LaRA: Layer-wise Representation Analyses for Detecting Data Contamination in RL Post-Training

Anonymous Authors[1]

Abstract

Reinforcement learning (RL) improves reasoning in large language models (LLMs) but can also induce contamination through reward-driven memorization. Existing contamination detection methods mainly rely on output-level signals, such as likelihood or entropy, which become unstable after RL post-training. We propose LaRA, a layer-wise representation analysis framework for contamination detection in RL-trained LLMs. LaRA introduces three complementary metrics—Representation Shift Magnitude (RSM), Directional Collapse (DC), and Representation Stability Index (RSI)—to measure perturbation sensitivity, directional concentration, and local representation variability under controlled perturbations. Our analyses show that contamination induces progressive geometric deviations across layers, characterized by amplified perturbation sensitivity, abnormal directional concentration dynamics, and unstable local variability. Based on these observations, we develop a layer-aware detection protocol that aggregates representation-level deviations across layers and metrics. Experiments on RL-trained reasoning models show that LaRA consistently improves contamination detection over existing output-level baselines.

1. Introduction

Large Language Models (LLMs) trained with reinforcement learning (RL) have demonstrated strong performance in complex reasoning tasks (Guo et al., 2025; Guha et al., 2025; Li et al., 2025b; Hochlehnert et al., 2025). However, this training paradigm introduces a critical but underexplored issue: data contamination during RL post-training (Tao et al., 2025; Wang et al., 2025; Wu et al., 2026). In this setting, a model may implicitly overfit to specific training samples through reward-driven updates, raising concerns about memorization, generalization, and evaluation reliability.

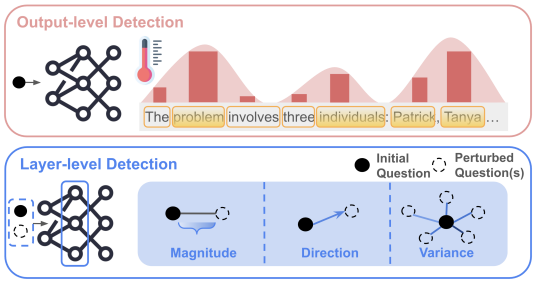


Figure 1. **Comparison of output vs. layer-level signals.** Output-level signals are sensitive to decoding and confounded by policy collapse, making them unreliable. Instead, layer-level representation geometry provides an improved robustness and interpretability when detecting contamination.

A natural way to formalize contamination detection is through the lens of membership inference attacks (MIA) (Wu & Cao, 2025), where the goal is to determine whether a given sample was included in the training data. While MIA has been extensively studied in supervised settings (Zhang et al., 2024; Shi et al., 2023; Xie et al., 2024), its application to RL remains non-trivial. Unlike pre-training or supervised fine-tuning (SFT), where memorization manifests as elevated likelihood or reduced perplexity (Gonen et al., 2023), RL modifies model behavior through reward-weighted policy updates, leading to fundamentally different signatures of contamination.

Recent work attempts to detect RL contamination using output-level signals, particularly entropy or divergence between reasoning trajectories (Tao et al., 2025). These approaches rely on the observation that contaminated samples often exhibit reduced behavioral deviation, such as similar reasoning paths between initial and critique stages. However, such signals suffer from key limitations. First, they are highly sensitive to decoding hyperparameters (*e.g.*, temperature, top-k), making them unstable. Second, entropy reduction can arise from general policy collapse rather than true memorization, even when trained on clean data (Dong et al., 2025). Third, existing methods do not verify whether RL training has sufficiently induced memorization, raising

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

concerns about the validity of detected signals.

These limitations reveal a deeper issue: objective-metric misalignment. RL optimizes expected reward, not token likelihood or entropy, making output-level metrics an indirect and potentially misleading proxy for contamination. This motivates a shift from output-based analysis to representation-level analysis, where training-induced changes may be more directly reflected.

Here, we propose a new framework for detecting data contamination in RL post-training via representation stiffness. *Our key hypothesis is that RL-induced memorization produces abnormal representation responses under controlled perturbations: locally insensitive to semantic variation, yet highly reactive to the removal of memorized information.* Thus, we introduce LaRA, a layer-wise representation perturbation framework, where we systematically remove critical information from inputs and measure how representations change relative to structurally similar samples.

Concretely, we construct structural control groups of semantically similar questions, apply consistent information removal, and analyze representation shifts across layers. We define three complementary metrics: (1) Representation Shift Magnitude (RSM) to measure sensitivity to perturbations, (2) Directional Collapse (DC) to capture alignment toward dominant transformation directions, and (3) Representation Stiffness Index (RSI) to quantify local invariance under small perturbations. Together, these provide complementary geometric signatures of contamination.

In summary, our contributions are as follows:

- We propose a novel representation-level framework as well as training and evaluation setup for detecting contamination via stiffness and rigidity.
- We additionally introduce a contamination-detection protocol and compare against output-level baselines.
- We provide empirical insights into how RL training affects representation geometry across layers.

2. LaRA: Layerwise Representation Analyses

2.1. Problem Definition

Membership Inference Attack. We consider the task of data contamination in the RL post-training phase of LLMs. Formally, this can be framed as a Membership Inference Attack (MIA) problem: given a model $\mathcal{M}$ that has undergone RL post-training and a sample x, the goal is to determine membership, where 1 indicates a member in the RL training dataset $\mathcal{D}_{\mathcal{RL}}$. A detector is a function $\mathcal{F}(M, x) \to \{0, 1\}$, where 1 indicates membership (contamination) and 0 indicates non-membership. The central question guiding our analyses is: *do layer-wise signals behave differently for member vs. non-member samples?* Given this, we introduce three metrics for further investigation below.

2.2. Experiment Setup

Contamination Dataset Curation. We construct a contamination evaluation set using open-source RL-trained models and their corresponding training data sources. Specifically, we use `Eurus-2-7B-PRIME` (Cui et al., 2025), which is trained on `Qwen2.5-Math-7B` (Yang et al., 2024) as the initial model. The corresponding dataset used during RL-training (`PRIME-RL/Eurus-2-RL-Data`) is opted to curate the contamination detection dataset as well; we curate a balanced dataset of 60 samples, consisting of 30 member and 30 non-member instances. Member samples are drawn from Olympiad-level mathematics problems among `PRIME-RL/Eurus-2-RL-Data`, while we use AIME 2026 (Balunović et al., 2025) problems as non-member problems. This results in a contamination detection dataset for controlled analysis of contamination effects while preserving comparable difficulty across splits. Refer to Appendix A.4 for details.

Training Dataset Curation. We construct two training datasets to analyze how contamination detection performance evolves under RL training. Each dataset contains samples with varying contamination signals by incorporating member instances from the Contamination Dataset with different exposure levels (*e.g.*, single and repeated occurrences). To ensure sufficient scale and diversity, we augment these with approximately 1K additional samples from the RL-MIA (Tao et al., 2025) Math dataset, including Olympiad-level problems. This results in training corpora that combine controlled exposure variations with diverse reasoning examples, enabling robust learning of contamination signals while preserving generalization. In the main results, we report performance using the dataset with single-occurrence members. Refer to Appendix A.4 for details.

2.3. Three Metrics for Representation Analyses

Metric 1: Representation Shift Magnitude. To quantify how strongly a model's internal representation responds to the removal of important information, we introduce Representation Shift Magnitude (RSM). Given an original question q_0, we construct a set of semantically similar questions

$$\mathcal{Q} = \{q_0, q_1, \ldots, q_K\},$$

where K denotes the number of generated semantic neighbors excluding the original question. For each question $q_i \in \mathcal{Q}$, we apply an importance-based blanking operator BLANKIMPORTANT that removes key information spans while preserving the overall question structure:

$$q_i^- \leftarrow \text{BLANKIMPORTANT}(q_i, k),$$

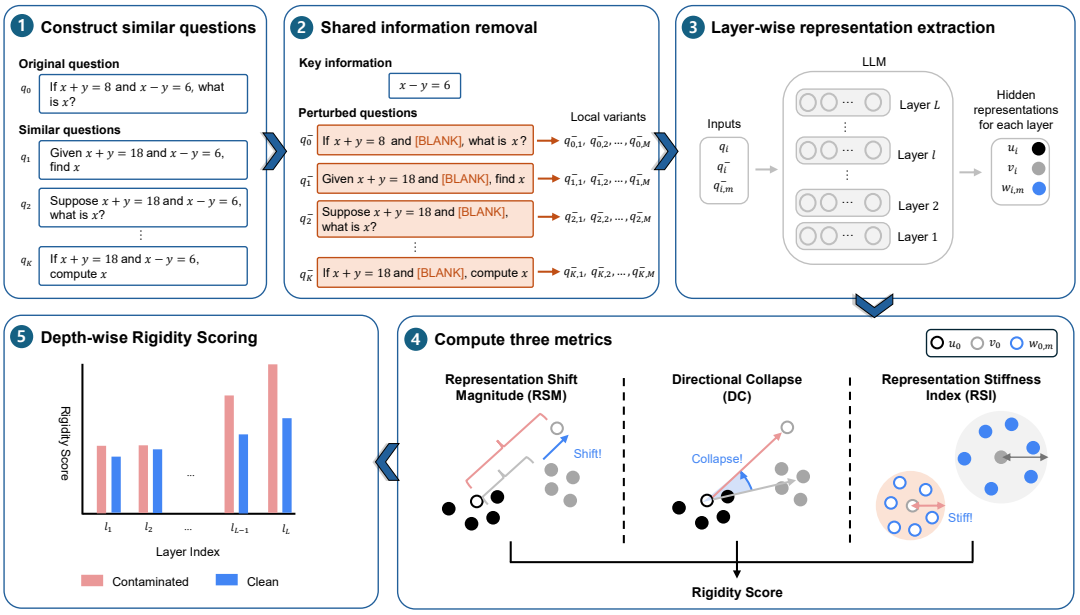


Figure 2. **Overview of LaRA, the proposed layer-wise representation geometry analysis framework.** Given an original question q_0, we generate semantically similar questions and remove shared key information to construct perturbed variants. Hidden representations are extracted across all transformer layers for original, perturbed, and paraphrased inputs. We then compute three complementary geometric metrics: Representation Shift Magnitude (RSM), Directional Collapse (DC), and Representation Stability Index (RSI), which characterize perturbation sensitivity, directional organization, and local representation variability under controlled perturbations.

