MEMBERSHIP INFERENCE ATTACKS ON LANGUAGE MODELS WITH MULTIPLE ALIGNMENT METHODS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are fine-tuned by different methods on different data during their alignment for knowledge acquisition or preference optimization. The data used in this stage may leak sensitive information. In this work, we propose membership inference attacks for multiple and multi-stage LLM alignment methods. We introduce a extensible set of signals covering different aspects of the alignment pipeline to map the sample data to a multi-dimensional vector. Depending on the availability of shadow models, we propose shadow-free **Envelope (ENV)** and shadow-based **shadow-calibrated likelihood ratio (SLR)** to construct the final composite score for membership inference. We evaluate the framework on five model families (GPT-2, Llama-3.2-1B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Llama-3.1-8B-Instruct) under two reward granularities(a continuous learned reward and a binary verifier) and multiple alignment pipelines. The extensive results indicate our proposed methods can achieve stronger average AUC than existing methods across all configurations.

1 INTRODUCTION

Large language models (LLMs) are fine-tuned using alignment methods such as supervised fine-tuning (SFT) (Ouyang et al., 2022), direct preference optimization (DPO) (Rafailov et al., 2023), negative preference optimization (NPO) (Zhang et al., 2024), group relative policy optimization (GRPO) (Shao et al., 2024), and proximal policy optimization (PPO) (Schulman et al., 2017). The LLMs are usually fine-tuning by more than one alignment methods using different data, like using SFT in the first stage for knowledge acquisition, and DPO, NPO, GRPO or PPO in the second stage for preference-based alignment. The training data used in these methods include instruction sets, preference pairs, and verifiable problems, which may contain user-generated or proprietary information. Membership inference attacks (MIAs) (Shokri et al., 2017) asks whether a given prompt was part of the training data. This question is more important for alignment data than for pretraining data, since alignment data is more likely to contain sensitive information.

Classic membership signals include likelihood (Yeom et al., 2018; Carlini et al., 2022), compression ratios (Carlini et al., 2021), and Min-K% (Shi et al., 2024). These signals were designed for models that memorize or overfit their training text. However, LLM alignment methods, including SFT and preference optimization, usually have very benign overfitting. Therefore, MIAs based on these classic signals can hardly distinguish training members from non-members. More recently, PRE-MIA (Feng et al., 2024) and DIBA (Liu et al., 2025b) show that this behavioral shift is measurable and exploitable, but each of them uses a few hand-picked statistics and one calibration mechanism, so they only work on some particular fine-tuning methods, reward granularities, and levels of attacker knowledge. Moreover, developing LLMs in reality commonly has multiple alignment stages rather than a single method, and model documentation rarely discloses the detailed settings in the full pipeline. For example, InstructGPT runs SFT, reward-model training, and PPO in sequence (Ouyang et al., 2022). The Llama 3 herd combines SFT, rejection sampling, and DPO (Grattafiori et al., 2024). Tülu 3 appends an RLVR stage with verifiable rewards to its DPO policy (Lambert et al., 2024). DeepSeek-R1 alternates supervised fine-tuning and rule-reward reinforcement learning (Guo et al., 2025). Therefore, an ideal membership inference attack method must work for multiple alignment algorithms when the training pipeline is *unknown* and possibly *multi-stage*.

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

This paper builds a unified framework for membership inference on fine-tuned LLMs. We introduce multiple signals of different categories, and each target sample is summarized as a multi-dimensional detector vector. This representation generalizes the signals of prior attacks is extensible by adding new families of signals. Depending on whether to use shadow models (Shokri et al., 2017), we introduce two scoring families that turn the vector into a membership score: a shadow-free **Envelope (ENV)** and a shadow-based **shadow calibrated likelihood ratio (SLR)**. ENV sorts the detector signal values, calibrates each rank against a held-out reference population, and scores the largest standardized excess. SLR blends statistical twin shadows through a fingerprint mechanism that recovers the target's training configuration without any alignment method label.

We evaluate the framework on five model families (GPT-2 Medium, Llama-3.2-1B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Llama-3.1-8B-Instruct), five alignment methods (SFT, DPO, NPO, GRPO, and PPO), and two reward granularities (a continuous learned reward model on HH-RLHF and a binary verifier on a GSM8K set). Across these settings, ENV outperforms shadow-free baselines and SLR outperforms shadow-based baselines in most cases. Multi-stage alignment experiments show that our multi-alignment MIA better captures training trace across multi-stages than baselines.

This paper makes the following contributions:

- We establish the foundation for membership inference attacks against multiple and multi-stage LLM alignment methods by introducing multiple signals that cover different axes of behavioral change in LLM alignment, including the prompt, the chosen and rejected likelihoods, the token logits, and the reward profile.
- Depending on the availability of shadow models, we propose two scoring families, including a shadow-free Envelope (ENV) and a shadow-based shadow-calibrated likelihood ratio (SLR), to blend the multi-dimensional signals into a score for membership inference.
- We evaluate on five alignment methods, two reward granularities, five model families, and multi-stage pipelines. Extensive experimental results indicate our proposed methods are broadly applicable and have superior performance in most settings.

Threat Model and General Settings. We consider an attacker that holds a pool of candidate prompts and must decide whether each prompt was used to train the target model. To make this decision, the attacker can use a frozen reference model and a set $\mathcal{P}$ of 100 prompts excluded from all training. The attacker does *not* know which configuration was used to train the target model. We consider both shadow-free and shadow-based settings. In shadow-based settings, our method uses a *shadow model*, which is an auxiliary model trained on data whose membership status is known. It lets the attacker observe how member and non-member prompts behave on a model trained under a known configuration, and then use that behavior to calibrate the attack on the target. In this work, we train one shadow model for each of the five training configurations considered, and each shadow model is trained on disjoint shadow data.

2 Related Work

Alignment Fine-Tuning of Language Models. Supervised fine-tuning fits the policy to a corpus of (prompt, response) pairs by next-token cross-entropy. Instruction tuning in this paradigm produced the first broadly usable aligned models (Ouyang et al., 2022). The RLHF family extends supervised fine-tuning with a preference signal. Reward-model RLHF trains a learned preference model to score policy rollouts, and PPO optimizes a clipped policy-gradient objective against that score under a KL anchor to the supervised policy (Schulman et al., 2017; Ouyang et al., 2022). DPO (Rafailov et al., 2023) removes the explicit reward model by rewriting the objective as a contrast between the policy's chosen- and rejected-response likelihoods against a frozen reference policy. NPO (Zhang et al., 2024) softens the contrast into a softplus loss that only pushes the rejected response down, introduced as an unlearning objective that avoids the catastrophic collapse of pure gradient ascent. GRPO (Shao et al., 2024) replaces the learned critic with a group baseline. Rollouts are sampled as a group, rewards are standardized within the group, and advantages are computed from the group statistics. The DAPO (Yu et al., 2026) extension adds a dynamic filter that zeroes the advantage of degenerate groups. RLVR (Liu et al., 2025a) replaces the learned reward model with verifiable rule-based rewards, trading reward generality for a ground-truth training signal.

Membership Inference for Language Models. Membership inference attacks predates large language models and is classically formulated as deciding whether a given example x was in the target model's training set. The canonical black-box attack of Shokri et al. (Shokri et al., 2017) trains a set of shadow models on data drawn from a distribution similar to the target's training distribution, queries them on known members and non-members, and uses their confidence vectors to train an attack classifier. Yeom et al. formalize membership advantage and relate it to overfitting (Yeom et al., 2018). Subsequent work broadened this picture to more general overfitting assumptions (Salem et al., 2018; Long et al., 2018). Beyond these, MIAs were extended to white-box access using gradients and activations (Nasr et al., 2019; Leino & Fredrikson, 2020), to label-only settings (Choquette-Choo et al., 2021), and to calibrated likelihood-ratio tests over multiple shadow models (Carlini et al., 2022). Together, these papers define the basic threat model, evaluation protocol, and attack signals of MIAs.

On language models the natural signal for MIA is per-token likelihood. LOSS (Yeom et al., 2018) carries the loss-threshold attack to text as the mean per-token log-probability, and the Zlib ratio (Carlini et al., 2021) normalizes the likelihood by the compressed size of the text, on the observation that memorized text is both likely and incompressible. LiRA (Carlini et al., 2022) casts membership as a hypothesis test calibrated by shadow models. Two modern signals target token-level memorization. Min-K% (Shi et al., 2024) averages the log-probabilities of the lowest-loss tokens. Neighbor (Mattern et al., 2023) normalizes a sample's likelihood by the likelihoods of its nearest neighbors in a paraphrase embedding space. These signals were designed for standard language models and do not transfer well to aligned policies.

Only two membership inference attacks target alignment training: PREMIA (Feng et al., 2024), a shadow-free method that scores preference displacement, and S-DIBA (Liu et al., 2025b), which casts membership as binary classification over five reward signals, using logistic regression and a random forest trained on a single shadow model.

3 PRELIMINARY

This section presents the notation shared by all attacks in the paper and describes the baselines used for comparison: classical LM membership signals, PREMIA, and S-DIBA. For each method, we describe the signal first and the score second.

Notation. The target model is a policy π_θ fine-tuned by some algorithms, such as SFT, DPO, NPO, GRPO, or PPO. We use $\mathcal{V}$ to denote the vocabulary and $\boldsymbol{y} = (y_1, ..., y_n)$ to represent an n-token response. For each position i, we define the per-token log-probability $p_i = \log \pi_\theta(y_i \mid y_{<i}, \mathbf{x})$ and the per-token predictive entropy $h_i = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid y_{<i}, \mathbf{x}) \log \pi_\theta(v \mid y_{<i}, \mathbf{x})$. For GRPO and PPO, we use a to represent the reward function. For any prompt, the model can generate $G = 20$ rollouts $\{\boldsymbol{r}_i\}_{i=1}^{G}$ from π_θ, each of which is scored by a.

The sample considered in MIA in this work is a triple $(\boldsymbol{x}, c, r)$ consisting of a prompt $\mathbf{x}$, a chosen response c, and a rejected response r. Therefore, we adopt datasets that provide the complete triple, such as HH-RLHF (Bai et al., 2022) and an IMDB-based preference dataset constructed in the same format as in Maas et al. (2011). Table 4 of Appendix A demonstrate the meanings of all notations.

Classic MIAs. Classical LM membership signals include LOSS, Zlib, Min-K%, Entropy, and Neighbor. Detailed explanations are deferred to Appendix B.1. These signals are applied in two modes: **self-gen**, which averages over $G{=}20$ responses generated by the target policy, and **static**, which is computed on the fixed chosen response that the sample carries. Neighbor is evaluated in static mode only. All five signals were designed to detect sorted text, as each quantifies how well the target predicts some text. Alignment training shifts behavior instead of storing text, so these classical signals do not perform well in alignment training.

PREMIA (Feng et al., 2024) and S-DIBA (Liu et al., 2025b) are membership inference attacks designed for the alignment training setting in LM. PREMIA uses the preference displacement: $s_{\mathrm{PREMIA}} = \max\Big(\big|\log \pi_\theta(c|\mathbf{x}) - \log \pi_{\mathrm{ref}}(c|\mathbf{x})\big|,\ \big|\log \pi_\theta(r|\mathbf{x}) - \log \pi_{\mathrm{ref}}(r|\mathbf{x})\big|\Big)$. S-DIBA formulates MIA in a five-dimensional feature space with features $a_{\max}$, a_{mean}, a_{std}, b_{KL}, and b_{token}, and trains a logistic regression and a random forest ensemble on a shadow model. Detailed definitions and explanations are deferred to Appendix B.2.

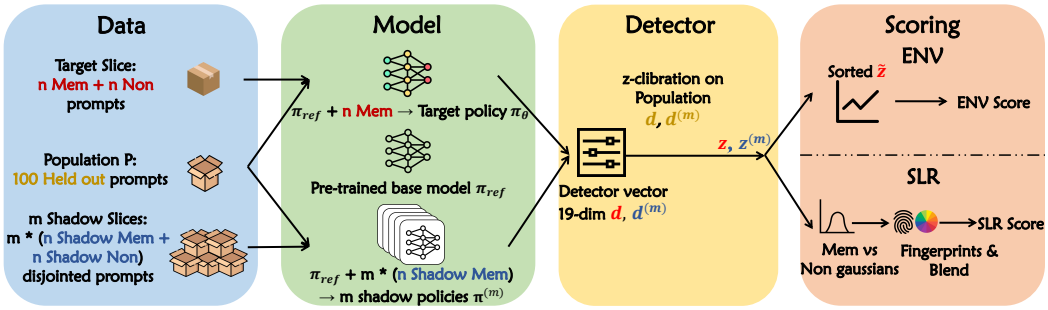

Figure 1: Overview of our MIA framework. The figure shows how data are partitioned into target, shadow, and population sets and how they flow through the pipeline. Each prompt is represented as a 19-dimensional detector vector, converted into a membership score by either the shadow-free ENV or the shadow-based SLR, and finally thresholded to decide membership.

The motivation of this work is that the signals used in classical MIAs are not comprehensive across various fine-tuning methods. For example, PREMIA reads only the preference displacement, so it captures the trace when the target is trained by DPO or NPO but misses the trace left by SFT or by reward-based objectives such as GRPO and PPO. S-DIBA reads only reward statistics, so it misses token-level and likelihood traces. Since the attacker does not know which fine-tunning method was used, a signal that reads only one category of behavior change cannot adapt to the target alignment. We therefore propose an attack that does not commit to a single category of behavior change.

4 METHODOLOGY

We propose two MIA methods for different settings: **SLR (shadow-calibrated likelihood ratio)** when shadow models are available and **ENV (Envelope)** otherwise. Each of them adapts a comprehensive set of signals to capture the effect of multiple fine-tuning methods. Section 4.1 describes how to construct the 19-dimensional detector vector for a prompt. Section 4.2 explains how to convert this detector vector into a score, which is then compared with a threshold to decide whether the prompt is a member. Figure 1 provides an overview of our proposed method.

