FROM REPRESENTATION TO READOUT: MECHANISTIC ANALYSIS OF MEMBERSHIP SIGNALS IN LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Membership inference aims to determine whether a particular example was included in a model's training data. Such membership signals are often broadly described as memorization, while the underlying mechanisms can involve both how information is represented internally and how it is accessed by the model to produce behaviors. We therefore present a mechanistic study of membership signals in large language models (LLMs). We introduce Decodable Information Score, Used Information Score, and Readout Accessibility Gap (DUG), a set of complementary measures that separately quantify membership information decodable from internal representations and the extent to which it is exposed through the model's readout. Across supervised fine-tuning (SFT) and reinforcement learning (RL)-based membership inference methods, membership-specific effects appear in both the amount of decodable information and readout under SFT, while under RL-based membership inference methods they appear primarily in readout. Activation patching further localizes this membership-specific effect to intermediate layers. To examine whether these findings extend to a more general RL setting, we further study reinforcement learning with verifiable rewards (RLVR) contamination, where the training reward is independent of the reference solution. In this setting, we find that behavioral changes can emerge without a corresponding increase in likelihood-based membership inference signals.

1 INTRODUCTION

Membership inference attacks (MIAs) aim to determine whether a given example belongs to a model's training data. They have been widely studied as a way to characterize and audit a model's exposure to training data. We refer to included examples as members and excluded examples as non-members (Shokri et al., 2017; Duan et al., 2024). Prior work uses signals from the behavior of large language models (LLMs), such as loss, perplexity, and model outputs, to infer membership (Xie et al., 2025; Carlini et al., 2021). More recent approaches further modify the target model weights to bring out membership-related behaviors, typically using reinforcement learning (RL) or supervised fine-tuning (SFT) to elicit membership signals (Yin et al., 2026; Fu et al., 2024). Such signals are often broadly referred to as memorization.

Interpretability studies of memorization have provided evidence that information associated with memorized content can manifest internally, and have shown that such representations can be actively steered at inference time to affect model behavior (Stoehr et al., 2024). Much of this work studies memorization through the model's ability to reproduce training content. Some studies group examples by their reproducibility and analyze the resulting differences (Chen et al., 2026; Wang et al., 2025). Others manipulate training exposure and examine the resulting changes in model behavior (Lee et al., 2022). These analyses primarily ask how well the model reproduces a given example, rather than directly comparing members and non-members to characterize how training membership is reflected in the model's internal representations. In particular, it remains unclear how these interventions alter how models expose information associated with their training data.

Motivated by this gap, we ask: *How do training interventions shape membership signals in language models? –by changing the information encoded in internal representations, by changing how that*

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.

information is read out, or both? To address this question, we introduce the **D**ecodable Information Score, **U**sed Information Score, and Readout Accessibility **G**ap, collectively termed **DUG**, a layer-wise set of complementary measures that quantify membership information in internal representations and its accessibility through the model's readout. We combine these measures with a difference-in-differences analysis to isolate membership-specific changes induced by training. We then apply this framework to SFT and RL-based MIA methods, further using activation patching to causally localize where the induced membership-specific effect forms, and examine whether these findings extend to a more general RLVR contamination setting.

To be specific, our main contributions can be listed as follows:

- We propose **DUG**, together with a difference-in-differences analysis that separates membership-specific changes from those shared with non-members, decomposes them into decodable information and model readout, and quantifies both across layers.
- Across controlled SFT and RL-based membership inference methods, we find that membership-specific changes under SFT emerge in both decodable information and readout, whereas RL-based MIA methods primarily affect the readout of membership-specific information (Section 5.2). We further use activation patching to causally localize this effect across layers (Section 5.5).
- We show, in a relatively general RLVR contamination setting where the reward is independent of the reference solution, that behavioral membership effects can emerge without a corresponding increase in likelihood-based MIA performance, thereby revealing a dissociation between behavioral and likelihood-based membership inference (Section 5.3).

2 Related Work

2.1 Membership Inference Attack in LLMs

MIA aims to determine whether a given data sample was included in a model's training data by exploiting differences in the model's behavior on member and non-member samples (Shokri et al., 2017; Duan et al., 2024). Prior work has explored methods that actively induce models to expose membership signals through targeted interventions, including both training-based and training-free approaches. Tao et al. (2025) argue that RL creates a detectable membership signal through policy collapse, as rewarded contaminated samples show more stable and less diverse reasoning when probed for membership. Similarly, Zhang et al. (2026) link RLVR-induced policy collapse to membership inference, showing that membership signals become increasingly detectable as RLVR training progresses. Yin et al. (2026) directly optimize models via RLVR for membership detection to elicit latent membership signals. These methods suggest that training can make membership signals more pronounced (Liu et al., 2026; Gwak et al., 2026). Controlled-contamination studies have also considered settings where only a small subset of evaluation data is introduced into training (Magar & Schwartz, 2022; Shi et al., 2024; Oren et al., 2024). However, these works primarily evaluate signal detectability, while how training changes their representation and readout of remains unexplored.

2.2 Interpretability of Memorization in LLMs

Memorization in language models has been studied primarily through behavioral tests of whether a model can reproduce or extract training data. Existing works have quantified memorization using likelihood- or perplexity-based measures (Xie et al., 2025; Carlini et al., 2021), and used membership inference to assess whether training examples exhibit distinguishable behavior from non-members (Yan et al., 2026; Duan et al., 2024; Tao & Shokri, 2025).

From a white-box perspective, prior work has investigated the internal mechanisms underlying what is broadly characterized as memorization, using mechanistic interpretability tools such as circuit discovery and activation steering to identify representations involved in initiating and maintaining memorized behavior (Makhija et al., 2026; Prabhakar et al., 2024). Related work on representation editing further shows that such internal representations can be intervened on to amplify or suppress specific behaviors (Stolfo et al., 2025). These studies provide evidence that memorization can be linked to identifiable and manipulable internal representations. However, it remains unclear how the membership signals detected by MIA relate to these internal representations, and whether training primarily changes the information stored in the representation or how that information is read out.

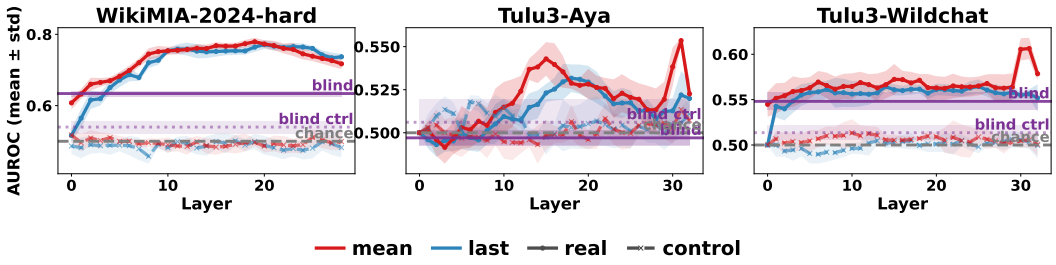


Figure 1: Layer-wise probe AUROC for membership prediction on three benchmarks.

