Detecting RLVR Training Data via Structural Convergence of Reasoning

Hongbo Zhang[1,2] **Leyang Cui**[2] **Jianhao Yan**[2] **Guangsheng Bao**[2]
Yue Zhang[*] **Shuo Yang**[3] **Yue Zhang**[2†]
[1] Zhejiang University [2] School of Engineering, Westlake University
[3] Keling AI
zhanghongbo@westlake.edu.cn, zhangyue@westlake.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) is central to training modern reasoning models, but the undisclosed training data raises concerns about benchmark contamination. Unlike pretraining methods, which optimize models using token-level probabilities, RLVR fine-tunes models based on reward feedback from self-generated reasoning trajectories, making conventional likelihood-based detection methods less effective. We show that RLVR induces a distinctive behavioral signature: prompts encountered during RLVR training result in more rigid and similar generations, while unseen prompts retain greater diversity. We introduce Min-kNN Distance, a simple black-box detector that quantifies this collapse by sampling multiple completions for a given prompt and computing the average of the k smallest nearest-neighbor edit distances. Min-kNN Distance requires no access to the reference model or token probabilities. Experiments across multiple RLVR-trained reasoning models show that Min-kNN Distance reliably distinguishes RL-seen examples from unseen ones and outperforms existing membership inference and RL contamination detection baselines. The project page is available at https://stevenzhb.github.io/detect-rlvr-data/.

1 Introduction

Recent advances in large language models (LLMs) have demonstrated remarkable improvements in reasoning performance, particularly on benchmarks with verifiable answers such as mathematics [14, 52, 43, 15, 37], coding [36, 19, 51], and symbolic problem solving [47]. A key driver behind these gains is reinforcement learning with verifiable rewards (RLVR) [14, 45, 48], in which models are trained on problems with automatically checkable outcomes, most commonly math and code tasks, and are optimized to produce correct final answers. By directly reinforcing reasoning trajectories that lead to correct solutions, RLVR enables models to refine their chain-of-thought generation beyond supervised fine-tuning. As a result, RLVR has become a cornerstone of modern post-training pipelines and now underpins many state-of-the-art reasoning models.

Despite impressive benchmark results, a growing body of evidence suggests that high reported reasoning performance does not always translate to robust generalization. When evaluation shifts to temporally newer, less exposed, or expert-level problems, model performance often drops substantially [31, 12, 22]. In practice, models that solve standard benchmark questions reliably may fail on newly released problems, alternative formulations of known tasks, or slightly more complex variants that require the same underlying reasoning. These patterns raise concerns that some gains reflect overspecialization during post-training, rather than consistently robust reasoning ability [53, 24, 1, 13, 33].

[*]The fifth author is Yue Zhang (章岳) and the last author is Yue Zhang (张岳).
[†]Corresponding author: zhangyue@westlake.edu.cn.

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

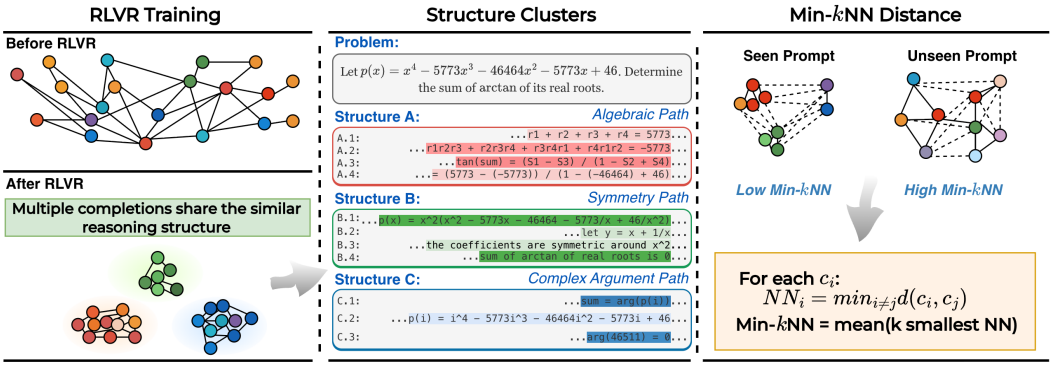


Figure 1: We study how RLVR induces structural convergence in reasoning trajectories. **Left**: RLVR compresses diverse reasoning paths into shared structural modes. **Middle**: multiple completions cluster into few reasoning structures. **Right**: Min-kNN Distance quantifies this collapse via nearest-neighbor edit distances, yielding low values for seen prompts and high values for unseen prompts.

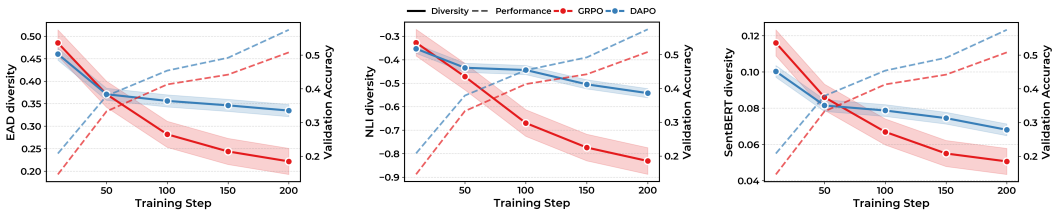
This issue is exacerbated by current open-source practices around RLVR. Many recent releases provide only RLVR-tuned models, without access to the base checkpoints or RL training data, making it difficult to assess whether benchmark problems or close paraphrases were encountered during training [28, 42]. As a result, risks of benchmark contamination and inflated generalization claims increase, while practitioners lack reliable tools to detect overlap with prior RLVR exposure. These challenges highlight the need for methods that can identify whether a specific example has appeared during RLVR training.

Prior work has shown that detecting training data exposure during pretraining or supervised fine-tuning is often feasible, as memorization under likelihood-based objectives leaves strong statistical traces detectable even in black-box settings [38, 54, 46, 8]. However, detecting exposure during RLVR training presents a different challenge. Unlike likelihood-based training, RLVR optimizes models through reward feedback on self-generated reasoning trajectories, which makes conventional token-level or likelihood-based signals ineffective. As demonstrated in our experiments (Section 5), traditional detection methods, which rely on perplexity and token-level statistics, fail to capture the changes in reasoning patterns induced by RLVR. To understand how RLVR reshapes model behavior, we analyze the evolution of generation diversity during training. As shown in Fig. 1, RLVR induces a systematic convergence in reasoning trajectories: multiple completions sampled for the same prompt seen during RL training yield increasingly similar generations, while unseen prompts retain high variability. This collapse concentrates on symbolic and algebraic reasoning components and compresses outputs into a small number of recurring structural modes rather than a single derivation. These stable structural patterns form a distinctive behavioral signature of RLVR exposure.

Building on this observation, we introduce Min-kNN Distance, a simple black-box method for detecting whether an example has appeared during RLVR training. As illustrated in Fig. 1, given a prompt, Min-kNN Distance samples multiple completions from the RLVR-tuned model and quantifies their structural diversity by computing edit distances among generated outputs. Since RLVR induces a convergence in reasoning diversity for seen examples, prompts that appeared during RLVR training yield consistently smaller Min-kNN Distance values than unseen prompts. The method requires only sampling access to the RLVR-tuned model and does not rely on token log probabilities or any reference models. Extensive experiments across multiple reasoning models and RLVR setups demonstrate that Min-kNN Distance reliably distinguishes RL-seen examples from unseen ones.