4.1 DETECTOR SIGNALS

The detector vector is built by enumeration rather than search. Fine-tuning changes a model's behavior in a limited number of ways. We identify 4 axes that cover these ways: the prompt, the chosen and rejected likelihoods, the token logits, and the reward profile. The change on each axis depends on the alignment objective. For example, SFT fits the chosen response token by token, so it leaves a token-level fit on the chosen response. Preference objectives such as DPO and NPO push the likelihoods of the chosen and rejected responses apart, so they leave a displacement on the preference pair. Reward-based objectives such as GRPO and PPO change the reward of the policy's own rollouts, so they leave a lifted reward profile or an overfit peak in that profile. Prior membership inference methods each read only one of these four families. They therefore capture the trace only when fine-tuning happens to change that family, and they miss the behavior changes that other objectives leave on the remaining axes. In multi-alignment membership inference, the attacker does not know which alignment method produced the target, so a single-family method cannot adapt and fails whenever the trace lies on a family it does not read.

We instead design signals to cover all 4 families. Each axis is represented by a small family of statistics, starting from a signal that targets this axis and completed by its natural relatives. This yields 19 signals, $D_1, \ldots, D_{19}$, organized into four groups and listed in Table 1. All signals are oriented so that a higher value indicates a higher likelihood of membership.

Membership inference is performed at the prompt level. For each triple $(\mathbf{x}, c, r)$, the attacker decides whether $\mathbf{x}$ was used during training, while c and r provide the response-level signal through which the aligned policy's behavior on $\mathbf{x}$ is observed. Some objectives use only part of the triple. We still

Table 1: The 19 detector signals. All signals are oriented so that a higher value indicates a higher likelihood of membership. Detailed definitions and justifications are in Appendix C.

ID	Definition	ID	Definition	ID	Definition
Prompt-level		*Token-level (cont.)*		*Reward-based (cont.)*	
D_1	$\frac{1}{\lvert\mathbf{x}\rvert}\sum_{t=1}^{\lvert\mathbf{x}\rvert}\log\pi_\theta(x_t \mid \mathbf{x}_{<t})$	D_7	$\frac{1}{n}\sum_i q_i$	D_{13}	$\max(\mathbf{r})-\bar{r}$
D_2	$\frac{1}{\lvert\mathcal{P}\rvert}\sum_{\mathbf{x}'\in\mathcal{P}}\mathbb{I}[\text{PPL}_\theta(\mathbf{x})<\text{PPL}_\theta(\mathbf{x}')]$	D_8	$\sqrt{\frac{1}{n}\sum_i(q_i-D_7)^2}$	D_{14}	$\bar{r}$
Likelihood-ratio		D_9	$\frac{\frac{1}{n}\sum_i(q_i-D_7)^3}{(\frac{1}{n}\sum_i(q_i-D_7)^2)^{3/2}}$	D_{15}	$\max(\mathbf{r})$
D_3	$\lvert(\ell_c-\tilde{\ell}_c)-(\ell_r-\tilde{\ell}_r)\rvert$	*Reward-based*		D_{16}	$\min(\mathbf{r})$
D_4	$\lvert\ell_r-\tilde{\ell}_r\rvert$	D_{10}	$\sqrt{\frac{1}{G}\sum_j(r_j-\bar{r})^2}$	D_{17}	$\frac{1}{k}\sum_{j:\,r_{(j)}\,\text{top-}k} r_{(j)}, k=\lfloor G/4\rfloor$
Token-level		D_{11}	$\frac{\max(\mathbf{r})-\bar{r}}{D_{10}+10^{-8}}$	D_{18}	$\text{clip}\big(\frac{\bar{r}}{D_{10}+10^{-8}},\,0,\,10^3\big)$
D_5	$\frac{1}{n}\sum_i p_i$	D_{12}	$\max(\mathbf{r})-\min(\mathbf{r})$	D_{19}	$\bar{r}_{\pi_\theta}(\mathbf{x})-\bar{r}_{\pi_{\text{ref}}}(\mathbf{x})$
D_6	$\text{Percentile}_{10}(\{p_i\})$				

compute the signals associated with the unused response, but since that response never influenced the policy's behavior, those signals carry little membership information. The detailed definitions, motivations, and connections to prior work are deferred to Appendix C.

4.2 MIA FOR MULTIPLE ALIGNMENT

All detector signals are standardized against the reference population. For a sample with standardized 19-dimensional detector vector $\mathbf{d}$, we compute $\mathbf{z}=(\mathbf{d}-\boldsymbol{\mu}_{\text{pop}})/(\boldsymbol{\sigma}_{\text{pop}}+10^{-8})$, where $\boldsymbol{\mu}_{\text{pop}}$ and $\boldsymbol{\sigma}_{\text{pop}}$ are the per-dimension mean and standard deviation of each signal over the reference population $\mathcal{P}$. Both the sample's detector vector and the population statistics therefore come from the same model. For the target model, the π_θ; for each shadow, the $\pi^{(m)}$. Each component of $\mathbf{z}$ is then converted to a percentile via the population empirical CDF for shadow-calibrated scoring. We keep the symbol $\mathbf{z}$ for the percentile-transformed vector, so all subsequent uses of $\mathbf{z}$ refer to the percentile version of the detector signals.

4.2.1 SHADOW-FREE: THE ENVELOPE (ENV)

The Envelope needs no alignment-specific shadow. It uses only the population of held-out prompts, which act as non-members. A sample is judged by how far its detector profile sits from the population's profile. The larger the distance, the more likely the sample is a member. The 19 signals do not all discriminate. Some carry the alignment trace while others stay flat. Since the attacker does not know which axes are informative, the score must identify them adaptively. A natural first choice is to take the single signal with the largest excess over the population. This works when one signal is strong enough on its own, but it fails when the informative signals are individually weak. In that case no single signal is convincing, and accurate detection requires several signals to agree.

Sorting the profile provides such an adaptive choice. It also makes the score invariant to which axis carries the trace. For a target sample let $\text{sort}(\mathbf{z})_\downarrow=(\tilde{z}_1\geq\tilde{z}_2\geq\cdots\geq\tilde{z}_{19})$ be its sorted profile, where $\tilde{z}_k$ is the raw sorted value at rank k. Sorting defines 19 strategies, where strategy k reads the k-th largest signal of the sample. The raw sorted values $\tilde{z}_k$ are not directly comparable across ranks: the value at rank 1 is on average larger than the value at rank 2, and so on. We therefore calibrate each rank against the same rank on the population's sorted profiles. The population $\mathcal{P}$ provides the per-rank mean and spread μ_k,σ_k. The sample's profile is standardized rank by rank, and the largest excess over the 19 ranks is the score:

$$\hat{s}_k=\frac{\tilde{z}_k-\mu_k}{\sigma_k+10^{-8}},\qquad s_{\text{ENV}}(\mathbf{z})=\max_{k=1}^{19}\hat{s}_k. \tag{1}$$

If the winning rank is k, every rank above k in the sorted order carries at least as large a raw value. After per-rank calibration, this means the top k signals are jointly elevated relative to the population, and k itself is chosen against the population. The per-rank population statistics μ_k and σ_k define an upper envelope over the sorted profile, and the score is the largest excess of the sample's profile above this envelope. This is why we name the method Envelope. **Envelope-simple** (ENV-simple) is the max of the unsorted z-scores, without order-statistic calibration. It is the Envelope's complementary case.

4.2.2 SHADOW-BASED: SLR

SLR uses shadow models for calibration. Each shadow model supplies member and non-member distribution estimates, and a fingerprint mechanism selects and blends the correct shadow calibration.

Fingerprint and blending. For each alignment method $m \in \{\text{SFT}, \text{DPO}, \text{NPO}, \text{GRPO}, \text{PPO}\}$, we train a *statistical twin (shadow model)* $\pi^{(m)}$ of the target. The shadow model $\pi^{(m)}$ is obtained by applying alignment method m to the frozen base model on a disjoint data slice (shadow data), using the same training configuration and sample count. For a sample with 19-dimensional detector vector $\mathbf{z}$, the shadow yields n_{mem} member detector vectors and n_{non} non-member detector vectors, both equal to n. From these vectors we estimate two Gaussian distributions, one for members and one for non-members:

$$\boldsymbol{\mu}_*^{(m)} = \frac{1}{n_*}\sum_{i=1}^{n_*} \mathbf{z}_{i,*}^{(m)}, \qquad \boldsymbol{\Sigma}_*^{(m)} = \text{LW}\big(\{\mathbf{z}_{i,*}^{(m)}\}\big), \qquad * \in \{\text{mem}, \text{non}\}, \tag{2}$$

where $\mathbf{z}_{i,*}^{(m)}$ is the i-th detector vector of class $*$ produced by the shadow model $\pi^{(m)}$. Each shadow model produces a 19-dimensional *fingerprint*, the mean detector vector over the population $\mathcal{P}$. The target fingerprint $\mathbf{f}_t$ is matched against the five shadow fingerprints. Blend weights are a softmax over negative squared distances (hyperparameter $\beta = 2.0$), and the blended distribution parameters are the weighted averages. The blend selects the correct calibration. Models trained by the same alignment method share a fingerprint, and the softmax concentrates on the matching shadow.

$$w_m = \frac{\exp(-\beta\|\mathbf{f}_t - \mathbf{f}^{(m)}\|^2)}{\sum_{m'}\exp(-\beta\|\mathbf{f}_t - \mathbf{f}^{(m')}\|^2)}, \qquad \hat{\boldsymbol{\mu}}_* = \sum_m w_m \boldsymbol{\mu}_*^{(m)},\ \hat{\boldsymbol{\Sigma}}_* = \sum_m w_m \boldsymbol{\Sigma}_*^{(m)}. \tag{3}$$

Scoring. Given the blended member and non-member Gaussian distributions estimated from the shadow models, the sample is scored by the joint-Gaussian log-likelihood ratio of its 19-dimensional detector vector $\boldsymbol{z}$ under the member and non-member distributions: $s_{\text{SLR}}(\mathbf{z}) = \log\mathcal{N}(\mathbf{z}; \hat{\boldsymbol{\mu}}_{\text{mem}}, \hat{\boldsymbol{\Sigma}}_{\text{mem}}) - \log\mathcal{N}(\mathbf{z}; \hat{\boldsymbol{\mu}}_{\text{non}}, \hat{\boldsymbol{\Sigma}}_{\text{non}})$, which expands to:

$$s_{\text{SLR-QDA}}(\mathbf{z}) = \frac{1}{2}\Big[(\mathbf{z}-\hat{\boldsymbol{\mu}}_{\text{non}})^\top \hat{\boldsymbol{\Sigma}}_{\text{non}}^{-1}(\mathbf{z}-\hat{\boldsymbol{\mu}}_{\text{non}}) - (\mathbf{z}-\hat{\boldsymbol{\mu}}_{\text{mem}})^\top \hat{\boldsymbol{\Sigma}}_{\text{mem}}^{-1}(\mathbf{z}-\hat{\boldsymbol{\mu}}_{\text{mem}}) + \log\frac{|\hat{\boldsymbol{\Sigma}}_{\text{non}}|}{|\hat{\boldsymbol{\Sigma}}_{\text{mem}}|}\Big]. \tag{4}$$

This full-covariance score is the quadratic discriminant analysis (QDA) instantiation of SLR, which we denote **SLR-QDA**. The covariance matrices are estimated with Ledoit–Wolf shrinkage. Its quality depends on the number of detector vectors available for members and for non-members, both equal to n. QDA also requires the inverse of a 19×19 covariance matrix, which is numerically unstable at small sample size. To avoid this instability, we propose **SLR-NB**. It assumes that the 19 signals are independent, so the covariance is diagonal and the score decomposes into per-dimension contributions. For the k-th dimension of the detector vector $\mathbf{z}$, let p_k be its value, and let $\hat{\mu}_{\text{mem},k}$ and $\hat{\sigma}_{\text{mem},k}$ be its mean and standard deviation under the blended member Gaussian. In practice, $\hat{\sigma}_{\text{mem},k}$ is computed as the square root of the k-th diagonal entry of $\hat{\boldsymbol{\Sigma}}_{\text{mem}}$. The quantities $\hat{\mu}_{\text{non},k}$ and $\hat{\sigma}_{\text{non},k}$ are defined analogously for the non-member Gaussian. Let

$$\ell_k(p_k) = -\frac{(p_k - \hat{\mu}_{\text{non},k})^2}{2\hat{\sigma}_{\text{non},k}^2} + \frac{(p_k - \hat{\mu}_{\text{mem},k})^2}{2\hat{\sigma}_{\text{mem},k}^2} + \frac{1}{2}\log\frac{\hat{\sigma}_{\text{non},k}^2}{\hat{\sigma}_{\text{mem},k}^2} \tag{5}$$

be the contribution of dimension k under a univariate Gaussian. Then $s_{\text{SLR-NB}}(\mathbf{z}) = \sum_{k=1}^{19}\ell_k(p_k)$. This score does not depend on the quality of a full covariance estimate and is therefore simpler and more stable, at the cost of assuming independence across signals. Building on NB, **SLR-NB-W** further assumes that dimensions differ in how much membership evidence they carry. It weights each dimension by its shadow-data AUC a_k with a hard gate at $\tau = 0.55$: $s_{\text{SLR-NB-W}}(\mathbf{z}) = \sum_{k=1}^{19} w_k\ell_k(p_k)$, where $w_k = a_k\,\mathbb{I}[a_k \geq \tau], \tau = 0.55$. QDA and NB both use the blended distribution parameters, so they already weight each alignment through the fingerprint mechanism. NB-W adds a second level of weighting, now over the 19 signal dimensions. Apart from NB-W, the simpler variants are the **SLR-top-K** methods. They restrict the score to the $K = 8$ dimensions $\mathcal{T}$ where the best matching shadow separates members best, so the score needs no dimension identity, only the rank profile: $s_{\text{SLR-NB(top-K)}}(\mathbf{z}) = \sum_{k\in\mathcal{T}}\ell_k(p_k)$.

5 EXPERIMENTS

Models and Datasets. We audit five model families: GPT-2 Medium, Llama-3.2-1B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Llama-3.1-8B-Instruct. GPT-2 Medium is the smallest and weakest model in the audit and is trained only on the IMDB sentiment dataset with the SieBERT sentiment reward model. The other four models are trained on two datasets that cover two reward landscapes: HH-RLHF with a continuous learned reward, and GSM8K with a binary verifiable reward, the RLVR setting. Both use Skywork-Reward-Llama-3.1-8B-v0.2 (Liu et al., 2024) as the reward model. Appendix D provides the detailed configurations.