where k denotes the number of inserted `[BLANK]` tokens. We additionally generate paraphrastic variants of each perturbed question while preserving the blank positions:

$$\{v_{i,1}, \ldots, v_{i,M}\} \sim \textsc{VariantGen}(q_i^-).$$

Refer to Appendix A.3 for details of the perturbation construction process.

Let $h_\ell(\cdot)$ denote the mean-pooled hidden representation extracted from transformer layer ℓ, where $\ell \in \mathcal{L} = \{0, 1, \ldots, L-1\}$. For each layer ℓ, we extract hidden representations:

$$u_i = h_\ell(q_i), \qquad w_i = h_\ell(q_i^-),$$

where $u_i, w_i \in \mathbb{R}^d$ and d is the hidden representation dimension.

We then compute the perturbation-induced representation shift:

$$\Delta_i = u_i - w_i,$$

and define its magnitude as:

$$S_i = \|\Delta_i\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm.

To measure how unusually large the original question's perturbation response is relative to its semantic neighbors, we standardize the original shift magnitude using statistics computed from the similar-question set:

$$zRSM_\ell = \frac{S_0 - \mu_S}{\sigma_S + \epsilon},$$

where

$$\mu_S = \frac{1}{K}\sum_{i=1}^{K} S_i, \qquad \sigma_S = \sqrt{\frac{1}{K-1}\sum_{i=1}^{K}(S_i - \mu_S)^2}.$$

Here, $\epsilon > 0$ is a numerical stability constant. A high $zRSM_\ell$ indicates that the original question exhibits a substantially larger representation shift under information removal compared to semantically similar questions, suggesting stronger perturbation sensitivity.

Metric 2: Directional Collapse. We next introduce Directional Collapse (DC) to characterize the directional organization of perturbation-induced representation changes. We first compute the average perturbation direction across semantically similar questions:

$$\bar{s}_\ell = \frac{1}{K}\sum_{i=1}^{K} \Delta_i,$$

where $\bar{s}_\ell \in \mathbb{R}^d$ represents the dominant perturbation direction shared across the semantic group.

Directional Collapse is then defined as:

$$DC_\ell = \frac{\Delta_0^\top \bar{s}_\ell}{(\|\Delta_0\|_2 + \epsilon)(\|\bar{s}_\ell\|_2 + \epsilon)}.$$

This quantity measures the cosine alignment between the original perturbation direction and the average perturbation direction of semantically similar questions. High DC_ℓ values indicate that perturbation responses are strongly aligned along a shared low-dimensional direction, whereas lower values indicate more distributed or heterogeneous perturbation dynamics.

Metric 3: Representation Stability Index. Finally, we measure local representation stability under semantically preserving perturbations through the Representation Stability Index (RSI). For each perturbed question q_i^-, we generate M paraphrastic variants while preserving the blank positions:

$$\{v_{i,1}, \ldots, v_{i,M}\} \sim \textsc{VariantGen}(q_i^-).$$

We then extract their hidden representations:

$$\phi_{i,m} = h_\ell(v_{i,m}),$$

where $\phi_{i,m} \in \mathbb{R}^d$.

Next, we compute the local representation centroid:

$$\bar{\phi}_i = \frac{1}{M} \sum_{m=1}^{M} \phi_{i,m},$$

and define the average local representation deviation:

$$R_i = \frac{1}{M} \sum_{m=1}^{M} \|\phi_{i,m} - \bar{\phi}_i\|_2.$$

We then standardize the original question's local variability relative to its semantic neighbors:

$$zRSI_\ell = \frac{R_0 - \mu_R}{\sigma_R + \epsilon},$$

where

$$\mu_R = \frac{1}{K} \sum_{i=1}^{K} R_i, \qquad \sigma_R = \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (R_i - \mu_R)^2}.$$

A high $zRSI_\ell$ indicates that the original question exhibits larger local representation variability relative to semantically similar questions under paraphrastic perturbations, while lower values indicate more locally stable representation behavior.

3. Layer-wise Representation Geometry under RL Contamination

To better understand how RL contamination alters internal computation, we analyze the layer-wise geometry of hidden representations under controlled information-removal perturbations. Specifically, we analyze three complementary representation metrics: (1) zRSM, which measures perturbation sensitivity relative to semantically matched controls, (2) DC, which captures directional concentration of perturbation-induced representation trajectories, and (3) zRSI, which quantifies local invariance under semantically preserving perturbations. Together, they characterize how contamination reshapes representation geometry throughout RL post-training.

3.1. Perturbation Sensitivity Analyses

We first analyze perturbation sensitivity using zRSM, which measures how strongly hidden representations respond to the removal of critical information relative to semantically matched controls. As shown in Figure 3(a), contaminated samples exhibit substantially larger perturbation-induced representation shifts than clean samples.

For `Eurus-2-7B-PRIME`, contaminated samples show a sharp increase in zRSM after the earliest layers and maintain consistently elevated sensitivity across most middle-to-late layers, whereas clean samples remain nearly flat throughout depth. For `LIMR`, the same overall separation is observed: contaminated samples remain above clean samples across most layers, despite a sharp localized drop around an early-to-middle layer. Thus, across both models, contaminated inputs are more sensitive to targeted information removal than clean inputs. Results suggest that contaminated samples occupy representation regions that are less robust to controlled perturbations.

3.2. Directional Concentration Analyses

We next analyze DC to examine whether perturbation-induced shifts are distributed across diverse directions or concentrated along group-level directions. Figure 4(b) shows that contaminated and clean samples exhibit clearly different directional dynamics, but the pattern is not a simple uniform increase for contaminated samples across layers.

In `Eurus-2-7B-PRIME`, contaminated samples start with higher directional concentration in the earliest layers, then undergo a sharp collapse around early layers before gradually increasing again in later layers. Clean samples follow a similar late-layer increase, but with weaker early-layer concentration and a different recovery profile. In `LIMR`, the contrast is stronger: clean samples maintain relatively high and stable directional concentration across depth, whereas contaminated samples sharply collapse after the first few

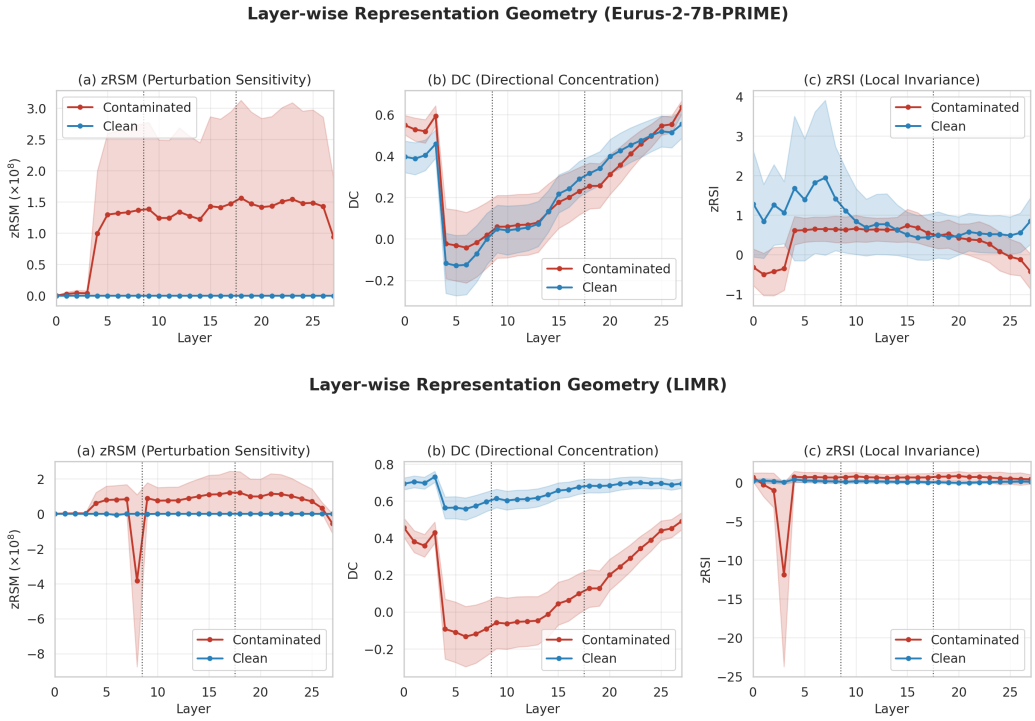


Figure 3. **Layer-wise representation geometry patterns between contaminated and clean samples across RL-trained models.** Across both models, contaminated samples consistently exhibit stronger perturbation sensitivity (zRSM), abnormal directional concentration dynamics (DC), and altered local representation variability patterns (zRSI) compared to clean samples.

layers and only gradually recover toward later layers.