3 Preliminary: Is Membership Linearly Decodable?

Before investigating how further training affects membership signals, we first study whether these signals are already encoded in publicly released models prior to any additional training, in a binary setting where members come from the model's own pre-training or post-training data. Specifically, we train linear probes on the residual-stream representations at each layer to assess the linear decodability of membership status from internal model representations. For pre-training membership, we evaluate Qwen2-7B on WikiMIA-2024-Hard (Hallinan et al., 2026), where members and non-members are separated by a temporal cutoff. For post-training membership, we examine the open Tulu3 model on the Aya (Singh et al., 2024) and WildChat (Zhao et al., 2024) subsets of its post-training mix, with members included in the mix and non-members held out.

Baselines and Controls. For each dataset, we compare our probe AUROC with a blind baseline that predicts membership from TF-IDF and surface text statistics alone without access to the model's representations (Das et al., 2025). We also retrain the probes after shuffling membership labels as a control. A probe is considered informative only if it outperforms the blind baseline while the shuffled control remains at chance level as expected.

Findings. Figure 1 reports per-layer probe AUROC on the base model, measuring whether a linear probe can distinguish member from non-member sequences. On WikiMIA-2024-hard, probe AUROC rises sharply in early layers and peaks at 0.78 around layer 20, substantially exceeding the blind baseline. On Tulu3-Aya and WildChat, probe AUROC reaches 0.543 and 0.57, respectively, in the middle layers, and rises again to 0.553 and 0.605 at the second-to-last layer before dropping at the final layer. The late-layer pattern aligns with prior findings of output collapse at the final layer and richer features at the second-to-last layer (Belrose et al., 2025; Skean et al., 2025).

These observations show that membership signals are linearly decodable from representations, though their strength and layerwise patterns differ across the two settings. However, binary probing alone does not reveal whether these differences arise from the encoded information, its readout, or both. Moreover, we include binary probing for the models studied in Section 5 in App. A.4, which provides limited resolution of the observed membership effects. This motivates our measures for disentangling these two aspects by measuring information at the level of individual target tokens.

4 The DUG Framework

In the previous section, we showed that membership is linearly decodable from model representations. Here, we move beyond binary membership classification to examine the information underlying this signal. We focus on the target tokens predicted by the model, using their target-predictive information to characterize membership rather than a membership label directly. We quantify how much membership-related information is decodable from the representations and how much is exposed through the model's readout at each layer. Their difference captures information that is encoded in the representations but not exposed by the readout.

4.1 Problem Statement

We study frozen language models obtained by training a base model θ_0. Membership is determined by a training set $\mathcal{T}$: evaluation questions that appear in $\mathcal{T}$ form the members $\mathcal{M}$, and the rest form the non-members $\mathcal{N}$, a partition shared by all models. A trained model θ_a is obtained by training the base model θ_0 on $\mathcal{T}$ under a specified training procedure, such as SFT or an RL-based MIA method. We denote the corresponding training procedure by a.

Each question p consists of a context c_p and a target sequence $y_p^\star$. We teacher-force each model on $c_p \| y_p^\star$ and pair every target token with the residual-stream representation at the preceding position, writing the i-th pair as $(h_i^{(\ell)}, y_i^\star)$ with $h_i^{(\ell)} \in \mathbb{R}^d$ at layer ℓ and $y_i^\star \in \mathcal{V}$, the vocabulary. Let $\mathcal{I}_p$ denote the pairs from question p. For cross-fitting, we split the pairs into K folds grouped by question, keeping each question's tokens together, with training and evaluation splits $\mathcal{S}_{\text{tr}}$ and $\mathcal{S}_{\text{ev}}$.

4.2 Proposed Framework

Motivation. We seek to quantify how much information about the ground-truth next token $y_i^\star$ is decodable from the residual representation $h_i^{(\ell)}$. Since this information is not directly measurable, we compare a probe trained on $h_i^{(\ell)}$ against a null baseline that has no access to it, and define their difference in cross-entropy as the **Decodable Information Score**. We introduce this concept to get a sense of how much target-relevant information there is to begin with, and how it changes with depth.

Null Baseline. For each fold, we define an intercept-only classifier $q_b = \text{softmax}(b)$ with $b \in \mathbb{R}^{|\mathcal{V}|}$, constant across positions. Fitting b on the training split $\mathcal{S}_{\text{tr}}$ recovers the empirical distribution of $y_i^\star$ over that split, so the resulting loss on the evaluation split,

$$\mathcal{L}_{\text{null}}(y_i^\star) = -\log q_b(y_i^\star), \qquad i \in \mathcal{S}_{\text{ev}}. \tag{1}$$

is what remains predictable from the marginal label distribution alone. It carries no dependence on ℓ, and we reuse it for another quantity defined below as well.

Decodable Information. We fit a linear probe $q_\phi(y \mid h) = \text{softmax}(Wh + \beta)$ at each layer of the model, and use $\mathcal{L}_\phi^{(\ell)}(y_i^\star) = -\log q_\phi(y_i^\star \mid h_i^{(\ell)})$ for its cross-entropy loss on the target token. We denote the decodable information score at layer ℓ as $D(\ell)$, and define it as the average reduction in cross-entropy relative to the null baseline on the same folds:

$$D(\ell) \;=\; \frac{1}{|\mathcal{S}_{\text{ev}}|} \sum_{i \in \mathcal{S}_{\text{ev}}} \Big[\mathcal{L}_{\text{null}}(y_i^\star) - \mathcal{L}_\phi^{(\ell)}(y_i^\star) \Big]. \tag{2}$$

The K-fold rotation evaluates each position with a probe fitted on the remaining folds. $D(\ell)$ therefore reflects how much of $y^\star$ a linear probe can recover from layer ℓ without having seen that position during fitting, indicating whether the information has entered the representation.

Used Information. We denote used information score as $U(\ell)$ to quantify how much target-relevant information is exposed by the residual stream at layer ℓ through the model's own unembedding W_U. We hold W_U frozen and read the residual stream through it:

$$\mathcal{L}_{\text{lens}}^{(\ell)}(y_i^\star) = -\log \text{softmax}\big(W_U \, \text{LN}_f(h_i^{(\ell)})\big)_{y_i^\star}. \tag{3}$$

As Equation 4 shows, $U(\ell)$ is defined as the corresponding reduction in cross-entropy. We use the same null baseline and the same folds as for $D(\ell)$. Since no parameters are fitted, $U(\ell)$ reflects what layer ℓ already contributes to the model's own prediction.

$$U(\ell) \;=\; \frac{1}{|\mathcal{S}_{\text{ev}}|} \sum_{i \in \mathcal{S}_{\text{ev}}} \Big[\mathcal{L}_{\text{null}}(y_i^\star) - \mathcal{L}_{\text{lens}}^{(\ell)}(y_i^\star) \Big], \tag{4}$$

where, intuitively, $D(\ell)$ asks what is present in the residual stream, while $U(\ell)$ asks what the model puts to use. We use the logit lens rather than a tuned lens (Belrose et al., 2025), because tuned-lens

translators are trained to match the final-layer distribution and thus absorb computation from later layers, and their trainable parameters can themselves introduce member/non-member differences unrelated to the representation. Details in App. B.