Experiment Configuration and Evaluation. The experiment configuration is deferred to Appendix D. For a given pair, we define its membership by comparing its score with a threshold. Although the threshold is mentioned, our primary metric is the Area Under the Receiver Operating Characteristic (AUC), which does not require setting a specific threshold. We therefore evaluate each method with the threshold-free AUC metric.

5.1 COMPARISON WITH BASELINES

In this section, We report, per target model, the MIA AUCs of our proposed methods and baselines. we present only the results for Llama-3.2-1B in the main text and defer the remaining results to Appendix E. Baselines include all of the MIA methods discussed in Section 3. The results are shown in Table 2. Since shadow-based methods combine multiple signals, their per-signal AUC tables are collected in Appendix G. The fingerprint blend weights for SLR methods across all models, together with the identification correctness of the target's alignment method, are reported in Appendix E.2, where we measure whether the fingerprint calibration matches the correct alignment class. Among all baselines, S-DIBA is the only one that combines multiple signals with a shadow model. Table 2 carry only its original form, i.e., the five-signal classifier calibrated by a single GRPO shadow. We further evaluate its two variants. The first adds the fingerprint blend as the alignment calibration, so that S-DIBA selects its calibration across all five shadows instead of relying on a single one. The second extends its five signals to the full 19-dimensional detector vector, so that S-DIBA reads the same signal space as our method. The fingerprint-weighted and extended full-detector-space S-DIBA variants are compared against its original form in the ablation study in Section F.2.

As shown in Table 2, the SLR kernels lead most column of both blocks, and SLR-NB-W is the strongest score. Each baseline reads only part of the behavioral change caused by alignment. The classic static variants match the kernels on the SFT columns, because the 1B base must memorize the task and its supervised trace appears on the fixed response. However, they fail on the RL columns where the trace is reward-based. PREMIA reads the supervised columns, where a preference objective pushed the chosen and rejected responses apart, but it stays near chance under the RL objectives (GRPO and PPO), which never see the preference pair. S-DIBA stays near chance overall, except that its GRPO-shadow calibration performs relatively well on GRPO and PPO, because these are similar RL-based methods. Across methods, the supervised columns are the sharpest. DPO and NPO leave traces in many signals, including token-level and behavioral changes across several families, and only methods that combine all signal dimensions jointly, ENV and SLR, can read all of them, so they perform best.

The two RL columns are the harder cases for MIA, and performance splits by reward granularity. Under the continuous reward, both RL traces are modest but readable. While under the binary reward, GRPO performs poorly. The reason is that every signal that reads behavior sees the same thing on both the member and non-member sides, so only the weighted gate (SLR-NB-W), which re-selects the static dimensions, still reads it. This happens because the binary reward landscape is simpler than the continuous one, and the model learns it so well that it solves members and held-out prompts equally. In PPO, the update direction at each step is determined by the critic model's value estimate. Our critic model is trained by warm-up, and its reward prediction is noisy and does not generalize well across prompts. This leads to slow and directionless movement under noisy gradients, that is, policy drift. As a result, PPO learns less, making it the hardest task, and all MIA methods perform modestly. The weaker performance of PPO compared with other alignment methods is also clear in the ROC curves. Figure 2 shows the full operating curves of the seven detectors. The summary AUCs are reported in Table 2.

Table 2: MIA AUC, Llama-3.2-1B. Left block: continuous reward. Right block: binary reward (RLVR). Bold marks the best AUC per alignment method. Avg columns are means over the five fine-tuning methods. the top block lists the classic LM-MIA baselines. **self-gen** averages over G=20 responses generated by the target policy, and **static** is computed on the fixed chosen response that the sample carries. The middle block lists shadow-free methods, with baseline PREMIA above our proposed ENV. The bottom block lists shadow-based methods, with baseline S-DIBA above our proposed SLR methods.

	HH-RLHF (continuous reward)						GSM8K (binary reward)					
Method	SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg
Classic signals												
LOSS (self-gen)	0.543	0.573	0.705	0.523	0.550	0.579	0.537	0.585	0.546	0.611	0.518	0.559
Zlib (self-gen)	0.513	0.596	0.709	0.536	0.550	0.581	0.539	0.533	0.504	0.546	**0.641**	0.553
Min-K% (self-gen)	0.514	0.514	0.667	0.535	0.555	0.557	0.500	0.518	0.502	0.649	0.526	0.539
Entropy (self-gen)	0.544	0.580	0.711	0.535	0.541	0.582	0.606	0.590	0.534	0.610	0.544	0.577
LOSS (static)	0.972	0.664	0.697	0.533	0.504	0.674	0.951	0.598	0.650	0.515	0.501	0.643
Zlib (static)	**0.985**	0.671	0.728	0.559	0.542	0.697	0.945	0.560	0.622	0.531	0.519	0.635
Min-K% (static)	0.970	0.634	0.648	0.517	0.519	0.658	0.506	0.512	0.539	0.611	0.521	0.538
Entropy (static)	0.939	0.647	0.666	0.520	0.533	0.661	0.654	0.521	0.548	0.565	0.617	0.581
Neighbor (static)	0.953	0.677	0.712	0.510	0.521	0.675	0.911	0.601	0.579	0.513	0.532	0.627
Shadow-free												
PREMIA	0.550	0.593	0.667	0.523	0.513	0.569	0.940	0.716	0.762	0.572	0.539	0.706
ENV	0.939	0.647	0.674	0.679	0.647	0.717	0.925	0.682	0.780	0.639	0.604	0.726
ENV-simple	0.955	0.606	0.686	0.579	0.548	0.675	0.875	0.638	0.884	0.684	0.529	0.722
Shadow-based												
S-DIBA	0.594	0.511	0.521	0.586	0.601	0.563	0.523	0.525	0.579	0.594	0.510	0.546
SLR-NB	0.970	**0.869**	0.846	0.721	**0.671**	0.815	0.923	0.924	0.972	0.594	0.608	0.804
SLR-QDA	0.973	0.861	0.821	0.515	0.520	0.738	0.933	0.922	0.948	0.575	0.624	0.800
SLR-NB-W	0.974	0.865	**0.848**	**0.750**	0.646	**0.817**	**0.978**	**0.925**	**0.983**	**0.783**	0.582	**0.850**
SLR-NB (top-K)	0.841	0.645	0.720	0.706	0.520	0.686	0.964	0.667	0.697	0.596	0.622	0.709
SLR-QDA (top-K)	0.763	0.635	0.727	0.676	0.631	0.686	0.933	0.649	0.734	0.609	0.626	0.710

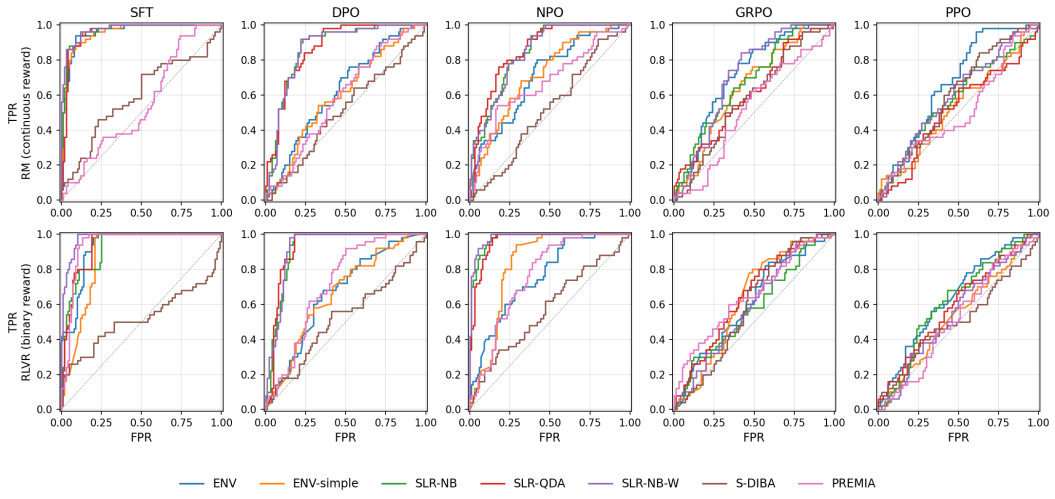

Figure 2: ROC curves for the Llama-3.2-1B. Top row: continuous reward. Bottom row: binary reward (RLVR).

5.2 Multi-Stage Alignment

The standard results above are single-stage, but real models are rarely aligned by a single method. Real pipelines train in sequence, such as the supervised fine-tuning followed by preference training of Zephyr (Tunstall et al., 2023), the supervised warm-up followed by RL of InstructGPT (Ouyang et al., 2022), and the three-stage pipeline of Tülu 3 (Lambert et al., 2024). This section examines such pipelines directly. We test four chained curricula on the Llama-1B continuous-reward side: SFT warm-up alone, SFT then DPO, SFT then GRPO, and SFT then GRPO then NPO. The shadow calibration is the same as in the previous single-stage tables. It contains one shadow per alignment,

each trained in one stage from the base model on 1000 shadow data that no target model saw in training. Each row of Table 3 reads the final-stage policy, so the target model is the π_θ obtained at the end of the chain. The reference column 1-stage SLR-NB shows a target model trained directly from the base model with only the final method, that is, a single-stage model with no earlier stages. Comparing the multi-stage row with this reference shows whether the earlier stages helped the final method's trace or diluted it. SLR calibrates its detectors on five shadows, each trained in a single stage. These are the same single-stage shadow models used in the previous sections. S-DIBA follows the previous sections and uses the GRPO shadow for calibration by default.

Table 3 reports the results on multi-stage fine-tuning. The token-logit family carries the trace through every curriculum, and chaining never erases it. This shows that the effect of SFT warm-up persists through the whole chain. PREMIA and S-DIBA are close to random. This means that in multi-stage alignment training, a signal that reads only part of the behavior and a method that calibrates on only one alignment cannot capture the multi-stage trace change. The shadow-free envelope and shadow-based SLR both perform much better than the baselines, and SLR is the strongest. This shows that SLR's fingerprint matching selects the most similar alignment among the five for calibration. SLR-NB performs close to its one-stage reference, which shows that the earlier stages neither helped nor diluted the final method's trace.

Across the four curricula, SFT followed by GRPO or DPO degrades all MIA methods, and GRPO gives the worst degradation. Adding NPO after GRPO then recovers some performance. DPO and GRPO mainly change the model's preference behavior. Once preference training is mixed in, the token-level trace from the SFT warm-up becomes weaker. The previous finding in Section 5.1 also show that RL-based alignment is a harder setting for MIA, so a final stage of GRPO degrades MIA more than DPO. When NPO is added after SFT then GRPO, the final stage becomes NPO and writes a new NPO trace, which raises MIA performance again. This suggests that the final-stage alignment has a larger effect on multi-stage MIA than the earlier stages, and that the final stage can overwrite the earlier trace with a new one.

Table 3: Multi-stage curricula, Llama-1B continuous-reward setting. MIAs read the final-stage policy of the multi-stage alignment chain. In the "1-stage SLR-NB" setting, the target model is trained by the final-stage alignment method alone, with no earlier stages.

Curriculum	ENV	SLR-NB	SLR-QDA	SLR-NB-W	PREMIA	S-DIBA	1-stage SLR-NB
SFT (init point)	0.939	0.970	0.973	0.974	0.550	0.570	0.970
SFT→DPO	0.806	0.940	0.878	0.947	0.550	0.563	0.869
SFT→GRPO	0.732	0.741	0.738	0.741	0.551	0.535	0.721
SFT→GRPO→NPO	0.795	0.804	0.805	0.809	0.592	0.503	0.846

6 Conclusion

We studied membership inference for fine-tuned language models. Different alignment methods leave traces on different axes. Prior attacks read only one category of change, so they fail when the target alignment leaves its trace elsewhere or when the pipeline chains several stages. We addressed this problem with a 19-dimensional detector vector that covers the prompt, the chosen and rejected likelihoods, the token logits, and the reward profile. We then proposed two scoring families: the shadow-free Envelope (ENV) and the shadow-based shadow-calibrated likelihood ratio (SLR). Experiments across five fine-tuning methods, two reward granularities, five model families, and multi-stage curricula show that the framework reads the behavioral shift left by alignment. The SLR methods lead most settings, SLR-NB-W is the strongest score, and the fingerprint match identifies the correct calibration in nearly every case. The shadow-free envelope remains competitive when no aligned shadow is available. Single-family baselines are sharp only on the axis they read and stay near chance elsewhere. Multi-stage curricula further show that chaining moves the trace rather than erasing it, and the framework follows the move while single-signal attacks stay pinned to their axis. The framework has limitations. SLR requires training one shadow per alignment method, which can be costly. Extending the detector to other alignment objectives, reducing shadow cost, and studying defenses against multi-axis membership inference are promising directions for future work.

7 AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript to improve readability and to assist with coding, including code debugging and implementation support. We have not used generative AI tools for tasks with required disclosure, and the required disclosure tasks are not applicable to this work. All AI-assisted text and code were reviewed, verified, and tested by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

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.

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)*, pp. 2633–2650, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pp. 1897–1914. IEEE, 2022.

Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pp. 1964–1974. PMLR, 2021.

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

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.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In *29th USENIX security symposium (USENIX Security 20)*, pp. 1605–1622, 2020.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*, 2024.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv preprint arXiv:2509.16679*, 2025a.

Yule Liu, Heyi Zhang, Jinyi Zheng, Zhen Sun, Zifan Peng, Jiaheng Wei, Tianshuo Cong, Yilong Yang, and Xinlei He. Auditing data membership in reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2511.14045*, 2025b.

Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A Gunter, and Kai Chen. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889*, 2018.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pp. 142–150, 2011.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, 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*, pp. 11330–11343, 2023.

Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pp. 739–753. IEEE, 2019.

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

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

Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

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

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.

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 *International Conference on Learning Representations*, volume 2024, pp. 51826–51843, 2024.