Observations indicate that contamination affects not only the magnitude of representation shifts, but also their geometric organization. Contaminated samples show abnormal directional concentration dynamics, especially early-layer collapse followed by late-layer recovery. However, because clean samples can also exhibit increasing directional concentration in later layers, DC should not be interpreted as a standalone detector. Instead, it is most informative when combined with perturbation sensitivity and local invariance.

3.3. Local Variability Analyses

Finally, we analyze local representation behavior using zRSI, which measures the variability of hidden representations across paraphrased variants of perturbed questions. Unlike zRSM, which captures the magnitude of perturbation-induced shifts, zRSI characterizes how stable local representation neighborhoods remain under semantically preserving perturbations.

Figure 3(c) shows that contaminated and clean samples exhibit substantially different local variability patterns across layers. In Eurus-2-7B-PRIME, clean samples display relatively stable and gradually decreasing zRSI trajectories throughout depth, whereas contaminated samples exhibit flatter and more irregular dynamics across layers. In LIMR, contaminated samples show a sharp deviation in early layers before stabilizing in later layers, while clean samples remain comparatively stable throughout depth.

These results suggest that contamination alters the local organization of representation neighborhoods under paraphrastic perturbations. While clean samples tend to preserve relatively consistent local representation behavior across layers, contaminated samples exhibit less stable local variability patterns, particularly in early-to-middle layers. This indicates that RL contamination affects not only the magnitude and direction of perturbation-induced representation shifts, but also the local geometric structure surrounding perturbed representations.

3.4. Cross-model and Cross-epoch Stability

Although the precise layer-wise trajectories differ across model families and RL checkpoints, Figures 3 and 4 reveal several stable patterns. First, $z\mathrm{RSM}$ is the most consistent separation signal: contaminated samples show substantially larger perturbation-induced representation shifts than clean samples across both `Eurus-2-7B-PRIME` and LIMR, and this separation remains visible across RL training stages. In `Eurus-2-7B-PRIME`, RL training further

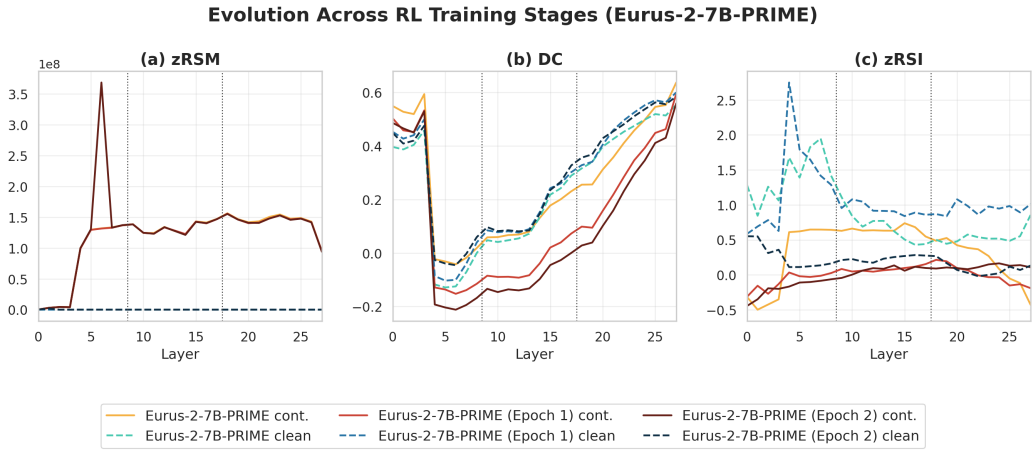


Figure 4. **Evolution of layer-wise representation geometry across RL training stages.** RL progressively amplifies contamination-associated geometric deviations, including elevated perturbation sensitivity, directional collapse dynamics, and altered local variability behavior.

amplifies this contaminated-only elevation, while clean samples remain close to zero across layers.

Second, DC shows that contamination changes the organization of perturbation trajectories, but this signal is less straightforward than zRSM. Across both models, contaminated samples exhibit abnormal directional dynamics, including early-to-middle layer collapse followed by later recovery. However, Figure 4 shows that RL training also increases directional alignment for clean samples in later layers, indicating that DC is partially confounded by global RL-induced representation alignment.

Third, zRSI indicates that contamination is associated with reduced or unstable local invariance, but the affected depth region varies across settings. In `Eurus-2-7B-PRIME`, clean samples show larger early-layer variability, whereas contaminated samples remain comparatively flatter and lower across depth. In `LIMR`, contaminated samples exhibit a sharp early-layer instability before becoming flat. Across RL checkpoints, invariance behavior becomes less monotonic, suggesting that local smoothness is also influenced by training-stage-specific representation regularization.

Overall, results show that contamination is not localized to a single universal layer or expressed through a strictly monotonic depth-wise trend. Instead, RL contamination appears as a distributed geometric shift across depth: contaminated representations become more sensitive to targeted perturbations, exhibit abnormal directional organization, and form less flexible local neighborhoods. This motivates using the three metrics jointly rather than relying on any single layer or metric as a standalone contamination indicator.

Implications for Contamination Detection. Taken together, these analyses show that RL contamination is not expressed as a single anomalous layer or a simple output-level memorization effect. Instead, it consistently manifests as a distributed geometric shift in how representations respond to controlled perturbations across network depth. Across models and RL checkpoints, contaminated samples repeatedly exhibit amplified perturbation sensitivity, abnormal directional concentration dynamics, and reduced or unstable local invariance relative to clean samples.

Meanwhile, the analyses also reveal that RL optimization introduces broader representation-level changes. In particular, later RL stages progressively increase directional alignment and modify local smoothness even for clean samples, partially overlapping with contamination-specific effects. Thus, contamination cannot be reliably characterized using a single metric, isolated layer, or fixed monotonic trend.

4. Contamination Detection Protocol

Motivated by the results in Section 3, we formulate contamination detection as a layer-aware representation anomaly detection problem. We find that contamination does not emerge at a single layer or through a uniform geometric pattern. Instead, contaminated samples exhibit distinct representation profiles across depth, including amplified perturbation sensitivity, abnormal directional concentration dynamics, and local variability under controlled perturbations.

These effects are both model-dependent and training-stage dependent. RL optimization itself progressively reshapes representation geometry even for clean samples, introducing broader alignment and smoothing effects that partially overlap with contamination-related behavior. Consequently, contamination should be characterized through *deviation from clean* geometric profiles across multiple metrics and

layers rather than from isolated layer-wise statistics.

Step 1: Clean-reference Robust Standardization. Let $\mathcal{M} = \{\mathrm{zRSM}, \mathrm{DC}, \mathrm{zRSI}\}$ denote the set of representation geometry metrics, let $\mathcal{L}$ denote the set of probed transformer layers, and let $m_\ell(x)$ denote the value of metric m at layer ℓ for sample x. The three metrics span several orders of magnitude in raw form, so we first apply a sign-preserving compression that tames their heavy-tailed regime while leaving values near zero essentially unchanged:

$$\tilde{m}_\ell(x) = \mathrm{sign}\big(m_\ell(x)\big)\ \log\big(1 + |m_\ell(x)|\big). \tag{1}$$

For each (m, ℓ) we estimate the clean reference *center* and *scale* from non-contaminated validation samples $\mathcal{D}^{\mathrm{clean}}$ using robust statistics that are themselves insensitive to outliers in the clean reference set:

$$\mu^{\mathrm{clean}}_{m,\ell} = \mathrm{median}\Big(\tilde{m}_\ell(x) : x \in \mathcal{D}^{\mathrm{clean}}\Big), \tag{2}$$

$$\sigma^{\mathrm{clean}}_{m,\ell} = 1.4826 \cdot \mathrm{MAD}\Big(\tilde{m}_\ell(x) : x \in \mathcal{D}^{\mathrm{clean}}\Big), \tag{3}$$

where the factor 1.4826 is the standard scaling that makes the median absolute deviation a consistent estimator of the standard deviation under Gaussian noise (Refer to Appendix A.8). The standardized geometric deviation of sample x at (m, ℓ) is then

$$z_{m,\ell}(x) = \frac{\tilde{m}_\ell(x) - \mu^{\mathrm{clean}}_{m,\ell}}{\sigma^{\mathrm{clean}}_{m,\ell} + \epsilon}, \tag{4}$$

with ϵ a small numerical constant. This formulation preserves the relative magnitude of geometric deviations while preventing a small number of extreme contaminated samples from inflating the clean-reference scale and washing out the signal for the rest of the population.