Readout Accessibility Gap. We denote the Readout Accessibility Gap as $G(\ell)$ and define it as the difference of the two scores:

$$G(\ell) \;=\; D(\ell) - U(\ell) \;=\; \frac{1}{|\mathcal{S}_{\mathrm{ev}}|} \sum_{i \in \mathcal{S}_{\mathrm{ev}}} \Big[\mathcal{L}^{(\ell)}_{\mathrm{lens}}(y_i^\star) - \mathcal{L}^{(\ell)}_{\phi}(y_i^\star) \Big]. \tag{5}$$

The null baseline cancels in this difference. A positive $G(\ell)$ means that a linear probe recovers more about $y^\star$ from layer ℓ than the model's own unembedding does, suggesting that information present in the residual stream is not yet fully expressed in the output distribution. This interpretation is most meaningful away from the final layer: at L, the logit lens uses the model's own output head, whereas the probe remains rank-limited, so $G(L) < 0$ is expected. Accordingly, $D(\ell)$ should be interpreted as a lower bound on the information available in the representation.

Difference-in-Differences. Beyond the information scores themselves, we ask whether training changes them differently for members and non-members, comparing each group's change in scores from the base to the trained model.

Let $X(\ell, p; \theta)$ denote any of the scores above, averaged over the pairs $i \in \mathcal{I}_p$ of question p. Training from θ_0 to θ_a moves this score by $\Delta X_p(\ell) = X(\ell, p; \theta_a) - X(\ell, p; \theta_0)$. We refer to the gap between the member and non-member averages of $\Delta X_p(\ell)$ as the difference-in-differences (DiD):

$$\mathrm{DiD}[X](\ell) \;=\; \frac{1}{|\mathcal{M}|} \sum_{p \in \mathcal{M}} \Delta X_p(\ell) \;-\; \frac{1}{|\mathcal{N}|} \sum_{p \in \mathcal{N}} \Delta X_p(\ell). \tag{6}$$

Here $\mathrm{DiD}[X]$ is computed for both D and U. Since $\mathrm{DiD}[\cdot]$ is linear in X, the DiD for G can be computed as $\mathrm{DiD}[G] = \mathrm{DiD}[D] - \mathrm{DiD}[U]$. A positive $\mathrm{DiD}[D]$ indicates that training made membership-relevant information more decodable from the representations, while a positive $\mathrm{DiD}[U]$ indicates that training amplified how strongly the readout uses that information.

5 Experiments

5.1 Experimental Setup

Design. To evaluate our framework across different training methods, we consider our primary setting as post-training, where we use AIME and Olympia Math (AI-MO, 2025) with Tulu2-7B (Ivison et al., 2023) and introduce membership through controlled SFT. As a robustness check, we additionally evaluate a pre-training setting on WikiMIA-2024-Hard with Qwen2-7B (Yang et al., 2024), where membership is determined by the temporal split of the corpus. For all datasets, members are randomly sampled. We consider the following training settings to characterize how membership manifests across different forms and degrees of training exposure. Details regarding experimental settings, datasets, and choices of models are listed in App. A.1.

❶ Base model, in which membership comes from pre-training naturally using the temporal split.

❷ SFT, in which we vary the number of training epochs to control the degree of exposure.

❸ RL-based MIA Methods, where we include the Active Data Reconstruction Attack (ADRA) and ADRA+ (Yin et al., 2026). Unlike the preceding methods, which simulate the introduction of membership data during training, ADRA is an active MIA method that applies RL-based reconstruction to elicit membership information from the contaminated model. ADRA+ further incorporates a membership prior from the loss-based MIA method Min-K%++ (Shi et al., 2024). We describe how these methods are initialized and trained in App. A.2.

5.2 Findings From DUG Framework

Base Model Validation. Figure 2 shows the three information scores across layers on the base model θ_0. $D(\ell)$ is already positive in early layers and increases with depth until it saturates. In

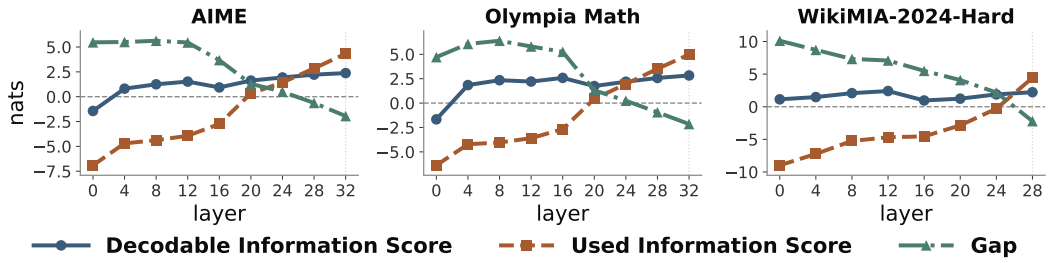


Figure 2: Results on base model.

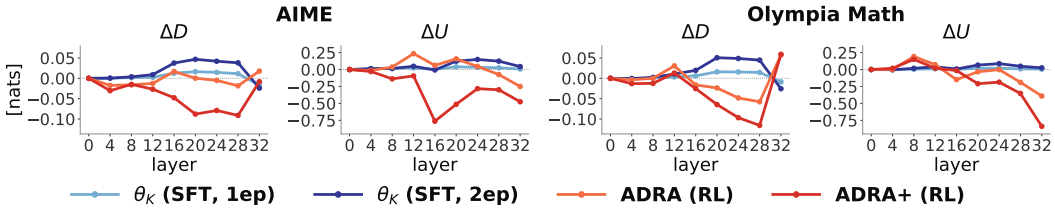


Figure 3: Change in Decodable Information and Used Information Score relative to θ_0.

contrast, $U(\ell)$ starts below the null baseline and rises with depth, suggesting that early residual representations are not yet well aligned with the unembedding space. Thus, the narrowing of $G(\ell)$ is primarily driven by $U(\ell)$ catching up with $D(\ell)$, consistent with later layers making encoded information accessible in the vocabulary space (Gurnee et al., 2026). Since early-layer $G(\ell)$ also reflects basis misalignment, we focus on its changes under training rather than its absolute value.

Information Shifts Under Training. Figure 3 compares the absolute changes in D and U from the base model θ_0 across the trained models θ_a. Here, we focus on how training changes the overall representation and defer member-specific changes to the diff-in-diff analysis.