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)*, pp. 3–18. IEEE, 2017.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*, 2023.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pp. 268–282. IEEE, 2018.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*, 2024.

A NOTATION

Table 4: Notation used throughout the paper. Symbols are listed in alphabetical order.

Symbol	Meaning
a	Reward function used to score rollouts.
a_k	Shadow-data AUC of detector dimension k, used in SLR-NB-W.
$a_{\max}$	Maximum reward among rollouts, an S-DIBA reward axis.
a_{mean}	Mean reward among rollouts, an S-DIBA reward axis.
a_{std}	Standard deviation of rewards, an S-DIBA reward axis.
AUC	Area under the receiver operating characteristic curve.
β	Softmax temperature in the fingerprint blend.
c	Chosen response in a preference triple.
$\mathbf{d}$	Raw 19-dimensional detector vector of a sample.
D_i	The i-th detector signal, $i = 1, \ldots, 19$.
$\mathbf{f}_t$	Fingerprint of the target model.
$\mathbf{f}^{(m)}$	Fingerprint of the shadow model for alignment method m.
G	Number of rollouts generated per prompt.
h_i	Per-token predictive entropy at position i.
K	Number of top dimensions retained by the top-K variants.
ℓ_c, ℓ_r	Log-likelihoods of the chosen and rejected responses.
$\tilde{\ell}_c, \tilde{\ell}_r$	Calibrated reference-model log-likelihoods of c and r.
$\ell_k(p_k)$	Contribution of dimension k under a univariate Gaussian in SLR-NB.
n	Number of tokens in a response or number of samples per class.
p_i	Per-token log-probability at position i.
$\mathcal{P}$	Reference population of held-out prompts.
$\text{PPL}_\theta(\mathbf{x})$	Prompt perplexity under policy π_θ.
q_i	Per-token log-ratio between the policy and the reference model.
r	Rejected response in a preference triple.
$\mathbf{r}$	Reward vector of the G rollouts.
s	Membership score produced by an attack.
$\mathcal{T}$	Index set of the top-K detector dimensions.
w_m	Blend weight assigned to shadow model m.
$\mathbf{x}$	Prompt.
y	Response, $y = (y_1, \ldots, y_n)$.
$\mathbf{z}$	Standardized and percentile-transformed detector vector.
π_θ	Target policy (fine-tuned model).
π_{ref}	Frozen reference or base model.
$\pi^{(m)}$	Shadow model trained with alignment method m.
$\mu_{\text{pop}}, \sigma_{\text{pop}}$	Per-dimension mean and standard deviation over the reference population.
$\mu_*^{(m)}, \Sigma_*^{(m)}$	Gaussian parameters for shadow m and class $* \in \{\text{mem}, \text{non}\}$.
$\hat{\mu}_*, \hat{\Sigma}_*$	Blended Gaussian parameters for class $*$.
$\sigma_{\text{mem},k}, \sigma_{\text{non},k}$	Standard deviations of dimension k under the member and non-member Gaussians.
τ	Hard-gate threshold in SLR-NB-W.
θ	Parameters of the target policy.
$\mathcal{V}$	Vocabulary.

B DETAILED CLASSICAL MEMBERSHIP SIGNALS

B.1 CLASSIC LM MEMBERSHIP SIGNALS

All classical membership signals for language models are computed over response tokens and oriented so that a higher score indicates a higher likelihood of membership. Let $y = (y_1, \ldots, y_n)$ be the response, with per-token log-probabilities $p_i = \log \pi_\theta(y_i | y_{<i}, \mathbf{x})$ and per-token predictive

entropies $h_i = -\sum_{v\in\mathcal{V}} \pi_\theta(v|y_{<i}, \mathbf{x}) \log \pi_\theta(v|y_{<i}, \mathbf{x})$. The four token-level signals are

$$s_{\mathrm{LOSS}} = \frac{1}{n}\sum_{i=1}^{n} p_i, \quad s_{\mathrm{Zlib}} = -\frac{\sum_{i=1}^{n} p_i}{\mathrm{zlib_bits}(y)}, \quad s_{\mathrm{MinK\%}} = \frac{1}{\lceil n/5 \rceil}\sum_{i\in\mathcal{T}} p_i, \quad s_{\mathrm{Entropy}} = -\frac{1}{n}\sum_{i=1}^{n} h_i, \tag{6}$$

where $\mathcal{T}$ denotes the set of the $\lceil n/5 \rceil$ tokens with the highest p_i, i.e., the lowest-loss 20% of tokens. The term $\mathrm{zlib_bits}(y)$ is the number of bits obtained by compressing the response string y with the zlib lossless compression algorithm. It is computed by encoding y as UTF-8 bytes, compressing these bytes with zlib, and multiplying the resulting byte length by 8. This quantity depends only on the text itself, not on the target model.

LOSS (Yeom et al., 2018) is the mean per-token log-probability of the response, measuring how well the model predicts its own response and thus the raw surface fit. Zlib (Carlini et al., 2021) normalizes the response log-likelihood by $\mathrm{zlib_bits}(y)$. A response receives a high score when the target assigns it high probability while zlib achieves little compression on it. Min-K% (Shi et al., 2024) is the mean log-probability of the lowest-loss 20% of tokens, providing evidence that even the model's least-confident tokens are fitted. Entropy is the negated mean predictive entropy, so that low uncertainty on the continuation itself serves as a memorization print.

Neighbor (Mattern et al., 2023) first embeds each prompt with RoBERTa-base (mean-pooled and L_2-normalized), retrieves the eight nearest prompts in the evaluation pool by cosine similarity, and computes

$$s_{\mathrm{Neighbor}}(\mathbf{x}) = s_{\mathrm{LOSS}}(\mathbf{x}) - \frac{1}{8}\sum_{\mathbf{x}'\in \mathrm{NN}_8(\mathbf{x})} s_{\mathrm{LOSS}}(\mathbf{x}'), \tag{7}$$

where $\mathrm{NN}_8(\mathbf{x})$ is the set of eight nearest prompts to $\mathbf{x}$ and $s_{\mathrm{LOSS}}(\mathbf{x})$ denotes the LOSS of the reference response associated with prompt $\mathbf{x}$. The subtraction removes the effect of prompt difficulty. An easy prompt raises the LOSS of every sample alike, whether it is a member or not. After subtracting the average LOSS of its neighbors, the remaining difference better reflects membership.

All classical LM membership signals are applied in two modes: **self-gen**, which averages over $G{=}20$ responses generated by the target policy, and **static**, which is computed on the fixed chosen response that the sample carries. Neighbor is evaluated in static mode only. All five signals were designed to detect stored text, as each quantifies how well the target predicts some text. Alignment training shifts behavior instead of storing text, so these classical signals do not perform well in alignment training.

B.2 Membership Signals for Alignment Training

PREMIA (Feng et al., 2024) and S-DIBA (Liu et al., 2025b) are membership inference attacks designed for the alignment training setting.

PREMIA does not require shadow models and needs only the policy and the frozen reference model. It also requires no generation. Its signal is the *preference displacement*, the absolute movement of the policy's likelihoods on the chosen and rejected responses away from the reference:

$$s_{\mathrm{PREMIA}} = \max\Big(\big|\log \pi_\theta(c|\mathbf{x}) - \log \pi_{\mathrm{ref}}(c|\mathbf{x})\big|,\ \big|\log \pi_\theta(r|\mathbf{x}) - \log \pi_{\mathrm{ref}}(r|\mathbf{x})\big|\Big). \tag{8}$$

It fires only where a preference objective pushed the pair apart on the training sample itself. It is therefore blind to objectives that never see the (c, r) pair, namely the RL objectives.

S-DIBA formulates MIA in a five-dimensional feature space. For each sample, G responses are generated and scored, yielding a reward vector $\mathbf{r}$ and the best generation $\hat{g} = \arg\max r_j$. The five features fall into two families. The reward axes are statistics of the target's own rollouts under the training reward, capturing the behavioral quality of the policy's output: $a_{\max} = \max(\mathbf{r})$, $a_{\mathrm{mean}} = \bar{r}$, and $a_{\mathrm{std}} = D_{10}$. The likelihood axes read the best response: $b_{\mathrm{KL}} = \log \pi_\theta(\hat{g}|\mathbf{x}) - \log \pi_{\mathrm{ref}}(\hat{g}|\mathbf{x})$ is the likelihood ratio displacement of the best rollout, and $b_{\mathrm{token}} = \frac{1}{|\hat{g}|}\sum_t \log \pi_\theta(\hat{g}_t|\mathbf{x}, \hat{g}_{<t})$ is its surface likelihood. A shadow model provides labeled five-dimensional vectors. A logistic regression (L_2, $C{=}1.0$) and a random forest (100 trees, depth 5) are trained and ensembled by soft voting with equal weights.

C Our proposed 19 Detector Signals: Details and Justification

The detector vector is built by enumeration rather than search. An alignment trace can appear on four axes: the prompt, the chosen and rejected likelihoods, the token logits, and the reward profile. Alignment training leaves a limited set of trace shapes on these axes. A preference objective, for example, leaves a one-sided edit on the rejected side or a selective rewrite across tokens, while a reward-based objective leaves a common-mode lift in the reward profile or an overfit peak in the best rollout. We design the signals to cover these shapes. Each axis is represented by a small family of statistics. Each family starts from a published signal that already targets this axis and is then completed by its natural relatives, including moments, percentiles, extrema, and counterfactuals, so that every known trace shape is captured by at least one statistic. The vector is also grounded in empirical evidence: the per-signal tables in Appendix G and the ablations of §F show what each family contributes.

The 19 signals are numbered $D_1, \ldots, D_{19}$ by their position in the detector vector and are organized into four groups. For a sample $(\mathbf{x}, c, r)$, with prompt $\mathbf{x}$, chosen response c, and rejected response r, let π_θ be the policy (the fine-tuned model) and π_{ref} the frozen base model. Reward signals are computed from $G = 20$ rollouts per sample, scored by the reward function the model was trained with, that is, a continuous RM score on the RM side and a binary verifier label on the RLVR side. Some objectives use only part of the triple. GRPO and PPO, for example, never train on the two responses. The detector still requires the complete triple and evaluates the full vector on it. A component the objective never trained on behaves like a constant: no training step moved it, so the signals built on it cannot separate members from non-members.

Prompt-level. Every alignment objective conditions on the prompt and backpropagates through its tokens. A training prompt is therefore one place where all five methods leave a direct imprint on the input side. D_1 reads that imprint as the raw surface fit. D_2 re-expresses the same fit as a rank within the held-out population, which needs no scale calibration:

$$D_1 = \frac{1}{|\mathbf{x}|}\sum_{t=1}^{|\mathbf{x}|} \log \pi_\theta(x_t \mid \mathbf{x}_{<t}), \qquad D_2 = \frac{1}{|\mathcal{P}|}\sum_{\mathbf{x}' \in \mathcal{P}} \mathbb{I}\big[\text{PPL}_\theta(\mathbf{x}) < \text{PPL}_\theta(\mathbf{x}')\big], \tag{9}$$

where $\mathcal{P}$ is the 100-prompt reference population and $\text{PPL}_\theta(\mathbf{x}) = \exp(-D_1)$. Neither signal is taken from PREMIA or DIBA. Both are standard prompt-perplexity statistics.

Likelihood-ratio. Preference objectives such as DPO and NPO train on chosen and rejected pairs. They push the pair's likelihoods apart relative to the frozen reference. The displacement of the preference pair is therefore where their trace concentrates. Let $\ell_c = \log \pi_\theta(c|\mathbf{x})$, $\ell_r = \log \pi_\theta(r|\mathbf{x})$, and let $\tilde{\ell}_c, \tilde{\ell}_r$ be the corresponding reference-model log-probabilities under a shadow-calibrated affine map $\tilde{\ell} = a \log \pi_{\text{ref}}(\cdot) + b$:

$$D_3 = \big|(\ell_c - \tilde{\ell}_c) - (\ell_r - \tilde{\ell}_r)\big|, \qquad D_4 = |\ell_r - \tilde{\ell}_r|. \tag{10}$$

D_3 folds the two-sided displacement into one non-negative statistic. D_4 isolates the rejected side, where NPO's one-sided update prints most strongly. D_3 and D_4 are exactly PREMIA's two preference-displacement components, whose score takes their maximum. DIBA's b_{KL} is a related likelihood ratio computed on the best generated completion instead of the preference pair.

Token-level. Sequence-level objectives aggregate their loss over the tokens of the chosen response. A memorized training response is therefore fitted token by token. Let $\mathbf{t} = (t_1, \ldots, t_n)$ be the tokenized chosen response appended to the prompt, with per-token log-probability $p_i = \log \pi_\theta(t_i|\mathbf{x}, \mathbf{t}_{<i})$ and per-token log-ratio $q_i = p_i - \log \pi_{\text{ref}}(t_i|\mathbf{x}, \mathbf{t}_{<i})$:

$$\begin{aligned} &D_5 = \frac{1}{n}\sum_i p_i, \qquad D_6 = \text{Percentile}_{10}\big(\{p_i\}\big), \qquad D_7 = \frac{1}{n}\sum_i q_i, \\ &D_8 = \sqrt{\tfrac{1}{n}\textstyle\sum_i (q_i - D_7)^2}, \quad D_9 = \frac{\frac{1}{n}\sum_i (q_i - D_7)^3}{(\frac{1}{n}\sum_i (q_i - D_7)^2)^{3/2}}. \end{aligned} \tag{11}$$

D_5 reads the mean fit. D_6 reads the 10th percentile, which is high only when even the least-confident tokens are fitted. This is the Min-K-style memorization signature. D_7 through D_9 are the first three

moments of the per-token log-ratio against the reference. D_7 captures an overall rewrite, D_8 a selective rewrite, and D_9 a one-sided edit. DIBA's b_{token} is the mean per-token log-probability of the best generated completion. D_5 is the same statistic read off the chosen response, extended here with the LR moments D_7 through D_9.