Step 2: Metric-specific Anomaly Alignment. Our analyses show that contamination affects each metric through a different geometric mechanism. Contaminated samples tend to exhibit elevated perturbation sensitivity in zRSM, abnormal directional concentration dynamics in DC, and reduced or unstable local invariance in zRSI. To account for these heterogeneous behaviors, we align each metric according to its contamination-associated deviation pattern:

$$\hat{z}_{m,\ell}(x) = \begin{cases} z_{m,\ell}(x), & m = \mathrm{zRSM}, \\ z_{m,\ell}(x), & m = \mathrm{DC}, \\ -z_{m,\ell}(x), & m = \mathrm{zRSI}. \end{cases} \tag{5}$$

For DC we use absolute deviations because both unusually high directional concentration and sharp directional collapse can indicate contamination-related geometric abnormalities; for zRSM and zRSI the contamination signal is directional and the alignment recovers a positive deviation in the contaminated direction.

Step 3: Layer-aware Aggregation. We aggregate the aligned per-(m, ℓ) deviations into a single per-sample contamination score by averaging their absolute magnitudes across the metric set $\mathcal{M}$ and the probed layer set $\mathcal{L}$:

$$S_{\mathrm{LaRA}}(x) = \frac{1}{|\mathcal{M}|\,|\mathcal{L}|} \sum_{m \in \mathcal{M}} \sum_{\ell \in \mathcal{L}} \hat{z}_{m,\ell}(x). \tag{6}$$

As each (m, ℓ) contribution is placed on a common, robust z-scale before aggregation, layer-localized abnormalities — such as early-layer instability, mid-layer directional collapse, or late-layer divergence — all contribute to $S_{\mathrm{LaRA}}(x)$ on a comparable scale. Clean samples are expected to remain close to the clean reference profile across merics and layers.

Interpretation. The proposed protocol characterizes contamination as a layer-aware geometric anomaly expressed through coupled changes in perturbation sensitivity, directional organization, and local invariance under controlled perturbations. A high contamination score indicates that a sample exhibits strong deviations from clean representation geometry distributed across the network and across multiple geometric facets simultaneously. In contrast, clean samples remain close to the clean reference profile at every (m, ℓ) and accumulate small standardized residuals overall.

Discussion. Our framework is motivated by four observations from the representation analyses. First, contamination emerges as a distributed geometric effect across layers rather than a single-layer artifact, motivating aggregation over all layers. Second, contamination produces heterogeneous metric behaviors, motivating metric-specific anomaly alignment instead of uniform standardization. Third, RL post-training reshapes representation geometry even for clean samples, making clean-reference standardization more reliable than absolute geometric statistics. Finally, the metrics are heavy-tailed and the clean reference distribution is estimated from a finite validation set, motivating signed-$\log(1 + |\cdot|)$ compression and median/MAD-based normalization for robust standardization. By combining robust compression, clean-reference normalization, metric-specific alignment, and layer-wise aggregation, S_{LaRA} captures contamination-related geometric deviations while remaining stable across models and training stages.

5. Contamination Detection Setup

5.1. Evaluation Metrics and Baselines

We evaluate contamination detection performance using standard metrics for membership inference attacks (MIA). **ROC-AUC** measures the model's ability to distinguish between member and non-member samples across all possible decision thresholds. It captures the overall separability of the two classes and is threshold-independent, making it a robust indicator of detection quality. **TPR@FPR=5%**

reports the true positive rate (i.e., correctly identified members) when the false positive rate (i.e., non-members incorrectly flagged as members) is fixed at 5%. This reflects performance in the low false-positive regime, which is critical in contamination detection where incorrectly labeling clean samples as contaminated is costly. These metrics provide a comprehensive evaluation of global discrimination ability (ROC-AUC) and practical operating performance under strict error constraints (TPR@FPR=5%). **Baselines** are six representative methods spanning likelihood-based, perturbation-based, and self-evaluation approaches. Refer to Appendix A.9 for further details.

6. Contamination Detection Results

Table 1 shows that LaRA consistently improves contamination detection when combined with output-level signals. In particular, when combined with the strongest output-level baseline, SELF-CRITIQUE (Tao et al., 2025), it achieves the best ROC-AUC and TPR@FPR=5% across all RL checkpoints, reaching 0.73/0.31 at initialization, 0.65/0.35 at epoch 1, and 0.79/0.38 at epoch 2.

These results directly support the main hypothesis of LaRA: contamination in RL-trained reasoning models is not fully captured by output-level statistics alone, but also manifests as layer-wise geometric deviations under controlled perturbations. While likelihood-based baselines such as Min-K, Min-K++, Recall, and CDD perform poorly and often collapse near random ranking, LaRA remains competitive across checkpoints despite relying solely on representation-level geometry.

Although PPL achieves relatively strong ROC-AUC scores, its behavior is less consistent across checkpoints and low-FPR regimes. This is likely because RL post-training partially preserves memorization-related likelihood signals from the base model while simultaneously reshaping reasoning behavior through reward optimization. As a result, perplexity can reflect general confidence calibration or policy sharpness rather than contamination-specific memorization. In contrast, LaRA explicitly measures how internal representations respond to controlled perturbations, making it more directly tied to contamination-induced geometric rigidity rather than output probability alone.

Importantly, LaRA provides complementary information to response-level self-evaluation. Although SELF-CRITIQUE is strong at the initial and final checkpoints, its performance drops substantially at epoch 1 (0.58 ROC-AUC), whereas LaRA remains stable (0.68 ROC-AUC, 0.73 F1). Combining both consistently improves low-FPR detection, indicating that layer-wise perturbation geometry captures contamination signals that are not fully reflected in generated reasoning trajectories. *Overall, the results validate the central contribution of this work: RL contamination induces distributed representation-level anomalies across depth, and aggregating perturbation sensitivity, directional organization, and local invariance yields a robust contamination signal during RL training.*

prime-rl/eurus-7b			
Method	**ROC-AUC**	**F1**	**TPR@FPR=5%**
Recall (Xie et al., 2024)	0.59	0.67	0.07
CDD (Dong et al., 2024)	0.38	0.67	0.00
Min-K (Shi et al., 2023)	0.32	0.67	0.07
Min-K++ (Zhang et al., 2024)	0.29	0.70	0.03
PPL (Gonen et al., 2023)	0.68	0.68	0.23
Self-Critique (Tao et al., 2025)	0.70	0.70	0.27
LaRA Score (Ours)	0.63	0.72	0.19
Self-Critique + LaRA Score (Ours)	**0.73**	**0.77**	**0.31**
prime-rl/eurus-7b (epoch 1)			
Recall (Xie et al., 2024)	0.56	0.67	0.13
CDD (Dong et al., 2024)	0.38	0.67	0.00
Min-K (Shi et al., 2023)	0.31	0.67	0.07
Min-K++ (Zhang et al., 2024)	0.26	0.67	0.03
PPL (Gonen et al., 2023)	**0.69**	0.68	0.23
Self-Critique (Tao et al., 2025)	0.58	0.67	0.07
LaRA Score (Ours)	0.68	**0.73**	0.19
Self-Critique + LaRA Score (Ours)	0.65	0.69	**0.35**
prime-rl/eurus-7b (epoch 2)			
Recall (Xie et al., 2024)	0.52	0.68	0.00
CDD (Dong et al., 2024)	0.38	0.67	0.00
Min-K (Shi et al., 2023)	0.33	0.67	0.07
Min-K++ (Zhang et al., 2024)	0.20	0.69	0.03
PPL (Gonen et al., 2023)	0.67	0.69	0.17
Self-Critique (Tao et al., 2025)	0.78	0.78	0.30
LaRA Score (Ours)	0.70	0.73	0.15
Self-Critique + LaRA Score (Ours)	**0.79**	**0.78**	**0.38**

Table 1. Results on three RL checkpoints of Eurus-2-7B-PRIME.

7. Conclusion

We introduced LaRA, a layer-wise representation analysis framework for detecting contamination in RL post-trained reasoning models. Unlike prior methods that rely on output-level likelihood or entropy signals, LaRA detects contamination through perturbation-induced representation geometry across layers. Our analyses show that contamination induces amplified perturbation sensitivity, abnormal directional concentration dynamics, and unstable local representation variability. Based on these findings, we develop a layer-aware detection protocol that aggregates geometric deviations across metrics and layers. Experiments across RL checkpoints show that representation-level signals complement output-level methods and improve contamination detection, particularly at low false-positive rates. Overall, our results suggest that contamination in RL-trained LLMs is systematically reflected in internal representation geometry, highlighting the value of representation-level approaches for auditing reasoning models.

Impact Statement

This work studies data contamination detection in RL post-trained large language models through layer-wise representation geometry analyses. As RL-trained reasoning models become increasingly common, contamination during post-training can undermine benchmark validity, inflate reported reasoning performance, and reduce the reliability of scientific comparisons. By introducing a representation-level framework for identifying contamination-related behaviors, our work aims to improve the transparency and trustworthiness of reasoning model evaluation.

