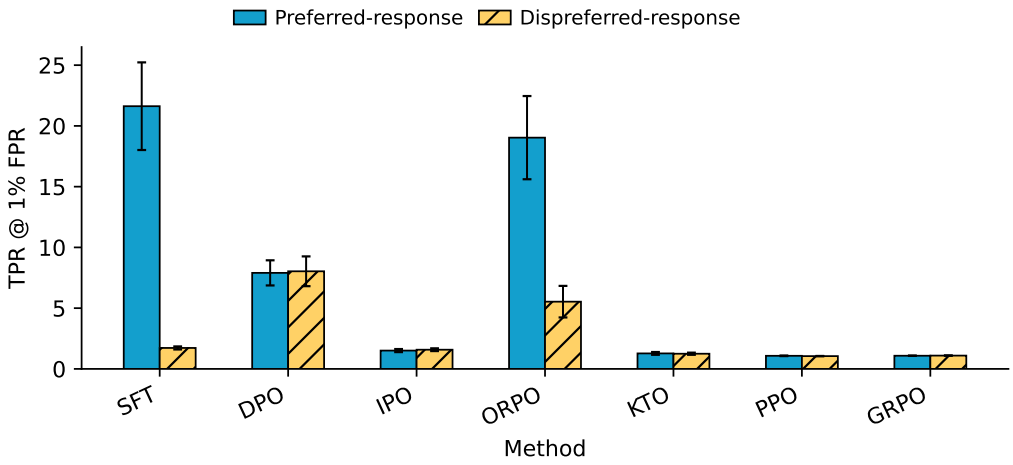


Figure 4: **Preferred-response leakage is strongest for SFT and ORPO.** The figure compares the preferred-response and dispreferred-response statistics. Preferred responses dominate for SFT and ORPO, while DPO, IPO, KTO, PPO, and GRPO are closer to parity.

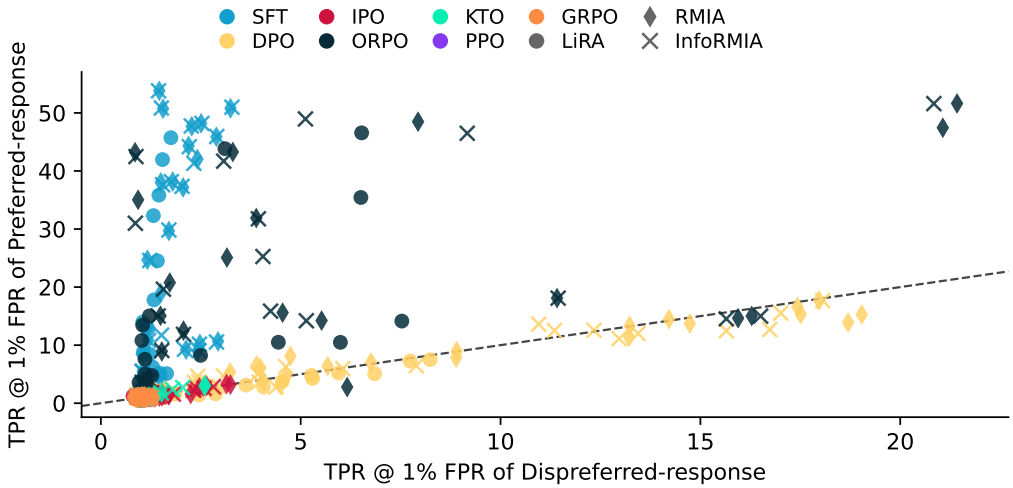


Figure 5: **SFT and ORPO drive preferred-response dominance.** The figure compares TPR at FPR$= 1\%$ for preferred-response and dispreferred-response scores across evaluated configurations. Points above the diagonal indicate stronger leakage from the preferred response, while points below the diagonal indicate stronger leakage from the dispreferred response. Most objectives lie near parity or slightly favor the dispreferred response, except SFT and ORPO.

other hyperparameters fixed. TPR at FPR$= 1\%$ FPR generally increases with sequence length, with ORPO showing the highest vulnerability across all lengths, while KTO remains consistently near the random baseline. Figure 8 shows that the leakage pattern interacts with the utility achieved by each objective.

H Extended feature analysis for MIAs

This appendix reports the extended feature analysis for the MIA scorers. Figures 9 and 10 compares score features across LiRA, RMIA, and InfoRMIA. Averaged over post-training objectives and settings, cross-entropy exposes the most membership signal, stable and hinge form the next tier, and Min-K%++ and logit trail. The MIA-scorer ordering is consistent with the main evaluation: RMIA

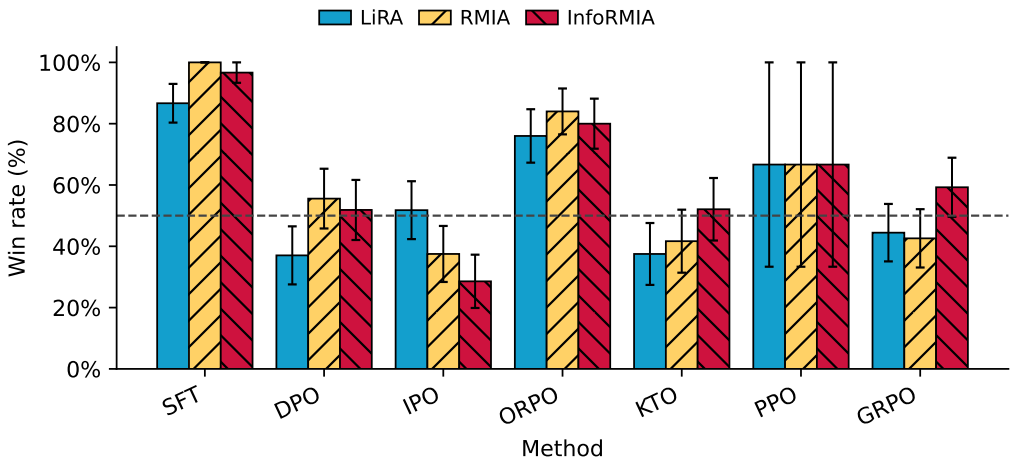


Figure 6: **Preferred-response dominance is scorer-robust only for SFT and ORPO.** Win rate at FPR = 1% with which the preferred-response statistic carries the larger single-response membership signal than the dispreferred-response statistic, broken down by MIA scorer.

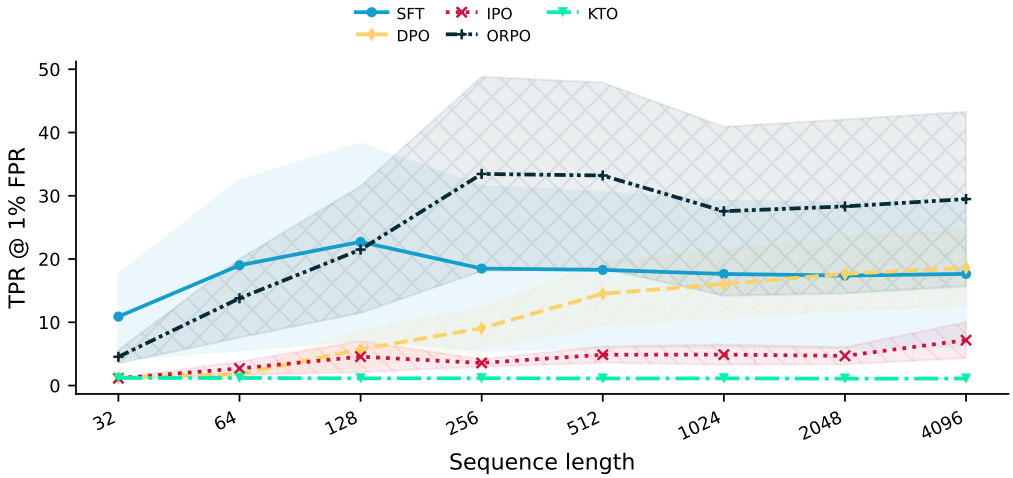


Figure 7: **Larger sequence length affects each post-training objective differently.** The plot reports TPR at FPR$= 1\%$ versus maximum training sequence length for each objective.

and InfoRMIA outperform LiRA across the feature variants, confirming that population normalization is a primary driver of attack strength rather than the choice of score feature alone. When broken down by post-training objective, the feature and scorer rankings are largely stable. SFT and ORPO remain the most exposed objectives regardless of the feature used, and IPO, KTO, PPO, and GRPO remain close to the baseline across all feature–scorer combinations.

Table 7 gives a holistic overview of adapting score features across MIA scorers: cross-entropy is the strongest feature choice overall, stable and hinge form the next tier, and Min-K%++ and logit trail. Table 8 gives the post-training-objective breakdown. In the aggregate comparison, cross-entropy reaches 8.7%, followed by hinge at 5.6% and stable at 5.5%; Min-K%++ and logit are lower at 4.4% and 3.4%. Thus the results support loss-based features as the most effective family, with cross-entropy strongest in the current table.

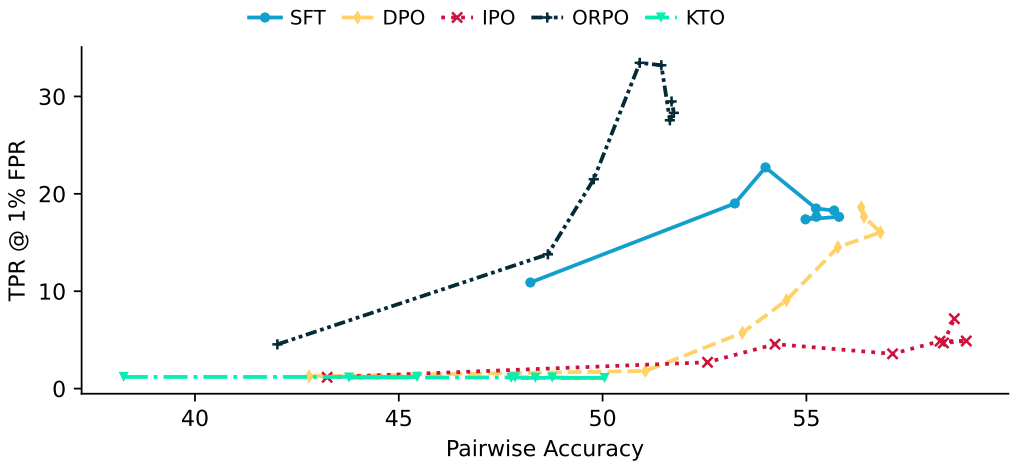


Figure 8: **Sequence length changes the privacy-utility trade-off.** Pairwise accuracy and TPR at FPR$= 1\%$ across maximum training sequence lengths. The leakage pattern depends on the post-training objective and the utility achieved at that length.

Table 7: **Feature and scorer comparison.** Average TPR at FPR$= 1\%$ for each score feature and reference-model MIA scorer, averaged over post-training objectives and settings. Bold entries mark the strongest scorer within each feature.