Our contributions are as follows:

- We provide the first systematic analysis of how RLVR reshapes reasoning behavior. We reveal that RLVR induces a *structural collapse* rather than mere token-level memorization, where reasoning trajectories for seen prompts concentrate into a limited set of recurring structural modes, particularly within symbolic and algebraic components.
- Based on these findings, we introduce Min-kNN Distance, a simple, black-box method for detecting RLVR exposure without requiring access to the token log probabilities.
- We conduct extensive experiments across diverse reasoning models, RL algorithms, and training setups, showing that Min-kNN Distance reliably distinguishes RL-seen examples



(a) EAD (b) NLI (c) Embedding

Figure 2: Evolution of generation diversity during RLVR training under DAPO and GRPO, measured in a per-input setting by three complementary metrics capturing lexical (EAD), logical (NLI), and semantic (embedding) diversity.

from unseen ones, even under challenging conditions like paraphrasing, distillation, and varying decoding configurations.

2 Preliminary

Problem Definition: We study the problem of detecting whether a given example has appeared during the RLVR training stage of a reasoning language model. Formally, given an RLVR-tuned model M_{RLVR} trained on an (unknown) RL dataset D_{RL}, the goal is to learn a detector h that determines whether an arbitrary query input x belongs to D_{RL}. The detector predicts membership as

$$h(x, M_{\text{RLVR}}) = \begin{cases} 1, & \text{if } x \in D_{\text{RL}} \\ 0, & \text{if } x \notin D_{\text{RL}} \end{cases}$$

This formulation follows the standard membership inference attack (MIA) setting, but differs from pretraining and supervised fine-tuning scenarios due to the reward-driven and self-generative nature of RLVR.

RLVR Training Process: During RLVR training, each example consists of a reasoning-intensive prompt x paired with a verifiable answer a. The model samples a CoT c followed with a final answer $\hat{a}$ from its policy $\pi_\theta(\cdot|x)$, producing output $o = (c, \hat{a})$.

A scalar reward $r(\hat{a}, a)$ is computed based on the correctness of the predicted answer by comparing $\hat{a}$ with a. The model parameters are updated to increase the expected reward

$$J(\theta) = \mathbb{E}_{(c,\hat{a})\sim\pi_\theta(\cdot|x)}[r(\hat{a}, a)],$$

and a typical policy gradient update takes the form

$$\nabla_\theta J(\theta) = \mathbb{E}_{(c,\hat{a})\sim\pi_\theta(\cdot|x)}[r(\hat{a}, a)\nabla_\theta \log \pi_\theta(c, \hat{a}|x)].$$

This optimization acts on the distribution of generated chains of thought rather than on the prompt itself, and no golden chain of thought is available for likelihood comparison. As a result, RLVR repeatedly adjusts the probability of reasoning trajectories according to their observed rewards. Unlike likelihood-based pretraining, RLVR optimizes models through reward feedback on self-generated reasoning trajectories, rather than optimizing on golden trajectories, making conventional token-level signals ineffective. This motivates a closer examination of how RLVR affects reasoning diversity.

3 Analyzing Reasoning Pattern under RLVR

Previous work has demonstrated that RLVR models tend to reduce the coverage of reasoning, leading to narrower reasoning trajectories compared to their base models [10, 50]. While these studies highlight the loss of reasoning diversity, they primarily focus on performance metrics rather than directly analyzing the changes in output structure. Notably, there has no systematic study quantifying how reasoning diversity evolves through the RLVR training process, especially for seen and unseen samples. To fill this gap, we focus on analyzing the dynamics of reasoning diversity in the outputs of RLVR models. We investigate how the diversity of reasoning trajectories decreases during RLVR training and explore how these changes can serve as a potential signal for detecting whether a specific

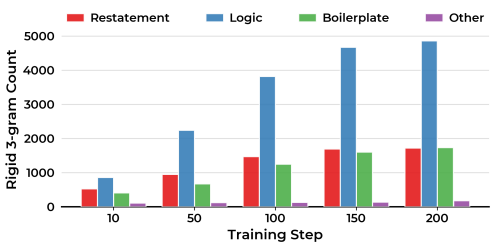


Figure 3: Rigid 3-gram category counts in GRPO training checkpoints. We retain 3-grams occurring in at least half of the completions for each prompt and classify them into four categories using GPT-4o.

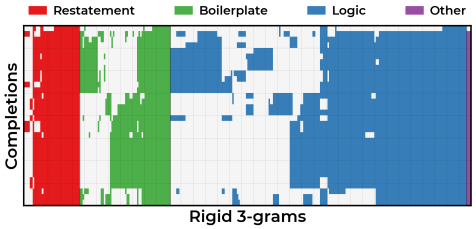


Figure 4: Occurrence of rigid 3-grams in at least three completions for one prompt. Rows are completions; columns are 3-grams. Red, green, blue, and purple indicate restatement, boilerplate, logic, and other fragments; white indicates absence. HAC reorders rows by shared fragments, revealing blocks of recurring structural modes.

example has been included in the training data. Specifically, we train a Qwen-2.5-7B-Base model [32] using two representative RLVR algorithms, DAPO [49] and GRPO [37] (Details in App. A.1). The DAPO algorithm employs a clip-higher strategy to explicitly encourage exploration during training. We conduct RLVR training on the DAPO dataset [49] for 200 rollout steps (approx. 5 epochs, reaching validation convergence) and analyze multiple intermediate checkpoints.

RLVR Induces Generation Rigidity. To quantify the impact of RLVR on output diversity, we use three complementary metrics, as proposed by Kirk et al. [18]: lexical, logical, and semantic diversity. Lexical diversity is measured using expectation-adjusted distinct n-grams (EAD) [20], which estimates the proportion of distinct n-grams while correcting for length bias, averaged over $n = \{1, 2, 3, 4, 5\}$. In particular, for a given prompt, we calculate the distinct n-grams from completions and normalize them by the expected number of n-grams. Logical diversity is assessed using an NLI-based metric, which measures the proportion of entailment and contradiction relations across pairs of randomly selected completions. This allows us to quantify how much the model's reasoning varies in terms of logical consistency. Finally, semantic diversity is evaluated through sentence-embedding diversity, defined as one minus the average pairwise cosine similarity between the embeddings of the generated outputs. Together, these metrics capture lexical, logical, and semantic diversity, respectively, providing a comprehensive view of how RLVR affects different aspects of generation behavior.

For each checkpoint, we sample 32 completions for 300 training prompts and compute the per-input diversity across all three metrics. As shown in Fig. 2a, we observe a steady decline in all three metrics throughout RLVR training under both DAPO and GRPO. This consistent decrease suggests that the space of reasoning trajectories conditioned on a fixed prompt narrows as the model undergoes RLVR, indicating that RLVR systematically leads to converged reasoning trajectories in lexical, logical, and semantic dimensions. We further analyze the diversity in a cross-input setting, where we assess the diversity across completions from all 300 prompts (Fig. 8 in App. A.2). We observe a similar declining trend in lexical diversity, further supporting the notion that RLVR reduces output diversity not only at the level of individual prompts but also across the global output distribution. This global reduction in diversity reflects the collapse of the reasoning space, concentrating it into a smaller set of recurring patterns, leading to more rigid outputs as training progresses.

RLVR Converges Symbolic Reasoning Segments. To understand which parts of the completions are most affected by RLVR training, we inspect *rigid 3-grams*, which recur across completions for the same prompt. For category counts, we retain those appearing in at least half of the completions, counting each 3-gram once per completion. Through this analysis, we identify repeated fragments that can be categorized into three main types. The first type is restatements of the problem, where the model habitually repeats the problem statement in the early part of the output. The second type is boilerplate connective phrases, such as "To solve this problem..." or "Let's denote the...". These phrases are structural fillers that do not contribute to the core reasoning but serve to connect the reasoning steps. The third type, symbolic or algebraic logic steps, represents the core reasoning

Table 1: Comparison of reasoning structure collapse between seen and unseen data. The first part shows the distribution of rigid 3-gram categories, while the second part displays the cumulative distribution of the number of reasoning structure clusters across different stages of RLVR training.

