DVD: A Robust Method for Detecting Variant Contamination in Large Language Model Evaluation

Anonymous ACL submission

Abstract

Evaluating large language models (LLMs) is increasingly confounded by *variant contamination*: the training corpus contains semantically equivalent yet lexically or syntactically altered versions of test items. Unlike verbatim leakage, these paraphrased or structurally transformed variants evade existing detectors based on sampling consistency or perplexity, thereby inflating benchmark scores via memorization rather than genuine reasoning. We formalize this problem and introduce **DVD** (**D**etection via **V**ariance of generation **D**istribution), a single-sample detector that models the local output distribution induced by temperature sampling. Our key insight is that contaminated items trigger alternation between a *memory-adherence* state and a *perturbation-drift* state, yielding abnormally high variance in the synthetic difficulty of low-probability tokens; uncontaminated items remain in drift with comparatively smooth variance. We construct the first benchmark for variant contamination across two domains Omni-MATH and SuperGPQA by generating and filtering semantically equivalent variants, and simulate contamination via fine-tuning models of different scales and architectures (Qwen2.5 and Llama3.1). Across datasets and models, **DVD** consistently outperforms perplexity-based, Min-k%++, edit-distance (CDD), and embedding-similarity baselines, while exhibiting strong robustness to hyperparameters. Our results establish variance of the generation distribution as a principled and practical fingerprint for detecting variant contamination in LLM evaluation. Code is available at `https://anonymous.4open.science/r/DVD2-7D97`.

1 Introduction

In recent years, large language models (LLMs) have exhibited explosive growth in capability, demonstrating transformative potential across a wide range of domains (Brown et al., 2020; Team et al., 2024; Touvron et al., 2023; Chowdhery et al., 2023; Achiam et al., 2023). However, their impressive performance relies heavily on massive web-scale corpora, which has brought a long-standing challenge into sharper focus: *data contamination* (Balloccu et al., 2024; Li et al., 2023; Chang et al., 2024; Cheng et al., 2025; Deng et al., 2023; Xu et al., 2024). Data contamination refers to unintended overlap between training data and evaluation benchmarks, which severely undermines the validity of empirical evaluation (Cheng et al., 2025). Such overlap can create an illusion of strong generalization and mislead research progress. When contaminated models are deployed in scientific investigations or real-world applications, their latent biases and hidden flaws may lead to incorrect scientific conclusions or catastrophic decisions, ultimately hindering technological advancement (Sainz et al., 2023).

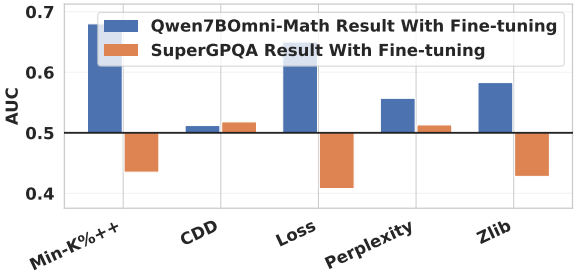

Figure 1: Performance of traditional contamination detection methods on the variant contamination identification task. In the figure, an AUC value below 0.5 indicates that the models predictions are inversely correlated with the true labels, while an AUC close to 0.5 suggests performance equivalent to random guessing.

With the widespread adoption of large-scale data augmentation and synthetic data generation (e.g., GPT-4o), a more subtle and potentially more dangerous form of contamination has gradually emerged, namely *variant contamination*. Vari-

ant contamination occurs when training data contains instances that are semantically equivalent to benchmark questions but have been rewritten at the lexical or structural level. Unlike exact duplicates, such variants can evade existing detection methods while still enabling models to effectively "memorize" the answers. To systematically study this phenomenon, we construct a new evaluation benchmark based on Omni-MATH (mathematical reasoning) (Gao et al., 2024) and SuperGPQA (general reasoning) (Du et al., 2025) by generating semantically equivalent variants through controlled transformations. Fine-tuning models of different scales and architectures on these contaminated datasets yields striking results: even when the training data contains only variants (with no exact duplicates), models can still achieve significantly inflated accuracy on the evaluation benchmarks. More importantly, widely used contamination detection methods fail in this setting and exhibit unstable performance (Figure 1). Existing approaches, such as perplexity-based metrics, Min-K%, and CDD, primarily rely on shallow surface-level features, including token probability distributions, embedding similarity, or surface perplexity patterns. However, prior work has shown that LLMs can be highly sensitive to minor phrasing changes, often leading to substantially different response behaviors (Lunardi et al., 2025; Sclar et al., 2023; Zhao et al., 2024). Consequently, in the variant contamination scenario, although the questions are semantically equivalent to the benchmarks, their carefully restructured surface forms weaken these shallow cues, making it difficult for existing methods to capture the true behavioral differences exhibited by models. This observation is consistent with our empirical findings.

To address these limitations, we propose **DVD** (*Detection via Variance of generation Distribution*), a variant contamination detection method based on the variance of generation distributions. DVD directly characterizes the core behavioral signatures induced by variant contamination by modeling fluctuations in the model's generation distribution across multiple stochastic decoding runs. Unlike existing methods that rely on static surface-level features, DVD focuses on dynamic response patterns in the model's uncertainty space. For uncontaminated questions, genuine reasoning processes typically produce relatively stable and smooth variance in the output distribution. In contrast, for contaminated questions, models frequently alternate between a *high-confidence memorization regime* and a *low-confidence exploratory reasoning regime*, resulting in pronounced distributional differences in the generation variance (Figure 2). By exploiting these dynamic behavioral differences, DVD is able to penetrate surface-level reformulations and directly identify contamination-induced anomalies, enabling robust and effective detection in the highly challenging variant contamination setting.

Extensive experiments demonstrate that DVD consistently outperforms baseline methods across datasets, domains, and model scales. For example, on SuperGPQA, DVD improves AUC by up to 0.22 over the strongest baseline (embedding similarity), while maintaining stable performance across model sizes from 1.5B to 32B parameters, as well as across both Qwen and Llama architectures. These results collectively establish DVD as a robust and efficient solution to the overlooked yet critical problem of variant contamination.