Feature	LiRA	RMIA	InfoRMIA
stable	2.4 ± 0.3	$\mathbf{5.5 \pm 0.5}$	3.9 ± 0.4
hinge	2.2 ± 0.4	$\mathbf{5.6 \pm 0.5}$	4.0 ± 0.4
logit	2.0 ± 0.3	$\mathbf{3.4 \pm 0.4}$	2.6 ± 0.2
min k++	1.1 ± 0.0	$\mathbf{4.4 \pm 0.5}$	2.6 ± 0.2
cross-entropy	2.6 ± 0.4	8.5 ± 0.9	$\mathbf{8.7 \pm 0.9}$

H.1 Why does LiRA underperform in our benchmark

Across the full attack-instance breakdown in Table 4, LiRA is consistently weaker than RMIA and InfoRMIA under the same record statistics. For the preferred-response statistic, LiRA achieves an aggregate TPR at FPR$= 1\%$ of only 4.3 ± 0.7, compared with 8.8 ± 1.1 for RMIA and 8.4 ± 1.1 for InfoRMIA. The gap is even larger for the signed response-pair statistic: LiRA reaches only 2.6 ± 0.4 on average, while RMIA and InfoRMIA reach 8.5 ± 0.9 and 8.7 ± 0.9, respectively. This pattern is particularly visible for DPO, where LiRA signed response-pair gives only 1.2 ± 0.0, while RMIA and InfoRMIA reach 12.3 ± 1.7 and 13.0 ± 1.9.

A likely explanation is the limited reference-model budget. Our benchmark uses four reference models per setting, which is computationally practical for a broad benchmark but leaves few IN and OUT samples for target-specific distribution estimation. This limitation matters most for LiRA because it directly estimates or compares the IN and OUT reference-score distributions for each target record. In contrast, RMIA and InfoRMIA use population-normalized variants of the same reference-model principle, which appears more stable in the limited reference model setup.

I Differentially Private Post-Training

This appendix describes how the post-training objectives are extended to DP over complete preference records using DP-SGD. The protected record is the complete preference example $z = (x, y_w, y_l)$. For SFT, DPO, IPO, and ORPO, each sampled record contributes one objective-specific gradient that can be clipped and noised. KTO is more delicate because its loss contains a batch-level reference point; this reference point must be privately released before it is reused inside the per-record loss.

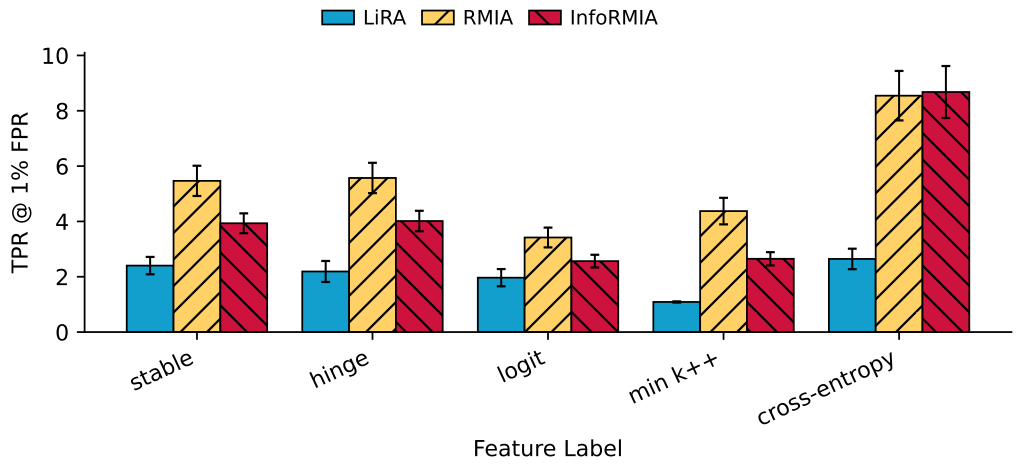


Figure 9: **Score feature and MIA scorer jointly determine leakage.** Average TPR at FPR$= 1\%$ for each score feature and reference-model MIA scorer, averaged over post-training objectives and settings. Cross-entropy is strongest overall, stable and hinge form the next tier, and Min-K%++ and logit trail.

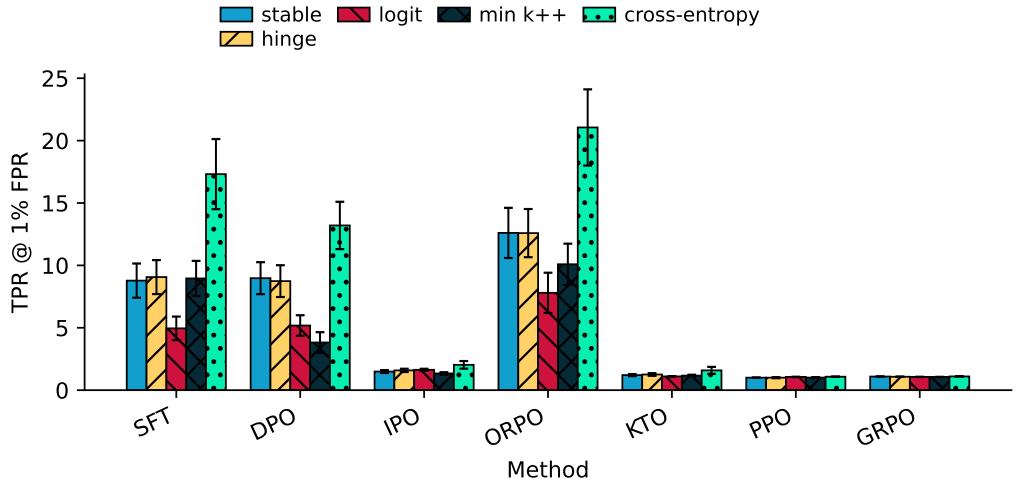


Figure 10: **Objective-level ordering is preserved across score features.** Average TPR at FPR$= 1\%$ broken down by post-training objective and score feature, using the strongest attack instance selected per run. SFT and ORPO remain the most exposed objectives across all features, while IPO, KTO, PPO, and GRPO stay close to the baseline regardless of the feature used.

A randomized training algorithm M is (ε, δ)-DP if, for any add/remove adjacent datasets $D \simeq D'$ and any measurable event E,

$$\Pr[M(D) \in E] \leq e^{\varepsilon} \Pr[M(D') \in E] + \delta. \tag{23}$$

We obtain private post-training variants with Poisson-subsampled DP-SGD. At step t, each target record is sampled independently with rate q, giving a sampled batch S_t. Let $\bar{m} = qN$ be the expected batch size. For each sampled record i, compute the objective-specific per-record gradient $g_i = \nabla_\theta \ell(\theta; z_i)$, clip it to norm C_g,

$$\bar{g}_i = g_i \min\left(1, \frac{C_g}{\|g_i\|_2}\right), \tag{24}$$

Table 8: **Feature comparison by objective.** Average TPR at FPR$= 1\%$ for each post-training objective and score feature, using the strongest attack instance selected per run. Bold entries mark the strongest feature within each objective.

Method	stable	hinge	logit	min k++	cross-entropy
SFT	8.8 ± 1.4	9.1 ± 1.4	5.0 ± 0.9	9.0 ± 1.4	$\mathbf{17.3 \pm 2.8}$
DPO	9.0 ± 1.3	8.7 ± 1.3	5.2 ± 0.8	3.8 ± 0.8	$\mathbf{13.2 \pm 1.9}$
IPO	1.5 ± 0.1	1.6 ± 0.1	1.6 ± 0.1	1.3 ± 0.1	$\mathbf{2.0 \pm 0.3}$
ORPO	12.6 ± 2.0	12.6 ± 1.9	7.8 ± 1.6	10.1 ± 1.6	$\mathbf{21.1 \pm 3.1}$
KTO	1.2 ± 0.1	1.3 ± 0.1	1.1 ± 0.0	1.2 ± 0.1	$\mathbf{1.6 \pm 0.3}$
PPO	1.0 ± 0.0	1.0 ± 0.0	$\mathbf{1.1 \pm 0.0}$	1.0 ± 0.0	$\mathbf{1.1 \pm 0.0}$
GRPO	$\mathbf{1.1 \pm 0.0}$	$\mathbf{1.1 \pm 0.0}$	$\mathbf{1.1 \pm 0.0}$	$\mathbf{1.1 \pm 0.0}$	$\mathbf{1.1 \pm 0.0}$
All	5.6 ± 0.6	5.7 ± 0.6	3.6 ± 0.4	4.4 ± 0.5	$\mathbf{9.3 \pm 1.0}$

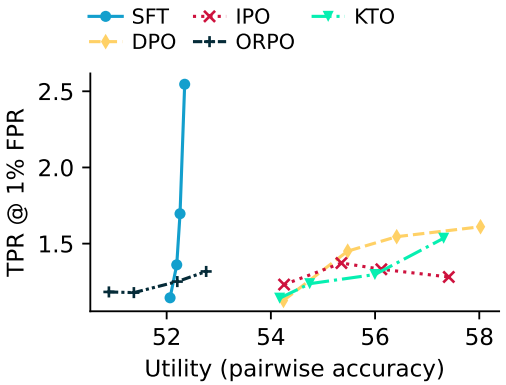


Figure 11: **DP privacy-utility trade-off.**

and update with the noisy mean

$$\tilde{g}_t = \frac{1}{\bar{m}} \sum_{i \in S_t} \bar{g}_i + \mathcal{N}\left(0, \sigma_g^2 \left(\frac{C_g}{\bar{m}}\right)^2 I\right). \tag{25}$$

The fixed-batch sensitivity of the mean clipped gradient is $C_g/\bar{m}$, and Poisson subsampling is handled by the privacy accountant. The optimizer update, learning-rate schedule, and released model parameters are post-processing of the noisy gradients. The target model and all reference models use the same private objective and the same accounting rule within each private setting, so the MIA evaluation changes only the training mechanism, not the membership game.

Clipping norm. Our DP runs clip per-record gradients to $C_g = 0.001$, as specified in Appendix J.2. Following Li et al. [24], we select a small clipping norm so that training operates in the regime where most per-record gradients are clipped, which is the recommended practice for DP fine-tuning of language models. To test the sensitivity to this choice, we repeat the full ε sweep on Qwen3-0.6B with two additional clipping norms, $C_g \in \{0.1, 1\}$. Tables 9 and 10 report TPR at FPR$= 1\%$ and pairwise accuracy averaged over the three datasets for $C_g \in \{0.001, 0.1, 1\}$. The relative impact of DP on vulnerability and utility remains similar across clipping norms. For example, SFT leakage rises with ε at every clipping norm (from 1.14 to 2.55 at $C_g = 0.001$, from 1.12 to 3.23 at $C_g = 0.1$, and from 1.19 to 3.11 at $C_g = 1$), and SFT remains the most exposed objective at $\varepsilon = 100$ in all three cases.

I.1 Differentially Private KTO

KTO is the nontrivial case for DP post-training. It uses the same membership-evaluation protocol as the other post-training objectives, but its loss contains a batch-level reference point that must be privatized before it is reused in the per-example loss.

Table 9: **Membership leakage under DP is robust to the clipping norm.** Average TPR at FPR$= 1\%$ on Qwen3-0.6B over the three datasets. Each cell reads $C_g = 0.001$ / 0.1 / 1, and the last column reports the largest gap between clipping norms in that row.