We find that training has a limited effect on decodable information in the residual stream, with ΔD from -0.10 to $+0.05$. In contrast, ΔU ranges from approximately -0.75 to 0.25 nats, suggesting that training primarily affects how effectively the model reads out information already encoded in the residual stream. Notably, both RL-based MIA methods consistently reduce ΔU relative to the base model, reaching as low as -0.83 nats in later layers, indicating suppressed readout. In contrast, SFT induces only small positive changes in both ΔD and ΔU. Within SFT, we further find that increasing the number of epochs leads to a gradual increase in D, consistent with prior findings that repeated exposure to the same training examples can increase memorization (Carlini et al., 2023).

Difference between Member and Non-member. To examine whether training affects members and non-members differently, Figure 4 reports the Difference-in-Differences (DiD) of D, U, and G across layers. For U, we observe a clear member-specific effect. Previously, we have shown that RL-based MIA methods generally suppress the model's readout. However, $\mathrm{DiD}[U]$ reveals that this suppression is substantially weaker for members than non-members. Specifically, ADRA+ yields positive $\mathrm{DiD}[U]$ values across most layers, reaching $+0.18$ nats at the final layer on AIME and $+0.20$ nats on Olympia Math. In comparison, $\mathrm{DiD}[D]$ does not exhibit a comparable member-specific shift. *This suggests that RL-based MIA methods amplify the readout of member-specific information, while the decodable information itself shows no substantial member-specific change.* This effect persists when DiD is computed relative to the SFT initialization, results in App. A.5.2.

Meanwhile, both $\mathrm{DiD}[D]$ *and* $\mathrm{DiD}[U]$ *show larger differences when the number of epochs increases under SFT.* This demonstrates a key distinction from RL-based MIA methods, which show a form of selective exemption for members in the aspect of readout suppression. One possible explanation is that the reward encourages the reconstruction trajectories of member problems to remain closer to $y^\star$, leading to less degradation in their readout. This differential suppression of readout may contribute to the membership signal observed by MIA.

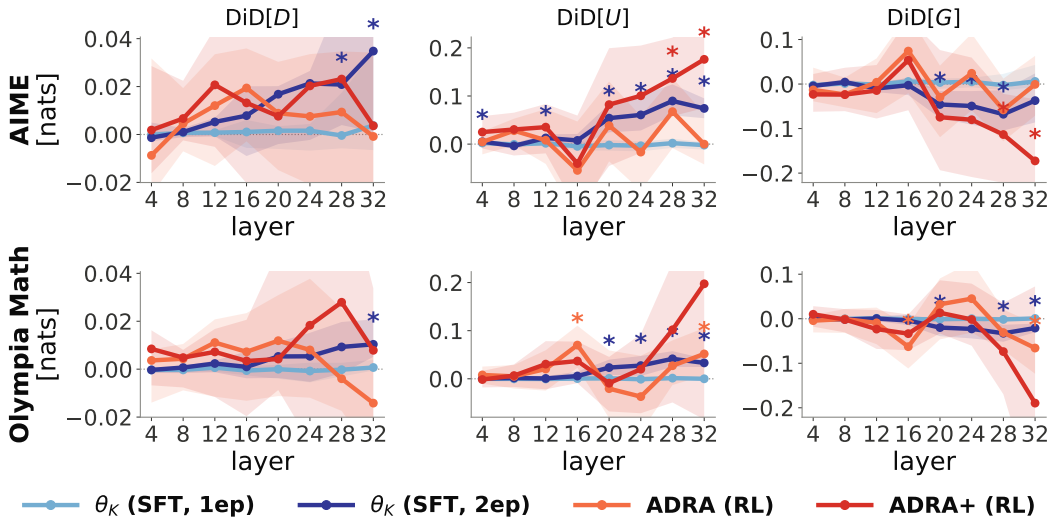


Figure 4: Diff-in-differences of membership-related changes in decodability (D), readout (U), and their gap (G) across layers. Shading shows 95% CIs and * denotes significance.

5.3 Case Study For RLVR

RL-based MIA methods take an attacker-centric perspective: given $x = (c, y^\star)$, they construct a reward function that encourages the model to reconstruct $y^\star$, eliciting latent membership signal. In practice, however, contamination takes a different form. RL in a post-training pipeline typically serves other objectives that have nothing to do with membership, e.g., improving mathematical or coding ability with verifiable rewards. In the course of pursuing these objectives, benchmark problems may end up in the RL prompt pool, thereby becoming members.

To examine whether membership still leaves traces in the representations when RL objective is independent of $y^\star$, we present a case study on RLVR contamination. Because this is a controlled comparison, we deliberately keep the reward simple rather than engineering it for accuracy. Our main goal is to observe whether membership signal is detectable under this paradigm.

Setup. We construct RLVR contamination on Qwen2-7B (θ_0) using the MATH-500 dataset (Lightman et al., 2024). Specifically, we mix 64 members into 640 GRPO training prompts (10%) and hold out 64 nonmembers. Both members and non-members are selected to satisfy $0 < \text{pass@32} < 1$ to ensure their GRPO advantages are nonzero. RLVR checkpoints at steps 10 and 20 are exposure-matched to SFT epochs 1 and 2. Further details in App. C.1.

Objective. For each prompt q, we sample $G = 32$ rollouts $\hat{y}^{(i)} \sim \pi_\theta(\cdot \mid q)$. Each rollout receives a verifiable reward:

$$r^{(i)} = \mathbf{1}\left[\text{verify}\left(\hat{y}^{(i)}, a^\star\right)\right] + \lambda\, \mathbf{1}\left[\hat{y}^{(i)} \text{ contains } \texttt{\textbackslash boxed\{\}}\right], \tag{7}$$

where verify tests symbolic equivalence with the gold answer $a^\star$ and the second term is a format bonus with $\lambda = 0.05$. The reward uses only the final answer and does not use the reference solution $y^\star$. We optimize with GRPO using group-normalized advantages $A^{(i)} = (r^{(i)} - \mu_r)/\sigma_r$.

Accuracy Comparison and Training Validity. We compare membership inference performance across the base model θ_0, SFT with one and two epochs, and RLVR at different steps. We evaluate two likelihood-based MIA scores, Loss and Min-k%, using AUROC. Table 1 summarizes the results.

We can observe that membership leaves a far stronger trace under SFT. Loss AUROC rises from 0.491 at base to 0.661 after one epoch and 0.908 after two, and Min-k% AUROC tracks it closely. Under RLVR the gain is far smaller, which we attribute to the training objective, in which the reward

Model	Loss AUROC	ΔAUROC	Min-k% AUROC	ΔAUROC
Base	0.491	–	0.492	–
SFT 1ep	0.661	+0.170 [+0.102, +0.245]	0.683	+0.191 [+0.109, +0.273]
SFT 2ep	0.908	+0.417 [+0.335, +0.508]	0.916	+0.424 [+0.337, +0.522]
RLVR (mid)	0.493	+0.002 [−0.008, +0.013]	0.490	−0.002 [−0.013, +0.008]
RLVR (later)	0.494	+0.003 [−0.009, +0.016]	0.500	+0.008 [−0.007, +0.025]

Table 1: Membership inference performance of Qwen2-7B on MATH-500 dataset. RLVR (mid) and RLVR (later) denote the intermediate and later RLVR checkpoints, respectively. ΔAUROC denotes the change in AUROC relative to the base model.