Our contributions are summarized as follows:

A Benchmark for Systematic Evaluation. We construct the first benchmark specifically designed for variant contamination detection, covering two representative domains: mathematical reasoning and general reasoning. Through controlled variant generation and filtering, this benchmark enables rigorous, reproducible evaluation of contamination detection methods across different models, scales, and target domains.

A Novel Detection Framework. We propose DVD, a training-free variant contamination detection method that relies solely on model generation behavior. By analyzing the variance of output distributions across multiple stochastic decoding runs, DVD captures anomalous fluctuations exhibited by models on contaminated queries, effectively penetrating surface-level paraphrasing to detect variant contamination. Experiments show that DVD significantly outperforms existing methods across models, scales, and domains, while remaining highly robust to decoding hyperparameters.

2 Related Work

Existing approaches for data contamination detection can be broadly divided into two categories.

Sampling and Output-Matching-Based Methods This line of research primarily relies on

the similarity between model generations and reference answers, or on detecting anomalous patterns within the output distribution. Representative works include reference-instance matching based on overlap measures (Golchin and Surdeanu, 2023); the CDD method, which conducts multiple random samplings alongside one greedy decoding under the same prompt, and uses the edit distance between greedy and stochastic outputs to approximate the output distribution and detect sharp modes caused by memorization (Khandelwal et al., 2019); and the DCQ method, which compares model preferences between original inputs and their perturbed variants to identify contamination (Golchin and Surdeanu, 2025). Moreover, membership inference has also been applied in this context, where the loss difference between a target sample and synthetic neighbors serves as an indicator of contamination (Mattern et al., 2023). Overall, these methods are effective for detecting verbatim memorization, yet remain limited by their reliance on shallow surface-level measures.

Perplexity-Based Methods In contrast to sampling- and matching-based approaches, another class of methods focuses on detecting contamination through the abnormally high confidence that models assign to seen samples. For example, the MIN-K% PROB method examines the average log-likelihood of low-probability tokens to determine whether a sample appears in the training set (Shi et al., 2023). Similarly, (Oren et al., 2023) demonstrates that a models ability to recall the order of training samples itself constitutes strong evidence of data leakage. Compared to the former category, perplexity-based methods provide a more direct quantification of model bias toward training data. However, their effectiveness is likewise constrained to verbatim memorization; once samples undergo semantic rewriting or structural perturbation, perplexity-level differences are often largely obscured, leading to a significant drop in detection performance.

Our Approach Motivated by the limitations of the above methods, we propose the DVD approach, which overcomes the dependence on shallow similarity measures or overall perplexity levels. Although CDD also relies on multiple samplings to construct an output distribution, its core remains restricted to edit-distance-based comparisons, failing to capture the true probabilistic dynamics underlying text generation. In contrast, DVD employs temperature sampling to generate multiple responses and systematically analyzes the variance of low-probability tokens, defined as synthetic difficulty. The key insight is that contaminated samples alternate between a "memorization-dependent state" and a "perturbation-drift state," resulting in substantially higher variance across generations. Uncontaminated samples, by contrast, remain consistently in the drift state, with variance reflecting only natural noise. By incorporating variance decomposition into a mixture-distribution framework, DVD fundamentally captures these deep probabilistic dynamics, thereby achieving superior performance in detecting semantic-variant contamination compared to existing methods.

3 Variant Contamination

This section introduces the formal definition of the **Variant Contamination Detection (VCD)** task (3.1) and describes in detail the construction of a benchmark dataset tailored for reliable variant contamination detection (3.2).

3.1 Task Definition

We define **variant contamination** as the scenario in which, during training, a model is exposed to samples that are logically equivalent to those in the test set but differ in surface form. Such variants may diverge in semantics, syntax, or narrative style, yet preserve the same underlying solution space, thereby allowing the model to perform as if it had previously observed the test instance.

Formally, let x denote a test instance and let f be a semantic abstraction function that extracts the core informational content of x. A variant of x is then rigorously defined as follows:

$$v = \tau(x), \quad \text{such that } f(v) = f(x), \qquad (1)$$

where τ is a transformation preserving the core semantics of x. If such a variant v appears in the training corpus of model M, we say that M is contaminated on test instance x. Importantly, unlike exact duplicates, variants may differ substantially from x in vocabulary, phrasing, or narrative structure, while remaining equivalent in required knowledge, logical dependencies, and trajectory.

The goal of the VCD task is thus to reliably identify, within a models test set, which instances x have been subject to contamination by semantically equivalent variants present in training.