Reward-based. Let $\mathbf{r} = (r_1, \ldots, r_G)$ be the rewards of the G rollouts. Four signals measure the *spread* of $\mathbf{r}$, which reflects how consistent the rollouts are:

$$D_{10} = \sqrt{\tfrac{1}{G}\textstyle\sum_j (r_j - \bar{r})^2}, \quad D_{11} = \frac{\max(\mathbf{r}) - \bar{r}}{D_{10} + 10^{-8}}, \\ D_{12} = \max(\mathbf{r}) - \min(\mathbf{r}), \quad D_{13} = \max(\mathbf{r}) - \bar{r}. \tag{12}$$

Spread statistics are shift-invariant. A common-mode reward lift, where the model scores every rollout of a training prompt higher, leaves them unchanged. Yet the lift is exactly what reward-based alignment produces for its training prompts. The remaining six signals therefore measure the *profile* of $\mathbf{r}$ directly:

$$D_{14} = \bar{r}, \quad D_{15} = \max(\mathbf{r}), \quad D_{16} = \min(\mathbf{r}), \quad D_{17} = \frac{1}{k} \sum_{j:\ r_{(j)} \text{ top-}k} r_{(j)}, \tag{13}$$

$$D_{18} = \text{clip}\Big(\frac{\bar{r}}{D_{10} + 10^{-8}},\ 0,\ 10^3 \Big), \quad D_{19} = \bar{r}_{\pi_\theta}(\mathbf{x}) - \bar{r}_{\pi_{\text{ref}}}(\mathbf{x}), \tag{14}$$

where $k = \lfloor G/4 \rfloor$ is the number of top-scoring rollouts averaged in D_{17}, that is, the top quarter of the G rollouts. The profile reads the behavioral lift itself. D_{14} is the mean lift. D_{15} and D_{16} are the best and worst rollout, which mark the overfit peak and the failure floor. D_{17} is the mean of the top-k rollouts, a stable high-quality core that is immune to outlier rollouts. D_{18} is the lift normalized by the spread and clipped to $[0, 10^3]$. D_{19} is the memorization lift itself: the trained model's mean reward minus the frozen base model's mean reward on the same prompt. It is the cleanest counterfactual of what alignment did to this sample. D_{14}, D_{15}, and D_{10} are exactly DIBA's a_{mean}, $a_{\max}$, and a_{std}. The remaining profile statistics, D_{16} through D_{19} and D_{11} through D_{13}, are new.

D Training Configurations

Models and Datasets. We audit five alignment methods, SFT, DPO, NPO, GRPO, and PPO, across five model families: GPT-2, Llama-3.2-1B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B-Instruct, and Llama-3.1-8B-Instruct (LoRA). GPT-2 is the smallest and weakest model in the audit. In this work, It is trained only on IMDB sentiment dataset with a continuous sentiment reward model, SieBERT. IMDB sentiment dataset is constructed from the original IMDB sentiment corpus by splitting each review into a prefix and a continuation, using the prefix as the prompt, generating two continuations with a language model, and labeling them with a sentiment classifier to obtain the chosen and rejected responses. The other four models are trained on two datasets that correspond to two different reward landscapes. (1) HH-RLHF provides a continuous reward: a learned reward model assigns each response a continuous score; (2) GSM8K provides a binary reward: each response is the reasoning process for a problem, and its correctness can be checked against ground truth, giving a $0/1$ reward. This is the Reinforcement Learning with Verifiable Rewards (RLVR) setting. Together, the two datasets cover both a continuous learned reward and a binary verifiable reward. Among the alignment methods, SFT, DPO, and NPO are supervised and have access to the chosen and rejected labels during training.

Data split. The target model is trained on a 2000-prompt *member* slice drawn from the training set. The five shadow models are trained on five disjoint 2000-prompt *member* slices, also drawn from the training set. Each shadow is therefore a statistical twin of the target: it uses an identical training configuration and the same number of samples, but its member data are disjoint from the target's member data and from those of the other shadows. *Non-member* prompts for the target and for all shadows are drawn from the test set, which is excluded from all training. The reference population is also drawn from the test set and is disjoint from other non-member data.

Training hyperparameters. Table 5 lists the continuous-reward side configurations. Table 6 lists the RLVR-side configurations. Table 7 lists the continuous-reward side configurations of the three extension models.

Table 5: Continuous-reward training configurations (Llama-3.2-1B). Common: bf16, AdamW with 100-step warmup, effective batch 16, max sequence 768 (prompt 384).

Method	Objective and hyperparameters
SFT	cross-entropy next-token loss; lr 10^{-4}; 2,500 steps
DPO	DPO loss, β_{DPO}=0.3; lr 5×10^{-6}; 2,500 steps
NPO	NPO softplus loss + chosen-pull anchor α=0.2; β_{DPO}=0.3; lr 5×10^{-6}; 2,500 steps
GRPO	group z-score advantage, G=16 (std floor 0.1, clip ± 1); DAPO dynamic filter (group std $< 0.1 \Rightarrow$ zero advantage); k3 KL estimator; β_{KL}=0.05; entropy coef 0.1; min-new-tokens 16; rollout length 64; lr 10^{-5}; 3,000 steps
PPO	faithful critic PPO: value model = base-LM + LoRA (r=16, α=32) + linear head; GAE(λ=0.95, γ=1.0) with the bandit reward at the last response token and terminal V=0; PPO2 advantage whitening per cycle; 4 epochs per window; per-token clip ϵ=0.2; value loss $0.5 \cdot \mathrm{MSE} + V$-clip 0.2; k3 KL; lr 10^{-5} (policy and critic, single optimizer); 3,000 steps

Table 6: Binary-reward (RLVR) training configurations. Common: rollout temperature T=0.3, 64-token generation budget, min-new-tokens 2, β_{KL}=0.3, k3 KL, entropy coef 0.05, DAPO dynamic filter, reward = 1 iff the extracted answer verifies; DPO/NPO additionally use β_{DPO}=0.3 and anchor α=0.2.

Method	Llama-3.2-1B	Qwen-1.5B / Qwen-3B	Llama-8B (LoRA)
SFT	CE on (problem, solution); lr 10^{-4}	lr 10^{-4}	lr 2×10^{-5}
DPO / NPO	lr 5×10^{-6}	lr 5×10^{-6}	lr 5×10^{-6}
GRPO	G=8; lr 10^{-6}	G=16; lr 10^{-6}	G=8; lr 8×10^{-6}
PPO	G=8; lr 10^{-6}; critic LoRA r=16	G=16; lr 10^{-6}	G=8; lr 8×10^{-6}
Batch / steps	4×accum 2; 1,500 steps (48 exposures)	8×accum 1; 2,000 steps (64 exposures)	1×accum 4; 800 steps (13 exposures)

E Additional Comparison Results with Baselines on More Models

E.1 Additional Comparison with Baselines

GPT-2 Medium (RM side). GPT-2 Medium is trained on the IMDB sentiment dataset with the SieBERT sentiment reward model. It has no RLVR block. Its detector uses all 19 signals, including the reward-profile group.

On GPT-2, the SLR kernels lead on the supervised columns and on DPO. SFT creates a strong supervised trace. Every kernel reads it near ceiling. SLR-QDA is best because the completion surface moves as a whole. DPO spreads its trace across many dimensions. Only the weighted kernel reads it clearly. NPO's trace is concentrated enough for the covariance-aware kernel to use it. The two RL columns are different. GRPO is read best by S-DIBA and PREMIA. Their GRPO shadow and preference statistic match the target's objective. The calibrated kernels stay near chance. A small model under a sentiment reward moves too little per prompt for the calibrated dimensions to separate. PPO is almost invisible to every detector. Its critic warm-up moves under a noisy reward prediction that carries no information about the training prompts. Only the uncalibrated envelope stays a little above chance. The reward lift D22 is the only single signal clearly above chance on every column, including the RL columns that the combined detectors miss. The results hold at 300 non-members: the weighted kernel keeps 0.951, 0.642, and 0.558 on SFT, DPO, and NPO, and stays near chance on the two RL columns. This matches the main panel.

Qwen-1.5B.

Table 7: Continuous-reward configurations of the three extension models (same configuration family as Llama-3.2-1B, with learning rates scaled per model family).

	Qwen-1.5B	Qwen-3B	Llama-8B (LoRA)
Steps	2,400	1,800	800
lr (SFT)	8×10^{-5}	6×10^{-5}	2×10^{-5}
lr (DPO / NPO)	5×10^{-6}	5×10^{-6}	5×10^{-6}
lr (GRPO / PPO)	10^{-5}	10^{-5}	8×10^{-6}
Rollouts per step	64	32	64
Parameterization	full fine-tune	full fine-tune	LoRA on frozen base

Table 8: MIA AUC, GPT-2 Medium (deliberately overfit). Bold marks the best AUC per alignment method (all ties bolded). The Avg column is the mean over the five methods.

Method	SFT	DPO	NPO	GRPO	PPO	Avg
Shadow-free						
PREMIA	0.741	0.503	0.535	0.633	0.507	0.584
ENV	0.870	0.529	0.576	0.517	0.537	0.606
ENV-simple	0.870	0.532	0.515	0.542	**0.576**	0.607
Shadow-based						
S-DIBA	0.559	0.573	0.561	**0.634**	0.533	0.572
SLR-NB	0.917	0.679	0.567	0.508	0.542	0.643
SLR-QDA	**0.944**	0.574	**0.650**	0.548	0.535	0.650
SLR-NB-W	0.932	**0.704**	0.597	0.532	0.541	**0.661**
SLR-NB (top-K)	0.850	0.577	0.514	0.540	0.513	0.599
SLR-QDA (top-K)	0.897	0.575	0.514	0.563	0.511	0.612

On Qwen-1.5B, the SLR kernels lead the supervised columns by a large margin. They read the RL columns weakly. SLR-NB-W is the strongest score. The baselines only win on specific columns. The classic static variants read the SFT columns, where the memorized surface appears. The self-generated variants read the two RL columns whose rollouts leak the policy shift while their fixed completions do not: continuous-reward PPO and binary GRPO. PREMIA reads the binary supervised columns. It wins the DPO column because the preference displacement is written directly on the pair. S-DIBA stays near chance except on the two PPO columns and the binary NPO column. Its features read the rollout statistics directly. Across methods, SFT is the sharpest column in both blocks. DPO and NPO are in the middle. The RL columns are the weakest. At this model size, the instruction-tuned base already solves the synthetic task. Binary GRPO only reshuffles rollout dispersion and leaves no per-prompt trace. The weighted gate misses it.

Qwen-3B.

On Qwen-3B, the SLR kernels lead the supervised columns in both blocks. The binary RL columns are read only by the full-dimension readers: the unweighted kernel and the S-DIBA classifier. Their traces lie in dimensions that the shadow sees as uninformative, so the gate discards them. On the continuous-reward side, GRPO leaves almost no trace. The base can solve the task without memorizing it, and the RL objective uses that ability. PPO is weak and miscalibrated. Its fingerprint resolves to the NPO shadow. This suppresses the kernels and leaves the shadow-free detectors ahead on that column. Among the baselines, the classic static variants read SFT alone. PREMIA reads the binary supervised columns. S-DIBA's best column is binary GRPO, where its shadow shares the objective.

Llama-8B.

The continuous-reward side is the hardest block in the paper. At 8B, the base already solves the task. The policy shift is too small to print on a fixed completion. Only SFT and NPO leave traces that the kernels read. On the binary side, the same model is read clearly on SFT and NPO. DPO stays weak. GRPO is caught by the static-surface classic signals and the shadow-free detectors, not by the kernels. The baselines are strongest here compared with the framework. The classic signals are the only readers above the noise band on the continuous-reward PPO column. The surface print on the fixed completion is all the trace leaves. PREMIA reads the binary SFT column. S-DIBA stays near

Table 9: MIA AUC, Qwen-1.5B. Left block: RM (continuous reward). Right block: RLVR (binary reward). Bold marks the best AUC per alignment method (all ties bolded). Avg columns are means over the five methods of their block (— where a method is missing). Classic signals (top block) report the better direction.

	RM (continuous reward)						RLVR (binary reward)					
Method	SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg
Classic signals												
LOSS (self-gen)	0.551	0.626	0.554	0.594	0.676	0.600	0.513	0.611	0.593	**0.681**	0.535	0.587
Zlib (self-gen)	0.527	0.614	0.551	0.610	0.676	0.596	0.501	0.598	0.590	0.680	0.545	0.583
Min-K% (self-gen)	0.540	0.595	0.554	0.579	**0.685**	0.591	0.519	0.643	0.539	0.623	0.565	0.578
Entropy (self-gen)	0.513	0.638	0.572	0.568	0.681	0.594	0.520	0.597	0.502	0.581	0.506	0.541
LOSS (static)	0.936	0.606	0.505	0.507	0.541	0.619	0.817	0.517	0.561	0.667	0.601	0.633
Zlib (static)	0.966	0.611	0.526	0.527	0.584	0.643	0.792	0.503	0.555	0.676	0.594	0.624
Min-K% (static)	0.899	0.603	0.572	0.552	0.603	0.646	0.520	0.517	0.542	0.512	0.539	0.526
Entropy (static)	0.906	0.601	0.567	0.582	0.593	0.650	0.641	0.558	0.542	0.569	0.570	0.576
Neighbor (static)	0.932	0.589	0.514	0.501	0.548	0.617	0.855	0.517	0.582	0.535	0.516	0.601
Shadow-free												
PREMIA	0.703	0.563	0.583	0.501	0.571	0.584	0.934	**0.869**	0.719	0.597	0.599	0.744
ENV	0.968	0.639	0.636	0.588	0.629	0.692	0.799	0.663	0.595	0.546	0.519	0.624
ENV-simple	0.984	0.611	0.565	0.505	0.592	0.651	0.768	0.771	0.612	0.515	0.517	0.637
Shadow-based												
S-DIBA	0.557	0.556	0.604	0.548	0.636	0.580	0.536	0.545	0.624	0.519	**0.634**	0.572
SLR-NB	0.992	**0.838**	0.814	**0.663**	0.604	0.782	0.933	0.790	0.922	0.551	0.622	0.764
SLR-QDA	0.992	0.814	0.813	0.515	0.588	0.744	0.952	0.726	0.924	0.506	0.541	0.730
SLR-NB-W	**0.993**	0.836	**0.819**	0.515	0.510	0.735	**0.977**	0.790	**0.940**	0.514	0.633	0.771
SLR-NB (top-K)	0.807	0.718	0.701	0.526	0.579	0.666	0.841	0.828	0.914	0.531	0.548	0.732
SLR-QDA (top-K)	0.752	0.684	0.689	0.535	0.547	0.641	0.838	0.830	0.913	0.519	0.516	0.723

chance throughout. This block also has the second identity failure: the continuous-reward GRPO blend resolves to the PPO shadow. This suppresses the kernels on that column.