	Layer 8	Layer 12	Layer 16	Layer 20	Layer 24	Layer 28
ΔD	−0.010	−0.007	−0.014	−0.016	−0.026	−0.083
ΔU	+0.079	+0.139	+0.217	+0.195	−0.001	−0.525

Table 2: Changes in decodability (ΔD) and readout (ΔU) of RLVR relative to the base model.

never references $y^\star$, so members are never directly fit. We can nonetheless provide evidence that RLVR trains effectively: across epochs, pool accuracy rises from 54.9% to 71.8%, with no signs of reward hacking, and the updates reach the members themselves (59 of 64 received at least one nonzero-advantage update). See Appendix C.2 for details.

Experimental Findings. As shown in Table 2, RLVR induces substantial changes at the representation level. At the final layer, U shifts by -0.53 nats, against -0.13 under SFT. These shifts, however, show little member-specific bias: under SFT, deep layer $\mathrm{DiD}[D]$ is $+0.042$ and $\mathrm{DiD}[U]$ is $+0.182$, whereas under RLVR they are $+0.002$ and $+0.028$.

To examine this further at the behavioral level, we evaluate the model on the 64 member questions and 64 held-out non-member questions, sampling 32 completions per question at temperature 1.0 and averaging per-question accuracy across the 64 items. Before training, the two sets are essentially indistinguishable, with accuracies of 0.141 and 0.132. After RLVR training, member accuracy rises to 0.170 while non-member accuracy falls to 0.088, widening the gap from 0.009 to 0.082 (95% CI [+0.008, +0.159]). Membership is thus directionally visible in behavior.

Why, then, are membership signals not readily detectable? A natural explanation is that the reward does not depend on $y^\star$, so training would not be expected to change its loss. Our results suggest otherwise: under RLVR, the loss on $y^\star$ increases by 0.24 nats at step 10 and 0.56 at step 20 for members. Non-members shift by the same amount, however, leaving a member/non-member gap below 0.01 nats, compared with -0.16 nats after SFT on the same model and data. RLVR thus changes this loss without making it membership-specific, leaving likelihood-based attacks little signal to detect.

This is not a property of RL, as RL-based MIA methods show a clear member signal ($\mathrm{DiD}[U] = +0.185$ on AIME). What matters is how much the training signal depends on the member's own text. In SFT the signal comes from the cross-entropy loss, whose target is the member's own text. In RL-based MIA methods it comes from what the reward itself sees of the member, and under RLVR the reward sees only the final answer. We therefore read this case study as evidence that the scope of likelihood-based MIA is narrower than commonly assumed. *Membership is hard to detect under low-supervision training. More precisely, likelihood-based attacks are most sensitive to training that directly affects the quantity they measure.*

5.4 Why not Steer?

Analysis from Section 5.2 shows that SFT changes both decodable information and readout, whereas RL-based MIA methods act primarily on readout. This raises a practical question: can the same gain be obtained at inference time using steering vectors? However, this requires assuming that the target behavior is controlled by a single linear direction, an assumption that does not necessarily extend to readout amplification. We therefore test whether steering vectors can selectively strengthen MIA.

Dataset	Layer	DiD$[D]$	95% CI	Crosses 0
AIME	4	-0.0013	$[-0.0146, +0.0124]$	✓
AIME	8	$+0.0218$	$[+0.0073, +0.0362]$	
Olympia Math	4	$+0.0005$	$[-0.0080, +0.0092]$	✓
Olympia Math	8	-0.0045	$[-0.0122, +0.0029]$	✓

Table 3: DiD$[D]$ (θ_a vs. θ_0) at layers below the patching window. DiD$[D]$ crosses zero in 3 of 4 cells, suggesting the window amplifies readout of an existing signal, though the AIME L8 cell indicates some contribution may be from storage.

SFT. We estimate the direction as the difference in mean residual-stream activations between member and nonmember texts at layer 24, which we choose as a representative mid-to-late layer. We define bootstrap consistency as the mean pairwise cosine similarity between directions estimated from bootstrap resamples. The SFT-induced direction is recovered stably (consistency 0.96). Single-layer injection yields no significant gain, with an AUROC of 0.56 and $p = 0.07$, whereas sustained injection across layers 24–27 yields a modest gain, with AUROC increasing to 0.60 and $p = 0.01$.

RL-based MIA. We attempt the extraction for RL-based MIA methods as well, varying the subspace used to estimate the direction. We consider the full residual stream, the LoRA subspace, and the leading principal component of the training-induced shift Δh. The RL-induced direction is not recovered consistently, with bootstrap consistency ranging from 0.44 to 0.54, compared with 0.96 for its SFT counterpart. Although Δh is nearly rank-one, with 78–95% of its variance concentrated in one component (2-3.5% for random directions), this dominant component is largely shared across members and nonmembers. This suggests that the RL-induced member-specific effect does not form a stable linear direction suitable for inference-time steering. Further details in App. D.

5.5 Activation Patching

To obtain causal evidence on where the membership-specific effect of RL-based MIA is formed, we perform activation patching. For a fixed input, we run θ_0 and θ_a in parallel and swap the residual stream at the suffix positions of layer ℓ in both directions.

Let $L_{\text{out}}(y_i^\star)$ denote the model's cross-entropy on $y_i^\star$ under its full-vocabulary softmax. For any model θ, we evaluate Eq. 6 by replacing θ_a with θ and denote the result by $\text{DiD}[X](\theta)$. We use $\text{DiD}[X]$ here as shorthand for $\text{DiD}[X](\theta_a)$. Let $\theta_{\text{fwd},\ell}$ denote θ_0 with its layer ℓ residual stream at the suffix positions replaced by that of θ_a, and $\theta_{\text{rev},\ell}$ denote θ_a with its layer ℓ residual stream replaced by that of θ_0. We compute forward and reverse patching ratio as:

$$R_{\text{fwd}}(\ell) = \frac{\text{DiD}[-L_{\text{out}}](\theta_{\text{fwd},\ell})}{\text{DiD}[-L_{\text{out}}]}, \qquad R_{\text{rev}}(\ell) = \frac{\text{DiD}[-L_{\text{out}}](\theta_{\text{rev},\ell})}{\text{DiD}[-L_{\text{out}}]}. \tag{8}$$

Therefore, a positive $R_{\text{fwd}}(\ell) - R_{\text{rev}}(\ell)$ indicates that the effect is already present by layer ℓ, while a negative value suggests that it emerges at later layers. We find that at $\ell = 8$, $R_{\text{fwd}} - R_{\text{rev}}$ is significantly negative on both AIME ($[-1.410, -0.481]$) and Olympia Math ($[-1.201, -0.627]$), suggesting that the effect is not yet present. DiD$[D]$ below this window is reported in Table 3. At $\ell = 20$, it is significantly positive on Olympia Math ($[+0.307, +0.931]$), whereas the AIME estimate is inconclusive ($[-3.036, +0.594]$) because its denominator is close to zero. These results suggest that the effect emerges between layers 8 and 20 on Olympia Math, and after layer 8 on AIME.