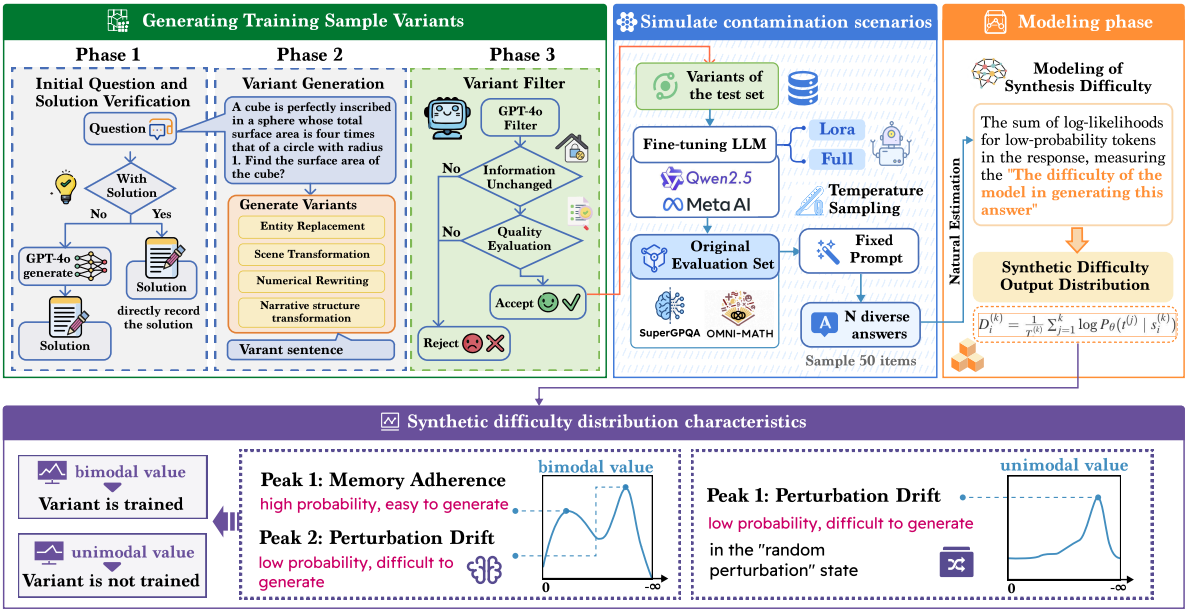

Figure 2: Our Method Pipeline

3.2 Benchmark Construction

Variant contamination commonly arises in the context of data augmentation. To systematically evaluate the extent of variant contamination in large language models (LLMs), we construct a dedicated benchmark dataset. The construction leverages mainstream data augmentation techniques (Shorten and Khoshgoftaar, 2019; Shorten et al., 2021; Maharana et al., 2022) and incorporates two widely used benchmarks: Omni-Math (Gao et al., 2024) and SuperGPQA (Du et al., 2025). As illustrated in Figure 2, we employ GPT-4o (Hurst et al., 2024) to produce semantically equivalent variants from the original problem.

Initial Question and Solution Verification In the initial stage, we first verify the original problemsolution pair (x, y). If the problem already comes with a standardized solution, it directly proceeds to the next step; Otherwise (e.g., in the SuperGPQA dataset), GPT-4o (Hurst et al., 2024) is employed to generate gold-standard answers. This ensures each problem is paired with a reference solution, forming the foundation for subsequent variant generation. Formally, given the training set:

$$D = \{(x_i, y_i)\}_{i=1}^{N}, \tag{2}$$

we take (x_i, y_i) as input in preparation for generating corresponding variants.

Variant Generation In this stage, we adopt mainstream data augmentation techniques (Shorten and Khoshgoftaar, 2019; Shorten et al., 2021; Maharana et al., 2022) to generate a set of semantically equivalent variants (x_v, y_v) for each original problem (see Table 1 and Figure 11). Specifically, we define a transformation set:

$$T = \{T_{\text{ent}}, T_{\text{scn}}, T_{\text{num}}, T_{\text{nar}}\}, \tag{3}$$

covering four categories: entity substitution, scenario conversion, numerical rewriting, and narrative restructuring. Through these surface-level transformations, we construct the variant set:

$$V(x_i) = \{v_i^{(1)}, \ldots, v_i^{(m)}\}, \quad \text{where } f(v_i^{(j)}) = f(x_i). \tag{4}$$

To guarantee semantic equivalence and correctness, rejection sampling is applied during generation (see Figure 10), with GPT-4o providing high-quality candidate variant answers.

Variant Filter Finally, GPT-4o is employed as a filter to conduct quality control over the generated variants (Liu et al., 2025). The filtering procedure consists of two steps: first, checking whether the information remains unchanged; second, performing a quality evaluation of the solution. Only when both conditions are satisfied is the variant pair (x_v, y_v) accepted. Ultimately, these high-quality variant samples are injected into the training set to simulate test contamination, enabling systematic evaluation of whether existing detection methods can identify variant-contaminated test instances.

Table 1: Variant generation strategies used to simulate contamination.

Method	Description
Entity substitution	Replace referents, variable names, and object categories while maintaining consistency in type and context.
Scenario transformation	Alter the background setting and narrative context, while preserving logical dependencies and constraint structures.
Numerical rewriting	Resample parameters under solvability constraints and update derivations and intermediate values for consistency.
Narrative structure transformation	Rearrange syntax or rewrite step-by-step analysis into a paragraph-style narrative while preserving semantic meaning.

4 Method

This paper proposes a method named DVD (**D**etection via **V**ariance of generation **D**istribution) grounded in modeling the distribution of model outputs. The core idea is to generate multiple responses under a fixed prompt using temperature sampling, thereby capturing fluctuations in low-probability regions of the models output distribution. These fluctuations serve as key signals for detecting contamination. More specifically, when a test sample appears in the training set, the model may operate in two distinct generative states. The first is memory adherence, where generation is guided by memorized templates internalized during training. The second is perturbation drift, where generation is primarily driven by stochastic perturbations introduced by temperature sampling, leading to free-form exploratory outputs. Memory adherence reflects the models reliance on training-based recall, while perturbation drift captures the natural randomness of unconstrained generation. If a test sample is contaminated, the model alternates between these two states, producing substantial variability in the conditional likelihoods of low-probability tokens. In contrast, for uncontaminated samples, the absence of reliable memory templates constrains the model to remain in a perturbation drift state, where tail-token probabilities mainly reflect inherent noise and thus exhibit only minor fluctuations. Based on this observation, we design the variance of synthetic difficulty as the contamination detection criterion.

4.1 Temperature Sampling

For each test sample x_i, we apply temperature sampling at test time under a fixed prompt p to generate N candidate responses $\{a_i^{(1)}, a_i^{(2)}, \ldots, a_i^{(N)}\}$. Each response is concatenated with the prompt to form the complete input:

$$s_i^{(k)} = (p, a_i^{(k)}). \tag{5}$$

Temperature sampling introduces stochastic perturbations, enabling the collection of diverse outputs for the same test sample. In uncontaminated cases, generation consistently remains in a perturbation drift state, and temperature perturbations do not alter the statistics of low-probability tokens. In contaminated cases, however, generation alternates between memory adherence and perturbation drift. Temperature perturbations amplify the disparity between template-based and non-template-based responses, causing tail tokens to exhibit more pronounced fluctuations.