Rigid Logic 3-gram Count				
Step	**50**	**100**	**150**	**200**
Seen	856	2,240	3,817	4,673
Unseen	840	1,684	2,639	3,476
Cumulative Distribution of Reasoning Structure Clusters				
Cluster Size	≤ 2	≤ 4	≤ 8	≤ 16
Seen	44.90%	67.35%	95.92%	100.00%
Unseen	22.45%	55.10%	89.80%	100.00%

components, typically involving algebraic manipulations, standardized mathematical transformations, or function definitions. For example, a model might express equations like "x = y + 2," apply mathematical laws, or define functions like "f(x) = x^2 + 3x," all of which involve relatively fixed structural forms that allow little variation during reasoning. These segments play a critical role in the core of the model's reasoning and tend to become more standardized as RLVR training progresses. For example, Fig. 9 in App. A.5 shows a representative case of these categories.

We extend this observation by sampling 50 training prompts at each checkpoint and extracting the high-frequency 3-grams. These 3-grams are categorized using GPT-4o (prompt is shown in App. A.3), allowing any unclassifiable 3-grams to be marked as "other." Human validation confirms strong agreement with GPT-4o labels (Cohen's $\kappa \approx 0.81$ on core categories), ensuring the robustness of this qualitative analysis (details in App. A.4). The results are plotted for each checkpoint, showing the number of recurring 3-grams. As shown in Fig. 3, we observe that symbolic logic fragments increase rapidly over the course of RLVR training, while restatement and boilerplate patterns exhibit slower growth. This indicates that RLVR significantly compresses the symbolic reasoning components, especially the logical tokens, which carry the model's reasoning structure. As these segments become more rigid, they form a core set of standardized reasoning steps, leaving less variability in the model's outputs. Thus, we conclude that RLVR primarily causes the collapse of symbolic reasoning segments.

RLVR Converges Reasoning into a Limited Set of Structural Modes. To explore how RLVR affects reasoning trajectories, we analyze whether it leads to a single deterministic path or multiple recurring structural modes. For the visualization in Fig. 4, we retain 3-grams appearing in at least three completions and construct a binary completion-by-3-gram occurrence matrix. Hierarchical agglomerative clustering (HAC) reorders the completion vectors so that outputs sharing recurring fragments appear together. The resulting blocks represent structural modes, with logical 3-grams often co-occurring with boilerplate phrases. Restatements of the problem appear across all completions. Fig. 1 (middle) provides a qualitative example of these recurring reasoning structures.

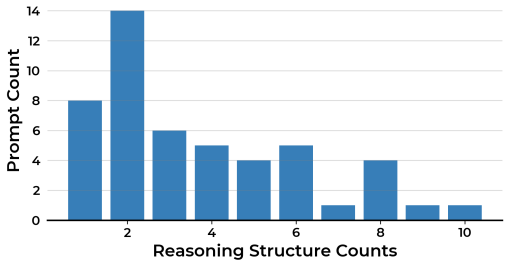


Figure 5: Distribution of the number of reasoning structure clusters across 50 training prompts.

To quantify these patterns, we apply HAC [26] to binary completion vectors indicating the occurrence of logic 3-grams for each of 50 training prompts, and compare cluster counts across RLVR checkpoints. The histogram in Fig. 5 shows that most prompts exhibit between two and four stable clusters of reasoning structures, with a few prompts showing even more. This result indicates that RLVR compresses reasoning into a limited number of tightly concentrated structural modes rather than a single canonical trajectory. Such rigidity is not inherently harmful: repeated structures may reflect useful learned procedures. We use it as a behavioral signal of exposure, not as a measure of reasoning quality.

Greater Rigidity in Seen Data Compared to Unseen Data. Finally, we compare the reasoning structure convergence between seen and unseen data. As shown in Tab. 1, both seen and unseen data exhibit reasoning convergence, but with notable differences. For the rigid 3-gram categories, the training set shows a greater increase in symbolic reasoning steps, indicating a more rigid model. In contrast, the unseen data exhibits a smaller increase in rigidity, suggesting a lower level of rigidity for unseen prompts. Regarding reasoning structure clusters, the cumulative distribution reveals that training prompts have a higher proportion of clusters with fewer reasoning structures, while unseen prompts retain more diverse reasoning paths. This suggests that RLVR induces stronger rigidity for seen data, while unseen prompts maintain more variability in reasoning.

4 Min-kNN Distance: A Simple Black-Box Detector for RLVR Exposure

Previous analyses show that RLVR reduces lexical, logical, and semantic diversity, while recurring 3-grams and clustering reveal convergence into a few structural modes. These observations motivate measuring local concentration: global diversity measures may obscure several compact modes whose completions remain far apart across modes. We therefore introduce Min-kNN Distance, a simple black-box statistic based on nearest-neighbor edit distances. Completions for **seen** prompts are expected to exhibit stronger local concentration than those for **unseen** prompts, yielding smaller distances.

Given a prompt x, we sample m completions from the RLVR-tuned model M_{RLVR},

$$O(x) = \{o_1, o_2, \ldots, o_m\}, \quad o_i \sim M_{\text{RLVR}}(\cdot \mid x),$$

and compute the pairwise normalized edit distance between every pair:

$$D_{ij} = d(o_i, o_j), \qquad 1 \le i, j \le m.$$

where $d(\cdot, \cdot)$ denotes the normalized Levenshtein edit distance between two completions, defined as the minimum number of insertions, deletions, and substitutions required to transform one sequence into the other, normalized by the length of the longer sequence. For each completion o_i, we define its nearest–neighbor distance:

$$\text{NN}_i(x) = \min_{j \neq i} D_{ij}.$$

Let $\text{NN}_{(1)}(x) \le \text{NN}_{(2)}(x) \le \cdots \le \text{NN}_{(m)}(x)$, be the sorted list of nearest-neighbor distances. We take the k smallest values to form a set Min-$k(x)$ and define the detection score:

$$\text{Min-kNN}(x) = \frac{1}{k} \sum_{t=1}^{k} \text{NN}_{(t)}(x). \tag{1}$$

Because RLVR collapse forces completions of seen prompts into a few structural modes, we expect

$$\text{Min-kNN}(x_{\text{seen}}) < \text{Min-kNN}(x_{\text{unseen}}).$$

A detector is obtained by thresholding $\text{Min-kNN}(x)$. The method requires only black-box sampling access to the RLVR-tuned model and does not use the token log probabilities, or reference models. In offline benchmark auditing, an auditor can calibrate a threshold using an appropriate control set and prioritize low-scoring prompts for further investigation. The score provides evidence of RLVR-associated exposure rather than definitive membership proof.

5 Experiments

5.1 Experimental Setup

Models and Datasets. We evaluate Min-kNN Distance under two complementary settings. For **Open-source RLVR models**, we evaluate SimpleRL-32B [52], DAPO-Qwen-32B [49], JustRL-DeepSeek-1.5B [15], and Open-Reasoner-Zero-7B [16], using 300 member and 300 non-member instances sampled from standard math benchmarks (e.g., AIME, MATH-500). **RL-MIA benchmark models.** Following the RL-MIA setup of [41], we include controlled contamination settings. Specifically, we train Qwen2.5-7B-Instruct and DeepSeek-Math-7B-Instruct [37] using the SAT and K&K settings to evaluate detection under precise contamination scenarios. Extended dataset details are provided in App. B.2.