Cross-panel findings. Three patterns hold across the eight learned panels.

Reward granularity does not change the main map. On both reward granularities, supervised methods are the sharp columns. RL columns are the mixed ones. When the two blocks differ, the difference comes from the objective granularity, not from the detectors.

Model size decides which channels carry the trace. The smallest model must memorize the tasks to solve them. Its supervised traces print on the completion surface, and the static classic signals read them. Mid-size models sit at the capacity edge. RL reshuffles dispersion without writing per-prompt traces. The largest model solves the tasks without memorization. Its traces are behavioral, spread across weakly informative dimensions, and read only by the prior-calibrated kernels. At the weakest column, the completion-surface print lasts longer than the behavioral one, and the classic signals take over.

Method families behave as families. SFT is the sharpest column on every panel. NPO is sharp on the binary side and middle on the continuous side. DPO is clear on smaller models and weak on larger ones. The RL methods are mixed, each with a recorded failure signature. GRPO fails when the corrected objective leaves nothing to move, when the gate discards its dimensions, or when only the shadow-free envelope still reads it. PPO fails when the policy drifts without member-specific content, borrows a mismatched prior, or is calibrated by a shadow too weak to transfer. The multi-stage study (§5.2) shows the same family patterns survive chaining.

E.2 Blend Weights and Automatic Identification

Every SLR score above is calibrated by a blend of the five shadow policies, weighted by a softmax over fingerprint distances. The weight vector is the framework's automatic identification of the target's alignment method, made without any alignment method label. The weight on each shadow is the confidence the fingerprint match puts on that shadow being the target's own method. A weight of

Table 10: MIA AUC, Qwen-3B. Left block: RM (continuous reward). Right block: RLVR (binary reward). Bold marks the best AUC per alignment method (all ties bolded). Avg columns are means over the five methods of their block (— where a method is missing). Classic signals (top block) report the better direction.

	RM (continuous reward)						RLVR (binary reward)					
Method	SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg
Classic signals												
LOSS (self-gen)	0.557	0.544	0.621	0.554	0.592	0.574	0.521	0.630	0.643	0.612	0.562	0.594
Zlib (self-gen)	0.560	0.544	0.618	**0.558**	**0.607**	0.577	0.546	0.631	0.640	0.590	0.572	0.596
Min-K% (self-gen)	0.512	0.519	0.549	0.520	0.594	0.539	0.562	0.632	0.572	0.570	0.573	0.582
Entropy (self-gen)	0.502	0.534	0.547	0.509	0.597	0.538	0.545	0.610	0.564	0.603	0.565	0.577
LOSS (static)	0.974	0.505	0.504	0.554	0.534	0.614	0.925	0.588	0.592	0.620	0.508	0.647
Zlib (static)	0.978	0.540	0.541	0.507	0.511	0.615	0.921	0.579	0.587	0.558	0.509	0.631
Min-K% (static)	0.866	0.505	0.501	0.513	0.509	0.579	0.558	0.551	0.544	0.505	0.509	0.533
Entropy (static)	0.851	0.539	0.546	0.532	0.507	0.595	0.716	0.510	0.511	0.533	0.581	0.570
Neighbor (static)	0.951	0.504	0.472	0.464	0.457	0.570	0.856	0.570	0.507	0.534	0.522	0.598
Shadow-free												
PREMIA	0.734	0.521	0.626	0.546	0.594	0.604	0.934	0.927	0.736	0.526	0.542	0.733
ENV	0.976	0.589	**0.680**	0.519	0.579	0.669	0.839	0.810	0.603	0.587	0.576	0.683
ENV-simple	0.982	0.572	0.603	0.520	0.538	0.643	0.894	0.931	0.668	0.545	0.511	0.710
Shadow-based												
S-DIBA	0.575	0.561	0.519	0.551	0.503	0.542	0.601	0.616	0.568	**0.636**	0.528	0.590
SLR-NB	0.986	0.577	0.655	0.536	0.554	0.662	0.935	**0.962**	0.941	**0.636**	**0.583**	0.811
SLR-QDA	0.981	0.585	0.678	0.507	0.582	0.667	0.950	0.931	0.948	0.557	0.561	0.789
SLR-NB-W	**0.988**	**0.616**	0.664	0.501	0.555	0.665	**0.966**	0.962	**0.956**	0.563	0.529	0.795
SLR-NB (top-K)	0.859	0.569	0.638	0.527	0.535	0.626	0.954	0.909	0.864	0.522	0.517	0.753
SLR-QDA (top-K)	0.759	0.580	0.634	0.540	0.538	0.610	0.950	0.921	0.869	0.508	0.568	0.763

1.0 on one shadow means the target's fingerprint matches that shadow's exactly, and the calibration is pure. Spread weights mean the target's trace sits between methods, and the calibration borrows from several priors. The identification is correct when the largest weight lands on the shadow of the target's own alignment method. Table 12 reports the weight vector of every evaluated column, with the largest weight per row bolded.

The identification is correct on nearly every column. It fails twice, on Qwen-3B's continuous-reward PPO, which resolves to the NPO shadow, and on Llama-8B's continuous-reward GRPO, which resolves to the PPO shadow. In both columns, the per-model analysis shows that the SLR kernels are suppressed by a mismatched prior. The remaining columns identify their method correctly, either purely or with only minor borrowing. The largest borrowings are Qwen-1.5B's continuous-reward PPO, which takes a third of its weight from the NPO shadow, and its binary DPO, which takes two fifths. The framework therefore requires no method label to calibrate itself, and the fingerprint match alone selects the correct prior in nearly every setting.

F Ablations

F.1 MIA vs. training stages

Here we study how detector scores change during training. We take checkpoints from each RLVR training run of Llama-1B. For each checkpoint, we compute the full detector panel on the learned member subset and also compute the per-prompt verify accuracy. Then we measure how each detector's score changes with the verify accuracy. The two failure patterns we saw earlier, PPO moving behavior without member content and GRPO having a silent behavioral channel, appear here as differences between the two curves.

We use five single-stage RLVR runs of Llama-1B: SFT, DPO, NPO, PPO, and a corrected GRPO run. For SFT, DPO, NPO, and PPO, we use eight checkpoints (150, 450, 500, 750, 1000, 1050, 1350, and 1500 steps). For the corrected GRPO run, we use every 50 steps across its 1500 steps. In

Table 11: MIA AUC, Llama-8B. Left block: RM (continuous reward). Right block: RLVR (binary reward). Bold marks the best AUC per alignment method (all ties bolded). Avg columns are means over the five methods of their block (— where a method is missing). Classic signals (top block) report the better direction.

	RM (continuous reward)						RLVR (binary reward)					
Method	SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg
Classic signals												
LOSS (self-gen)	0.568	0.550	0.528	0.503	0.573	0.544	0.601	0.638	0.565	0.521	0.576	0.580
Zlib (self-gen)	0.577	**0.579**	0.539	0.509	0.574	0.556	0.527	0.569	0.515	0.523	0.517	0.530
Min-K% (self-gen)	0.526	0.471	0.561	0.491	0.556	0.521	0.540	0.570	0.567	0.516	0.549	0.548
Entropy (self-gen)	0.537	0.542	0.512	0.516	0.571	0.536	0.517	0.624	0.669	0.520	0.580	0.582
LOSS (static)	0.571	0.527	0.543	0.510	0.517	0.534	0.584	0.504	0.590	0.585	0.552	0.563
Zlib (static)	0.593	0.551	0.570	0.571	**0.596**	0.576	0.558	0.508	0.567	0.507	0.523	0.533
Min-K% (static)	0.544	0.500	0.548	0.488	0.531	0.522	0.539	0.547	0.568	**0.665**	**0.625**	0.589
Entropy (static)	0.554	0.541	0.561	0.520	0.529	0.541	0.515	0.549	0.530	0.659	0.548	0.560
Neighbor (static)	0.560	0.475	0.559	0.543	0.502	0.528	0.506	0.513	0.567	0.602	0.568	0.551
Shadow-free												
PREMIA	0.554	0.550	0.502	0.517	0.515	0.528	0.884	0.571	0.513	0.655	0.519	0.628
ENV	0.554	0.538	0.539	0.505	0.502	0.528	0.717	0.662	0.765	0.624	0.547	0.663
ENV-simple	0.506	0.565	0.525	0.566	0.534	0.539	0.779	0.620	0.713	0.601	0.545	0.652
Shadow-based												
S-DIBA	0.529	0.551	0.592	0.503	0.517	0.538	0.511	0.625	0.510	0.505	0.555	0.541
SLR-NB	**0.747**	0.510	0.590	0.551	0.510	0.582	0.951	**0.666**	**0.951**	0.546	0.521	0.727
SLR-QDA	0.702	0.530	0.574	**0.590**	0.544	0.588	**0.964**	0.560	0.931	0.627	0.545	0.725
SLR-NB-W	0.746	0.577	**0.603**	0.518	0.514	0.592	0.959	0.638	0.935	0.534	0.500	0.713
SLR-NB (top-K)	0.552	0.575	0.560	0.516	0.509	0.542	0.909	0.593	0.905	0.625	0.537	0.714
SLR-QDA (top-K)	0.557	0.548	0.532	0.519	0.536	0.538	0.910	0.616	0.902	0.615	0.514	0.711

Table 12: Blend weights of the SLR calibration, one row per evaluated column. Each weight is the softmax mass the fingerprint match assigns to the shadow policy of that alignment method, so each row is the framework's automatic identification of the target's method. The largest weight per row is bolded. A row whose largest weight is not its own method is a failed identification. GPT-2's panel predates the fingerprint blend and carries no weights. Left: Llama-1B and Qwen-1.5B. Right: Qwen-3B and Llama-8B.

Model	Side	Target	SFT	DPO	NPO	GRPO	PPO
Llama-1B	RM	SFT	**0.994**	0	0.004	0	0.002
	RM	DPO	0	**1.000**	0	0	0
	RM	NPO	0	0	**0.966**	0	0.034
	RM	GRPO	0	0	0	**1.000**	0
	RM	PPO	0	0	0.001	0.004	**0.995**
	RLVR	SFT	**1.000**	0	0	0	0
	RLVR	DPO	0	**0.999**	0.001	0	0
	RLVR	NPO	0	0.023	**0.898**	0	0.079
	RLVR	GRPO	0	0	0	**1.000**	0
	RLVR	PPO	0	0	0.008	0	**0.991**
Qwen-1.5B	RM	SFT	**0.999**	0	0	0.001	0
	RM	DPO	0	**1.000**	0	0	0
	RM	NPO	0	0	**0.998**	0	0.002
	RM	GRPO	0	0	0.011	**0.987**	0.003
	RM	PPO	0	0	0.372	0.001	**0.627**
	RLVR	SFT	**1.000**	0	0	0	0
	RLVR	DPO	0	**0.597**	0.403	0	0
	RLVR	NPO	0.003	0.048	**0.945**	0	0.005
	RLVR	GRPO	0	0	0	**1.000**	0
	RLVR	PPO	0	0	0	0.004	**0.996**

Model	Side	Target	SFT	DPO	NPO	GRPO	PPO
Qwen-3B	RM	SFT	**1.000**	0	0	0	0
	RM	DPO	0	**0.999**	0	0	0
	RM	NPO	0	0	**0.999**	0	0.001
	RM	GRPO	0	0	0	**0.999**	0.001
	RM	PPO	0.245	0.006	**0.571**	0.001	0.178
	RLVR	SFT	**1.000**	0	0	0	0
	RLVR	DPO	0	**0.997**	0.003	0	0
	RLVR	NPO	0	0.012	**0.987**	0	0.001
	RLVR	GRPO	0	0	0	**0.983**	0.017
	RLVR	PPO	0	0	0.009	0.001	**0.990**
Llama-8B	RM	SFT	**1.000**	0	0	0	0
	RM	DPO	0	**0.786**	0.213	0	0.001
	RM	NPO	0	0.035	**0.964**	0	0
	RM	GRPO	0.359	0.030	0.069	0	**0.543**
	RM	PPO	0	0.001	0	0	**0.999**
	RLVR	SFT	**1.000**	0	0	0	0
	RLVR	DPO	0	**0.963**	0.037	0	0
	RLVR	NPO	0.247	0.001	**0.727**	0.004	0.021
	RLVR	GRPO	0	0	0	**0.996**	0.004
	RLVR	PPO	0.001	0	0	0.009	**0.990**

total, we have 62 checkpoint evaluations. For each, we compute the detector panel and the verify accuracy on the learned member subset.

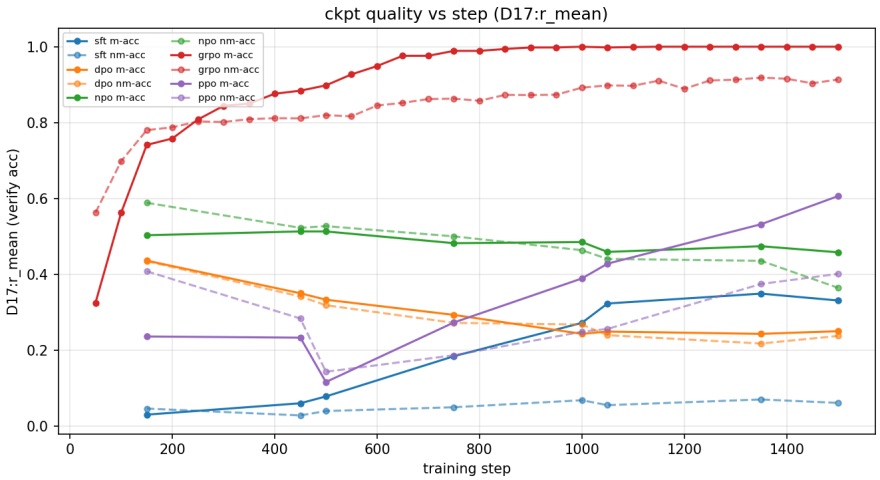

Figure 3: Member and non-member verify accuracy over training stages (Llama-1B, RLVR side, learned member subset). The corrected GRPO run opens far below the non-member level and climbs monotonically to ceiling accuracy. SFT members climb slowly from near zero with a widening gap. DPO and NPO erode. PPO climbs steadily, its member gap turning clearly positive.