More broadly, our findings suggest that contamination in RL-trained models is reflected not only in output behavior, but also in internal representation geometry. This may motivate future work on representation-level auditing, interpretability, and reliability analysis for post-trained LLMs. In particular, methods that characterize perturbation sensitivity and representation rigidity may help identify failure modes that are difficult to observe from outputs alone.

At the same time, representation-level analysis methods may introduce dual-use concerns. Techniques that probe internal activations could potentially be adapted for stronger membership inference attacks or for extracting information about training distributions. Although our framework is designed for aggregate contamination auditing rather than recovery of individual training samples, advances in representation analysis may nevertheless increase privacy-related risks if misused. We therefore encourage future work on safeguards, controlled evaluation protocols, and privacy-aware auditing methodologies.

Our experiments are conducted exclusively on publicly available models and datasets within a controlled research setting. We do not release methods intended to reconstruct memorized data or reveal sensitive user information. Overall, we believe this work contributes toward more reliable and rigorous evaluation of RL-trained reasoning systems while highlighting the importance of studying the broader implications of representation-level auditing techniques.

References

Balunović, M., Dekoninck, J., Petrov, I., Jovanović, N., and Vechev, M. Matharena: Evaluating llms on uncontaminated math competitions, February 2025. URL https://matharena.ai/.

Bi, J., Yan, D., Wang, Y., Huang, W., Chen, H., Wan, G., Ye, M., Xiao, X., Schuetze, H., Tresp, V., and Ma, Y. Reasoning self-evaluation via trajectory dynamics modeling, 2026. URL https://openreview.net/forum?id=Qtq5YjnIlB.

Choi, H. K., Khanov, M., Wei, H., and Li, Y. How contaminated is your benchmark? quantifying dataset leakage in large language models with kernel divergence. *arXiv preprint arXiv:2502.00678*, 2025.

Cui, G., Yuan, L., Wang, Z., Wang, H., Zhang, Y., Chen, J., Li, W., He, B., Fan, Y., Yu, T., et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Dong, Y., Jiang, X., Liu, H., Jin, Z., Gu, B., Yang, M., and Li, G. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12039–12050, 2024.

Dong, Y., Jiang, X., Tao, Y., Liu, H., Zhang, K., Mou, L., Cao, R., Ma, Y., Chen, J., Li, B., et al. Rl-plus: Countering capability boundary collapse of llms in reinforcement learning with hybrid-policy optimization. *arXiv preprint arXiv:2508.00222*, 2025.

Gonen, H., Iyer, S., Blevins, T., Smith, N. A., and Zettlemoyer, L. Demystifying prompts in language models via perplexity estimation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 10136–10148, 2023.

Guha, E., Marten, R., Keh, S., Raoof, N., Smyrnis, G., Bansal, H., Nezhurina, M., Mercat, J., Vu, T., Sprague, Z., et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gwak, M., Son, G., and Kim, J. Revisiting the uniform information density hypothesis in llm reasoning traces. *arXiv preprint arXiv:2510.06953*, 2025.

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, 1986.

Hao, S., Sukhbaatar, S., Su, D., Li, X., Hu, Z., Weston, J., and Tian, Y. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Hochlehnert, A., Bhatnagar, H., Udandarao, V., Albanie, S., Prabhu, A., and Bethge, M. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. *arXiv preprint arXiv:2504.07086*, 2025.

Huber, P. J. and Ronchetti, E. M. *Robust Statistics*. Wiley, 2 edition, 2009.

Kang, Z., Zhao, X., and Song, D. Scalable best-of-n selection for large language models via self-certainty. *arXiv preprint arXiv:2502.18581*, 2025.

Kwak, M. and Kim, J. Gap-k%: Measuring top-1 prediction gap for detecting pretraining data. *arXiv preprint arXiv:2601.19936*, 2026.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Lee, B. W., Padhi, I., Ramamurthy, K. N., Miehling, E., Dognin, P., Nagireddy, M., and Dhurandhar, A. Programming refusal with conditional activation steering. *arXiv preprint arXiv:2409.05907*, 2024.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023.

Li, M. Z., Agrawal, K. K., Ghosh, A., Teru, K. K., Santoro, A., Lajoie, G., and Richards, B. A. Tracing the representation geometry of language models from pretraining to post-training. *arXiv preprint arXiv:2509.23024*, 2025a.

Li, X., Zou, H., and Liu, P. Limr: Less is more for rl scaling. *arXiv preprint arXiv:2502.11886*, 2025b.

Maronna, R. A., Martin, R. D., Yohai, V. J., and Salibián-Barrera, M. *Robust Statistics: Theory and Methods (with R)*. Wiley, 2 edition, 2019.

OpenAI. Gpt-4o mini: advancing cost-efficient intelligence, 2024.

OpenRouter. Openrouter: Unified api for AI models, 2024. URL https://openrouter.ai.

Roh, Y., Cho, H., and Kim, J. Embracing anisotropy: Turning massive activations into interpretable control knobs for large language models. *arXiv preprint arXiv:2603.00029*, 2026.

Rousseeuw, P. J. and Croux, C. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993.

Rousseeuw, P. J. and Leroy, A. M. *Robust Regression and Outlier Detection*. Wiley, 1987.

Sheng, G., Zhang, C., Ye, Z., Wu, X., Zhang, W., Zhang, R., Peng, Y., Lin, H., and Wu, C. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., and Zettlemoyer, L. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

Tao, Y., Wang, T., Dong, Y., Liu, H., Zhang, K., Hu, X., and Li, G. Detecting data contamination from reinforcement learning post-training for large language models. *arXiv preprint arXiv:2510.09259*, 2025.

Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., and MacDiarmid, M. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Wang, H., Li, H., Ko, B., and Zhang, H. On the fragility of benchmark contamination detection in reasoning models. *arXiv preprint arXiv:2510.02386*, 2025.

Wang, Y., Zhang, P., Yang, B., Wong, D. F., and Wang, R. Latent space chain-of-embedding enables output-free llm self-evaluation. *arXiv preprint arXiv:2410.13640*, 2024.

Wu, H. and Cao, Y. Membership inference attacks on large-scale models: A survey. *arXiv preprint arXiv:2503.19338*, 2025.

Wu, M., Zhang, Z., Dong, Q., Xi, Z., Zhao, J., Jin, S., Fan, X., Zhou, Y., Lv, H., Zhang, M., et al. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 33944–33952, 2026.

Wurgaft, D., Rager, C., Kowal, M., Shyam, V., Feucht, S., Bhalla, U., Haklay, T., Bigelow, E., Sarfati, R., McGrath, T., Lewis, O., Merullo, J., Goodman, N., Fel, T., Geiger, A., and Lubana, E. S. Manifold steering reveals the shared geometry of neural network representation and behavior, 2026. URL https://arxiv.org/abs/2605.05115.

Xie, R., Wang, J., Huang, R., Zhang, M., Ge, R., Pei, J., Gong, N. Z., and Dhingra, B. Recall: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8671–8689, 2024.

Yang, A., Zhang, B., Hui, B., Gao, B., Yu, B., Li, C., Liu, D., Tu, J., Zhou, J., Lin, J., et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Zhang, J., Sun, J., Yeats, E., Ouyang, Y., Kuo, M., Zhang, J., Yang, H. F., and Li, H. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024.

Zhao, X., Kang, Z., Feng, A., Levine, S., and Song, D. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

A. Appendix

A.1. Related Work

A.1.1. Memorization and Data Contamination Detection

Data contamination detection is commonly framed as a special case of membership inference attacks (MIAs) (Wu & Cao, 2025), which were originally introduced to measure memorization and privacy risks in machine learning models. These methods exploit behavioral differences between training and non-training samples, typically using output-level statistics such as likelihood or perplexity Gonen et al. (2023); Xie et al. (2024); Zhang et al. (2024); Shi et al. (2023); Kwak & Kim (2026). With the rise of large language models, data contamination has received increasing attention due to its impact on benchmark validity. Most existing detection methods focus on the pre-training and supervised fine-tuning (SFT) stages, where learning relies heavily on memorization. In these regimes, memorization induces strong likelihood-based signals – such as unusually low perplexity – that many detectors are designed to capture. In contrast, during RL-based post-training, models are optimized through reward-driven exploration of reasoning trajectories rather than direct likelihood maximization. This decoupling of behavior from output probability weakens conventional MIA signals, making contamination detection substantially more challenging. Recent work has begun to address this gap by proposing entropy-based signals to detect contamination in RL-trained models (Tao et al., 2025), leveraging differences in output uncertainty between contaminated and uncontaminated samples. While this approach provides initial evidence that post-training contamination can still manifest in model behavior, it remains limited to output-level uncertainty measures and does not account for how RL reshapes internal representations. As a result, such methods may struggle when entropy differences are subtle or confounded by exploration dynamics, motivating alternative approaches that probe contamination through internal model behavior rather than surface-level statistics alone.