4.2 Synthetic Difficulty Modeling

To quantify such fluctuations, we define the notion of **synthetic difficulty**. For each sequence $s_i^{(k)}$, we select the k least probable tokens in the response, compute the sum of their log-likelihoods, and normalize by sequence length $T_i^{(k)}$:

$$D_i^{(k)} = \frac{1}{T_i^{(k)}} \sum_{j=1}^{k} \log P_\theta\left(t^{(j)} \mid s_i^{(k)}\right). \tag{6}$$

This statistic captures local uncertainty in the tail region of the distribution. Unlike global perplexity, tail-token probabilities are more sensitive to the presence of training-set memorization. If a test sample is contaminated, $D_i^{(k)}$ varies markedly across generations due to the alternation between memory adherence and perturbation drift. If uncontaminated, tail probabilities primarily reflect noise, yielding relatively stable values of $D_i^{(k)}$ across multiple generations.

Given the synthetic difficulty set $\{D_i^{(1)}, D_i^{(2)}, \ldots, D_i^{(N)}\}$, we define the DVD

indicator as their sample variance:

$$\mathrm{DVD}_i = \frac{1}{N}\sum_{k=1}^{N}\left(D_i^{(k)} - \overline{D}_i\right)^2, \quad \overline{D}_i = \frac{1}{N}\sum_{j=1}^{N} D_i^{(j)}. \tag{7}$$

This indicator characterizes the fluctuation of synthetic difficulty. According to the variance decomposition principle for mixture distributions, if a test sample is contaminated, the distribution of synthetic difficulty can be regarded as a mixture of memory states and drift states, which differ in expectation, thereby inflating the overall variance. If uncontaminated, synthetic difficulty arises from a single state, and variance remains low.

More specifically, contaminated samples can be modeled as a mixture of two latent generation states: the memory-adhering state ($Z = M$) dominated by training memorization, and the unconstrained perturbation-drift state ($Z = U$). Let $\pi_M = \Pr(Z = M)$, $\pi_U = \Pr(Z = U)$, with $\pi_M + \pi_U = 1$. Then,

$$\mu = \pi_M \mu_M + \pi_U \mu_U, \tag{8}$$

$$\mathrm{Var}(X) = \pi_M\left(\sigma_M^2 + (\mu_M - \mu)^2\right) + \pi_U\left(\sigma_U^2 + (\mu_U - \mu)^2\right). \tag{9}$$

Here, μ_M and μ_U denote the expectations under the memory and drift states, respectively. Since the memory state relies on templates encountered during training, its synthetic difficulty is generally lower than that of the drift state, i.e., $\mu_M > \mu_U$ empirically. By the decomposition of within-group and between-group variance, if the two states differ substantially in expectation, the overall variance of the mixture will exceed that of a single distribution. This theoretical grounding demonstrates the effectiveness of our method in distinguishing contaminated from uncontaminated samples.

5 Experiments

In this section, we simulate a variant-contamination scenario based on the constructed variant dataset (see Section 3.2) and perform a comprehensive evaluation of our method against a range of baseline approaches under this setting.We exclude closed-source models from our study because they are not practically trainable/fine-tunable, making it difficult to simulate the variant-contamination setting. Detailed experimental configurations are provided in Section 5.1, large language model fine-tuning details are described in Section A.1, and the experimental results are reported in Section 5.2.

5.1 Experimental Setup

Model selection: To comprehensively assess the robustness of our variant-contamination detection method, we compare models along multiple dimensions: parameter scale (Qwen2.5-1.5B-Instruct (Team, 2024), Qwen2.5-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Team, 2024), Qwen2.5-32B-Instruct (Team, 2024)), architecture (Qwen2.5 vs. Llama3.1 (Dubey et al., 2024)), and fine-tuning strategy (full-parameter fine-tuning vs. LoRA fine-tuning).

Baselines: To validate the effectiveness of our method, we compare it with the following baselines: 1)*Embedding Similarity* (Dong et al., 2024): computes the similarity between answers using embeddings produced by the base model; 2)*Perplexity* (Li et al., 2023): computes the perplexity of the original answer given the prompt; 3)*Min-k% Probability* (Shi et al., 2023): computes the average probability over the lowest k% token probabilities of the original answer given the prompt; 4)*Min-k%++ Probability* (Zhang et al., 2025): an enhanced variant of Min-k%, which normalizes and calibrates token log-probabilities using statistics (mean and standard deviation) of the class distribution over the model vocabulary, and takes the average over the lowest k% calibrated scores as the detection score; 5)*CDD* (Dong et al., 2024): measures the sharpness of the output distribution via edit distance; 6)*Zlib* (Zhang et al., 2025): computes the Zlib compression entropy of the original answer given the prompt; 7)*Loss* (Zhang et al., 2025): computes the loss of the original answer given the prompt. The hyperparameters specific to our method were set as follows: the number of minimum-probability tokens *k* was fixed at 20, and the number of samples *N* was set to 50.

5.2 Experimental Results

We evaluate the performance of our proposed method, **DVD**, against several baselines on two distinct datasets: *Omni-MATH* and *SuperGPQA*. The results, measured by AUC, are summarized in Tables 2 and 3. Across all settings, DVD consistently and significantly outperforms all baseline approaches, demonstrating superior detection accuracy and cross-domain robustness.

Superiority Over Log-Probability and Loss-Based Baselines: Traditional detection methods such as *Loss*, *Perplexity*, and *Zlib* exhibit highly

Method	Omni-MATH				SuperGPQA			
	Qwen1.5B	Qwen3B	Qwen7B	Qwen32B	Qwen1.5B	Qwen3B	Qwen7B	Qwen32B
Min-K%++	0.694	0.693	0.680	0.681	0.422	0.463	0.435	0.436
CDD	0.494	0.495	0.512	0.507	0.496	0.504	0.518	0.503
Min-K	0.538	0.560	0.572	0.578	0.501	0.492	0.511	0.539
Perplexity	0.544	0.549	0.557	0.556	0.517	0.520	0.513	0.517
Loss	0.626	0.637	0.650	0.635	0.404	0.406	0.408	0.409
Zlib	0.573	0.576	0.583	0.581	0.425	0.430	0.428	0.427
EM	0.521	0.506	0.533	0.505	0.531	0.529	0.524	0.521
DVD (Ours)	0.744	0.747	0.734	0.667	0.770	0.708	0.740	0.743