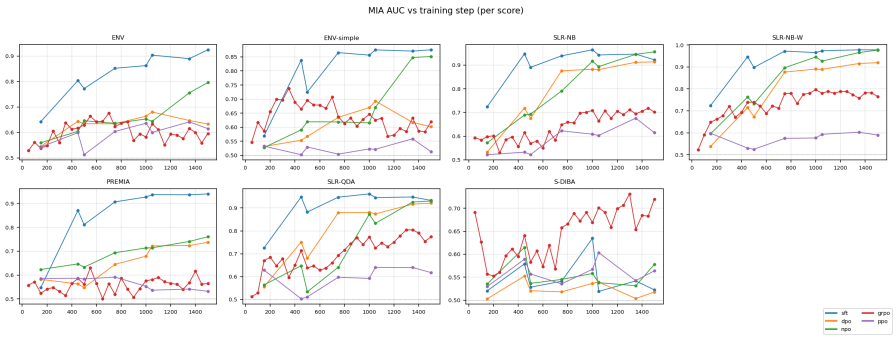

Figure 4: Detector scores over the same checkpoints. The static-calibrated kernels and PREMIA deepen with training in SFT, DPO, and NPO. In the corrected GRPO run the static kernels climb with the behavioral recovery, SLR-QDA and SLR-NB-W from chance to their final values, while PREMIA stays near chance. PPO moves the static kernels only weakly.

The main finding is the sign of the correlation. A positive correlation means the detector score goes up when members learn to verify. A negative correlation means the score goes up when learning does not happen. S-DIBA is the only detector with a positive correlation, and it is the strongest. All detectors based on static or preference surfaces have negative correlations. The reason is that on this pretrained base, the trace and the ability are learned in different courses. The courses that write a deep static trace (SFT, DPO, NPO) have low, stalled, or eroding verify accuracy. The course that reaches ceiling accuracy is the corrected GRPO run. So across the sweep, *trace depth* and *behavioral gain* go in opposite directions. Any detector that reads the trace inherits this opposite relationship. The negative sign is not a calibration failure. It simply shows which channel the detector reads.

Within each run, the relationship is clearer. DPO and NPO show the trade-off in pure form: the trace gets deeper while verify accuracy drops or stalls. The model memorizes at the cost of competence. SFT is a mild case: its trace deepens while accuracy slowly climbs from the floor, as the base gradually memorizes the hard subset. The corrected GRPO run is the key case. Its behavioral readout climbs from well below the non-member level to ceiling. The static kernels climb with it: SLR-QDA and SLR-NB-W go from chance to their final values. In this run, trace and behavior move together. This is why the pooled correlations of the SLR kernels are close to zero. PREMIA

Table 13: Correspondence between each detector score and the member verify accuracy across the stage sweep: Pearson and Spearman correlation over the 62 checkpoint evaluations (Llama-1B, RLVR side, learned member subset).

Detector	Pearson r	Spearman ρ
S-DIBA	$+0.74$	$+0.77$
SLR-NB	-0.43	-0.22
SLR-QDA	-0.20	-0.09
SLR-NB-W	-0.13	-0.04
ENV	-0.39	-0.38
ENV-simple	-0.12	-0.02
PREMIA	-0.56	-0.49

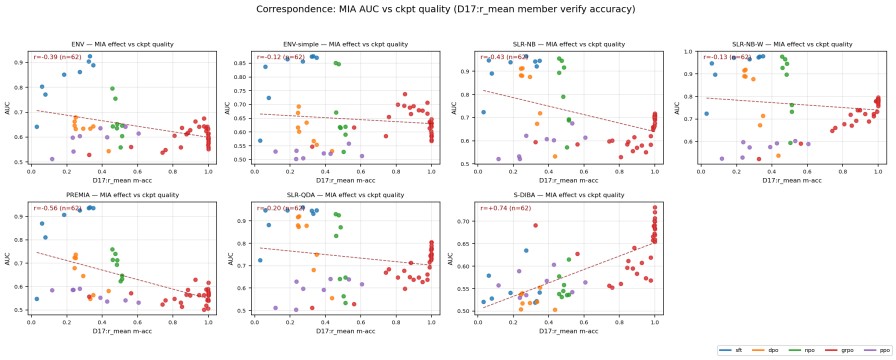

Figure 5: Detector score against member verify accuracy, one panel per detector of Table 5, with the correlation annotated.

stays near chance in both RL runs. It reads the preference displacement that only supervised courses write, so its negative correlation remains in both orderings.

The two failure patterns from the panel evaluations appear again, but only PPO's pattern survives the sweep. PPO moves the behavioral readout, and its member gap becomes clearly positive, while the static kernels move only weakly. This is behavioral movement without member content. The corrected GRPO run is a partial exception. Its behavioral channel is silent at the checkpoint used in the panel evaluation, but in the later half of the run, the static trace deepens along with the behavioral climb. This means learning eventually prints on content. Here, we look at training depth within one model, where the competition between memorization and competence dominates. The sign flip between the reward-statistics detector and the static kernels is the clearest evidence that each detector reads its own channel, and that the panel as a whole misses neither. The static kernels record every stage where the trace deepens, and the reward-statistics detector records every stage where behavior moves.

F.2 DIBA CLASSIFIER VARIANTS

The main tables use the original five-axis S-DIBA. Table 14 extends the same classifier to the full detector vector, with and without the SLR fingerprint. We compare each variant against the SLR kernels on average. Each DIBA variant reports its five per-method AUCs and their mean. SLR is summarized by the mean of its three kernels across the five methods.

The results in Table 14 shows findings as follow. Adding the SLR fingerprint to the original five-signal S-DIBA gives almost no gain, because the fingerprint alone cannot help when the classifier only reads five signals and does not cover all four families. Extending S-DIBA to the full 19-dimensional detector vector improves over the original S-DIBA, showing that our 19-signal choice is better than DIBA's five signals. When DIBA combines both innovations (S-DIBA-SLR-19d), it comes close to our SLR methods. This shows that the gain comes from our two design choices

Table 14: DIBA extension ablation. SLR (avg): the mean of the three SLR kernels across the five alignment methods (fifteen values per side); S-DIBA (avg): the mean of the original five-axis S-DIBA across the five methods. Each DIBA variant reports its five per-method AUCs and their average.

Side	SLR (avg)	S-DIBA (avg)	S-DIBA-SLR						S-DIBA-19d						S-DIBA-SLR-19d					
			SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg	SFT	DPO	NPO	GRPO	PPO	Avg
Llama-3.2-1B																				
RM	0.790	0.563	0.570	0.555	0.523	0.586	0.527	0.552	0.563	0.530	0.656	0.554	0.558	0.572	0.985	0.876	0.867	0.554	0.602	0.777
RLVR	0.818	0.546	0.593	0.582	0.528	0.594	0.608	0.581	0.813	0.504	0.725	0.671	0.574	0.657	0.975	0.926	0.976	0.671	0.664	0.842
Qwen-1.5B																				
RM	0.754	0.580	0.531	0.535	0.531	0.548	0.556	0.540	0.723	0.682	0.693	0.591	0.579	0.654	0.996	0.844	0.826	0.591	0.562	0.764
RLVR	0.755	0.572	0.504	0.555	0.520	0.501	0.607	0.537	0.730	0.639	0.837	0.565	0.554	0.665	0.975	0.833	0.924	0.565	0.586	0.777
Qwen-3B																				
RM	0.664	0.542	0.522	0.553	0.501	0.561	0.508	0.529	0.841	0.579	0.528	0.524	0.560	0.606	0.990	0.669	0.683	0.523	0.502	0.673
RLVR	0.799	0.590	0.544	0.553	0.568	0.637	0.573	0.575	0.540	0.582	0.562	0.657	0.600	0.588	0.958	0.938	0.960	0.657	0.523	0.807
Llama-8B																				
RM	0.587	0.538	0.665	0.545	0.534	0.542	0.519	0.561	0.515	0.617	0.589	0.547	0.522	0.558	0.749	0.540	0.620	0.600	0.551	0.612
RLVR	0.722	0.541	0.530	0.651	0.671	0.505	0.621	0.595	0.509	0.642	0.594	0.673	0.627	0.609	0.970	0.607	0.953	0.673	0.556	0.752
GPT-2 Medium (RM side only)																				
RM	0.651	0.572	0.513	0.522	0.527	0.634	0.528	0.545	0.748	0.513	0.511	0.604	0.516	0.579	0.786	0.559	0.605	0.604	0.618	0.634

working together, not from an advantage of DIBA itself: the full 19 signals and the SLR fingerprint are complementary, and the fingerprint needs a comprehensive signal set to reach its full effect.

G Per-Signal Detector Tables

Per-signal AUCs for every model and side. Bold marks the best score per column across all rows of the table, the detector signals and the reference baselines (DIBA axes, PREMIA) alike. All tied bests are bolded.

GPT-2 Medium (RM side).

Table 15: Per-signal AUC, GPT-2 Medium (deliberately overfit). Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.510	0.633	0.631	0.615	0.644
Prompt	D2: RMIA-Frac	0.509	0.632	0.634	0.611	0.646
LR	D3: LR-Asym	0.859	0.521	0.546	0.565	0.553
LR	D4: |LR-Rej|	0.526	0.509	0.540	0.592	0.520
Token	D5: TokenMean	**0.993**	0.513	0.516	0.504	0.532
Token	D6: Token10pct	0.954	0.518	0.509	0.520	0.510
Token	D7: TokenLR-Mean	0.797	0.529	0.503	0.632	0.507
Token	D8: TokenLR-Std	0.549	0.521	0.514	0.519	0.525
Token	D9: TokenLR-Skew	0.546	0.529	0.516	0.547	0.528
Reward	D10: RewardStd	0.624	0.615	0.530	0.548	0.522
Reward	D11: Reward-Adv	0.679	0.588	0.506	0.653	0.522
Reward	D12: Reward-Range	0.647	0.626	0.541	0.556	0.514
Reward	D13: RewardGap-Mean	0.579	0.618	0.509	0.543	0.519
Reward	D14: r-mean	0.579	0.617	0.512	0.544	0.519
Reward	D15: r-max	0.518	0.503	0.551	0.564	0.538
Reward	D16: r-min	0.643	0.665	0.567	0.550	0.504
Reward	D17: r-topk	0.555	0.502	0.508	0.593	0.550
Reward	D18: r-snr	0.561	0.622	0.503	0.536	0.522
Reward	D19: r-lift	0.823	**0.888**	**0.678**	**0.815**	0.663
DIBA	$a_{\max}$	0.518	0.503	0.551	0.564	0.538
DIBA	a_{mean}	0.579	0.617	0.512	0.544	0.519
DIBA	a_{std}	0.624	0.615	0.530	0.548	0.522
DIBA	b_{KL}	0.549	0.539	0.528	0.566	0.648
DIBA	b_{token}	0.566	0.555	0.503	0.581	**0.667**
	PREMIA	0.741	0.503	0.535	0.633	0.507

Llama-3.2-1B, RM side.

Table 16: Per-signal AUC, Llama-3.2-1B, RM side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO
Prompt	D1: PrefStr	0.606	0.560	0.604	0.502
Prompt	D2: RMIA-Frac	0.605	0.560	0.602	0.502
LR	D3: LR-Asym	0.947	0.800	0.746	0.516
LR	D4: |LR-Rej|	0.552	0.540	0.690	0.508
Token	D5: TokenMean	**0.995**	**0.808**	0.797	0.530
Token	D6: Token10pct	0.990	0.787	0.770	0.506
Token	D7: TokenLR-Mean	0.992	0.806	**0.847**	0.526
Token	D8: TokenLR-Std	0.919	0.777	0.778	0.527
Token	D9: TokenLR-Skew	0.808	0.776	0.740	0.541
Reward	D10: RewardStd	0.537	0.532	0.528	0.557
Reward	D11: Reward-Adv	0.506	0.580	0.609	0.546
Reward	D12: Reward-Range	0.542	0.523	0.530	0.545
Reward	D13: RewardGap-Mean	0.519	0.516	0.543	0.565
Reward	D14: r-mean	0.500	0.507	0.564	0.753
Reward	D15: r-max	0.524	0.519	0.517	0.743
Reward	D16: r-min	0.553	0.526	0.583	0.633
Reward	D17: r-topk	0.520	0.509	0.561	0.743
Reward	D18: r-snr	0.500	0.500	0.500	0.500
Reward	D19: r-lift	0.730	0.803	0.765	**0.780**
DIBA	$a_{\max}$	0.524	0.519	0.517	0.743
DIBA	a_{mean}	0.500	0.507	0.564	0.753
DIBA	a_{std}	0.537	0.532	0.528	0.557
DIBA	b_{KL}	0.528	0.512	0.521	0.529
DIBA	b_{token}	0.508	0.516	0.562	0.508
	PREMIA	0.550	0.593	0.667	0.523

Llama-3.2-1B, RLVR side.

Qwen-1.5B, RM side.

Qwen-1.5B, RLVR side.

Qwen-3B, RM side.

Qwen-3B, RLVR side.

Llama-8B, RM side.

Llama-8B, RLVR side.