A.1.2. Representation Dynamics in LLMs

Recent work has increasingly started leveraging representation dynamics in LLMs to study behavior beyond previous output-level likelihoods or their relevant metrics (Kang et al., 2025; Gwak et al., 2025; Zhao et al., 2025). In particular, works such as Bi et al. (2026); Wang et al. (2024); Hao et al. (2024); Li et al. (2025a) approaches analyzing internal states or their evolution across layers to characterize behavior during post-training. Another line of work investigates how semantic and behavioral properties are encoded within internal representations, showing that hidden-state directions can be used to steer, detect, or selectively modulate model behavior (Turner et al., 2023; Lee et al., 2024; Li et al., 2023; Roh et al., 2026; Wurgaft et al., 2026). These studies suggest internal activations contain structured behavioral signals that reveal contamination-related characteristics.

Beyond reasoning and analyzing encoded semantic and behavioral properties, internal representations have also been explored for data contamination analysis. Kernel Divergence Score (Choi et al., 2025), which quantifies contamination by measuring how fine-tuning on a benchmark dataset alters the similarity structure of sample embeddings. This approach demonstrates that internal representations can encode signals related to data reuse and memorization. However, KDS operates at the dataset level, relies on an explicit supervised fine-tuning (SFT) intervention, and is not formulated as a membership inference attack on individual instances.

Importantly, existing representation-based approach in data contamination has largely been studied in SFT settings, where representation shifts induced by fine-tuning are pronounced. Whether analogous internal signals can be used to detect contamination in the RL post-training regime remains underexplored. In contrast to SFT, RL post-training optimizes models via reward-driven trajectory exploration, potentially leading to different and more subtle representation dynamics. This gap motivates the use of layer-wise representation stiffness as a membership-relevant signal for contamination detection in RL-post-trained models.

A.2. Algorithm

Algorithm 1 LaRA: Per-sample Layer-wise Representation Geometry Extraction

Input: Original question q_0; similar-question generator SIMILARGEN; importance-based blanking operator BLANKIMPORTANT producing k `[BLANK]` tokens; LLM paraphrase generator VARIANTGEN that preserves `[BLANK]` positions; mean-pooled hidden-state extractor $h_\ell(\cdot)$; layer set $\mathcal{L} = \{0, 1, \ldots, L-1\}$ (every transformer layer); number of similar questions K; number of paraphrase variants M; number of `[BLANK]` tokens k; numerical floor ϵ

Output: Per-layer geometric scores $\{\text{zRSM}_\ell, \text{DC}_\ell, \text{zRSI}_\ell\}_{\ell \in \mathcal{L}}$

$\{q_1, \ldots, q_K\} \leftarrow \text{SIMILARGEN}(q_0)$ $\mathcal{Q} \leftarrow \{q_0, q_1, \ldots, q_K\}$

foreach $q_i \in \mathcal{Q}$ **do**

  $q_i^- \leftarrow \text{BLANKIMPORTANT}(q_i,\, k)$ $\{v_{i,1}, \ldots, v_{i,M}\} \leftarrow \text{VARIANTGEN}(q_i^-)$

end

foreach $\ell \in \mathcal{L}$ **do**

  `/* (1) Representation Shift Magnitude → zRSM */`

  for $i = 0$ **to** K **do**

    $u_i \leftarrow h_\ell(q_i)$ $w_i \leftarrow h_\ell(q_i^-)$ $\Delta_i \leftarrow u_i - w_i$ $S_i \leftarrow \|\Delta_i\|_2$

  end

  $\mu_S \leftarrow \frac{1}{K} \sum_{i=1}^{K} S_i$; `// similars only`

  $\sigma_S \leftarrow \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (S_i - \mu_S)^2}$; `// sample std`

  $\text{zRSM}_\ell \leftarrow \dfrac{S_0 - \mu_S}{\sigma_S + \epsilon}$

  `/* (2) Directional Collapse → DC */`

  $\bar{s}_\ell \leftarrow \frac{1}{K} \sum_{i=1}^{K} \Delta_i$; `// mean shift over similars`

  $\text{DC}_\ell \leftarrow \dfrac{\Delta_0^\top \bar{s}_\ell}{(\|\Delta_0\|_2 + \epsilon)\,(\|\bar{s}_\ell\|_2 + \epsilon)}$; `// cosine`

  `/* (3) Representation Stability Index → zRSI */`

  for $i = 0$ **to** K **do**

    for $m = 1$ **to** M **do**

      $\phi_{i,m} \leftarrow h_\ell(v_{i,m})$

    end

    $\bar{\phi}_i \leftarrow \frac{1}{M} \sum_{m=1}^{M} \phi_{i,m}$ $R_i \leftarrow \frac{1}{M} \sum_{m=1}^{M} \|\phi_{i,m} - \bar{\phi}_i\|_2$; `// mean L2 distance`

  end

  $\mu_R \leftarrow \frac{1}{K} \sum_{i=1}^{K} R_i$ $\sigma_R \leftarrow \sqrt{\frac{1}{K-1} \sum_{i=1}^{K} (R_i - \mu_R)^2}$ $\text{zRSI}_\ell \leftarrow \dfrac{R_0 - \mu_R}{\sigma_R + \epsilon}$

end

return $\{\text{zRSM}_\ell, \text{DC}_\ell, \text{zRSI}_\ell\}_{\ell \in \mathcal{L}}$

A.3. Prompts for Generating Similar and Perturbed Questions

To analyze representation dynamics under controlled perturbations, we use a three-stage prompt pipeline consisting of: (1) generating structurally similar questions, (2) identifying removable key information, and (3) generating paraphrased perturbation variants.

As shown in Figure 5, we first generate semantically similar math problems that preserve the same reasoning structure and difficulty while modifying numerical values. This produces structurally matched control groups for representation comparison.

As shown in Figure 6, we then identify a generalized information component (e.g., "the time duration" or "the initial quantity") that can be consistently removed across related problems without revealing the actual value. The resulting perturbed questions are used to measure representation sensitivity under targeted information deletion.

Finally, Figure 7 illustrates the prompt used to generate paraphrased variants of perturbed questions while preserving the exact position and semantic role of the `[BLANK]` placeholder. These variants enable measurement of local representation variability for computing RSI.

Prompt for Generating Similar Questions

You are a math problem generator.
Given an original math problem, create \{num_questions\} similar problems that:

1. Follow the EXACT same structure and solution method as the original
2. Use DIFFERENT numerical values (change all numbers to make the problem unique)
3. Maintain the same difficulty level
4. Have the same type of solution approach
5. Are valid, solvable problems

Original Problem:
\{original_question\}
Generate \{num_questions\} similar problems. For each problem:

- Change ALL numerical values to create unique scenarios
- Keep the problem structure and mathematical concepts identical
- Ensure the problem remains solvable and realistic
- Make sure the new numbers create valid mathematical relationships

Output ONLY a JSON array of \{num_questions\} similar problems, where each element is a string containing the full problem text. Do not include solutions or explanations, only the problems.

Format your response as:
["{}"Problem 1 text here...", "{}"Problem 2 text here..."]

Figure 5. Prompt used for generating structurally similar math questions while preserving the original reasoning process and difficulty.

Prompt for Generating Perturbed Questions

You are a question editor that identifies key information to remove from math problems.
Given a math problem, identify ONE key piece of information that should be removed. Describe this information in a way that can be consistently applied to similar problems.

For example:

- "the total number of residents/people"

- "the initial quantity"

- "the final result value"

- "the time duration"

- "the distance measurement"

Original Problem:

\{original_question\}

Output ONLY a short description of what information type should be removed (e.g., "the total number of residents"). Do not include the actual value or explain why, just describe the information type in 5-10 words.

Figure 6. Prompt used for identifying removable key information shared across semantically related math problems.

Prompt for Generating Perturbed Variants

You are a text rewriter that creates paraphrased versions of math problems. Given an incomplete math problem with [BLANK] placeholders, create \{num_variants\} paraphrased versions that:

1. Preserve the EXACT position and meaning of [BLANK] - do NOT move or change [BLANK]

2. Use different wording and phrasing while maintaining the same mathematical meaning

3. Keep the same structure and logical flow

4. Do NOT reveal what the blank should be

5. Maintain all mathematical relationships and constraints

Incomplete Problem:

\{incomplete_question\}

Output ONLY a JSON array of \{num_variants\} strings with the paraphrased versions. Do not include explanations or notes.

Figure 7. Prompt used for generating variants of perturbed questions while preserving the semantic role of missing information.

A.4. Details of Curated Datasets

We summarize the details of curated datasets in Table 2 and provide examples of it in Table 3.