Table 2: Performance comparison of different detection methods on Omni-MATH and SuperGPQA datasets with full Fine-tuning (1 epoch). EM denotes the Embedding-similarity method. The notation QwenXB in the table refers to the Qwen2.5-XB-Instruct model, where X denotes the model's parameter count

Method	Omni-MATH				SuperGPQA			
	Qwen1.5B	Qwen3B	Qwen7B	Qwen32B	Qwen1.5B	Qwen3B	Qwen7B	Qwen32B
Min-K%++	0.646	0.621	0.648	0.608	0.415	0.456	0.364	0.369
CDD	0.501	0.503	0.513	0.507	0.520	0.517	0.584	0.600
Min-K	0.549	0.505	0.531	0.536	0.497	0.519	0.414	0.341
Perplexity	0.572	0.543	0.567	0.557	0.478	0.495	0.404	0.370
Loss	0.572	0.556	0.567	0.563	0.379	0.372	0.334	0.333
Zlib	0.549	0.542	0.550	0.550	0.409	0.400	0.378	0.376
EM	0.590	0.608	0.544	0.563	0.580	0.594	0.599	0.712
DVD (Ours)	0.771	0.745	0.731	0.715	0.674	0.737	0.700	0.751

Table 3: Performance comparison of detection methods on Omni-MATH and SuperGPQA with Lora Fine-tuning (10 epochs). EM denotes the Embedding-similarity method. The notation QwenXB in the table refers to the Qwen2.5-XB-Instruct model, where X denotes the model's parameter count

unstable performance across different scenarios. For instance, while the *Loss* method achieves moderate results on Omni-MATH (e.g., AUC of 0.626 to 0.664 in Table 2), its performance deteriorates sharply on the SuperGPQA dataset, with AUC values significantly below the random-chance threshold of 0.5 (e.g., 0.333 to 0.409 in Table 3). Importantly, an AUC below 0.5 indicates that the models predictions are systematically inverted relative to the true labels. Specifically, samples with higher loss are more likely to be incorrectly classified as clean, while those with lower loss are misidentified as contaminated. Similarly, *Perplexity* and *Zlib* yield AUCs consistently in the range of 0.4 to 0.5 on SuperGPQA, reflecting a similar tendency to produce judgments that contradict the actual contamination status. This clearly demonstrates that simple likelihood- or compression-based metrics are highly sensitive to data distribution and training configurations, lacking not only robustness but also the basic reliability required for effective detection of variant contamination.

Analysis of Min-K and Min-K%++: The *Min-K* and *Min-K%++* methods, which focus on the likelihood of the least probable tokens, show a specialized but fragile advantage. *Min-K%++* is the strongest baseline on the Omni-MATH dataset, achieving AUCs between 0.608 and 0.694. However, its effectiveness diminishes on SuperGPQA, where it drops as low as 0.278 (Table 3). This indicates that while focusing on outlier token probabilities helps in structured mathematical domains, it fails to generalize to complex, semantic-heavy reasoning tasks where the "variant" nature of the data is not captured by local token statistics.

Comparison with Distributional and Similarity Measures: We also compared our method against *CDD* (based on edit distance) and *Embedding-similarity*. *CDD* consistently performs near the level of random guessing (AUC ≈ 0.50) on Omni-MATH, suggesting that surface-level text fluctuations are insufficient for detection in diverse mathematical contexts. On SuperGPQA, *CDD* improves slightly (up to 0.600) but remains uncompetitive. *Embedding-similarity* proves to be a more robust baseline, particularly on SuperGPQA where it achieves AUCs between 0.521 and 0.712. Nevertheless, it still lags behind DVD by a significant margin. This confirms that while semantic similarity captures some distribu-

tional shifts, it cannot match the discriminative power of DVDs difficulty-fluctuation modeling.

Effectiveness Under Different Fine-tuning Regimes: The experimental results across Full Fine-tuning (1 epoch) and LoRA (10 epochs) highlight DVD's versatility. In the challenging 10-epoch LoRA setting on Omni-MATH (Table 3), DVD achieves an AUC of 0.771 on Qwen-1.5B, while the next best baseline (*Min-K%++*) only reaches 0.646. Even when the model is heavily fine-tuned, DVD effectively captures the "synthetic difficulty" signatures that distinguish contaminated variants from clean data.

5.3 Ablation Study

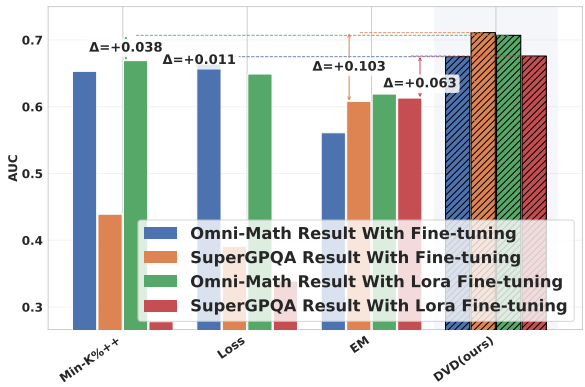

Figure 3: Performance comparison of the DVD method and other baselines on the Llama architecture (using the Llama3.1-8B-Instruct model). EM denotes the Embedding-similarity method. A complete visualization of the results is provided in Figure 9 of Appendix A.5.

Robustness Across Model Scales and Architectures: Our method exhibits remarkable stability across different model sizes (from Qwen-1.5B to 32B) and architectures (Qwen and Llama). As shown in Table 2, DVD maintains high AUCs (0.667–0.747) regardless of the parameter count. Notably, on the Llama-8B model, DVD consistently provides a substantial gain over the best baselines (shown in Figure 4). While other methods like *Min-K%++* or *Embedding-similarity* fluctuate wildly depending on the model scale, DVDs performance remains consistently high, validating its architecture-agnostic nature.