Table 17: Per-signal AUC, Llama-3.2-1B, RLVR side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.571	0.525	0.529	0.516	0.505
Prompt	D2: RMIA-Frac	0.572	0.526	0.530	0.517	0.506
LR	D3: LR-Asym	0.740	**0.876**	0.885	0.541	0.501
LR	D4: |LR-Rej|	0.636	0.715	0.809	0.508	**0.533**
Token	D5: TokenMean	0.985	0.781	0.963	0.500	0.504
Token	D6: Token10pct	0.976	0.766	0.958	0.511	0.510
Token	D7: TokenLR-Mean	0.965	0.782	0.943	0.546	0.515
Token	D8: TokenLR-Std	**0.989**	0.750	**0.972**	**0.591**	0.508
Token	D9: TokenLR-Skew	0.970	0.745	0.949	0.523	0.523
Reward	D10: RewardStd	0.520	0.509	0.521	0.510	0.500
Reward	D11: Reward-Adv	0.520	0.511	0.518	0.511	0.506
Reward	D12: Reward-Range	0.520	0.512	0.518	0.508	0.504
Reward	D13: RewardGap-Mean	0.520	0.511	0.518	0.511	0.506
Reward	D14: r-mean	0.556	0.528	0.526	0.522	0.510
Reward	D15: r-max	0.556	0.522	0.526	0.510	0.514
Reward	D16: r-min	0.536	0.510	0.508	0.518	0.518
Reward	D17: r-topk	0.556	0.525	0.529	0.516	0.509
Reward	D18: r-snr	0.556	0.528	0.526	0.522	0.510
Reward	D19: r-lift	0.553	0.505	0.501	0.552	0.510
DIBA	$a_{\max}$	0.556	0.522	0.526	0.510	0.514
DIBA	a_{mean}	0.556	0.528	0.526	0.522	0.510
DIBA	a_{std}	0.520	0.509	0.521	0.510	0.500
DIBA	b_{KL}	0.508	0.522	0.535	0.535	0.531
DIBA	b_{token}	0.520	0.510	0.538	0.520	0.503
	PREMIA	0.958	0.733	0.763	0.534	0.524

Table 18: Per-signal AUC, Qwen-1.5B, RM side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.550	0.585	0.507	0.558	0.557
Prompt	D2: RMIA-Frac	0.550	0.585	0.507	0.558	0.559
LR	D3: LR-Asym	0.949	0.757	0.680	0.523	0.511
LR	D4: |LR-Rej|	0.550	0.550	0.604	0.586	0.503
Token	D5: TokenMean	**0.997**	**0.786**	**0.785**	0.542	0.534
Token	D6: Token10pct	**0.997**	0.770	0.769	0.526	0.508
Token	D7: TokenLR-Mean	0.986	0.768	0.740	0.537	0.520
Token	D8: TokenLR-Std	0.950	0.742	0.775	0.565	0.544
Token	D9: TokenLR-Skew	0.880	0.560	0.658	0.638	0.527
Reward	D10: RewardStd	0.582	0.506	0.506	0.562	0.624
Reward	D11: Reward-Adv	0.662	0.546	0.581	0.571	0.579
Reward	D12: Reward-Range	0.620	0.500	0.531	0.584	0.618
Reward	D13: RewardGap-Mean	0.647	0.528	0.542	0.577	0.656
Reward	D14: r-mean	0.555	0.570	0.607	0.598	0.659
Reward	D15: r-max	0.562	0.542	0.593	0.575	0.636
Reward	D16: r-min	0.593	0.571	0.590	0.638	0.629
Reward	D17: r-topk	0.515	0.552	0.605	0.584	0.646
Reward	D18: r-snr	0.500	0.505	0.505	0.546	0.545
Reward	D19: r-lift	0.710	0.740	0.706	**0.725**	**0.732**
DIBA	$a_{\max}$	0.562	0.542	0.593	0.575	0.636
DIBA	a_{mean}	0.555	0.570	0.607	0.598	0.659
DIBA	a_{std}	0.582	0.506	0.506	0.562	0.624
DIBA	b_{KL}	0.557	0.531	0.546	0.583	0.533
DIBA	b_{token}	0.559	0.543	0.554	0.516	0.594
	PREMIA	0.703	0.563	0.583	0.501	0.571

Table 19: Per-signal AUC, Qwen-1.5B, RLVR side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.630	0.503	0.537	0.514	0.563
Prompt	D2: RMIA-Frac	0.631	0.503	0.537	0.514	0.564
LR	D3: LR-Asym	0.818	**0.911**	0.846	0.514	0.520
LR	D4: |LR-Rej|	0.535	0.862	0.677	0.531	0.563
Token	D5: TokenMean	**0.981**	0.788	**0.966**	0.597	0.550
Token	D6: Token10pct	0.973	0.760	0.957	0.625	0.601
Token	D7: TokenLR-Mean	0.939	0.788	0.931	0.508	0.618
Token	D8: TokenLR-Std	0.959	0.724	0.958	0.524	0.563
Token	D9: TokenLR-Skew	0.964	0.623	0.952	0.509	0.568
Reward	D10: RewardStd	0.507	0.505	0.545	0.535	0.538
Reward	D11: Reward-Adv	0.508	0.505	0.547	0.536	0.515
Reward	D12: Reward-Range	0.510	0.505	0.545	0.535	0.550
Reward	D13: RewardGap-Mean	0.508	0.505	0.547	0.536	0.515
Reward	D14: r-mean	0.528	0.542	0.518	0.547	0.552
Reward	D15: r-max	0.525	0.540	0.500	0.510	0.545
Reward	D16: r-min	0.515	0.535	0.545	0.545	0.505
Reward	D17: r-topk	0.528	0.542	0.507	0.515	0.566
Reward	D18: r-snr	0.528	0.542	0.518	0.546	0.551
Reward	D19: r-lift	0.512	0.577	0.530	**0.801**	**0.751**
DIBA	$a_{\max}$	0.525	0.540	0.500	0.510	0.545
DIBA	a_{mean}	0.528	0.542	0.518	0.547	0.552
DIBA	a_{std}	0.507	0.505	0.545	0.535	0.538
DIBA	b_{KL}	0.562	0.552	0.518	0.568	0.571
DIBA	b_{token}	0.566	0.519	0.524	0.641	0.574
	PREMIA	0.934	0.869	0.719	0.597	0.599

Table 20: Per-signal AUC, Qwen-3B, RM side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.536	0.547	0.544	0.555	0.570
Prompt	D2: RMIA-Frac	0.536	0.548	0.545	0.555	0.567
LR	D3: LR-Asym	0.959	0.553	0.624	0.535	0.513
LR	D4: |LR-Rej|	0.532	0.599	0.589	0.502	0.517
Token	D5: TokenMean	**0.998**	0.641	0.592	0.538	0.517
Token	D6: Token10pct	**0.998**	0.618	0.602	0.540	0.511
Token	D7: TokenLR-Mean	0.992	0.635	0.610	0.526	0.580
Token	D8: TokenLR-Std	0.929	0.625	0.535	0.528	0.536
Token	D9: TokenLR-Skew	0.945	0.502	0.574	0.538	0.558
Reward	D10: RewardStd	0.662	0.504	0.643	0.574	0.527
Reward	D11: Reward-Adv	0.509	0.535	0.514	0.552	0.574
Reward	D12: Reward-Range	0.659	0.527	0.610	0.574	0.511
Reward	D13: RewardGap-Mean	0.654	0.518	0.619	0.552	0.524
Reward	D14: r-mean	0.521	0.507	0.525	0.501	0.532
Reward	D15: r-max	0.584	0.528	0.561	0.500	0.531
Reward	D16: r-min	0.612	0.515	0.502	0.501	0.545
Reward	D17: r-topk	0.566	0.518	0.560	0.500	0.517
Reward	D18: r-snr	0.500	0.520	0.502	0.500	0.500
Reward	D19: r-lift	0.727	**0.736**	**0.772**	**0.723**	**0.748**
DIBA	$a_{\max}$	0.584	0.528	0.561	0.500	0.531
DIBA	a_{mean}	0.521	0.507	0.525	0.500	0.532
DIBA	a_{std}	0.662	0.504	0.643	0.574	0.527
DIBA	b_{KL}	0.541	0.578	0.558	0.558	0.505
DIBA	b_{token}	0.518	0.569	0.531	0.571	0.504
	PREMIA	0.734	0.521	0.626	0.546	0.594

Table 21: Per-signal AUC, Qwen-3B, RLVR side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.655	0.566	0.556	0.545	**0.635**
Prompt	D2: RMIA-Frac	0.656	0.565	0.556	0.546	**0.635**
LR	D3: LR-Asym	0.689	**0.960**	0.838	0.634	0.503
LR	D4: |LR-Rej|	0.766	0.929	0.744	0.558	0.533
Token	D5: TokenMean	**0.979**	0.861	**0.972**	0.546	0.575
Token	D6: Token10pct	0.964	0.855	0.950	0.563	0.526
Token	D7: TokenLR-Mean	0.941	0.860	0.943	0.554	0.502
Token	D8: TokenLR-Std	0.941	0.839	0.935	0.542	0.513
Token	D9: TokenLR-Skew	0.965	0.776	0.939	0.507	0.502
Reward	D10: RewardStd	0.521	0.508	0.532	0.540	0.543
Reward	D11: Reward-Adv	0.520	0.509	0.534	0.540	0.544
Reward	D12: Reward-Range	0.520	0.510	0.535	0.540	0.540
Reward	D13: RewardGap-Mean	0.520	0.509	0.534	0.540	0.544
Reward	D14: r-mean	0.600	0.525	0.508	0.550	0.567
Reward	D15: r-max	0.600	0.525	0.515	0.510	0.530
Reward	D16: r-min	0.580	0.515	0.520	0.550	0.570
Reward	D17: r-topk	0.602	0.524	0.510	0.520	0.541
Reward	D18: r-snr	0.600	0.525	0.508	0.550	0.567
Reward	D19: r-lift	0.716	0.613	0.647	**0.750**	0.627
DIBA	$a_{\max}$	0.600	0.525	0.515	0.510	0.530
DIBA	a_{mean}	0.600	0.525	0.508	0.550	0.567
DIBA	a_{std}	0.521	0.508	0.533	0.540	0.543
DIBA	b_{KL}	0.537	0.608	0.601	0.605	0.547
DIBA	b_{token}	0.510	0.595	0.593	0.552	0.627
	PREMIA	0.934	0.927	0.736	0.526	0.542

Table 22: Per-signal AUC, Llama-8B, RM side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.535	0.507	0.545	0.507	0.559
Prompt	D2: RMIA-Frac	0.534	0.506	0.544	0.507	0.560
LR	D3: LR-Asym	0.511	0.550	0.556	0.539	0.519
LR	D4: |LR-Rej|	0.518	0.522	0.520	0.565	0.514
Token	D5: TokenMean	0.665	0.539	0.601	0.505	0.519
Token	D6: Token10pct	0.651	0.524	0.606	0.512	0.524
Token	D7: TokenLR-Mean	0.578	0.602	0.536	0.534	0.519
Token	D8: TokenLR-Std	0.526	0.544	0.503	0.505	0.537
Token	D9: TokenLR-Skew	0.507	0.556	0.515	0.535	0.526
Reward	D10: RewardStd	0.699	0.508	0.546	0.558	0.576
Reward	D11: Reward-Adv	0.510	0.510	0.529	0.530	0.517
Reward	D12: Reward-Range	0.670	0.513	0.551	0.567	0.581
Reward	D13: RewardGap-Mean	0.713	0.516	0.513	0.558	0.594
Reward	D14: r-mean	0.626	0.557	0.580	0.538	0.560
Reward	D15: r-max	0.524	0.543	0.569	0.550	0.538
Reward	D16: r-min	0.660	0.547	0.590	0.515	0.552
Reward	D17: r-topk	0.558	0.544	0.575	0.550	0.551
Reward	D18: r-snr	0.545	0.520	0.510	0.500	0.514
Reward	D19: r-lift	**0.760**	**0.793**	**0.758**	**0.758**	**0.817**
DIBA	$a_{\max}$	0.524	0.543	0.569	0.550	0.538
DIBA	a_{mean}	0.626	0.557	0.580	0.538	0.560
DIBA	a_{std}	0.699	0.508	0.546	0.558	0.576
DIBA	b_{KL}	0.574	0.526	0.514	0.583	0.534
DIBA	b_{token}	0.535	0.540	0.523	0.551	0.575
	PREMIA	0.554	0.550	0.502	0.517	0.515

Table 23: Per-signal AUC, Llama-8B, RLVR side. Bold: best per column.

Group	Signal	SFT	DPO	NPO	GRPO	PPO
Prompt	D1: PrefStr	0.532	0.542	0.533	0.538	0.564
Prompt	D2: RMIA-Frac	0.533	0.543	0.533	0.539	0.564
LR	D3: LR-Asym	0.605	0.590	0.517	0.570	0.560
LR	D4: |LR-Rej|	0.642	0.505	0.565	0.588	0.569
Token	D5: TokenMean	**0.969**	0.603	**0.934**	0.557	0.501
Token	D6: Token10pct	0.953	0.609	0.915	0.535	0.512
Token	D7: TokenLR-Mean	0.875	0.591	0.851	0.664	0.518
Token	D8: TokenLR-Std	0.886	0.614	0.828	0.594	0.526
Token	D9: TokenLR-Skew	0.729	0.601	0.770	0.638	0.504
Reward	D10: RewardStd	0.549	0.603	0.501	0.560	0.555
Reward	D11: Reward-Adv	0.536	0.594	0.516	0.571	0.600
Reward	D12: Reward-Range	0.550	0.615	0.510	0.550	0.535
Reward	D13: RewardGap-Mean	0.535	0.594	0.516	0.571	0.600
Reward	D14: r-mean	0.645	0.645	0.802	0.578	0.617
Reward	D15: r-max	0.610	0.650	0.710	0.505	0.510
Reward	D16: r-min	0.560	0.535	0.700	0.555	0.545
Reward	D17: r-topk	0.623	0.656	0.742	0.525	0.555
Reward	D18: r-snr	0.645	0.645	0.801	0.578	0.617
Reward	D19: r-lift	0.794	**0.742**	0.862	**0.801**	**0.814**
DIBA	$a_{\max}$	0.610	0.650	0.710	0.505	0.510
DIBA	a_{mean}	0.645	0.645	0.802	0.578	0.617
DIBA	a_{std}	0.549	0.603	0.501	0.560	0.555
DIBA	b_{KL}	0.515	0.504	0.525	0.541	0.527
DIBA	b_{token}	0.574	0.512	0.540	0.546	0.664
	PREMIA	0.869	0.503	0.570	0.668	0.507