Dataset	Source / Composition	Members	Non-Members	Exposure	Purpose
Contamination Eval	Eurus RL Data + AIME 2026	**30**	**30**	-	Evaluation
Training Set	30 members + 970 RL-MIA Math samples	**30**	**970**	Once	RL training analysis

Table 2. **Overview of the curated contamination evaluation and training datasets.** The evaluation set is balanced with 30 member and 30 non-member Olympiad-level math problems. The training set contains 1K samples, consisting of 30 member instances exposed once and 970 additional RL-MIA Math (Tao et al., 2025)samples.

data_source	prompt	answer	member	metadata
Contamination evaluation samples				
aime26	[{role: system, content: When tackling complex reasoning tasks, you have access to actions such as ASSESS, ADVANCE, VERIFY, SIMPLIFY, SYNTHESIZE, PIVOT, and OUTPUT.}, {role: user, content: Patrick started walking at a constant rate from school to the park. Tanya ran 2 miles per hour faster than Patrick, Jose bicycled 7 miles per hour faster than Tanya, and all three arrived at the same time. Find m+n.}]	277	0	–
numina_amc_aime	[{role: system, content: When tackling complex reasoning tasks, you have access to actions such as ASSESS, ADVANCE, VERIFY, SIMPLIFY, SYNTHESIZE, PIVOT, and OUTPUT.}, {role: user, content: The highest price is $8.50 and the lowest price is $5.50. Calculate the percent by which the highest price is more than the lowest price.}]	70%	1	–
RL training sample				
olympiads	[{role: system, content: Your task is to follow a systematic, thorough reasoning process before providing the final solution. Structure your response into two sections: Thought and Solution. In the Thought section, present your reasoning using <think>{thoughts}</think>. In the Solution section, provide the final logical answer, optionally in \boxed{} format.}, {role: user, content: Determine the largest even positive integer which cannot be expressed as the sum of two composite odd positive integers.}]	38	–	{style: "rule"}

Table 3. **Samples from the curated contamination evaluation and RL training datasets.** Each instance contains the data source, structured conversational prompt, ground-truth answer, membership label when applicable, and metadata annotations.

A.5. Implementation Details

A.5.1. Training Details

We fine-tune the base model using Group Relative Policy Optimization (GRPO) within the VeRL (Sheng et al., 2024) training framework. Training is conducted for 2 epochs with a learning rate of 1×10^{-6}, train batch size 128, validation batch size 512, maximum prompt length 1024, and maximum response length 4096. We enable gradient checkpointing and dynamic batch sizing during optimization, with a per-GPU token budget of 16384 tokens. For rollout generation, we use vLLM (Kwon et al., 2023) with 4 sampled responses per prompt at temperature 1.0, while validation uses temperature 0.6. We do not apply explicit KL regularization during training. All training experiments are performed on $8\times$ NVIDIA A6000 GPUs.

A.5.2. Inference Details

We conduct two types of evaluation: reasoning evaluation and contamination evaluation.

Reasoning Evaluation. We conduct reasoning evaluation in Section A.6.2 to test training robustness. Here, we generate 5 sampled responses per example and report $\mathrm{pass@5}$, where a prediction is considered correct if at least one sampled response matches the ground-truth answer after extracting the final boxed or numeric answer. In addition, we compute the mean length-normalized token log-probability of the gold answer under the model:

$$\frac{1}{T}\sum_{t=1}^{T}\log p(a_t \mid \mathrm{prompt}, a_{<t}),$$

where T denotes the answer length. Log-probabilities are computed using either a standard Transformers forward pass or vLLM-based prompt log-probability scoring.

Contamination Evaluation. Contamination evaluation follows a two-stage pipeline. First, we generate model responses together with token-level statistics such as log-probabilities and entropies. Second, we compute contamination detection scores using both output-based and representation-based methods.

The evaluated baselines include Min-K%, PPL, CDD, Recall, and Self-Critique. For representation-based methods, LaRA constructs semantically related and perturbed variants of each question using gpt-4o-mini-generated (via OpenRouter API) (OpenAI, 2024; OpenRouter, 2024) paraphrases and incomplete-question variants, from which representation-level signals such as RSI/RSM and directional collapse are derived.

We evaluate member versus non-member separability using AUROC, TPR at fixed FPR (0.05). All inference and evaluation experiments are performed on $4\times$ NVIDIA RTX 3090 GPUs.

A.6. Validation on Training Setup

Before analyzing contamination-related representation dynamics, we first validate whether the GRPO-based RL post-training setup induces meaningful policy adaptation. Since our primary goal is to study how contamination signatures emerge after RL post-training rather than to optimize the RL algorithm itself, we focus on verifying whether the trained checkpoints exhibit non-trivial optimization and behavioral changes compared to the initial model. We use three checkpoints corresponding to different stages of RL post-training: the initial model before additional training (**epoch 0**), the checkpoint after the first training epoch (**epoch 1 – step 8**), and the checkpoint after the second training epoch (**epoch 2 – step 15**).

A.6.1. Training Logs

Figure 8a shows the mean critic score throughout training. We observe a consistent increase in the critic score as optimization progresses, indicating that the policy increasingly generates responses preferred by the reward objective. This suggests that the RL signal provides meaningful optimization pressure and that it undergoes substantial adaptation during post-training.

We additionally analyze optimization-related diagnostics to assess the stability of the training dynamics. Figure 8c shows the policy gradient clipping fraction during training. The clipping fraction remains near zero throughout most of training, suggesting that clipping rarely activates during optimization. While this does not prevent empirical policy improvement, it indicates that the policy updates may not be strongly constrained by clipping-based regularization.

In addition to these training diagnostics, we also observe improvements in downstream evaluation performance after RL post-training, further supporting that the trained checkpoint is behaviorally distinct from the initial model. Collectively, these observations indicate that the model is sufficiently updated to analyze whether contamination-related representation signatures emerge after RL post-training.

At the same time, the optimization diagnostics suggest that the training process is not perfectly stabilized. In particular, the combination of near-zero clipping fraction and extremely large gradient norms implies that policy updates may not be sufficiently constrained during optimization. Therefore, we do not interpret the current setup as a fully stabilized RL training regime. Instead, we treat the resulting checkpoint as an empirically improved post-trained model that undergoes substantial policy adaptation, which is sufficient for studying contamination-related representation dynamics in RL post-training. Despite imperfect optimization stability, the checkpoints exhibit clear behavioral and representational divergence sufficient for controlled contamination analyses.

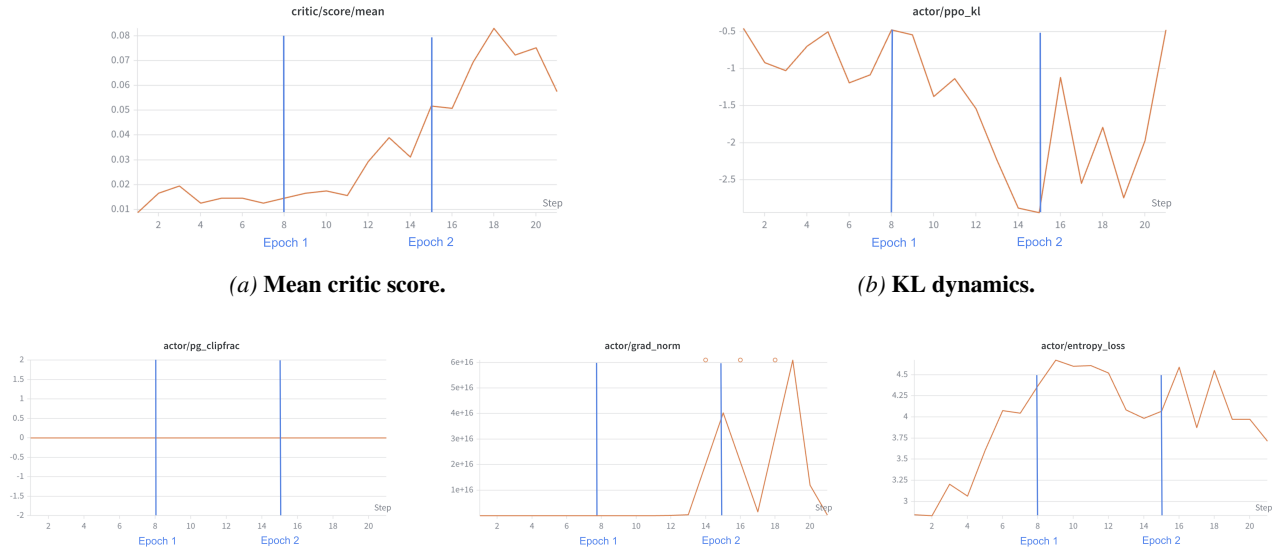
A.6.2. Performance Evaluation of Trained Models on Member vs. Non-member

We further evaluate whether RL post-training induces different behaviors on member and non-member samples by comparing answer accuracy and token-level confidence between the two groups.

Evaluation setup. We evaluate the RL-trained open-source model `Eurus-2-7B-PRIME` and its corresponding base model `Qwen2.5-Math-7B`. Following prior contamination analyses, we partition evaluation samples into *member* and *non-member* subsets based on whether the underlying samples originate from the training distribution used during RL post-training. For each sample, we compute: (i) **Pass@5**, which measures whether the correct answer appears among five sampled generations, and (ii) the length-normalized token-level log-probability of the generated answer conditioned on the prompt in Section A.5.2.

Results. Table 4 reports results for `Qwen2.5-Math-7B` and three `Eurus-2-7B-PRIME` checkpoints (Initial, Epoch 1, Epoch 2). The member–non-member Pass@5 gap increases monotonically with RL post-training: 23.3 points for the base model, 30.0 points at initialization, and 36.7 points at both Epoch 1 and Epoch 2. In particular, non-member Pass@5 collapses to zero (0/30) after RL training, while member Pass@5 increases from 33.3% to 36.7%. Since overall Acc@5 remains fixed at 18.3% across all PRIME checkpoints, the trend reflects a redistribution of performance toward seen samples rather than a general capability improvement.