Table 2: AUC results for RLVR data detection across RL-tuned models. **BB** indicates whether a method operates with sampling-only access. DS-KK and DS-SAT denote DeepSeek-Math-7B-Instruct models RLVR-trained under the K&K and SAT settings, respectively; QW-KK and QW-SAT denote QW-2.5-7B-Instruct models RLVR-trained under the same settings. The best result is shown in **bold** and the second best is underlined.

		Open-source RLVR Models				RL-MIA Models				
Method	**BB**	**DAPO**	**JustRL**	**ORZ**	**SimpleRL**	**DS-KK**	**QW-KK**	**DS-SAT**	**QW-SAT**	**Avg.**
PPL	✗	0.63	0.61	0.72	0.72	0.59	0.56	0.44	0.54	0.60
Min-K%	✗	0.37	0.43	0.29	0.28	0.53	0.44	0.53	0.47	0.42
Min-K%++	✗	0.37	0.48	0.56	0.39	0.56	0.53	0.66	0.52	0.51
CDD	✓	0.52	0.51	0.36	0.45	0.49	0.46	0.50	0.47	0.47
Recall	✗	0.50	0.52	0.51	0.62	0.54	0.52	0.56	0.47	0.53
Self-Critique	✗	0.58	0.51	0.48	0.49	0.54	0.53	0.43	0.49	0.51
Min-kNN Distance	✓	**0.72**	**0.66**	**0.75**	**0.76**	**0.80**	**0.68**	**0.68**	**0.57**	**0.70 (+17%)**

Baselines. We compare Min-kNN Distance against existing membership inference methods, including PPL, Min-K% [38], Min-K%++ [54], Recall [46], and CDD [8], as well as Self-Critique [41]. Detailed mechanisms for all baselines are deferred to App. B.4. Following standard practice [38, 41], we evaluate all methods using the Area Under the ROC Curve (AUC).

Implementation and Evaluation Metrics For each prompt, Min-kNN Distance samples 32 completions from the RLVR-tuned model, computes all pairwise edit distances among the generated outputs, and averages the k smallest distances to obtain the final score. Unless otherwise stated, we set $k = 10$, which we find to be stable across models and training regimes. The baseline experimental setup follows the methodology in Tao et al. [41]. Other details are shown in App. B.3. We evaluate detection performance using the Area Under the ROC Curve (AUC), a threshold-independent metric widely adopted in membership inference and data contamination studies [38, 41, 54].

5.2 Main Results

Tab. 2 reports AUC performance across all models.

Min-kNN Distance consistently outperforms all baseline methods. Min-kNN Distance achieves the highest AUC across all evaluated models, with an average score of 0.70, which represents a 17% relative improvement over the strongest baseline. In contrast, probability-based and consistency-based baselines often exhibit unstable or near-random performance. Blind baselines also perform below Min-kNN Distance on DAPO (App. C.3). The signal persists after controlling for observed solve rate (App. C.4).

Min-kNN Distance is robust across different RLVR algorithms. Min-kNN Distance consistently performs well across models trained with different RLVR algorithms, including GRPO (SimpleRL-32B), DAPO (DAPO-Qwen-32B), and PPO (Open-Reasoner-Zero-7B) [35]. This stability suggests that the signal detected by Min-kNN Distance is not dependent on the RL algorithm.

Min-kNN Distance generalizes across model scales. From 1.5B to 32B parameters, Min-kNN Distance consistently maintains strong detection performance, despite significant variations in model size and training dynamics. Notably, Min-kNN Distance operates in a fully black-box setting, consistently outperforming methods that require access to token probabilities or internal likelihoods. These results demonstrate that RLVR-induced structural collapse provides a strong, model-agnostic signal for exposure detection.

5.3 Ablation

We analyze the sensitivity of Min-kNN Distance to its hyperparameters. For k and sampled completions, we report mean AUC over eight RLVR-trained models, while temperature varies on SimpleRL. Comparisons with greedy-anchored statistics are provided in App. C.5.

Min-kNN Distance is robust to k over a wide range. As shown in Fig. 6 (left), performance improves quickly as k increases and then saturates, with noticeable degradation only when k is too small to reliably capture collapsed modes.

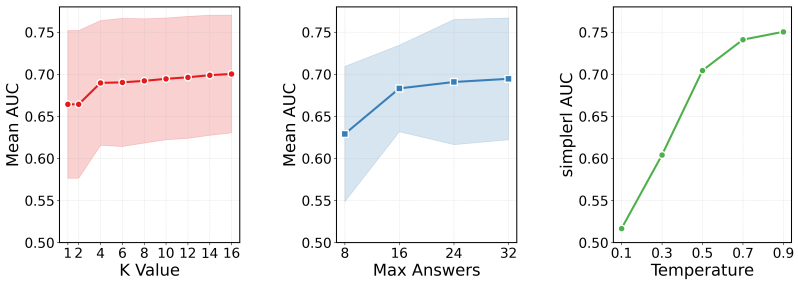


Figure 6: Ablation studies of Min-kNN Distance. (a) Effect of k. (b) Effect of the number of sampled completions. (c) Effect of sampling temperature.

Sampling more completions consistently improves detection. Fig. 6 (middle) shows steady AUC gains as sampled completions increase. Additional samples make collapsed structural patterns easier to expose, though improvements eventually plateau.

Higher decoding temperatures strengthen the detection signal. In Fig. 6 (right), higher temperatures yield stronger detection, while lower temperatures reduce output variability and partially obscure structural collapse. We also observe that models with extensive RLVR require smaller k for stable performance, whereas lower temperatures make Min-kNN Distance more sensitive to k.

5.4 Analysis

Robustness to Paraphrased Prompts. In practice, a detector should remain effective when training prompts are paraphrased rather than exactly repeated. We therefore evaluate the robustness of Min-kNN Distance under the DAPO-Qwen-32B setting by paraphrasing 300 RLVR training prompts using GPT-4o and testing detection performance on these paraphrased inputs. As shown in Tab. 3, Min-kNN Distance remains highly stable under paraphrasing, with AUC decreasing only slightly from 0.72 to 0.71, and other baselines exhibit similar changes. This indicates that the signal exploited by Min-kNN Distance is not tied to surface-level prompt forms, but instead reflects a structural collapse in reasoning induced by RLVR that generalizes across semantically equivalent prompts.

Table 3: Detection performance before and after paraphrasing RLVR training prompts under the DAPO setting.

Method	Original	Paraphrased
PPL	0.63	0.66
Self-Critique	0.58	0.57
Min-kNN Distance	**0.72**	**0.71**

Detecting Distillation Prompts. Recent reasoning models are often distilled from RLVR-tuned models, where the RLVR-trained teacher model generates multiple reasoning trajectories, and the student is trained on these outputs. We analyze whether prompts used during distillation can be detected. We conduct this analysis on OpenR1-Distill-7B [17]. We evaluate detection performance on two sets of prompts: 300 held-out evaluation problems and 300 prompts used during the distillation process. Using the same detection setup as in our main experiments, Min-kNN Distance achieves an AUC of 0.76 on distillation prompts, outperforming other baselines (Tab. 4). These results suggest that the structural collapse induced by RLVR is partially transferred to the distilled model, allowing Min-kNN Distance to detect prompts used during distillation.

Table 4: Detection performance on distillation prompts (OpenR1-Distill-7B).