Method	$\varepsilon = 0.1$	$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$	Max spread over C_g
SFT	1.14 / 1.12 / 1.19	1.36 / 1.50 / 1.59	1.70 / 1.94 / 2.01	2.55 / 3.23 / 3.11	0.68
DPO	1.13 / 1.12 / 1.17	1.45 / 1.41 / 1.34	1.55 / 1.57 / 1.53	1.61 / 1.72 / 1.69	0.11
IPO	1.23 / 1.14 / 1.13	1.37 / 1.37 / 1.44	1.33 / 1.55 / 1.58	1.28 / 1.70 / 1.71	0.43
ORPO	1.18 / 1.18 / 1.23	1.18 / 1.42 / 1.36	1.25 / 1.64 / 1.60	1.32 / 1.81 / 1.81	0.49
KTO	1.14 / 1.12 / 1.17	1.24 / 1.27 / 1.28	1.30 / 1.44 / 1.40	1.54 / 1.67 / 1.61	0.14

Table 10: **Utility under DP is robust to the clipping norm.** Average pairwise accuracy on Qwen3-0.6B over the three datasets. Each cell reads $C_g = 0.001$ / 0.1 / 1, and the last column reports the largest gap between clipping norms in that row.

Method	$\varepsilon = 0.1$	$\varepsilon = 1$	$\varepsilon = 10$	$\varepsilon = 100$	Max spread over C_g
SFT	52.1 / 50.8 / 50.8	52.2 / 50.4 / 50.3	52.3 / 50.5 / 50.5	52.4 / 50.5 / 50.5	1.9
DPO	54.2 / 55.0 / 55.0	55.5 / 55.8 / 55.8	56.4 / 56.5 / 56.5	58.0 / 57.7 / 57.7	0.8
IPO	54.3 / 55.0 / 55.0	55.4 / 55.8 / 55.7	56.1 / 56.5 / 56.5	57.4 / 57.8 / 57.8	0.7
ORPO	50.9 / 54.4 / 54.5	51.4 / 53.7 / 53.7	52.2 / 53.7 / 53.7	52.8 / 54.4 / 54.3	3.6
KTO	54.2 / 55.3 / 55.2	54.7 / 55.6 / 55.6	56.0 / 56.2 / 56.2	57.3 / 57.0 / 57.0	1.1

We use a paper-faithful cyclic estimator of the KTO reference point. Let the private record be one preference example $t_i = (x_i, y_{w,i}, y_{l,i})$. At step t, sample a Poisson batch $S_t = (t_1, \ldots, t_s)$ with sampling rate q and expected batch size $\bar{m} = qN$. Define the cyclic donor index $d(i) = i + 1$ for $i < s$ and $d(s) = 1$. For clipping threshold C_z, compute

$$c_i^w = \mathrm{clip}\left(\log \frac{\pi_\theta(y_{w,d(i)} \mid x_i)}{\pi_{\mathrm{ref}}(y_{w,d(i)} \mid x_i)}, -C_z, C_z \right), \tag{26}$$

$$c_i^l = \mathrm{clip}\left(\log \frac{\pi_\theta(y_{l,d(i)} \mid x_i)}{\pi_{\mathrm{ref}}(y_{l,d(i)} \mid x_i)}, -C_z, C_z \right), \tag{27}$$

and combine the two channels as $u_i = (c_i^w + c_i^l)/2$, so $u_i \in [-C_z, C_z]$. The cyclic dependence means that removing one target example changes two old cyclic summands and one new bridge summand. The add/remove sensitivity of the summed statistic is therefore $3C_z$, and the sensitivity of the mean release is $3C_z/\bar{m}$. If the two channels are summed without averaging, the corresponding sensitivity is $6C_z/\bar{m}$.

The private KTO reference point is

$$z_t^{\mathrm{priv}} = \max\left(0, \frac{1}{\bar{m}} \sum_{i=1}^{s} u_i + \mathcal{N}\left(0, \sigma_z^2 \left(\frac{3C_z}{\bar{m}} \right)^2 \right) \right) \tag{28}$$

for the averaged two-channel version. This released scalar is detached before it is used in the loss. Algorithm 1 summarizes the full private KTO update.

For accounting, the cyclic reference release has sensitivity $3C_z/\bar{m}$ when the two channels are averaged, or $6C_z/\bar{m}$ when the two clipped channels are summed. The gradient release has sensitivity $C_g/\bar{m}$. When both releases use the same Poisson batch, the two fixed-sample Gaussian mechanisms are composed first and Poisson amplification is applied once to the composed step.

J Experimental Details

J.1 General Experimental Details

The main benchmark uses UltraFeedback Binarized [11], HH-RLHF [4], and Chatbot Arena [10]. The main model grid contains Qwen3-0.6B, Qwen3-1.7B, Gemma 3 270M, and Gemma 3 1B; the larger-model subset contains Qwen3-4B and Qwen3-8B. The main objective grid contains SFT, DPO, IPO, ORPO, KTO, PPO, and GRPO; the larger-model subset uses SFT, DPO, IPO, and ORPO.

Algorithm 1 DP KTO step with a cyclic reference release.

Require: model π_θ, reference policy π_{ref}, target universe size N, sampling rate q
Require: expected batch size $\bar{m} = qN$, clipping thresholds C_z, C_g, noise multipliers σ_z, σ_g
sample a Poisson batch $S_t = (t_1, \ldots, t_s)$ with $t_i = (x_i, y_{w,i}, y_{l,i})$
for $i = 1, \ldots, s$ **do**
  set $d(i) \leftarrow i + 1$ if $i < s$, and $d(s) \leftarrow 1$
  compute clipped cyclic log-ratios c_i^w and c_i^l as defined above
  $u_i \leftarrow (c_i^w + c_i^l)/2$
end for
sample $\xi_z \sim \mathcal{N}(0, \sigma_z^2(3C_z/\bar{m})^2)$
$z_t^{\text{priv}} \leftarrow \max\big(0, \bar{m}^{-1} \sum_{i=1}^{s} u_i + \xi_z\big)$
detach z_t^{priv}
for $t_i \in S_t$ **do**
  compute $r_i^+(\theta)$ and $r_i^-(\theta)$
  $\ell_i \leftarrow \ell_i(\theta; z_t^{\text{priv}})$
  $g_i \leftarrow \nabla_\theta \ell_i$
  $\bar{g}_i \leftarrow g_i \min(1, C_g/\|g_i\|_2)$
end for
sample $\xi_g \sim \mathcal{N}(0, \sigma_g^2(C_g/\bar{m})^2 I)$
$\tilde{g}_t \leftarrow \bar{m}^{-1} \sum_{i \in S_t} \bar{g}_i + \xi_g$
update θ with $\tilde{g}_t$

For each dataset, model, and training configuration, the candidate universe contains 20,000 target preference records. Following the reference-model budget used in [35], each audited setting uses $K = 4$ reference models, so each metric is scored over $4 \times 20{,}000$ model–record evaluations. We evaluate four record statistics: preferred response, dispreferred response, scalar preference margin, and signed response-pair contrast. Each record statistic is scored with LiRA, RMIA, and InfoRMIA, yielding one attack instance per statistic–scorer pair. Unless otherwise stated, aggregate results average over the datasets, models, and configurations included in the corresponding experiment, and standard errors are computed over the setting-level means in that aggregate.

The PEFT grid uses Qwen3-0.6B and covers SFT, DPO, IPO, ORPO, and KTO with eight adapter variants. These runs use the same target records, record statistics, privacy metric, and utility metric as full fine-tuning, changing only the adaptation parameterization.

J.2 DP Training

The DP experiments extend the post-training objectives with DP-SGD. The implementation uses `dp-transformers` with Opacus for per-sample gradients, clipping, DP optimizer wrapping, and sampling. Privacy is tracked with the Opacus RDP accountant and the `prv_accountant` package. The protected unit is adding or removing one training tuple. All the DP runs use Qwen3-0.6B, the three benchmark datasets, objectives SFT, DPO, IPO, ORPO, and KTO, privacy budgets $\varepsilon \in \{0.1, 1.0, 10.0, 100.0\}$, clipping norm 0.001, and $\delta = 1/|\mathcal{D}_{\text{train}}|$.

J.3 Response-Side Analysis

The response-side analysis applies the same single-response attack construction separately to preferred and dispreferred responses. This diagnostic supports the response-localization results in Appendix F.

J.4 Sequence-Length Study

The sequence-length study uses Gemma 3 1B IT and varies the maximum training sequence length over $\{32, 64, 128, 256, 512, 1024, 2048, 4096\}$. This isolates the effect of training context length while keeping the audit protocol fixed. Appendix G reports the corresponding privacy-utility analysis.

Table 11: Computational cost (Training time in hrs.) for training each reference model. **Indicators: * 2-GPU setup. ** 4-GPU setup. *** Single GPU with micro batching to avoid OOM errors.**

Model	SFT	DPO	IPO	ORPO	KTO*	PPO***	GRPO**
Gemma3 270M	1.02	1.3	1.26	1.16	1.47	24.44	2.17
Gemma3 1B	1.55	2.53	2.48	2.27	2.95	30.72	8.35
Qwen3 0.6B	2.21	2.77	2.77	2.42	4.02	26.52	4.95
Qwen3 1.7B	0.90	1.32	1.44	0.92	1.55	30.21	5.27
Qwen3 4B*	2.22	2.62	2.61	2.57	4.73	–	–
Qwen3 8B**	3.59	4.71	4.76	3.92	5.35	–	–

J.5 Canary Study

The canary study inserts 200 controlled records per run into Qwen3-0.6B experiments across all three datasets and SFT, DPO, IPO and ORPO. The canary families perturb different parts of a preference record, including the prompt and responses; Appendix K gives the full family construction.

J.6 Compute Resources

Table 11 reports wall-clock training time, in hours, for each model and post-training objective primarily on NVIDIA A100 GPUs and NVIDIA H100 GPUs for GRPO specifically. Since one audited setting uses 4 model trainings in total, the full training cost of the audit is approximately 4 times the per-model runtime reported, plus a small scoring overhead.

In the standard experiments, all models are trained on 20,000 samples with a maximum sequence length of 1024 tokens, so the runtime differences are driven mainly by the model size and post-training objective rather than by changes in the dataset configuration.

All numbers presented in this section are considered from successfully completed runs only.

K Canary Construction and Evaluation

This appendix defines the canary families used in the privacy analysis. Each canary family isolates a different controlled perturbation of the preference record, so the canary study can distinguish ordinary record insertion from prompt corruption, response corruption, preferred/dispreferred swapping, and high-entropy memorization. Conceptually, the families separate classifier-like label/order corruption from language-model-like memorization of rare strings. The text boxes below show examples from each canary family.

Normal canaries (Normal). This family inserts ordinary-looking preference records without corrupting the prompt or responses. It provides a natural-data baseline for canary insertion rather than a deliberately adversarial perturbation.

Prompt. Q: Write the right answer to the question based on the context passage. His eyes were open and his head bobbed around at an impossible angle. He was sitting in about forty feet of water...
Preferred. Dan, the protagonist, got a coke out of the cooler.
Dispreferred. Dan got coke out of the cooler.