Length-normalized token log-probabilities exhibit a non-monotonic calibration pattern. The base model assigns higher confidence to member generations than non-member generations (-2.08 vs. -2.44), consistent with accuracy. However, this ordering reverses at the Initial PRIME (-8.54 vs. -7.88) and Epoch 1 (-9.06 vs. -8.50) checkpoints, where the model assigns higher confidence to non-member outputs despite achieving zero correct answers on them. By Epoch 2, the ordering reverses again (-8.85 vs. -9.12). This indicates that likelihood confidence becomes unstable during RL post-training and



(a) **Mean critic score.** *(b)* **KL dynamics.**

(c) **Policy gradient clipping fraction.** *(d)* **Actor gradient norm.** *(e)* **Entropy loss.**

Figure 8. **Training diagnostics during GRPO-based RL post-training.** We analyze the optimization dynamics using critic score, KL divergence, clipping fraction, gradient norm, and entropy loss. The critic score progressively increases and the KL divergence changes throughout training, indicating substantial policy adaptation. At the same time, near-zero clipping fraction and extremely large gradient norms suggest that the optimization dynamics are not fully stabilized despite empirical performance improvements.

Model	Overall Pass@5	Pass@5 (%)		LogP(answer | prompt)	
		Member	Non-member	Member	Non-member
Qwen2.5-Math-7B (Base)	15.0 (9/60)	26.7 (8/30)	3.3 (1/30)	-2.08	-2.44
Eurus-2-7B-PRIME (Initial)	18.3 (11/60)	33.3 (10/30)	3.3 (1/30)	-8.54	-7.88
Eurus-2-7B-PRIME (Epoch 1)	18.3 (11/60)	36.7 (11/30)	0.0 (0/30)	-9.06	-8.50
Eurus-2-7B-PRIME (Epoch 2)	18.3 (11/60)	36.7 (11/30)	0.0 (0/30)	-8.85	-9.12

Table 4. **Comparison between the RL-trained open-source model and its base model on member and non-member samples.**

does not reliably track correctness across checkpoints.

Overall, RL training consistently amplifies the member–non-member accuracy gap, whereas confidence-based signals vary substantially across training stages. These results motivate representation-level analyses that capture contamination dynamics more stably than output-level confidence metrics.

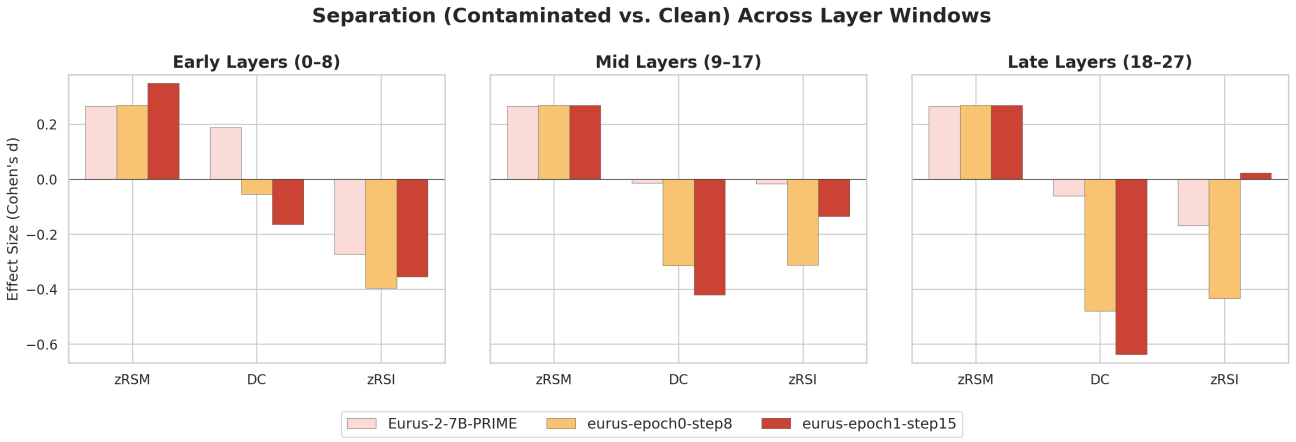


Figure 9. **Effect-size separation across early, mid, and late layer windows of `Eurus-2-7B-PRIME`.** Contaminated samples are consistently elevated on zRSM but progressively lower on DC and zRSI in mid-to-late layers, with the widened gap indicating that RL training amplifies a layer-selective representational signature of contamination.

A.7. Analyses on Layer-wise Trends on `Eurus-2-7B-Prime`

Figure 9 shows contaminated-vs-clean separation (Cohen's d) across early (0–8), mid (9–17), and late (18–27) layers over three RL checkpoints. Three consistent trends emerge.

First, **zRSM remains nearly unchanged across both depth and training**. All checkpoints show stable positive separation ($d \approx 0.27$–0.35), indicating that perturbation sensitivity is largely preserved throughout RL fine-tuning.

Second, **DC becomes increasingly negative as RL training progresses**, especially in deeper layers. While the initial checkpoint shows weak or near-zero separation, Epoch 1 and Epoch 2 exhibit progressively larger negative effects, with the strongest separation appearing in late layers ($d \approx -0.65$ at Epoch 2). This suggests that RL training amplifies directional-collapse behavior on contaminated samples, particularly in deeper representations.

Third, **zRSI is strongest in early and mid layers but weakens in late layers after training**. Early-layer separation remains consistently negative across checkpoints, whereas the late-layer signal gradually diminishes and nearly disappears by Epoch 2. This indicates that local-invariance differences are primarily concentrated in shallower representations.

Overall, the figure reveals a clear layer-conditioned structure: zRSM is stable across training, DC is progressively amplified by RL in deeper layers, and zRSI is concentrated in earlier layers. These complementary trends motivate combining all three metrics in LaRA rather than relying on a single layer-wise signal.

A.8. Justifications of the Scaling Factor Metric in Contamination Detection Protocol

The factor $1.4826 = 1/\Phi^{-1}(0.75)$ is the standard Fisher-consistency constant used in robust statistics to make the median absolute deviation (MAD) a consistent estimator of the standard deviation under a Gaussian reference (Hampel et al., 1986; Huber & Ronchetti, 2009; Rousseeuw & Croux, 1993). Specifically, if $X \sim \mathcal{N}(\mu, \sigma^2)$, then $\text{median}\,|X-\mu| = \sigma\,\Phi^{-1}(0.75)$, so multiplying the empirical MAD by $1/\Phi^{-1}(0.75) \approx 1.4826$ recovers σ in the large-sample limit.

Two properties of this scaling are particularly important for our protocol. First, it places the robust scale estimate on the same numerical units as the ordinary sample standard deviation, so the standardized deviations $z_{m,\ell}(x)$, the metric-specific alignments $\hat{z}_{m,\ell}(x)$, and the aggregated score $S_{\text{LaRA}}(x)$ retain their interpretation as approximate Gaussian-style z-scores. Consequently, replacing the standard deviation with a robust scale estimator does not implicitly retune downstream thresholds or alter the semantic interpretation of the score.

Second, unlike the sample standard deviation, which has a 0% breakdown point and can be driven arbitrarily large by a single extreme outlier, the MAD achieves a 50% breakdown point and therefore remains stable even when the clean-reference pool $\mathcal{D}^{\text{clean}}$ contains a substantial fraction of atypical samples (Hampel et al., 1986; Rousseeuw & Leroy, 1987). This is particularly relevant for representation-geometry metrics, whose raw distributions are often heavy-tailed even after signed-$\log(1+|\cdot|)$ compression. The resulting robustness-efficiency trade-off is well established in the robust statistics literature: while the MAD is less asymptotically efficient than the sample standard deviation under perfectly Gaussian noise, it provides substantially improved stability under contamination and heavy-tailed deviations (Huber & Ronchetti, 2009; Maronna et al., 2019).

A.9. Contamination Detection Baselines

The representative baselines we use to compare against our contamination detection protocol are as follows:

(1) Recall (Xie et al., 2024), which probes memorization by measuring the model's ability to regenerate ground-truth answers under controlled prompting; (2) CDD (Dong et al., 2024), which detects contamination via discrepancies in model predictions under input or prompt perturbations, based on the intuition that memorized samples are less sensitive to such changes; (3) Min-K% Prob (Shi et al., 2023), a likelihood-based metric that averages the log-probability over the lowest-probability tokens in a sequence, assuming memorized samples exhibit fewer low-confidence tokens; (4) Min-K++ (Zhang et al., 2024), which extends Min-K% with improved normalization and calibration for greater robustness across settings; (5) PPL (Gonen et al., 2023), which measures sequence-level likelihood via perplexity, where unusually low values indicate potential memorization; and (6) Self-Critique (Tao et al., 2025), which leverages the model's own reflective reasoning to assess contamination based on the confidence and consistency of its self-evaluation.