Robustness of DVD Across Training Epochs and Model Scales: As shown in Figure 4, each data point represents the average result over ten independent runs to mitigate the influence of randomness. DVD demonstrates consistently high robustness across different training epochs. When evaluated on various instruction tuned large language models, including the Qwen2.5 series and Llama3.1-8BInstruct, DVD achieves strong AUC scores for detecting model variant contamination as early as the initial training stages. Moreover, detection performance generally improves with increasing model scale; for instance, Qwen2.5 32B Instruct attains a higher and more stable AUC. Collectively, these results indicate that DVD exhibits consistent and reliable detection capability across models of varying sizes and architectures, as well as across different training epochs.

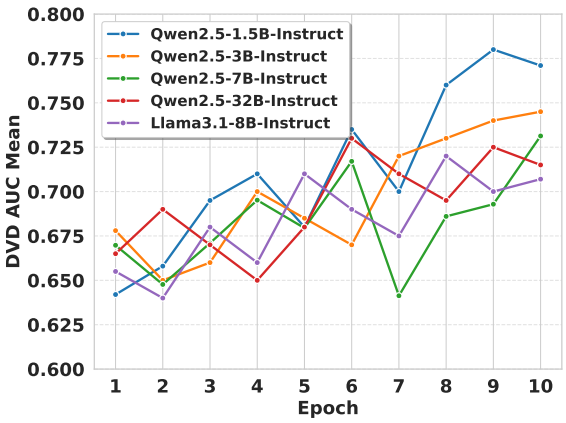

Figure 4: Average performance of the DVD method across different epochs.

6 Conclusion

This work systematically uncovers the overlooked problem of variant contamination in large language models, establishes the first benchmark dedicated to this issue, and proposes DVD as a principled solution. DVD effectively identifies contaminated samples by modeling fluctuations in synthesis difficulty across multiple generations, significantly outperforming conventional approaches based on log probability, distributional properties, and similarity metrics. Evaluated on the proposed benchmark, which encompasses Omni-MATH and SuperGPQA, DVD demonstrates consistently high accuracy and strong cross-domain robustness across diverse training configurations and model scales, offering a reliable tool to mitigate contamination risks and enable fairer, more trustworthy evaluation of large language models.

7 Limitation

For tasks with open-ended answers, underspecified problem statements, or multiple valid reasoning paths, temperature sampling naturally induces greater output diversity, thereby elevating the baseline level of variance in the generation distribution. As a result, when comparing DVD scores across tasks or domains, it is generally necessary to adopt unified prompt templates, length constraints, and decoding configurations, and to apply task- or category-specific calibration or relative scoring schemes to ensure interpretability and comparability. Moreover, because DVD derives its signal from local uncertainty fluctuations in low-probability tokens, it is sensitive to generation length, stopping criteria, and answer formats, all of which can alter the composition and proportion of tail tokens and thus affect the stability of synthetic difficulty estimation. At a practical level, DVD constructs a local generation distribution via repeated sampling, which necessitates balancing the number of samples against detection stability in environments with limited query budgets or restricted access. Finally, the benchmark construction and contamination simulation in this work rely primarily on LLM-generated and LLM-filtered semantic variants injected through fine-tuning; while this setup is reproducible and well controlled, variant contamination in real pretraining corpora may arise from more diverse sources-such as cross-domain reuse, templated rewriting, heterogeneous annotation styles, or retrieval-augmented pipelineswhose behavioral signatures and decision boundaries warrant further systematic characterization in future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Simone Balloccu, Patrícia Schmidtová, Mateusz Lango, and Ondřej Dušek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. *arXiv preprint arXiv:2402.03927*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Yuxing Cheng, Yi Chang, and Yuan Wu. 2025. A survey on data contamination for large language models. *arXiv preprint arXiv:2502.14425*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2023. Investigating data contamination in modern benchmarks for large language models. *arXiv preprint arXiv:2311.09783*.

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

Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, King Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, and 1 others. 2025. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *arXiv preprint arXiv:2502.14739*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

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

Shahriar Golchin and Mihai Surdeanu. 2025. Data contamination quiz: A tool to detect and estimate contamination in large language models. *Transactions of the Association for Computational Linguistics*, 13:809–830.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Yucheng Li, Frank Guerin, and Chenghua Lin. 2023. An open source data contamination report for large language models. *arXiv preprint arXiv:2310.17589*.

Haoxiong Liu, Yifan Zhang, Yifan Luo, and Andrew C Yao. 2025. Augmenting math word problems via iterative question composing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24605–24613.

Riccardo Lunardi, Vincenzo Della Mea, Stefano Mizzaro, and Kevin Roitero. 2025. On robustness and reliability of benchmark-based evaluation of llms. *arXiv preprint arXiv:2509.04013*.

Kiran Maharana, Surajit Mondal, and Bhushankumar Nemade. 2022. A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, 3(1):91–99.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*.

Yonatan Oren, Nicole Meister, Niladri S Chatterji, Faisal Ladhak, and Tatsunori Hashimoto. 2023. Proving test set contamination in black-box language models. In *The Twelfth International Conference on Learning Representations*.

Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. Nlp evaluation in trouble: On the need to measure llm data contamination for each benchmark. *arXiv preprint arXiv:2310.18018*.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2023. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*.

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

Connor Shorten and Taghi M Khoshgoftaar. 2019. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48.

Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. 2021. Text data augmentation for deep learning. *Journal of big Data*, 8(1):101.

Gemini Team, R Anil, S Borgeaud, Y Wu, JB Alayrac, J Yu, R Soricut, J Schalkwyk, AM Dai, A Hauth, and 1 others. 2024. Gemini: A family of highly capable multimodal models, 2024. *arXiv preprint arXiv:2312.11805*, 10.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ruijie Xu, Zengzhi Wang, Run-Ze Fan, and Pengfei Liu. 2024. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Yang, and Hai Li. 2025. Min-k%++: Improved baseline for pre-training data detection from large language models. In *International Conference on Representation Learning*, volume 2025, pages 64845–64862.

Yukun Zhao, Lingyong Yan, Weiwei Sun, Guoliang Xing, Shuaiqiang Wang, Chong Meng, Zhicong Cheng, Zhaochun Ren, and Dawei Yin. 2024. Improving the robustness of large language models via consistency alignment. *arXiv preprint arXiv:2403.14221*.