Method	AUC
PPL	0.70
Self-Critique	0.65
Min-kNN Distance	**0.76**

Robustness across Different RL Data Sources. We further analyze the robustness of Min-kNN Distance across different types of RLVR training data. We conduct this analysis on Nemotron-Research-Reasoning-Qwen-1.5B [19], which is trained with a mixture of math, coding data during RLVR. From its RL training set, we sample 100 prompts from math and 100 from code as member examples. To construct unseen data, we use 100 problems from Beyond-AIME for math and 100 validation examples from Eurus Coding [7] for code. As shown in Tab. 5, Min-kNN Distance achieves strong performance on both math (AUC 0.80) and coding (AUC 0.69), outperforming other methods. However, RL-phase contamination detection is more challenging for code, likely due to the higher diversity and flexibility in coding tasks compared to the more structured nature of math problems.

Table 5: Detection performance on code and math data under ProRL setting.

Method	Math	Code
PPL	0.40	0.36
Self-Critique	0.75	0.47
Min-kNN Distance	**0.80**	**0.69**

Dual-Stage Contamination Analysis We follow the dual-stage contamination detection experiment setup from Tao et al. [41], using the GSM8K dataset [6], which is known for its pretraining contamination. RL-phase contamination is simulated by injecting a portion of the test set into the RL training process and fine-tuning with PPO. Two data subsets are created for contamination detection. The first subset, representing lower pretraining contamination, is created by selecting the bottom portion of items based on pretraining-contamination scores (using PPL). The second subset is a random-control subset, sampled uniformly to match the sample size of the first subset, controlling for sample size effects on AUC.

As shown in Fig. 7, we compare RL-phase contamination effects across these subsets. The green line represents performance on the lower pretraining contamination subset, while the gray line shows performance on the random-control subset. The results indicate that Min-kNN Distance performs significantly better on the lower pretraining contamination subset, while the random-control subset shows reduced performance. This suggests that Min-kNN Distance is more effective at detecting RL-phase contamination when pretraining contamination is lower, validating our hypothesis that RL-phase contamination is easier to detect in data with weaker pretraining contamination signals. Moreover, Min-kNN Distance can be used alongside common pretraining detection methods to help determine the training phase in which the data appeared.

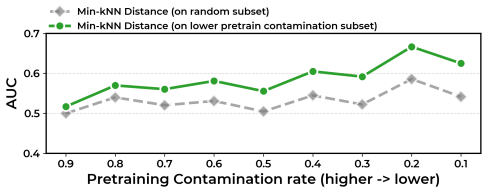


Figure 7: Dual-stage contamination analysis. The green line shows Min-kNN Distance RL data detection performance on the lower pretraining contamination subset, selected based on the bottom-q quantile of PPL scores. The gray line shows performance on the random-control subset, matched in size to control for sample size effects on AUC.

Extended Analyses. We defer detailed extended analyses to App. C. **Pre-RL Distribution Shift:** AUC increases from 0.60 on the base model (Qwen2.5-7B) to 0.75 after RLVR (ORZ), suggesting that RLVR amplifies a weak pre-existing structural signal (App. C.1). **Template Sensitivity:** Evaluating Min-kNN Distance under an unseen formatting template yields robust detection (AUC 0.78), demonstrating it captures deep intrinsic reasoning structures rather than overfitting to superficial output tags (App. C.2). **Cost and Scalability:** While multi-sample generation introduces expected overhead, evaluating 600 prompts on a 32B model takes only $\sim$15 minutes using vLLM on 8$\times$H800 GPUs. This confirms Min-kNN Distance maintains a highly favorable efficiency-accuracy trade-off in practice (App. D).

6 Related Work

Detect Data Exposure in Language Models. A growing body of work investigates whether large language models have been trained on specific data, driven by concerns about benchmark contamination, privacy leakage, and evaluation reliability. Most existing methods focus on pretraining or supervised fine-tuning, where data exposure shows up as statistical signals like low perplexity or

probability outliers [38, 8, 54, 46, 4, 23, 9, 21, 40]. Reinforcement learning post-training introduces a different challenge. In RLVR, models are optimized via reward feedback on self-generated reasoning, making likelihood-based methods less effective. Tao et al. [41] proposed self-critique to detect RL data contamination by identifying entropy collapse. Our work complements this by showing that RLVR induces a collapse in reasoning trajectory diversity for seen prompts, using structural similarity across multiple generations for RLVR data exposure detection, with a fully black-box approach.

Effects of Reinforcement Learning on Generation Diversity. Reinforcement learning with preference or alignment objectives consistently reduces generation diversity compared to supervised fine-tuning or non-preference methods. This effect has been observed across lexical, syntactic, semantic, and conceptual dimensions, indicating a compression of the model's output space [18, 29, 39, 27, 2, 34, 30]. Studies show that this reduction in diversity is linked to decreased creativity and exploratory behavior. For example, Mohammadi [25] show that RLHF lowers token-level entropy and embedding-space diversity, driving models toward fewer attractor states, which negatively impacts open-ended generation tasks. Casper et al. [5] discuss mode collapse in RL-trained models, attributing it to reward maximization. More recent work, like Murthy et al. [27], shows that RLHF and RLAIF methods reduce a model's ability to capture conceptual diversity in human responses, even with high surface-level quality. These findings suggest that diversity reduction is not merely a trade-off but a fundamental shift caused by reinforcement-based alignment. Unlike prior work, which focuses on diversity collapse as a trade-off in alignment, our study examines how this collapse manifests structurally in reasoning trajectories and uses it to detect RLVR training data exposure.

7 Conclusion

We investigate training data exposure in RL-post-trained reasoning models, identifying a structural collapse in reasoning trajectories during RLVR and proposing Min-kNN Distance, a black-box detection method leveraging this. Min-kNN Distance requires no access to training data or model internals. Experiments across various models and RL algorithms show Min-kNN Distanceconsistently outperforms baselines, providing a robust signal for contamination detection. Limitations and broader impact of our work are outlined in App. E and App. F.

Acknowledgements and Disclosures

This publication has been supported by the National Natural Science Foundation of China (NSFC) Key Project under Grant Number 62336006.

References

[1] Zeyuan Allen-Zhu. ICML 2024 Tutorial: Physics of Language Models, July 2024. Project page: `https://physics.allen-zhu.com/`.

[2] Oron Anschel, Alon Shoshan, Adam Botach, Shunit Haviv Hakimi, Asaf Gendler, Emanuel Ben Baruch, Nadav Bhonker, Igor Kviatkovsky, Manoj Aggarwal, and Gerard Medioni. Group-aware reinforcement learning for output diversity in large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 32382–32403, 2025.

[3] ByteDance-Seed. Beyondaime: Advancing math reasoning evaluation beyond high school olympiads. `https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME(https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME)`, 2025.

[4] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.

[5] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems

and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

[6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[7] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[8] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. *arXiv preprint arXiv:2402.15938*, 2024.

[9] Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. Practical membership inference attacks against fine-tuned large language models via self-prompt calibration. *arXiv preprint arXiv:2311.06062*, 2023.

[10] Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*, 2025.

[11] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL `https://arxiv.org/abs/2410.07985`.

[12] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*, 2024.

[13] Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*, 2023.

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

[15] Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

[16] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[17] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL `https://github.com/huggingface/open-r1`.

[18] Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

[19] Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025.

[20] Siyang Liu, Sahand Sabour, Yinhe Zheng, Pei Ke, Xiaoyan Zhu, and Minlie Huang. Rethinking and refining the distinct metric. *arXiv preprint arXiv:2202.13587*, 2022.

[21] Zhenhua Liu, Tong Zhu, Chuanyuan Tan, Bing Liu, Haonan Lu, and Wenliang Chen. Probing language models for pre-training data detection. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1576–1587, 2024.

[22] Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Christos Thrampoulidis, Leonid Sigal, and Renjie Liao. Leveraging online olympiad-level math problems for llms training and contamination-resistant evaluation. *arXiv preprint arXiv:2501.14275*, 2025.