Random strings (RandStr). This family replaces the prompt, preferred response, and dispreferred response with random strings. It is the high-entropy language-model canary: it tests whether rare, semantically unrelated content becomes detectable after training.

Prompt. `l0IneW'*QRgurR*Q`Fua3l/WiaO~5nfl\sqcL3,9H8eNR%9A:...`
Preferred. `qv\oY^.C;z"xej8>VE...`
Dispreferred. `e RvaIoyOjL"lnB\:QY7;3zkLoE;DKjr>f2W_2t3KVZ/R-A"...`

Mislabeled preference examples (MisPref). This family copies a real preference example and swaps the preferred and dispreferred responses. It is analogous to a mislabeled classification example and tests whether the reversed preference order itself becomes detectable.

> **Prompt.** I am trying to build typescript typings for a mod for Minecraft that runs an embedded version of JavaScript (via Mozilla Rhino). The mod uses 3 folders to load scripts depending on the environment...
> **Preferred.** To get the `@kubejs/startup` module to see the types in `@kubejs/core`, you can add `typeRoots` or `types` to the `tsconfig.json`...
> **Dispreferred.** I think I can help you with this. First, let me clarify a few things about how TypeScript typings work...

Random prompt with real responses (RandP + RealR). This family replaces the prompt with a random prompt while keeping the preferred and dispreferred responses from a sampled real example. It isolates prompt corruption while preserving realistic response text; comparing it with RealP + RandR tests whether prompt or response corruption dominates.

> **Prompt.** `83,nf/#y(bv4}&b|WE4[in|"4,'wqWTT@cL...`
> **Preferred.** Of course! I'd be happy to help you expand on your story. Please go ahead and give me a brief description of the first part of your story...
> **Dispreferred.** Sure, why not? Let's work together to create an engaging story. You start off by describing the setting and main character...

Real prompt with random responses (RealP + RandR). This family keeps a real prompt and replaces both responses with random responses. It tests whether response corruption is memorized even when the prompt remains realistic, and should be read against RandP + RealR as the response-side counterpart.

> **Prompt.** Is there a hierarchy or ranking system among Milanese dialects, and if so, how is this determined?
> **Preferred.** `&O|^70i)OM>:/4x&sm&'LIT :OO_j8)Sc!wM4Xm:4jT^fy)4Z1Wd...`
> **Dispreferred.** `I=(7`aB|8rh4O\]OWp-Y}y1mTj[?~t#F=n%g!KB{QKTPR...`

Real prompt with random preferred response (RealP + RandPref). This family keeps a real prompt and replaces only the preferred response with a random response. It tests high entropy specifically in the preferred, low-loss response channel, and should be read against RealP + RandR to separate one-sided preferred-response corruption from corruption of both responses.

> **Prompt.** How long does an average course of chiropractic treatment last, and how often might a patient need to schedule appointments?
> **Preferred.** `%G``L,r?>%T-|qFI?WMH4p[#_`*_*#*%s+ZZ9#>A.\v;6<M...`
> **Dispreferred.** It depends on the condition that is being treated. Most people see improvement within the first few weeks of treatment...

Canaries are appended to the candidate training universe before the reference-model split is sampled. They are therefore treated like ordinary examples when IN and OUT reference sets are formed. In the $K = 4$ reference-model setting [35], each canary is assigned to two reference models and omitted from two reference models, matching the balanced IN/OUT construction used by the LiRA-style evaluation. This makes the canary results directly comparable to the main results. Table 12 reports the canary results by objective and canary family. We also compare different response scores for different canary types in Figure 12.

We also compare different canaries against preferred-response and dispreferred-response TPR at FPR$= 1\%$ scores in Figure 14. The RandStr canary family has the highest dispreferred-response leakage, with one configuration reaching a TPR above 12% on the dispreferred side while the preferred-response TPR remains near zero.

Table 12: **Canary results by family.** TPR at FPR$= 1\%$ by post-training objective and family.

Method	Normal	RandStr	MisPref	RandP + RealR	RealP + RandR	RealP + RandPref
SFT	2.0	1.3	3.5	3.7	2.6	**3.1**
DPO	3.2	9.3	**12.8**	**6.1**	**7.9**	**3.1**
IPO	**4.5**	**10.6**	9.5	2.1	4.4	2.6
ORPO	3.2	3.4	1.5	1.9	2.8	1.5
All	3.2	6.2	6.8	3.5	4.4	2.6

K.1 Additional canary diagnostics

The main text uses canaries as mechanism probes for localizing membership evidence within the prompt and response components of a preference record. Here, we report the full canary-type breakdown and discuss two additional diagnostic views: preference-order corruption and high-entropy memorization.

Preference-order corruption canaries (MisPref) keep the text of a real preference example fixed but swap the preferred and dispreferred sides. They are strongest for DPO, reaching 12.8% TPR at FPR=1%, and remain strong for IPO at 9.5%. This indicates that, for some pairwise objectives, the assignment of responses to the preferred or dispreferred side can itself create detectable membership evidence.

High-entropy string canaries (RandStr) replace the prompt and responses with rare random strings. They are strongest for IPO at 10.6% and also increase leakage for DPO to 9.3%. Thus, although IPO is much less exposed on ordinary records in the main benchmark, it can still show substantial membership evidence on controlled high-entropy or corrupted records. We interpret this as a stress-test result rather than as evidence that these canary types are optimal for auditing.

Table 12 reports the full objective-by-canary-type breakdown.

L PEFT Privacy-Utility Frontier

This appendix reports the PEFT privacy-utility frontier used in the PEFT analysis in Section 5.4. The main text summarizes utility in Table 3; Figure 13 shows the full scatter of pairwise accuracy against TPR at FPR$= 1\%$. Table 13 shows that VeRA [23] and AdaLoRA [42] achieve the highest Pareto counts, although no adapter dominates uniformly. The two methods reach this position through different parameterizations but a shared property: both restrict the effective amount of trainable update applied to the frozen weights more aggressively than the remaining adapters in the grid.

VeRA replaces the trainable low-rank factors A and B with frozen random projections shared across layers and learns only two scaling vectors $b \in \mathbb{R}^{d_{\text{out}}}$ and $d \in \mathbb{R}^r$ per adapted module [23]. The trainable parameter count is therefore $\mathcal{O}(d_{\text{out}} + r)$ per layer rather than $\mathcal{O}(r(d_{\text{in}} + d_{\text{out}}))$ as in LoRA [21], which is roughly one to two orders of magnitude smaller at the ranks used in our sweep. The model can therefore only rescale a fixed random adapter basis, which sharply limits how far it can move toward any individual record.

AdaLoRA achieves a comparable end-of-training capacity reduction through a different mechanism. Each update is parameterized as $\Delta W = B\,\mathrm{diag}(E)\,A$ with $A \in \mathbb{R}^{r \times d_{\text{in}}}$, $E \in \mathbb{R}^r$, and $B \in \mathbb{R}^{d_{\text{out}} \times r}$, mimicking a singular value decomposition [42]. During training, a rank allocator computes a sensitivity-uncertainty importance score for each singular triple and hard-masks low-importance entries of E on a cubic budget schedule. In our sweep, we initialize each adapter at rank 32 and set the target rank to 8, so the per-layer trainable parameter count is high at the start of training but the effective rank of the surviving update is reduced by roughly a factor of four by the end of each reference-model run. The model is therefore constrained to move in a small number of input-output directions selected during training, rather than in the full rank-32 subspace available throughout training to a fixed-rank LoRA configuration.

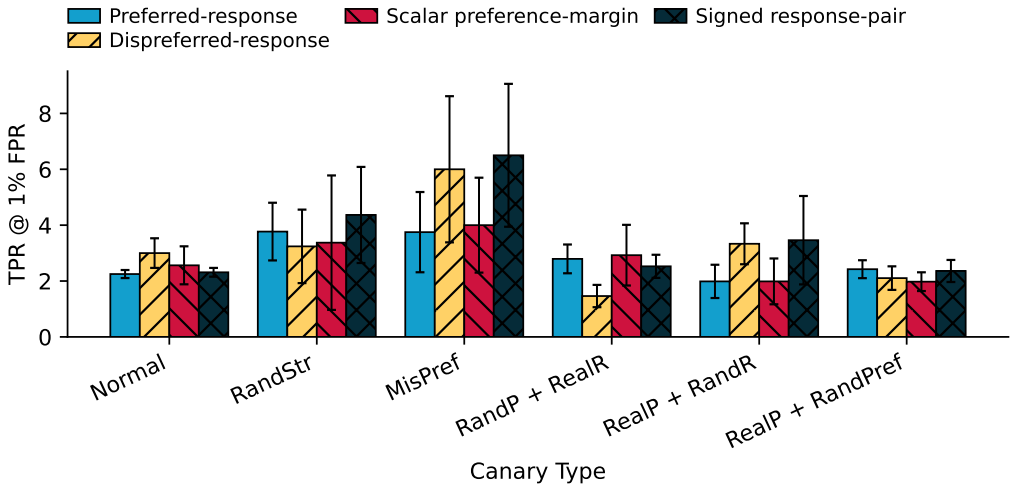


Figure 12: **Canary leakage varies by record statistic.** TPR at FPR$= 1\%$ by canary type and record statistic, separating which statistic exposes each controlled perturbation.

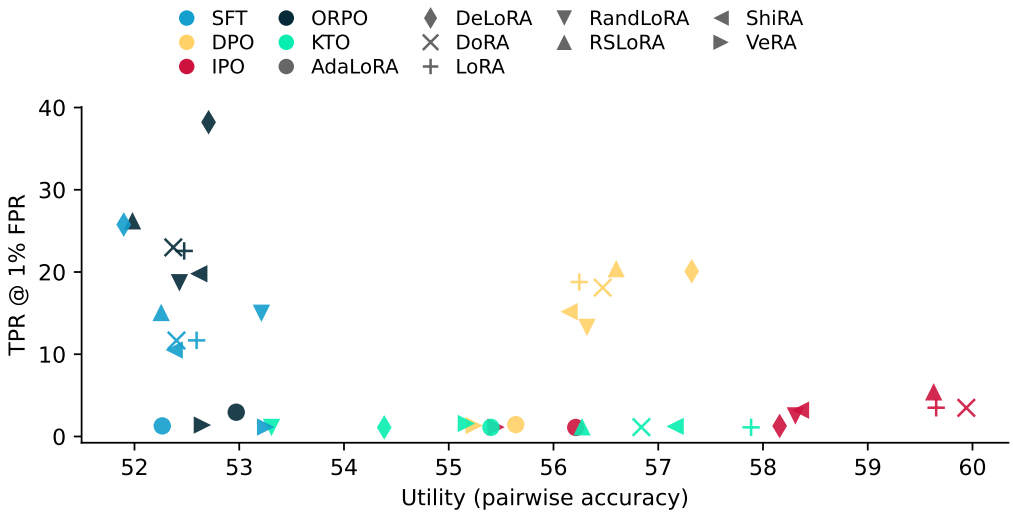


Figure 13: **PEFT moves models along a broad privacy-utility frontier.** Pairwise accuracy versus TPR at FPR$= 1\%$ for the evaluated adapter configurations.