A Appendix

A.1 Fine-tuning -details

To emulate variant contamination, we fine-tune the above models on our constructed variant-contamination dataset. To model resource-constrained scenarios, we adopt LoRA for parameter-efficient adaptation; this training is conducted on a single NVIDIA A800 GPU with the following settings: LoRA rank 8; Adam optimizer; 10 training epochs; initial learning rate $1e5$; a cosine learning-rate scheduler with a warmup ratio of 0.1; per-GPU batch size 2; gradient accumulation steps 1; and bfloat16 precision. To model quality-prioritized scenarios, we additionally perform full-parameter fine-tuning on two NVIDIA A800 GPUs, using: Adam optimizer; 1 training epoch; initial learning rate $1e5$; a cosine learning-rate scheduler with a warmup ratio of 0.1; per-GPU batch size 2; gradient accumulation steps 1; and bfloat16 precision

A.2 Hyperparameter Sensitivity Analysis

To evaluate the sensitivity of the proposed DVD method to the key hyperparameter M (i.e., the minimum number of low-probability tokens considered when calculating the synthetic difficulty), we conducted extensive experiments on the Qwen2.5-3B-Instruct model and the Omni-MATH variant dataset. The experimental results (Figure 5) reveal both the effectiveness and moderate sensitivity of the method to M.

Strong Performance with a Clear Optimal Region The detection performance of the DVD method, measured by ROC AUC, peaks at **0.751** when $M = 22$. Notably, even at nearby values-such as $M = 20$ (AUC = 0.747)performance remains high, indicating a **well-defined and broad performance peak** rather than extreme fragility. Across the full tested range (approximately $M = 5$ to $M = 35$), AUC scores stay consistently above 0.58 and reach a maximum of 0.751, demonstrating that the method is effective over a wide hyperparameter regime.

Presence of a High-Performance Interval Although the curve is not completely flat, a **robust high-performance interval** exists around $M \in [18, 26]$, where AUC remains above 0.74. This suggests that while fine-tuning M can yield marginal gains, users can still achieve near-optimal detection performance by selecting M within this practical intervalavoiding the need for exhaustive search while maintaining strong results.

Clear Advantage Over Baseline Methods Critically, even the **lower end** of the observed AUC range (e.g., $\sim$0.58 at extreme M values) is comparable to or exceeds the performance of baseline methods such as Min-K%++ Prob (0.693), Perplexity (0.549), and CDD (0.495). More importantly, the **peak performance (0.751)** substantially outperforms all baselines, confirming that the DVD methods superiority is both significant and realizable with reasonable hyperparameter choices.

Theoretical Interpretation The unimodal shape of the AUC curve aligns with the underlying mechanism of DVD:

- When M is too small (e.g., $M < 15$), the synthetic difficulty is estimated from too few tokens, leading to high variance and unreliable detection signals.
- When M is too large (e.g., $M > 30$), the inclusion of medium-probability tokens dilutes the signal from truly "difficult" (low-probability) tokens, slightly degrading discriminative power.
- Around $M = 22$, the method strikes an optimal balancecapturing enough low-probability tokens for stable variance estimation while avoiding noise from less informative tokens.

This behavior reflects a **principled trade-off** inherent in the design of DVD, rather than arbitrary sensitivity. The existence of a clear, high-performing region further supports the methods practical utility.

A.3 Case Study

The three representative cases examined above provide a mechanistic explanation for the macroscopic performance trends observed in Figure 6. They demonstrate that the effectiveness of a detection method is not arbitrary but is determined by the intrinsic alignment between its underlying mechanism and the nature of the contamination. The superior performance of our DVD method stems from its unique capacity to probe the model's internal "cognitive state," enabling it to penetrate surface-level textual variations and identify the essential signal of memorization.

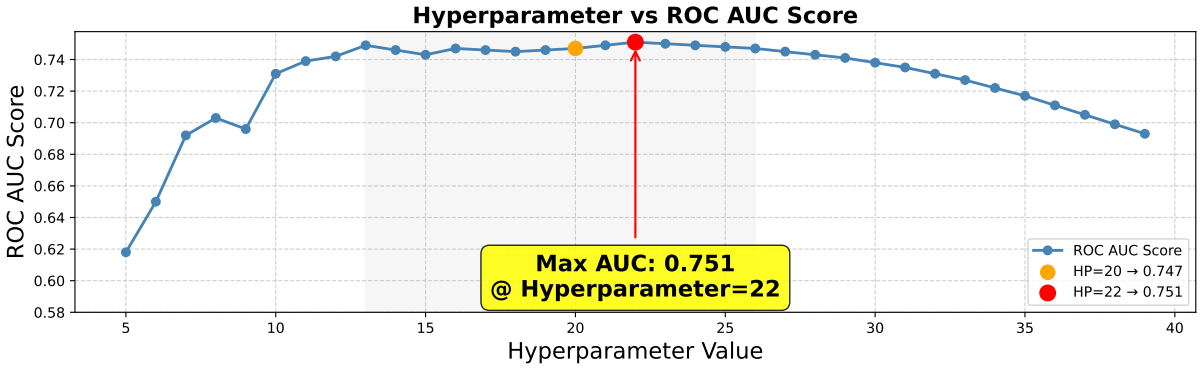

Figure 5: The DVD method demonstrates remarkable robustness across a wide range of hyperparameters.

A.3.1 Deeply Transformed Variants

Cases 1 and 2 represent deeply transformed variants where all four transformation methods (entity substitution, scenario conversion, etc.) are applied. Although the surface forms are completely different, such as changing "students" to "participants" and completely altering the narrative order, the core semantics remain consistent.

Sampling and Output-Matching-Based Methods , such as CDD, perform poorly is that they rely on surface form matching and fail to effectively capture deep logical equivalence. In Case 1, there is a significant surface difference between the original and variant problems, such as from "16 students took part in a competition" to "A group of sixteen participants joined a contest." Although the core semantics of the two problems remain consistent, the surface text changes significantly. The CDD method detects based only on surface symbol similarity, and when faced with surface-level changes (such as replacing some vocabulary or altering the narrative structure), CDD fails to recognize the deep similarities between these problems. CDD misjudges this surface difference as "output inconsistency," resulting in detection failure.