[23] Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826, 2022.

[24] Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations*, 2024.

[25] Behnam Mohammadi. Creativity has left the chat: The price of debiasing language models. *arXiv preprint arXiv:2406.05587*, 2024.

[26] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 2(1):86–97, 2012.

[27] Sonia Krishna Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11241–11258, 2025.

[28] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

[29] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

[30] Ethan Perez, Sam Ringer, Kamile Lukosiute, Karina Nguyen, Edwin Chen, Scott Heiner, Craig Pettit, Catherine Olsson, Sandipan Kundu, Saurav Kadavath, et al. Discovering language model behaviors with model-written evaluations. In *Findings of the association for computational linguistics: ACL 2023*, pages 13387–13434, 2023.

[31] Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating llms on 2025 usa math olympiad. *arXiv preprint arXiv:2503.21934*, 2025.

[32] A Yang Qwen, Baosong Yang, B Zhang, B Hui, B Zheng, B Yu, Chengpeng Li, D Liu, F Huang, H Wei, et al. Qwen2. 5 technical report. *arXiv preprint*, 2024.

[33] Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, 2023.

[34] Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. Whose opinions do language models reflect? In *International conference on machine learning*, pages 29971–30004. PMLR, 2023.

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

[36] ByteDance Seed, Yuyu Zhang, Jing Su, Yifan Sun, Chenguang Xi, Xia Xiao, Shen Zheng, Anxiang Zhang, Kaibo Liu, Daoguang Zan, et al. Seed-coder: Let the code model curate data for itself. *arXiv preprint arXiv:2506.03524*, 2025.

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

[38] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. *arXiv preprint arXiv:2310.16789*, 2023.

[39] Stewart Slocum, Asher Parker-Sartori, and Dylan Hadfield-Menell. Diverse preference learning for capabilities and alignment. In *The Thirteenth International Conference on Learning Representations*, 2025.

[40] Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for llm benchmark data contamination. *arXiv preprint arXiv:2503.16402*, 2025.

[41] 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. *arXiv preprint arXiv:2510.09259*, 2025.

[42] ByteDance Seed Team. Seed-oss open-source models. `https://github.com/ByteDance-Seed/seed-oss`, 2025.

[43] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

[44] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.

[45] Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

[46] Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. Recall: Membership inference via relative conditional log-likelihoods. *arXiv preprint arXiv:2406.15968*, 2024.

[47] Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

[48] Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

[49] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[50] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[51] Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

[52] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

[53] Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, et al. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836, 2024.

[54] Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. Min-k%++: Improved baseline for detecting pre-training data from large language models. *arXiv preprint arXiv:2404.02936*, 2024.

A Analysis Details

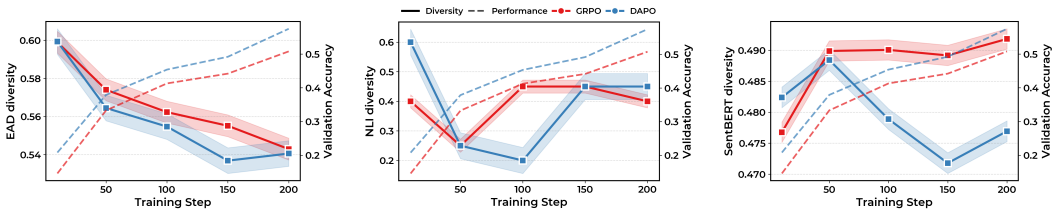
A.1 Training Details

For the analysis model, we train Qwen-2.5-7B-Base using the GRPO and DAPO algorithms on the DAPO training data. The training hyperparameters are shown in Tab. 6. The model is trained for a total of 200 rollout steps, which is approximately equivalent to 6 epochs.

Table 6: Hyperparameters used for training Qwen-2.5-7B-Base on GRPO and DAPO.

HyperParameter	**GRPO/DAPO**
Actor learning rate	1.0×10^{-6}
Train batch size	512
Max prompt length	2048
Max generation length	20480
Temperature	1.0
Samples per prompt	16
Tensor model parallel	4
Mini batch size	32
Use KL loss	No

A.2 Cross-Input Generation Diversity



(a) EAD (b) NLI (c) Embedding

Figure 8: Evolution of generation diversity during RLVR training under DAPO and GRPO, measured in a cross-input setting by three complementary metrics capturing lexical (EAD), logical (NLI), and semantic (embedding) diversity.

A.3 Prompt For Labeling

Labeling Rigid N-gram Prompt

```
{
    "task": (
        "You will be given: (1) a problem statement, (2) one sample model
            answer for that problem, and (3) a list of n-grams extracted
            from multiple model answers to the same problem. Your job is to
            label EACH n-gram with exactly one category from: restatement,
            logic, boilerplate, other.

        Labeling rules:
        1) restatement: the n-gram repeats or paraphrases the problem
            statement. If the n-gram appears verbatim in the problem
            statement, it MUST be labeled restatement.
        2) logic: the n-gram expresses problem-specific reasoning, math
            relations, formulas, constraints, or derived quantities.
        3) boilerplate: generic reasoning template language (e.g., 'we need
            to', 'let us', 'therefore', 'in conclusion'), or
            domain-agnostic filler.
        4) other: everything else that does not fit the above.

        Important:
        - Use the problem statement and the sample answer as context.
        - The sample answer includes inline markers [[...]] around matching
            n-grams.
        - Do not invent new n-grams or change the provided strings.
        - Output ONLY a JSON object mapping each input n-gram to one of the
            labels.
        - Include all n-grams, even if uncertain.
        - Keep the n-gram strings exactly as given."
    ),
    "problem": problem,
    "sample_answer": annotated_answer,
    "ngrams": ngrams,
}
```

A.4 Human Validation of LLM Labeling Robustness

We utilize GPT-4o to categorize frequent 3-grams into four structural types: *Restatement*, *Logic*, *Boilerplate*, and *Other*. To assess the robustness and reliability of this LLM-based labeling, we conducted a human validation study.

We uniformly sampled 100 instances from the extracted 3-grams and invited three PhD-level researchers with expertise in reasoning models to independently assign categories. We then measured the inter-annotator agreement among the human experts, as well as the agreement between the human consensus (majority vote) and the GPT-4o predictions. The statistical results, measured using Cohen's κ and Fleiss' κ, are summarized in Table 7 and Table 8.

Table 7: Agreement metrics including all categories (Restatement, Logic, Boilerplate, Other).

Metric	**Value**
Human–Human (avg. pairwise Cohen's κ)	0.6758
Human–Human (Fleiss' κ)	0.6734
Human (majority) vs GPT (Cohen's κ)	0.6239

We observed that the "Other" category mainly corresponds to boundary or ambiguous cases (e.g., subtle transitions between reasoning steps), where both human annotators and GPT-4o exhibited

higher disagreement. When excluding the 20 samples classified as "Other", the agreement scores improved significantly, as shown in Table 8.

Table 8: Agreement metrics excluding the ambiguous "Other" category.

Metric	**Value**
Human–Human (avg. pairwise Cohen's κ)	0.8033
Human–Human (Fleiss' κ)	0.8026
Human (majority) vs GPT (Cohen's κ)	0.8100

The strong agreement ($\kappa \approx 0.80$) on the core structural categories demonstrates that the reasoning space collapse identified in our qualitative analysis is highly consistent and robust.

A.5 Case on Rigidity

Fig. 9 presents the final checkpoint of GRPO training, showing three answers to a single prompt, with n-grams that appear more than 8 times highlighted.