M Hyperparameter and PEFT Sensitivity

This appendix contains additional sensitivity analyses for preference-objective hyperparameters and LoRA rank. These plots are not used as the primary evidence for the main claims, but they test whether the conclusions are stable under common training choices. The hyperparameter sweeps show non-monotonic leakage patterns. The LoRA-rank sweep shows that low-rank adaptation can move the model along a privacy-utility frontier rather than simply reproducing full fine-tuning behavior. Figure 15 gives the LoRA-rank privacy-utility view. The objective-hyperparameter sweeps are reported separately for DPO β, IPO τ, KTO β, and ORPO λ in Figure 16, Figure 17, Figure 18, and Figure 19. The Preferred-response win rate is reported for the same sweeps in Figure 20, Figure 21, Figure 22 and Figure 23.

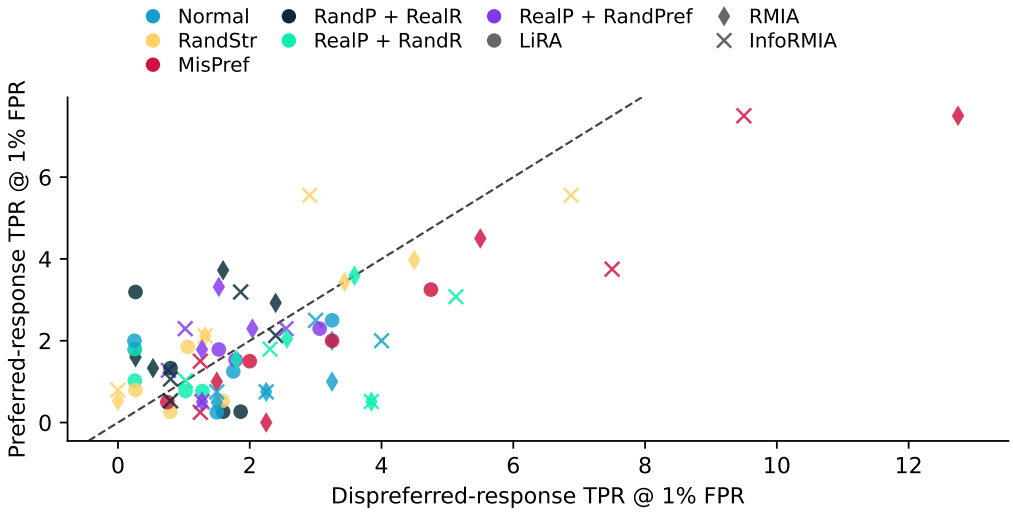


Figure 14: **Dispreferred responses carry more membership signal than preferred responses for most canary families.** Preferred-response versus dispreferred-response TPR at FPR$= 1\%$ across canary families and attack instances.

Table 13: **Pareto count by adapter and post-training objective.** The table reports the number of Pareto-optimal configurations out of three evaluated settings for each adapter–objective combination, with the total count across all objectives in the rightmost column. Bold entries mark the highest Pareto count within each objective.

Adapter	SFT	DPO	IPO	ORPO	KTO	Total
Full FT	1/3	1/3	2/3	1/3	0/3	5/15
LoRA	2/3	0/3	0/3	0/3	**2/3**	4/15
RSLoRA	1/3	2/3	1/3	1/3	0/3	5/15
DoRA	1/3	2/3	1/3	1/3	1/3	6/15
VeRA	**3/3**	**3/3**	1/3	**3/3**	1/3	**11/15**
AdaLoRA	1/3	**3/3**	**3/3**	**3/3**	1/3	**11/15**
RandLoRA	1/3	**3/3**	2/3	1/3	1/3	8/15
ShiRA	2/3	0/3	1/3	1/3	1/3	5/15
DeLoRA	1/3	2/3	**3/3**	1/3	**2/3**	9/15

N Low-FPR Operating Points

This appendix reports stricter operating points than the main TPR at FPR $= 1\%$. These tables test whether the conclusions hold in higher-precision regimes where false membership accusations are even rarer. Table 14 reports TPR at FPR $= 0.1\%$, and Table 15 reports TPR at FPR $= 0.01\%$.

O Area Under the ROC Curve

This appendix reports area under the receiver operating characteristic curve (AUC) as a threshold-agnostic companion metric. The main text focuses on TPR at FPR$= 1\%$ because privacy evaluation is most relevant when false positives must remain rare. AUC is useful for checking whether the broad objective ordering is stable beyond the fixed-FPR operating point. AUC ordering mirrors the TPR at FPR$= 1\%$ results from the main evaluation. Figure 24 shows that SFT and ORPO achieve the highest AUC, followed by DPO and IPO, while KTO, PPO, and GRPO remain close to random guessing across all attack instances. Table 16 gives the full AUC table.

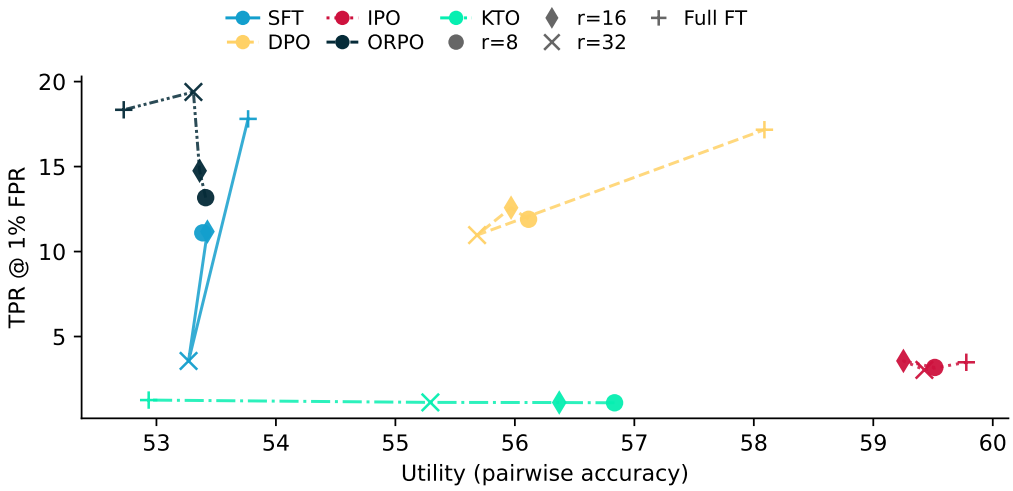


Figure 15: **LoRA rank changes the privacy-utility operating point.** The figure compares full fine-tuning and LoRA ranks using pairwise accuracy and TPR at FPR$= 1\%$. Full fine-tuning and higher-capacity configurations can have substantially larger leakage, while lower-rank configurations often reduce measured leakage at the cost of objective-dependent utility changes.

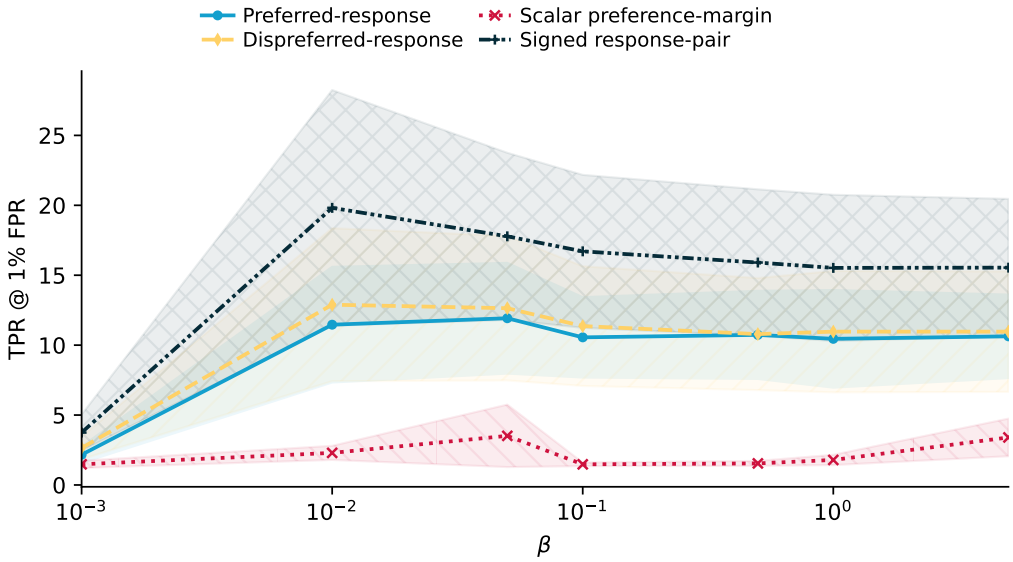


Figure 16: **DPO hyperparameter sensitivity.** TPR at FPR$= 1\%$ across the DPO β sweep for the selected attack instances.

P Broader Impacts

Preference-based post-training is increasingly used to adapt LLMs with data from users, annotators, and deployed systems. Such records may contain sensitive prompts, private responses, proprietary tasks, or confidential preference judgments. This work contributes to safer deployment by showing that membership privacy should be audited at the level of the full preference record, rather than only through a single preferred response or scalar preference score. By providing a benchmark across post-training objectives, PEFT methods, and differentially private training, our study gives practitioners more concrete guidance for identifying where privacy risk arises and for selecting adaptation strategies that better balance utility and privacy. Our findings may also help standardize

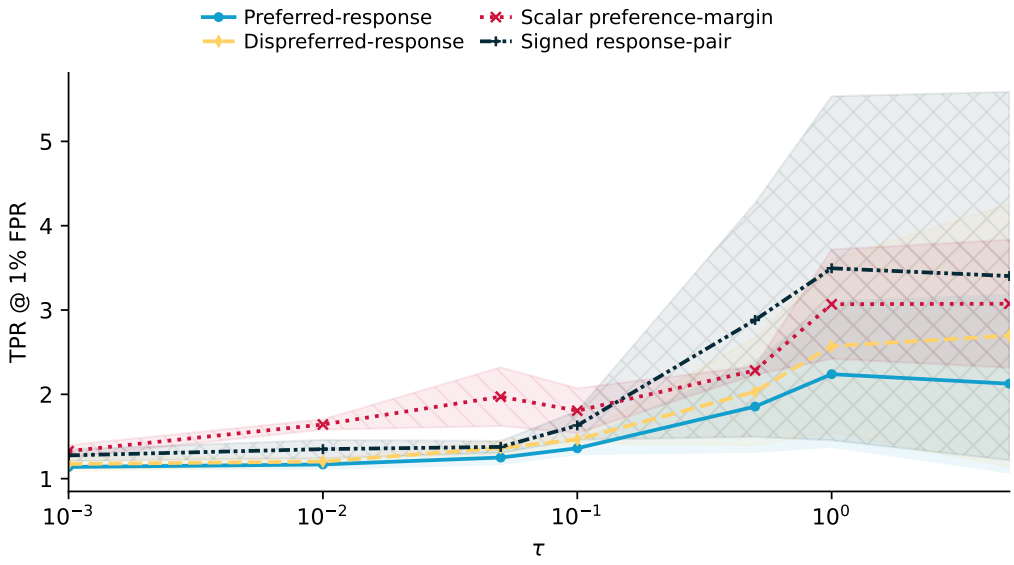


Figure 17: **IPO hyperparameter sensitivity.** TPR at FPR$= 1\%$ across the IPO τ sweep for the selected attack instances.