6 Discussion and Conclusion

Our results suggest that membership signals depend not only on what is represented but also on how it is read out. SFT changes both decodable information and readout, whereas RL-based MIA methods act primarily on readout. Under our RLVR case study, behavioral changes do not necessarily translate into membership-specific likelihood signals when the reward is independent of the reference solution, leaving open whether similar patterns arise for non-likelihood-based MIA methods.

AI USE STATEMENT

In this work, we used generative AI tools for polishing writing, specifically to improve readability and language; retrieval and discovery, specifically to locate sources for a small number of related works; research ideation or execution, specifically for feedback on experiments, coding and figure generation; and drafting sections of the paper, specifically for summarizing experimental settings and data in the appendix. We have not used generative AI tools for discovering research topics, identifying research gaps, formulating questions for literature surveys, summarizing existing literature, developing theoretical models or conceptual frameworks. Tasks of generating synthetic datasets, formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting with writing proofs are not applicable to this work.

We have reviewed all AI-assisted work. AI-edited code was verified and tested, and AI-assisted writing, descriptions of experimental settings, and extracted related-work content were manually reviewed for accuracy and consistency with the cited sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

AI-MO. Ai-mo olympiad reference dataset. Hugging Face Dataset, 2025. URL https://huggingface.co/datasets/AI-MO/olympiads-ref.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens, 2025. URL https://arxiv.org/abs/2303.08112.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models, 2021. URL https://arxiv.org/abs/2012.07805.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models, 2023. URL https://arxiv.org/abs/2202.07646.

Bowen Chen, Namgi Han, and Yusuke Miyao. A comparative analysis of llm memorization at statistical and internal levels: Cross-model commonalities and model-specific signatures, 2026. URL https://arxiv.org/abs/2603.21658.

Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models, 2025. URL https://arxiv.org/abs/2406.16201.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models?, 2024. URL https://arxiv.org/abs/2402.07841.

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Mia-tuner: Adapting large language models as pre-training text detector, 2024. URL https://arxiv.org/abs/2408.08661.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL https://transformer-circuits.pub/2026/workspace/index.html.

Minju Gwak, Minseo Kwak, Dongseok Lee, Guijin Son, Alan Ritter, and Jaehyung Kim. Lara: Layer-wise representation analysis for detecting data contamination in rl post-training, 2026. URL https://arxiv.org/abs/2605.29888.

Skyler Hallinan, Jaehun Jung, Melanie Sclar, Ximing Lu, Abhilasha Ravichander, Sahana Ramnath, Yejin Choi, Sai Praneeth Karimireddy, Niloofar Mireshghallah, and Xiang Ren. The surprising effectiveness of membership inference with simple n-gram coverage, 2026. URL https://arxiv.org/abs/2508.09603.

Hamish Ivison, Yizhong Wang, Valentina Pyatkin, Nathan Lambert, Matthew Peters, Pradeep Dasigi, Joel Jang, David Wadden, Noah A. Smith, Iz Beltagy, and Hannaneh Hajishirzi. Camels in a changing climate: Enhancing lm adaptation with tulu 2, 2023. URL https://arxiv.org/abs/2311.10702.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. Investigating data contamination for pre-training language models, 2024. URL https://arxiv.org/abs/2401.06059.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8424–8445, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.577. URL https://aclanthology.org/2022.acl-long.577/.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

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, 2026. URL https://arxiv.org/abs/2511.14045.

Inbal Magar and Roy Schwartz. Data contamination: From memorization to exploitation, 2022. URL https://arxiv.org/abs/2203.08242.

Disha Makhija, Manoj Ghuhan Arivazhagan, Vinayshekhar Bannihatti Kumar, and Rashmi Gangadharaiah. Neural breadcrumbs: Membership inference attacks on LLMs through hidden state and attention pattern analysis. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5608–5620, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-380-7. doi: 10.18653/v1/2026.eacl-long.262. URL https://aclanthology.org/2026.eacl-long.262/.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KS8mIvetg2.

Akshara Prabhakar, Thomas L. Griffiths, and R. Thomas McCoy. Deciphering the factors influencing the efficacy of chain-of-thought: Probability, memorization, and noisy reasoning. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 3710–3724, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.212. URL https://aclanthology.org/2024.findings-emnlp.212/.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2024. URL https://arxiv.org/abs/2310.16789.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models, 2017. URL https://arxiv.org/abs/1610.05820.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje F. Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Moura, Dominik Krzemiński, Hakimeh Fadaei, Irem Ergun, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Vu Chien, Sebastian Ruder, Surya Guthikonda, Emad Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. Aya dataset: An open-access collection for multilingual instruction tuning. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11521–11567, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.620. URL https://aclanthology.org/2024.acl-long.620/.

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Nikul Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. Layer by layer: Uncovering hidden representations in language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WGXb7UdvTX.

Niklas Stoehr, Mitchell Gordon, Chiyuan Zhang, and Owen Lewis. Localizing paragraph memorization in language models, 2024. URL https://arxiv.org/abs/2403.19851.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. Improving instruction-following in language models through activation steering, 2025. URL https://arxiv.org/abs/2410.12877.

Jiashu Tao and Reza Shokri. (token-level) informia: Stronger membership inference and memorization assessment for llms, 2025. URL https://arxiv.org/abs/2510.05582.

Yongding Tao, Tian Wang, Yihong Dong, Huanyu Liu, Kechi Zhang, Xiaolong Hu, and Ge Li. Detecting data contamination from reinforcement learning post-training for large language models, 2025. URL https://arxiv.org/abs/2510.09259.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IQxBDLmVpT.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, Yanwei Fu, Qin Liu, Songyang Zhang, and Qi Zhang. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination, 2025. URL https://arxiv.org/abs/2507.10532.

Chulin Xie, Yangsibo Huang, Chiyuan Zhang, Da Yu, Xinyun Chen, Bill Yuchen Lin, Bo Li, Badih Ghazi, and Ravi Kumar. On memorization of large language models in logical reasoning. In Kentaro Inui, Sakriani Sakti, Haofen Wang, Derek F. Wong, Pushpak Bhattacharyya, Biplab Banerjee, Asif Ekbal, Tanmoy Chakraborty, and Dhirendra Pratap Singh (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 2742–2785, Mumbai, India, December 2025. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics. ISBN 979-8-89176-298-5. doi: 10.18653/v1/2025.ijcnlp-long.148. URL https://aclanthology.org/2025.ijcnlp-long.148/.