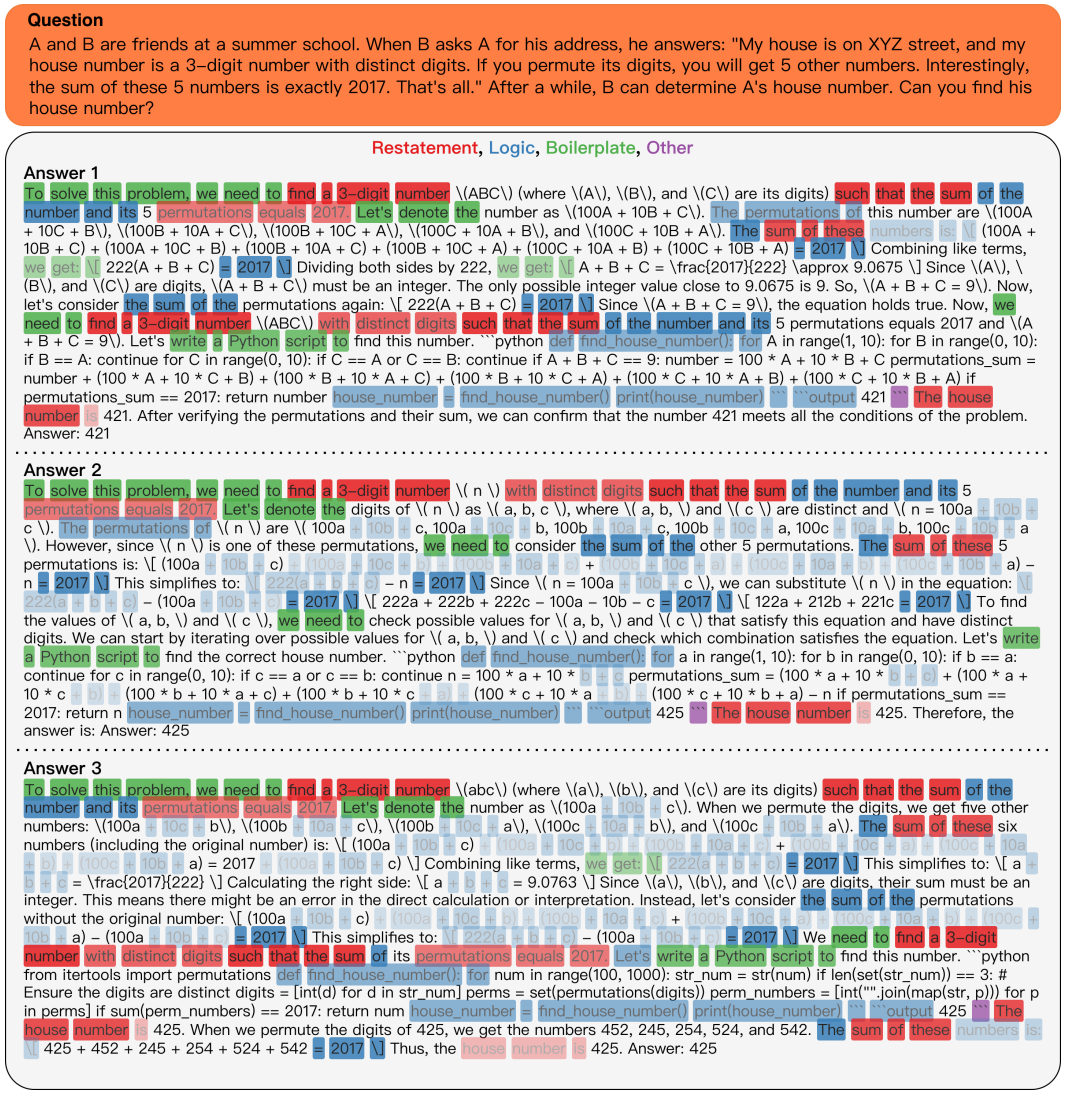


Figure 9: Example of repeated n-grams extracted from completions of a single prompt. High-frequency n-grams naturally cluster into four categories: restatement, symbolic or algebraic logic steps, boilerplate phrases, and other tokens. The shading of the n-grams indicates their frequency, with darker colors representing higher repetition.

B Experimental Details

B.1 Training Details

For the RL-MIA benchmark, we follow Tao et al. [41] to train Qwen2.5-7B-Instruct and DeepSeek-Math- 7B-Instruct with the following hyperparameters:

Table 9: Hyperparameter settings for Qwen2.5-7B-Instruct and Deepseek-math-7b-Instruct in RL-MIA.

Parameter	Qwen2.5-7B-Instruct	Deepseek-math-7b-Instruct
Actor learning rate	1.0×10^{-6}	1.0×10^{-6}
Train batch size	128	128
Max prompt length	1024	1024
Max generation length	4096	3072
Temperature	1.0	1.0
Samples per prompt (n)	8	8
Tensor model parallel (TP)	2	2
Mini batch size	2	2
Use KL loss	No	No

B.2 Data Details

For the evaluation of open-source RLVR models, we construct a balanced evaluation set consisting of 300 seen (member) instances and 300 unseen (non-member) instances. The 300 unseen reasoning problems are meticulously sourced to avoid overlap with the RL training data of any evaluated model. Specifically, the unseen set is composed of: AIME 2024 (30 problems), AIME 2025 (30 problems), Beyond-AIME [3] (100 problems), Omni-Math [11] (100 problems), and MATH-500 [44] (40 problems). This balanced composition provides a clean and fair benchmark for detecting RLVR data in real-world models.

For the controlled RL-MIA benchmark setting, we follow the data partition protocol established by Tao et al. [41]. We utilize the SAT and K&K datasets, injecting a specific subset of evaluation instances into the base RL training corpus to serve as member (seen) data. Additional synthetic or unexposed samples are reserved as non-member (unseen) data. The detailed statistics of these splits, including the base corpus size, the number of injected items, and the total detection tasks, are summarized in Table 10.

Table 10: RL-MIA data splits for training and evaluation (Selected).

Source	Base RL Corpus (size)	Injected items	Train Size	Occurrences	Detection Tasks
K&K	K&K train: **950**	K&K test: **50**	950 + 50	3	**100**
SAT	SAT train: **450**	SAT test: **50**	450 + 50	3	**100**

B.3 Implementation Details

In all experiments, we use vLLM and deploy all models on 8 H800 GPUs with TP=8 or TP=4. During sampling, the default temperature is set to 0.7, and top_p is set to 0.95. Each model's output is capped at a max token length of 1024. We found that this token limit provides sufficient detection performance while also speeding up detection time.

B.4 Baseline Mechanisms

In the main text, we compare Min-kNN Distance against several representative membership inference baselines. Most of these methods were originally proposed for detecting pretraining data, while Self-Critique targets the RL post-training phase:

- **PPL:** Uses the standard perplexity of the prompt as a proxy for memorization. Lower perplexity indicates a higher likelihood of the data being seen during training.
- **Min-K% Prob [38]:** Computes the average probability of the $k\%$ tokens with the lowest likelihood in the sequence, based on the assumption that unseen examples contain more low-probability outlier tokens.
- **Min-K%++ [54]:** An improved extension of Min-K% that calibrates token probabilities by normalizing them against the unconditional probability of the tokens, reducing the impact of naturally rare words.

- **Recall [46]:** Measures likelihood shifts under unrelated prefixing. It compares the marginal likelihood of the sequence with its conditional likelihood given a generic prefix.
- **CDD [8]:** Compares the edit distance between stochastically sampled generations and greedy sampling. A sharp peak in the distance distribution or unusually high similarity suggests the model has memorized the exact sequence.
- **Self-Critique [41]:** Specifically designed for RL data contamination detection. It compares the token-level entropy sequences between a model's initial generated response and its subsequent self-critique. High similarity in the entropy profiles is interpreted as evidence of RLVR contamination.