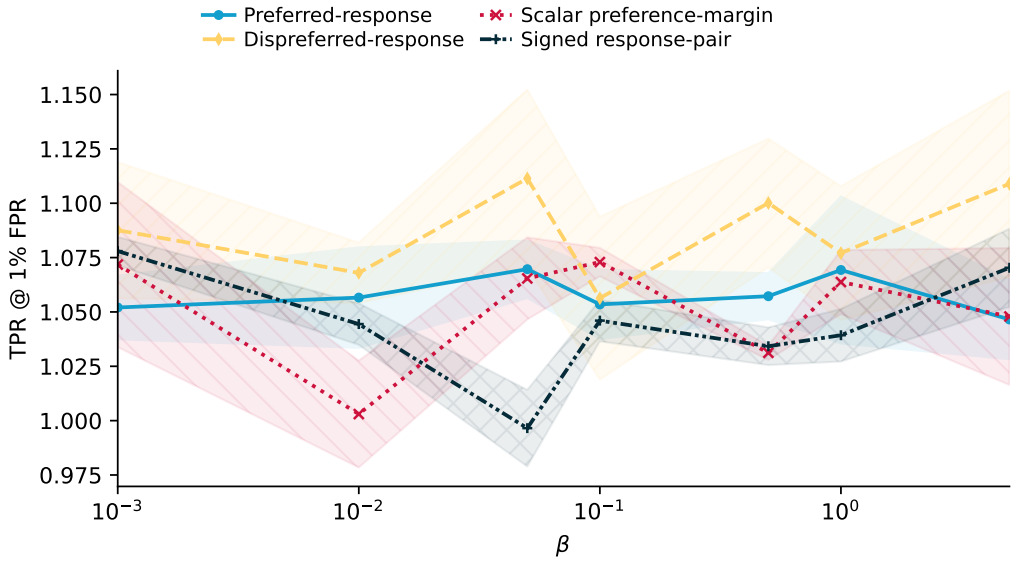


Figure 18: **KTO hyperparameter sensitivity.** TPR at FPR$= 1\%$ across the KTO β sweep for the selected attack instances.

privacy evaluations for aligned and preference-trained models. In particular, the results suggest that privacy audits should report which part of a preference record carries membership signal and should evaluate leakage jointly with downstream utility. This can support more transparent model release decisions, especially in domains where preference data may encode sensitive user behavior, institutional policies, or expert judgments.

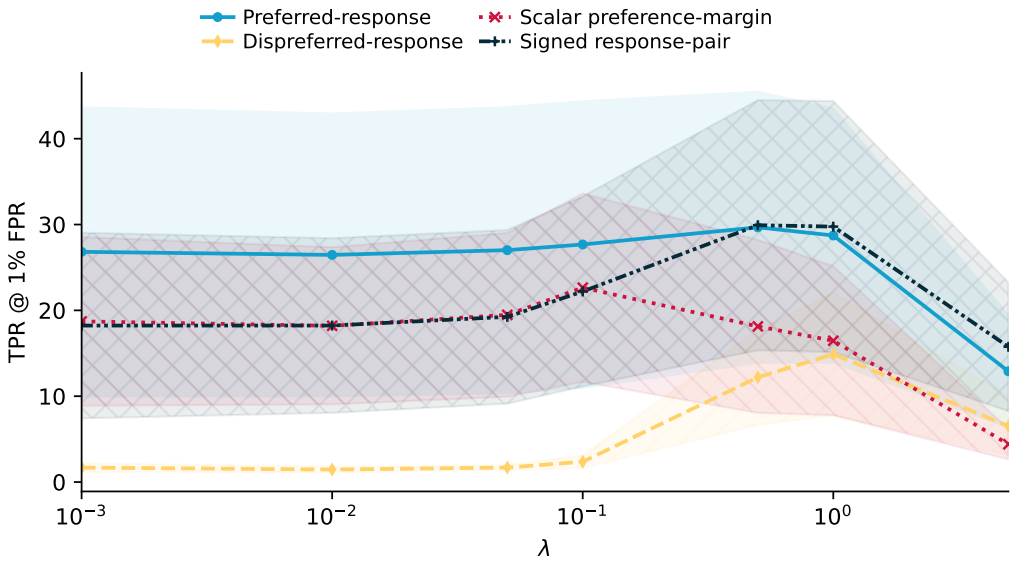


Figure 19: **ORPO hyperparameter sensitivity.** TPR at FPR$= 1\%$ across the ORPO λ sweep for the selected attack instances.

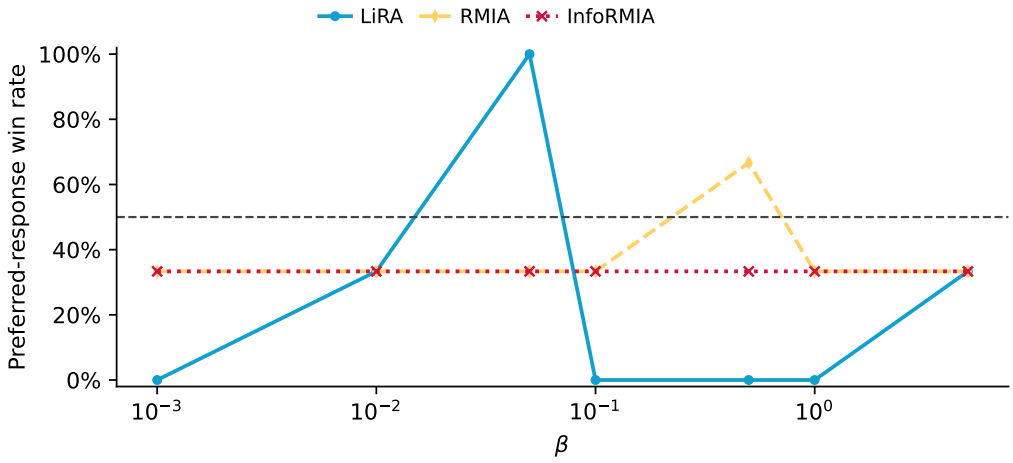


Figure 20: **Preferred-response win rate across DPO β.** The figure reports how often the preferred response carries the larger single-response membership signal across the DPO β sweep, broken down by MIA scorer.

Q Number of Reference Models

The main evaluation trains four reference models for each setting, as detailed in Appendix J.1. To study how the number of reference models K affects the measured leakage, we evaluate Qwen3-0.6B on Chatbot Arena and UltraFeedback with up to $K = 128$ reference models. Table 17 reports TPR at FPR$= 1\%$ for $K \in \{8, 16, 32, 64, 128\}$. The largest advantage comes from increasing the number of reference models up to $K = 32$, while beyond that, the increase is much smaller.

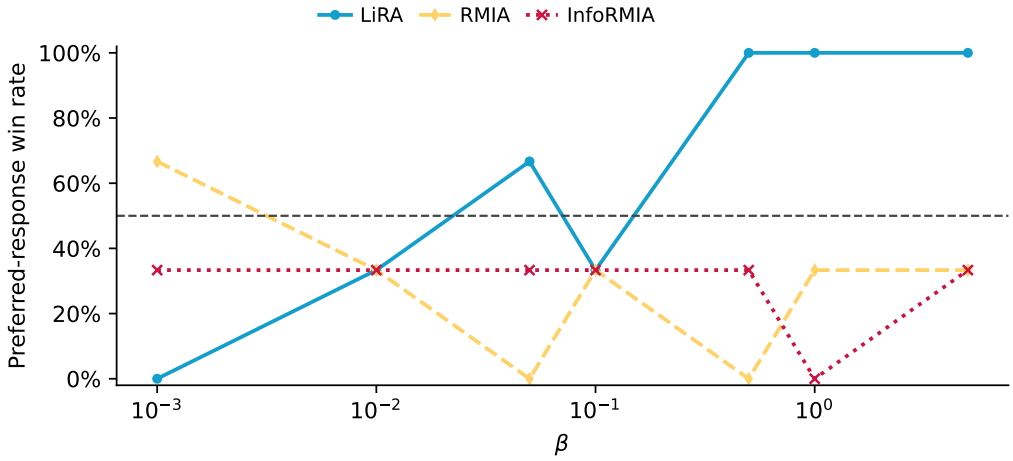


Figure 21: **Preferred-response win rate across IPO τ.** The figure reports how often the preferred response carries the larger single-response membership signal across the IPO τ sweep, broken down by MIA scorer.

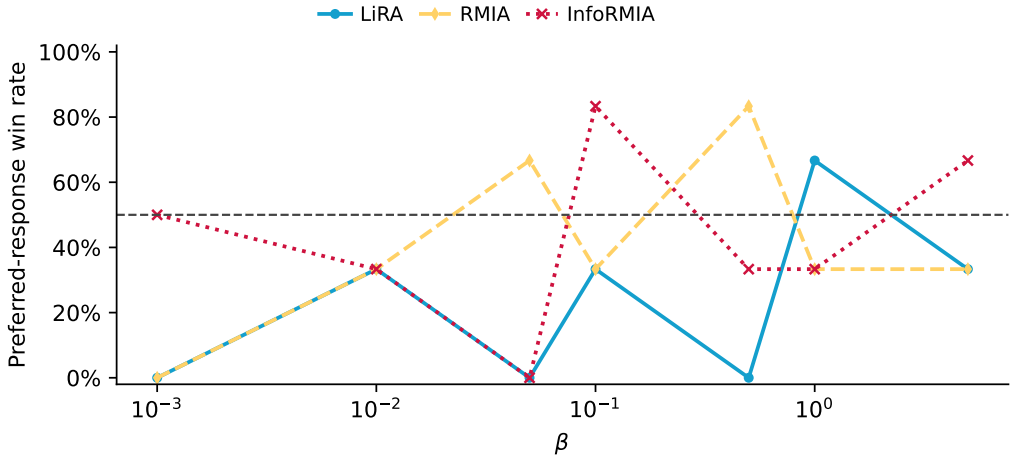


Figure 22: **Preferred-response win rate across KTO β.** The figure reports how often the preferred response carries the larger single-response membership signal across the KTO β sweep, broken down by MIA scorer.

Table 14: **The attack-instance comparison can change at FPR $= 0.1\%$.** Average TPR at FPR $= 0.1\%$ on the main evaluation set. Rows are attack instances, columns are post-training objectives, and entries report mean $\pm$ SE.