Lecheng Yan, Ruizhe Li, Guanhua Chen, Qing Li, Jiahui Geng, Wenxi Li, Longyue Wang, and Chenyang Lyu. Spurious rewards paradox: Mechanistically understanding how rlvr activates memorization shortcuts in llms, 2026. URL https://arxiv.org/abs/2601.11061.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang,

Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024. URL https://arxiv.org/abs/2407.10671.

Junjie Oscar Yin, John X. Morris, Vitaly Shmatikov, Sewon Min, and Hannaneh Hajishirzi. Learning to detect language model training data via active reconstruction, 2026. URL https://arxiv.org/abs/2602.19020.

Hongbo Zhang, Yue Yang, Jianhao Yan, Guangsheng Bao, Yue Zhang, and Yue Zhang. Detecting rlvr training data via structural convergence of reasoning, 2026. URL https://arxiv.org/abs/2602.11792.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. Wildchat: 1m chatgpt interaction logs in the wild, 2024. URL https://arxiv.org/abs/2405.01470.

A DUG EXPERIMENTS

A.1 DATASET AND MODEL

Dataset. For the controlled-contamination setting, we use two math datasets, AIME and Olympia Math. For AIME, we collect problems and solutions from 2021–2025. Since each year contains only 30 problems, this gives 120 member examples and 30 non-member examples. For each seed, we draw 32 examples, with 16 from each group. For Olympia Math, we divide the problems into member and non-member groups using a random partition, and draw 64 examples per seed, with 32 from each group. For WikiMIA, members and non-members are defined by a temporal cutoff. From each group, we draw 64 examples for evaluation. All examples used as members are randomly selected from their respective member pools.

Our evaluation sets are intentionally small, reflecting a light-contamination setting we target, where only a limited number of benchmark items leak into training. This mirrors how contamination typically arises in practice. It also follows controlled-contamination protocols in prior work, which inject a limited number of test examples into continued pre-training or fine-tuning to study their detectability (Shi et al., 2024; Jiang et al., 2024; Oren et al., 2024). Heavier contamination would shift the focus of our analysis.

Choice of Models. A controlled study requires us to know whether exposure to the evaluation data comes from our training (Duan et al., 2024). This is difficult to establish for recent models, whose training corpora are not fully disclosed and may already contain widely circulated benchmarks (Wu et al., 2025). We therefore use earlier base models and measure membership effects relative to the base model, so that prior exposure shared by members and non-members is accounted for.

A.2 EXPLANATION ON RL-BASED MIA METHODS

Throughout, θ_0 denotes the base model before any of our training. RL-based MIA methods are optimized on the same members and non-members as the other settings. In the post-training setting (AIME and Olympia Math), they are initialized from the SFT model after one epoch, so that membership is introduced before the attack; in the pre-training setting (WikiMIA-2024-Hard), they are initialized from θ_0. All DiD values are computed relative to θ_0. Since DiD is linear in the trained model's scores, the difference between the RL-based MIA methods and SFT (1 ep) curves equals the DiD of RL-based MIA methods relative to their initialization, i.e., the effect of RL alone.

A.3 FRAMEWORK DETAILS

Output vocabulary. For probing, predicting over the full vocabulary would make the probe's output head dominate its parameter count, especially for Qwen2-7B ($|\mathcal{V}| = 152{,}064$). We therefore restrict the output space to the tokens observed in each dataset's suffix pool plus a single UNK class: 1,328 classes for AIME and 2,630 for Olympia Math (both on tulu-2-7b, $|\mathcal{V}| = 32{,}000$), and 6,005 for WikiMIA-2024-Hard. The same restriction is applied to the null baseline and to the logit lens.

Hence the probe, the logit lens, and the null baseline are all evaluated on the same output space, and $D(\ell)$, $U(\ell)$, and $G(\ell)$ are directly comparable.

Probe capacity. At each layer, we fit a linear probe. For AIME and Olympia Math, we use full rank probe. For WikiMIA-2024-Hard, a full-rank probe would be heavily over-parameterized relative to the available training samples, so we cap the rank at $r = 256$. This cap is held fixed across all layers within this dataset, so the layerwise profiles remain internally comparable.

Controls and inference. Probes are fit using 5-fold GroupKFold cross-validation, with problems as groups. For each layer, the weight decay is selected from $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}$ using a shuffle-label control, choosing the smallest value for which the shuffled-target probe yields $|D_{\text{shuffle}}| \leq 0.05 H_{\text{pool}}$. This threshold corresponds to 0.270, 0.295, and 0.338 nats for AIME, Olympia Math, and WikiMIA, respectively. Confidence intervals and DiD tests are computed from 1,000 problem-level bootstrap samples, with members and non-members resampled independently.

A.4 Binary Membership Detection

Following our preliminary experiments as in Sec. 3, we also examine whether a binary probe can detect membership-related changes in the trained models. Specifically, we fit a logistic regression on mean- and last-token residual-stream features at each layer, after PCA ($k = 30$) and residualization on suffix length, using Olympia Math member/non-member splits of Sec. 5.1, and compare against a shuffled-label control and the loss-based MIA AUROC of the same models.

Across layers, probe AUROC on θ_0 ranges from 0.29 to 0.55, while the shuffled-label control remains near 0.50. For SFT, the probe yields AUROC of 0.30–0.55, with no clear improvement over θ_0. At this scale, SFT produces a modest increase in loss-based MIA performance (AUROC 0.61), while the token-level measures of DUG reveal membership-specific changes (e.g., $\mathrm{DiD}[U]$ at late layers, as in Figure 4). These results suggest that per-example AUROC may be less sensitive to membership effects at this sample scale, motivating our use of token-level measures. We therefore do not use the binary probe to draw conclusions about RL-based MIA methods.

A.5 Further Results

A.5.1 Results on WikiMIA-2024-Hard.

Figure 5 reports the difference-in-difference estimates on WikiMIA-2024-Hard. SFT reproduces the main-text pattern, in which both $\mathrm{DiD}[D]$ and $\mathrm{DiD}[U]$ grow with epochs and are significant at most layers. RL-based MIA methods show the same readout-specific effect, with the asterisks in the figure indicating statistical significance, though in a weaker form.

We attribute the weaker RL effects to two factors. First, RL here starts from pre-training exposure, where the membership-specific signal appears relatively difficult to further amplify. Our preliminary results likewise suggest that signals established during the pre-training stage are less amenable to amplification by subsequent training. Second, ADRA+ gates the ground-truth suffix with a Min-K%++ prior whose standalone AUROC is 50.6 on WikiMIA-2024-Hard, compared with 78.3 on AIME and 60.2 on Olympia Math (Yin et al., 2026). At chance level, the gate is unrelated to membership, so no member-specific effect accumulates. ADRA+ effect sizes follow the same ordering as the prior AUROC, though three datasets cannot establish this causally.