C Extended Analyses and Control Experiments

C.1 Controlling for Pre-RL Distribution Shift

To examine how RLVR changes the detection signal, we compare performance on the same data split before (Qwen2.5-7B) and after (ORZ) RLVR training. As shown in Table 11, pre-RL performance reaches an AUC of 0.60, indicating a weak pre-existing signal, while post-RL performance increases to 0.75. This increase suggests that RLVR amplifies the structural signal, although it does not establish that the signal originates exclusively from RLVR.

Table 11: Detection performance before and after RLVR training.

Setting	AUC
Pre-RL (Qwen2.5-7B Base)	0.60
Post-RL (ORZ)	0.75

C.2 Sensitivity to Fixed Output Templates

RLVR models often use fixed formatting tags (e.g., `<think>`). To ensure Min-kNN Distance captures underlying reasoning structures rather than superficial templates, we evaluated ORZ using an unseen prompt format (the Qwen-DAPO template). As shown in Table 12, performance remains stable and even improves ($0.75 \rightarrow 0.78$), demonstrating our method's robustness against formatting variations.

Table 12: Detection performance under different prompt templates.

Setting	AUC
ORZ (Original Template)	0.75
ORZ (DAPO Template)	0.78

C.3 Blind Baselines on the Open-Source Split

To examine whether simple length or difficulty features distinguish seen and unseen prompts, we evaluate three blind baselines on the DAPO evaluation split: prompt token length, base-model response token length, and base-model solve rate. These baselines do not query the RLVR-trained model. As shown in Table 13, prompt length performs near chance, while base-response length and solve rate provide weak separation (AUC 0.575 and 0.588). All three baselines perform below Min-kNN Distance (AUC 0.72), showing that these individual features provide weaker detection signals on the same split.

C.4 Controlling for Difficulty and Correctness

To examine whether the detection signal is explained by problem difficulty or correctness, we stratify the DAPO evaluation prompts by the model's observed solve rate. We evaluate a balanced subset with equal numbers of seen and unseen prompts within each stratum, a full-set conditional AUC, and the subset of prompts for which all sampled completions are incorrect. The conditional AUC averages within-stratum AUCs, weighted by the number of seen–unseen pairs in each stratum.

Table 13: Blind baseline performance on the DAPO open-source split. The 95% confidence intervals are the 2.5th and 97.5th percentiles of AUCs obtained by separately bootstrapping seen and unseen prompts.

Feature	AUC	95% CI
Prompt token length	0.495	[0.447, 0.540]
Base-response token length	0.575	[0.530, 0.621]
Base solve rate	0.588	[0.546, 0.633]

Table 14: Detection performance after controlling for observed solve rate on DAPO. Counts indicate the numbers of seen and unseen prompts. We report AUCs with 95% confidence intervals.

Evaluation	Seen/Unseen	PPL	Min-kNN Distance
Solve-rate matched	250/250	0.696 [0.650, 0.744]	0.708 [0.661, 0.755]
Conditional (full set)	300/300	0.704 [0.657, 0.749]	0.731 [0.686, 0.774]
Consistently unsolved	202/250	0.707 [0.661, 0.754]	0.735 [0.690, 0.780]

Confidence intervals are obtained by separately bootstrapping seen and unseen prompts and taking the 2.5th and 97.5th percentiles of the resulting AUCs, with additional solve-rate stratification for conditional analyses. As shown in Table 14, Min-kNN Distance retains AUCs of 0.708–0.735, including on consistently unsolved prompts. This suggests that the signal is not solely explained by correctness or observed difficulty. PPL also performs well under these controls, with AUCs close to those of Min-kNN Distance. Topic and intrinsic solution-path diversity remain imperfectly controlled in the open-source split.

C.5 Comparison with Greedy-Anchored Statistics

To examine the contribution of the aggregation statistic, we compare Min-kNN Distance with CDD and two continuous greedy-anchored variants on DAPO. CDD uses a thresholded-peak statistic around a greedy completion. The continuous variants retain this anchor but compute either the mean distance or the mean of the 10 smallest distances from sampled completions to the greedy completion. All variants use the same prompts, decoding setup, distance normalization, and sampled completions.

Table 15: Comparison of aggregation statistics on DAPO. The continuous greedy-anchored variants are diagnostic ablations rather than the original CDD statistic.

Statistic	AUC
CDD	0.52
Greedy mean distance	0.691
Greedy Min-10 distance	0.701
Min-kNN Distance	0.72

As shown in Table 15, continuous aggregation recovers most of the detection signal, while Min-kNN Distance provides a further improvement. This suggests that CDD's thresholding discards useful convergence information, and that anchor-free nearest-neighbor aggregation better captures the local concentration of multiple structural modes.

D Cost Analysis

We calculate the average detection time per sample using 32 completions on Open-Reasoner-Zero-7B, as shown in Tab. 16. While Min-kNN Distance requires more time compared to methods like PPL and DIME, with 6.65 seconds per item for 32 completions, the increased time is expected given the need to sample multiple completions for structural diversity analysis. This time is still reasonable and comparable to existing methods, making Min-kNN Distance an effective approach for RLVR data exposure detection.

Table 16: Comparison of average detection latency per sample.

Method	Avg. Latency (s/item)
PPL / Min-K / Min-K++	1.43
DIME	2.22
Self-Critique	2.67
CDD / Consistency	4.29
Min-kNN Distance ($n = 16$)	4.56
Min-kNN Distance ($n = 32$)	6.65

E Limitations

While Min-kNN Distance provides an effective solution for detecting RLVR exposure, there are some limitations. It relies on the assumption that RLVR-induced structural collapse is observable across various models and tasks. Additionally, the method requires generating multiple completions per prompt, which can increase computational cost. Finally, Min-kNN Distance focuses on detecting structural collapse but does not directly address broader concerns like potential biases in training data.

F Broader Impact

This work advances the trustworthy evaluation of large language models (LLMs). As Reinforcement Learning with Verifiable Rewards (RLVR) becomes a standard paradigm for training advanced reasoning models, undisclosed training data raises severe concerns about benchmark contamination. By providing a simple, black-box method to detect RLVR exposure, our work empowers the open-source community to independently audit models. This promotes transparency, ensures fair comparisons on leaderboards, and helps verify whether models have genuinely acquired generalized reasoning capabilities or merely memorized fixed structural patterns.

NeurIPS Paper Checklist

1. **Claims**
   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Answer: [Yes]
   Justification: We have included the claims made in the abstract and introduction.
   Guidelines:
   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.
2. **Limitations**
   Question: Does the paper discuss the limitations of the work performed by the authors?
   Answer: [Yes]
   Justification: We have added a separate Limitations section in appendix.
   Guidelines:
   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.

- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We have fully disclosed all the information needed to reproduce the main experimental results.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.

- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The code and data will be all open-sourced.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: We specified all testing details in the paper.

Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Results are averaged over multiple runs; details of variability are reported in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We have provided the computer resources needed to reproduce the experiments.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: We have added a separate Ethical and Broader Impact section in appendix.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper posed no such risk.

    Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: We have cited the original papers that produced the code packages and datasets.

Guidelines:

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

13. **New assets**

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

Answer: [Yes]

Justification: We added details of the code as part of submission via structured templates.

Guidelines:

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

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

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

Answer: [Yes]

Justification: We describe the human annotation procedure and validation details in appendix (Human Validation of LLM Labeling Robustness).

Guidelines:

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

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

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

Answer: [No]

Justification: The annotation study involved minimal risk and did not require/obtain IRB approval; participants were PhD researchers.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLM is not used for the core methodology, scientific rigor, or originality of the research

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.