Attack	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
LiRA Preferred-response	3.1 ± 1.3	0.4 ± 0.1	0.1 ± 0.0	2.6 ± 1.2	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	1.1 ± 0.3
RMIA Preferred-response	$\mathbf{12.6 \pm 2.6}$	2.7 ± 0.5	0.2 ± 0.0	9.0 ± 2.4	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{4.2 \pm 0.7}$
InfoRMIA Preferred-response	12.3 ± 2.6	2.5 ± 0.5	0.2 ± 0.0	9.3 ± 2.5	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{4.2 \pm 0.7}$
LiRA Dispreferred-response	0.1 ± 0.0	0.4 ± 0.1	0.1 ± 0.0	0.2 ± 0.1	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.2 ± 0.0
RMIA Dispreferred-response	0.2 ± 0.0	2.8 ± 0.6	0.2 ± 0.0	1.1 ± 0.4	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.7 ± 0.1
InfoRMIA Dispreferred-response	0.2 ± 0.0	2.4 ± 0.5	0.2 ± 0.0	0.9 ± 0.3	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.6 ± 0.1
LiRA Scalar preference-margin	1.0 ± 0.4	0.1 ± 0.0	0.2 ± 0.0	1.6 ± 0.6	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.5 ± 0.1
RMIA Scalar preference-margin	2.0 ± 0.7	0.2 ± 0.0	0.3 ± 0.0	1.8 ± 0.6	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.7 ± 0.2
InfoRMIA Scalar preference-margin	1.9 ± 0.7	0.1 ± 0.0	0.3 ± 0.1	1.5 ± 0.4	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.7 ± 0.2
LiRA Signed response-pair	2.0 ± 0.7	0.1 ± 0.0	0.1 ± 0.0	0.3 ± 0.1	0.1 ± 0.0	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	0.5 ± 0.1
RMIA Signed response-pair	6.3 ± 1.4	$\mathbf{6.0 \pm 1.2}$	$\mathbf{0.4 \pm 0.1}$	$\mathbf{9.6 \pm 2.1}$	$\mathbf{0.2 \pm 0.1}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	3.7 ± 0.5
InfoRMIA Signed response-pair	8.5 ± 1.9	5.2 ± 1.1	0.3 ± 0.1	7.6 ± 1.8	$\mathbf{0.2 \pm 0.1}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{0.1 \pm 0.0}$	3.7 ± 0.6

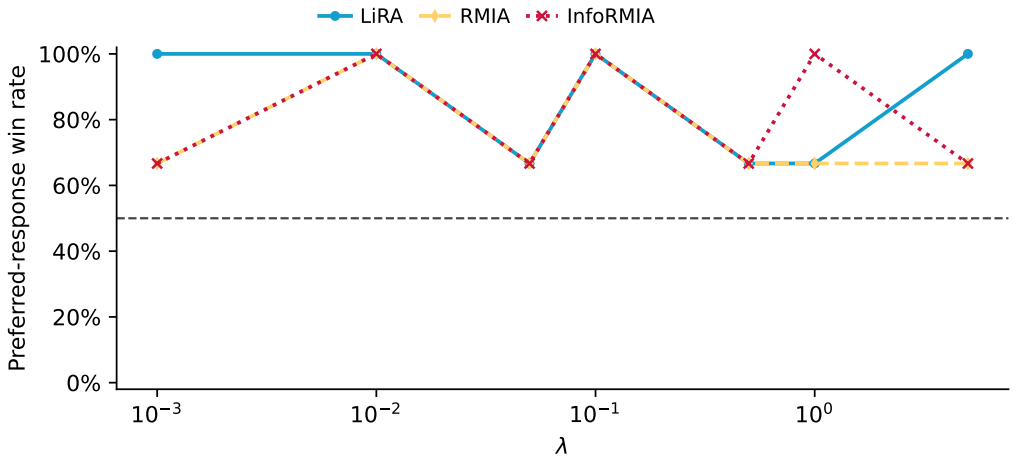


Figure 23: **Preferred-response win rate across ORPO λ.** The figure reports how often the preferred response carries the larger single-response membership signal across the ORPO λ sweep, broken down by MIA scorer.

Table 15: **The attack-instance comparison can change at FPR $= 0.01\%$.** Average TPR at FPR $= 0.01\%$ on the main evaluation set. Rows are attack instances, columns are post-training objectives, and entries report mean $\pm$ SE.

Attack	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
LiRA Preferred-response	0.8 ± 0.4	0.0 ± 0.0	0.0 ± 0.0	0.2 ± 0.1	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.2 ± 0.1
RMIA Preferred-response	$\mathbf{4.5 \pm 1.2}$	0.9 ± 0.3	0.0 ± 0.0	3.1 ± 1.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	1.5 ± 0.3
InfoRMIA Preferred-response	4.4 ± 1.2	0.9 ± 0.3	0.0 ± 0.0	3.4 ± 1.1	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	1.5 ± 0.3
LiRA Dispreferred-response	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.0 ± 0.0
RMIA Dispreferred-response	0.0 ± 0.0	0.6 ± 0.2	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
InfoRMIA Dispreferred-response	0.0 ± 0.0	0.5 ± 0.2	0.0 ± 0.0	0.1 ± 0.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
LiRA Scalar preference-margin	0.2 ± 0.1	0.0 ± 0.0	0.0 ± 0.0	0.5 ± 0.2	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
RMIA Scalar preference-margin	0.3 ± 0.1	0.0 ± 0.0	0.0 ± 0.0	0.4 ± 0.2	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
InfoRMIA Scalar preference-margin	0.3 ± 0.1	0.0 ± 0.0	$\mathbf{0.1 \pm 0.0}$	0.3 ± 0.1	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
LiRA Signed response-pair	0.3 ± 0.2	0.0 ± 0.0	0.0 ± 0.0	0.0 ± 0.0	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.1 ± 0.0
RMIA Signed response-pair	2.8 ± 0.6	$\mathbf{3.3 \pm 0.8}$	$\mathbf{0.1 \pm 0.0}$	$\mathbf{4.4 \pm 1.4}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{1.8 \pm 0.3}$
InfoRMIA Signed response-pair	2.8 ± 0.9	1.5 ± 0.6	0.0 ± 0.0	1.1 ± 0.4	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	$\mathbf{0.0 \pm 0.0}$	0.9 ± 0.2

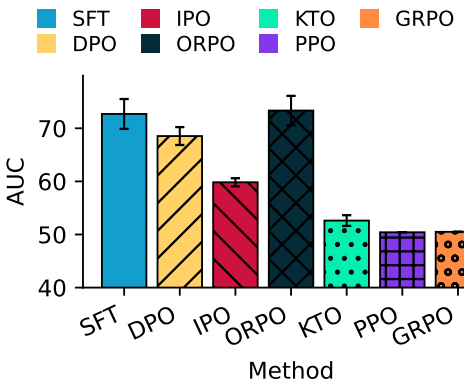


Figure 24: **AUC by post-training objective.** Average AUC for each post-training objective, using the strongest attack instance selected per run.

Table 16: **AUC is a threshold-agnostic companion metric.** Average AUC on the main evaluation set. Rows are attack instances, columns are post-training objectives, and entries report mean $\pm$ SE.

Attack	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
LiRA Preferred-response	66.4 ± 2.5	54.3 ± 0.8	50.4 ± 0.1	64.4 ± 2.3	50.5 ± 0.1	$\mathbf{50.4 \pm 0.2}$	$\mathbf{50.3 \pm 0.0}$	56.1 ± 0.8
RMIA Preferred-response	$\mathbf{74.4 \pm 2.9}$	62.6 ± 1.4	52.4 ± 0.5	$\mathbf{73.2 \pm 2.9}$	51.4 ± 0.6	50.1 ± 0.0	50.2 ± 0.0	60.7 ± 1.1
InfoRMIA Preferred-response	74.2 ± 2.9	59.9 ± 1.3	51.8 ± 0.4	72.7 ± 2.9	51.1 ± 0.4	50.2 ± 0.1	50.2 ± 0.0	60.0 ± 1.1
LiRA Dispreferred-response	51.1 ± 0.2	54.2 ± 0.8	50.4 ± 0.1	54.4 ± 1.1	50.4 ± 0.1	$\mathbf{50.4 \pm 0.1}$	50.2 ± 0.0	51.7 ± 0.3
RMIA Dispreferred-response	53.5 ± 0.5	61.7 ± 1.4	52.7 ± 0.5	61.4 ± 2.1	51.2 ± 0.5	50.2 ± 0.1	50.2 ± 0.0	55.0 ± 0.5
InfoRMIA Dispreferred-response	53.5 ± 0.5	59.5 ± 1.3	52.0 ± 0.5	59.9 ± 1.9	50.9 ± 0.3	50.1 ± 0.0	50.2 ± 0.0	54.2 ± 0.5
LiRA Scalar preference-margin	67.4 ± 2.5	57.4 ± 1.1	53.8 ± 0.6	67.5 ± 2.4	50.7 ± 0.3	50.2 ± 0.1	$\mathbf{50.3 \pm 0.0}$	57.9 ± 0.8
RMIA Scalar preference-margin	71.0 ± 2.7	61.9 ± 1.4	$\mathbf{59.2 \pm 0.7}$	71.3 ± 2.6	51.6 ± 0.6	50.2 ± 0.1	50.2 ± 0.0	60.9 ± 0.9
InfoRMIA Scalar preference-margin	70.9 ± 2.7	57.6 ± 1.2	56.9 ± 0.6	67.3 ± 2.3	51.2 ± 0.4	50.2 ± 0.1	50.2 ± 0.0	59.1 ± 0.9
LiRA Signed response-pair	66.0 ± 2.5	50.9 ± 0.2	50.2 ± 0.0	61.1 ± 1.8	50.2 ± 0.0	50.1 ± 0.0	$\mathbf{50.3 \pm 0.0}$	54.9 ± 0.7
RMIA Signed response-pair	67.8 ± 2.2	$\mathbf{68.3 \pm 1.7}$	54.9 ± 0.7	72.4 ± 2.7	$\mathbf{52.4 \pm 1.0}$	50.3 ± 0.0	50.2 ± 0.0	60.9 ± 0.9
InfoRMIA Signed response-pair	72.0 ± 2.7	67.6 ± 1.8	54.5 ± 0.8	73.0 ± 2.8	52.3 ± 1.0	50.2 ± 0.0	50.2 ± 0.0	$\mathbf{61.6 \pm 1.0}$

Table 17: **More reference models strengthen the audit, with diminishing returns beyond $K = 32$.** TPR at FPR$= 1\%$ on Qwen3-0.6B trained on Chatbot Arena and UltraFeedback. Rows are post-training objectives, and columns are the number of reference models K.

Method	$K = 8$	$K = 16$	$K = 32$	$K = 64$	$K = 128$
SFT	19.8	23.1	24.8	25.6	26.1
DPO	17.3	17.4	17.9	19.5	20.4
IPO	9.9	13.7	15.8	17.4	18.4
ORPO	24.9	27.8	28.8	29.2	29.5

R Record-Statistic Comparison Without Per-Cell Selection

The per-instance TPR at FPR$= 1\%$ values without any selection are exactly those of Table 4, where every row is one attack instance and no per-cell argmax is involved. To compare the record statistics directly, Table 18 reports the difference Δ between the signed response-pair and each single statistic under the same scorer. Overall, the signed response-pair is significantly stronger than the dispreferred-response statistic and than the scalar preference-margin for RMIA and InfoRMIA, by $+5.4$ to $+6.0$ and by $+4.6$ to $+5.0$. Its advantage is general but setting-dependent. It is largest where the objective contrasts the two responses, reaching $+15.2$ on ORPO and $+11.0$ on DPO, and it reverses on SFT against the preferred-response statistic, by -7.3 and -4.3, since SFT takes gradients on the preferred response only. No statistic is therefore best everywhere, but the signed response-pair is the only one that stays competitive under every objective that leaks, so omitting it under-reports the leakage.