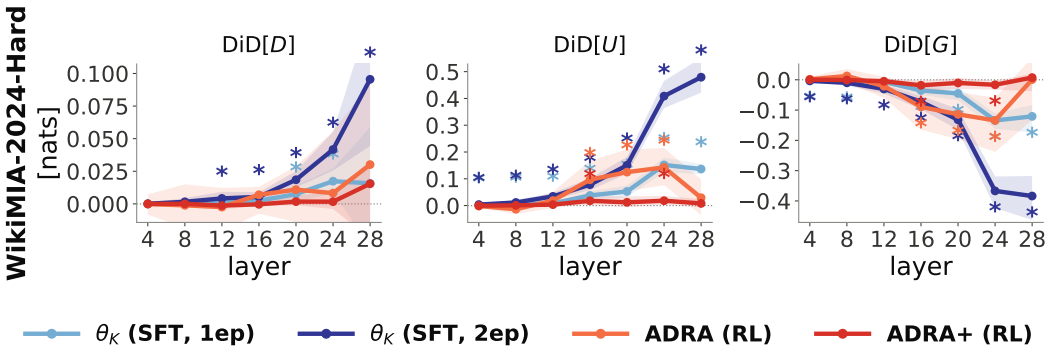


Figure 5: Diff-in-differences of membership-related changes in decodability (D), readout (U), and their gap (G) across layers.

A.5.2 DiD relative to the SFT 1ep.

In the post-training setting, RL-based MIA methods are initialized from the SFT (1 ep) model, so their DiD relative to θ_0 also includes the effect of SFT. We already report the DiD for SFT (1 ep) in the main text for reference. For completeness, we additionally compute Eq. 6 using the SFT (1 ep) model in place of θ_0 on AIME and Olympia Math.

As shown in Figure 6, $\mathrm{DiD}[U]$ remains positive at mid-to-late layers, reaching $+0.18$ nats on AIME and $+0.20$ nats on Olympia Math, whereas $\mathrm{DiD}[D]$ stays very close to zero. This is consistent with

Section 5.2 and supports our conclusion that RL-based MIA methods amplify membership-specific effects primarily through readout rather than through decodable information.

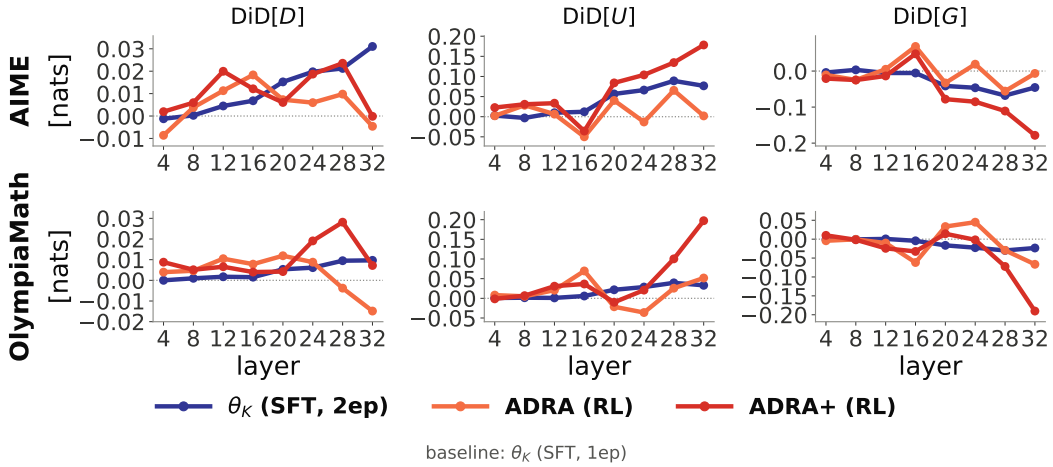


Figure 6: Further results on membership-specific changes. DiD scores for D and U using SFT (1 ep) as the reference model.

B Why Not a Tuned Lens?

We also implemented a tuned lens (Belrose et al., 2025), which learns an affine translator (A_ℓ, c_ℓ) per layer by minimizing the KL divergence to the final-layer distribution p_{final}, using the same folds as the probe. We did not adopt it for two reasons.

The translator absorbs later computation. Because A_ℓ is trained to reproduce p_{final}, the tuned lens measures how well layer ℓ can be mapped onto the model's final prediction, rather than what the model reads out at layer ℓ. Empirically, U_{tuned} exceeds D at all layers, yielding a negative G throughout.

Trainable readouts can introduce membership differences. The bias c_ℓ learns a fold-level constant related to the mean of p_{final}. Because members and non-members differ in final-layer confidence, this constant can induce a member/non-member difference in U_{tuned}. The difference is largest at layer 0 (0.196 nats), where the representation contains little membership information, and decreases to 0.005 nats at the final layer, close to the logit-lens value of 0.004. This suggests that trainable readouts may introduce membership differences beyond those reflected in the model's own output head, motivating the use of the logit lens for U.

C RLVR Case Study

C.1 Experimental Settings

Members and nonmembers are both drawn from MATH-500 problems with $0 < \text{pass@32} < 1$, ensuring GRPO's group-normalized advantage is nonzero (Shao et al., 2024). The 64 members are mixed into 576 problems from the training set, yielding 640 prompts; the nonmembers never enter training at any stage. Each record retains only the problem statement and the final answer. With 64 prompts per optimizer step, ten steps form one pool epoch in which every member appears exactly once, so RLVR at steps 10 and 20 is exposure-matched to SFT at 1 and 2 epochs. Training ran until policy entropy fell below half its step-1 value.

C.2 Supplementary Training Validity

Mean pool accuracy rises from 54.9% in epoch 1 to 71.8% in epoch 2, with per-step accuracy increasing from 56.5% to 83.4%. Mean reward rises correspondingly from 0.613 to 0.884, while reward variance and the fraction of groups with non-zero advantage decrease, consistent with the policy approaching the task ceiling.

C.3 Future Directions

Our analysis of the RLVR case study focuses on likelihood-based signals. Evaluating non-likelihood-based detection methods is orthogonal to our question and remains a direction for future work, along with broader training objectives.

D Additional Details on Activation Steering

SFT. For bootstrap consistency, we use 100 resamples of the 32 construction pairs. The resulting directions have a mean pairwise cosine similarity of 0.96.

To examine the limited effect of single-layer injection, we measure how much of the injected component remains at the final layer. The corresponding measure is 0.05, compared with 0.75 for random directions, indicating that the injected component is largely weakened through the remaining layers.

For sustained injection, we evaluate the intervention on 183 held-out pairs. Reversing the injection direction yields chance-level performance.