Table 18: **The signed response-pair is the only statistic that stays competitive wherever leakage occurs.** Difference Δ in TPR at FPR$= 1\%$ between the signed response-pair and each single record statistic under the same scorer, without per-cell selection. Each difference is tested with a paired Wilcoxon signed-rank test with Benjamini-Hochberg correction, where * denotes $q < 0.10$ and ** denotes $q < 0.01$.

Scorer	Record statistic	SFT	DPO	IPO	ORPO	KTO	PPO	GRPO	All
LiRA	Δ vs preferred-response	-2.7^{*}	-1.4^{**}	0.0	-6.2^{**}	0.0	-0.1	0.0	-1.7^{**}
LiRA	Δ vs dispreferred-response	$+6.4^{**}$	-1.7^{**}	0.0	$+1.5^{*}$	0.0	0.0	0.0	$+1.0$
LiRA	Δ vs scalar preference-margin	$+1.4$	0.0	-0.6^{**}	-3.2	0.0	0.0	0.0	-0.4
RMIA	Δ vs preferred-response	-7.3^{**}	$+4.4^{**}$	$+0.5^{*}$	$+1.6$	$+0.4^{*}$	0.0	0.0	-0.3^{*}
RMIA	Δ vs dispreferred-response	$+12.6^{**}$	$+4.3^{**}$	$+0.4$	$+15.2^{**}$	$+0.4^{*}$	0.0	0.0	$+5.4^{**}$
RMIA	Δ vs scalar preference-margin	$+5.7$	$+10.4^{**}$	-0.3	$+12.2^{**}$	$+0.6$	0.0	0.0	$+4.6^{**}$
InfoRMIA	Δ vs preferred-response	-4.3^{**}	$+6.2^{**}$	$+0.5^{**}$	-0.3	$+0.2^{*}$	0.0	0.0	$+0.3^{*}$
InfoRMIA	Δ vs dispreferred-response	$+15.1^{**}$	$+6.2^{**}$	$+0.4$	$+13.7^{**}$	$+0.3$	0.0	0.0	$+6.0^{**}$
InfoRMIA	Δ vs scalar preference-margin	$+8.2^{*}$	$+11.0^{**}$	-0.4	$+11.1^{**}$	$+0.4$	0.0	0.0	$+5.0^{**}$

S Preference Gap and Leakage

The benchmark datasets differ in how far apart the preferred and the dispreferred response of a record are. To quantify this, we randomly select 2,000 records per dataset, score the per-token log-likelihood margin of each record under the pretrained Qwen3-0.6B, and average over the records:

$$g = \left| \frac{1}{n} \sum_{i=1}^{n} \left(\frac{1}{|y_{w,i}|} \log p_0(y_{w,i} \mid x_i) - \frac{1}{|y_{l,i}|} \log p_0(y_{l,i} \mid x_i) \right) \right|. \tag{29}$$

Table 19 reports TPR at FPR$= 1\%$ for the three objectives that use the response pair, together with the Spearman correlation ρ against this preference gap. A larger preference gap, as in HH-RLHF, yields an overall lower MIA success than the smaller gaps of UltraFeedback and Chatbot Arena, and this trend is also captured by the negative correlation with g. To further verify this finding, we repeat the same analysis on the synthetic canaries of Table 2 and score g for each canary type under the same pretrained model. Table 20 shows that a smaller preference gap yields higher leakage for every objective. We hypothesize that a pair whose two responses are near-equivalent carries no rule determining which one a rater should pick, so it behaves like a noisy label, and noisy labels tend to be memorized [16].

Table 19: **Datasets with a smaller preference gap leak more.** Preference gap g and TPR at FPR$= 1\%$ per dataset for the objectives that use the response pair, with the Spearman correlation against g in the last row.

Dataset	g	DPO	IPO	ORPO	Mean
UltraFeedback	0.146	16.4	3.6	33.8	17.9
HH-RLHF	0.322	8.1	2.0	22.7	10.9
Chatbot Arena	0.157	16.1	3.5	10.8	10.2
ρ with g		-1.00	-1.00	-0.50	-0.50

Table 20: **Canary types reproduce the preference-gap trend.** Preference gap g and TPR at FPR$= 1\%$ per canary type, with the Spearman correlation against g in the last row. The canary types are those of Table 2.

Canary	g	DPO	IPO	ORPO	Mean
Normal	0.007	3.2	4.5	3.2	3.7
MisPref	0.007	12.8	9.5	1.5	7.9
RealP + RandR	0.031	7.9	4.4	2.8	5.0
RandStr	0.091	9.3	10.6	3.4	7.8
RandP + RealR	0.368	6.1	2.1	1.9	3.4
RealP + RandPref	4.146	3.1	2.6	1.5	2.4
ρ with g		-0.49	-0.58	-0.09	-0.70

T Per-Model Breakdown of the Larger-Model Subset

The standard errors of Table 5 are computed per setting, that is, across models and datasets, which summarizes the trend of each scorer and post-training objective in a single number. Tightening them would therefore require more models and datasets rather than more reference models. When the same runs are reported individually per model and dataset, the standard errors are computed over the reference models and are already tight. Table 21 gives this breakdown for Qwen3-4B on Chatbot Arena, where the largest standard error is 0.7 on estimates that reach 36.6. Qwen3-8B shows the same pattern, with a largest standard error of 0.8.

We further vary the number of reference models K for this subset. Table 22 reports the same standard error as a function of K, averaged over the 12 attack instances and the four objectives of each model. The standard error is already small at $K = 4$, and doubling the number of reference models to $K = 8$ reduces it to 0.23, so four reference models are sufficient here.

Table 21: **Per-model standard errors are tight.** TPR at FPR$= 1\%$ for Qwen3-4B on Chatbot Arena, reported as mean $\pm$ SE over the reference models. Rows are attack instances and columns are post-training objectives.

Scorer	Record statistic	SFT	DPO	IPO	ORPO
LiRA	Preferred-response	1.8 ± 0.2	2.0 ± 0.1	1.0 ± 0.1	1.0 ± 0.1
RMIA	Preferred-response	36.6 ± 0.5	13.0 ± 0.3	1.4 ± 0.1	24.8 ± 0.3
InfoRMIA	Preferred-response	36.6 ± 0.5	13.0 ± 0.3	1.4 ± 0.1	24.8 ± 0.3
LiRA	Dispreferred-response	1.0 ± 0.1	2.3 ± 0.2	1.1 ± 0.1	1.0 ± 0.0
RMIA	Dispreferred-response	1.6 ± 0.1	13.7 ± 0.3	1.5 ± 0.1	1.2 ± 0.1
InfoRMIA	Dispreferred-response	1.6 ± 0.1	13.7 ± 0.3	1.5 ± 0.1	1.2 ± 0.1
LiRA	Scalar preference-margin	2.2 ± 0.1	2.0 ± 0.1	3.0 ± 0.1	2.1 ± 0.1
RMIA	Scalar preference-margin	3.8 ± 0.4	4.2 ± 0.5	4.8 ± 0.3	3.1 ± 0.3
InfoRMIA	Scalar preference-margin	3.8 ± 0.4	4.2 ± 0.5	4.8 ± 0.3	3.1 ± 0.3
LiRA	Signed response-pair	7.8 ± 0.3	1.2 ± 0.1	1.0 ± 0.0	3.4 ± 0.2
RMIA	Signed response-pair	32.9 ± 0.6	20.7 ± 0.2	1.7 ± 0.2	24.0 ± 0.4
InfoRMIA	Signed response-pair	25.7 ± 0.7	23.3 ± 0.1	1.4 ± 0.1	19.1 ± 0.4

Table 22: **Four reference models already suffice on the larger-model subset.** Mean standard error of TPR at FPR$= 1\%$ as a function of the number of reference models K, averaged over the 12 attack instances and the four post-training objectives of each model.

K	Qwen3-4B mean SE	Qwen3-8B mean SE
2	0.39	0.41
3	0.35	0.37
4	0.32	0.34
5	0.29	0.31
6	0.27	0.28
7	0.25	0.26
8	0.23	0.25

U Limitations

This work focuses on membership privacy in preference-based post-training and studies this risk through a controlled empirical benchmark. While membership inference is a standard and informative tool for privacy auditing, it captures one important class of leakage rather than every possible privacy concern. Other complementary analyses, such as extraction-based audits or attribute-inference evaluations, could further enrich the assessment of post-training privacy. Our experiments span several preference datasets, open model families, and post-training objectives, but they do not cover every model or deployment setting. In particular, we focus on open models for which controlled post-training and privacy auditing are feasible. Extending the benchmark to larger proprietary systems, additional preference-data sources, and broader adaptation pipelines would be a valuable direction for future work.

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: an empirical membership-privacy analysis of preference-based post-training, with results on attack representation, post-training objectives, PEFT, DP, sequence length, and canaries.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix U discuss the limitations of the work.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

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: Justification: The paper contains one formal anchor-invariance lemma for the scalar preference-margin statistic, proved in Appendix B. The rest of the paper is empirical.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

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: Sections 3 and 4 and appendix J.1 specify the target examples, reference-model protocol, datasets, models, methods, metrics, and additional evaluation settings. Appendix J.2 gives the DP accounting and implementation details used for the saved DP grid.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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: [Yes].

Justification: We release an anonymized version of the code and provide reproduction instructions in the supplementary material, including the environment file, data-access and preprocessing steps, and the commands used to run the main training and attack experiments. The released code reproduces the main proposed-method and baseline results; upstream public datasets and pretrained model weights are accessed from their original providers rather than redistributed.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be evaluated negatively simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

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

Answer: [Yes]

Justification: Section 4 and appendix J.1 give the datasets, models, methods, candidate universe, reference-model sampling rule, metrics, PEFT methods, sequence-length sweep, and canary setup; Appendix J.2 adds the DP batch, clipping, accounting, and objective-specific details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

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: [Yes]

Justification: The aggregate tables report mean $\pm$ SE where exported values are available, and Appendix J.1 states the scored-record count and aggregation convention used for standard errors.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

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 J.6 describes the execution time for the main experiments and the type of hardware used.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

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 authors fully comply with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We discuss the societal impact of our work in Appendix P.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

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., pretrained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The research relies on publicly available models and datasets, presenting no significant risk of misuse. Therefore, no specific safeguards were required. We also do not release trained models, private datasets, or membership-labeled sensitive artifacts. The released code is intended for auditing on public datasets. Potential misuse of membership-inference techniques is discussed in Appendix P.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

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: Our released code does not include source code or binary files from external libraries. Therefore, no additional permissions or license files are required for bundled third-party software. We use only open-source datasets and models, all of which are cited in the paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [Yes]

    Justification: We release the code introduced in this paper alongside documentation in the supplementary material/README. The documentation includes the environment setup, experiment commands, data preparation instructions and expected outputs. The released assets are anonymized for submission.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

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: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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: This work does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

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 rigorousness, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The evaluated language models are the object of study, not an original or non-standard component used to develop the method.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.