Perplexity-Based Methods perform poorly when faced with surface-level changes or structural perturbations, because they are highly sensitive to changes in the surface form of the text. These methods typically assess whether a model has memorized certain samples by measuring the perplexity of low-probability tokens. However, perplexity-based methods mainly rely on the model's confidence in generating known samples, neglecting the deeper semantics of the text. In Case 2, when the original problem is changed in variant names and narrative order (e.g., replacing i, j with m, n), this change does not fundamentally alter the mathematical structure of the problem, but the perplexity method may mistakenly classify it as non-memorized due to the model's lower confidence in generating these changes.

The DVD method is successful by probing the internal cognitive state of the model during generation, rather than analyzing the output text. Despite the surface differences, the model has memorized the core logical template for solving these problem types. When generating answers, it exhibits high confidence at the key reasoning steps and final answer. This results in low and stable "constitutive difficulty" values across samples, leading to a high variance score. Thus, DVD effectively detects contamination by identifying the model's familiarity with the underlying mathematical structure, bypassing surface-level noise.

A.3.2 Simply Transformed Variants

Case 3 is a simple variant that involves only entity substitution and numerical rewriting. The mathematical problem (an application of the Cauchy-Schwarz inequality) remains identical, with only the variable names and the scenario changed.

Sampling and Output-Matching-Based Methods, CDD, success was achieved in Case 3 because of the high text and semantic similarity between the original and variant problems, with only a few differences in specific phrases and variant names.

Perplexity-Based Methods perform poorly when facing surface-level changes or structural perturbations. Among multiple methods, only Embedding Similarity successfully identified the contamination, highlighting their fragility. Changes in specific tokens (variables, numbers) are sufficient to alter the probability distribution. For ex-

Original Question	Variant Problem	Transform Method	Detection Method
16 students took part in a competition. All problems were multiple choice style. Each problem had four choices. It was said that any two students had at most one answer in common, find the maximum number of problems?	A group of sixteen participants joined a contest where all the questions were of multiple-choice format. Each question had four possible answers. It was stated that no two participants had more than one answer in common. What is the largest possible number of questions?	Entity Substitution: ✓ Scenario conversion:✓ Numerical rewriting:✓ Narrative restructuring: ✓	Ours: Yes CDD: No Min-K: No Min-K%++: No Perplexity: No Loss: No Zlib: No Embedding Similarity: No
Find, with proof, the maximum positive integer \\(k\\) for which it is possible to color \\(6k\\) cells of a \\(6 \times 6\\) grid such that, for any choice of three distinct rows \\(R_{1}, R_{2}, R_{3}}\\) and three distinct columns \\(C_{1}, C_{2}, C_{3}\\), there exists an uncolored cell \\(c\\) and integers \\(1 \\leq i, j \\leq 3\\) so that \\(c\\) lies in \\(R_{i}\\) and \\(C_{j}\\).	In a six by six grid, find, with proof, the largest integer \\(k\\) such that you can color six thousand \\(k\\) squares, ensuring that for any selection of three distinct rows \\(R_1, R_2, R_3\\) and three distinct columns \\(C_1, C_2, C_3\\), there is always one uncolored square \\(x\\) and integers \\(1 \leq m, n \leq 3\\) so that \\(x\\) lies in \\(R_m\\) and \\(C_n\\).	Entity Substitution: ✓ Scenario conversion: ✓ Numerical rewriting: ✓ Narrative restructuring: ✓	Ours: Yes CDD: No Min-K: No Min-K%++: No Perplexity: No Loss: No Zlib: No Embedding Similarity: No
Given that a,b,c,d,e are real numbers such that\n $a+b+c+d+e=8$, \n$a^2+b^2+c^2+d^2+e^2=16$. \nDetermine the maximum value of e.	Consider that x, y, z, w, v are real numbers such that\\ $x + y + z + w + v = 10$,\\ $x^2 + y^2 + z^2 + w^2 + v^2 = 20$.\\ Find the largest possible value of v.	Entity Substitution: ✓ Scenario conversion: ✓ Numerical rewriting: ✗ Narrative restructuring: ✗	Ours: Yes CDD: Yes Min-K: No Min-K%++: No Perplexity: No Loss: No Zlib: No Embedding Similarity: Yes

Figure 6: Compare the effectiveness of different detection methods on different variants

ample, changing a, b, c, d, e to x, y, z, w, v, and changing 8 and 16 to 10 and 20. These specific token changes are enough to significantly alter the models computation of the probability distribution of the entire sequence. The model has seen $a + b + c + d + e = 8$, but has not seen $x + y + z + w + v = 10$, so it perceives the latter sequence as having a slightly lower probability.

The DVD method performs excellently in such a simple entity substitution scenario, further demonstrating that by probing the models internal cognitive state during generation, our method effectively identifies and captures deep logical structures and semantic consistency.

A.4 Statistical Evidence for Generation States: Memory Adherence and Perturbation Drift

This section provides the detailed statistical and experimental foundation for introducing the core generation states: **Memory Adherence** and **Perturbation Drift**. These states are not theoretical assumptions but stable, objectively observed modes of behavior resulting from a systematic statistical analysis of the model's generation process on contaminated and uncontaminated samples.

Experimental Setup

To quantify the model's generation mechanism, we performed multiple repeated samplings for 100 randomly selected samples (including both contaminated and clean examples) at a fixed sampling temperature τ (e.g., $\tau = 0.8$). The primary statistical quantity analyzed is the distribution of the **log-likelihood of the sum of K-min token** at each generation step. This distribution characterizes the model's propensity to generate tokens with varying degrees of confidence and quality.

Bimodal Structure in Contaminated Samples

As shown in 7, for samples subject to variant contamination, the distribution of the sum of K-min token log-likelihood consistently exhibits a **pronounced and repeatable bi-modal structure**. This characteristic structure is direct evidence that the model's generation process, when exposed to contamination, is not governed by a single random mechanism but dynamically switches between two distinct modes.

The analysis of the bi-modal structure reveals the following:

1. **The First Peak (High-Confidence Region):** This peak is consistently located in the

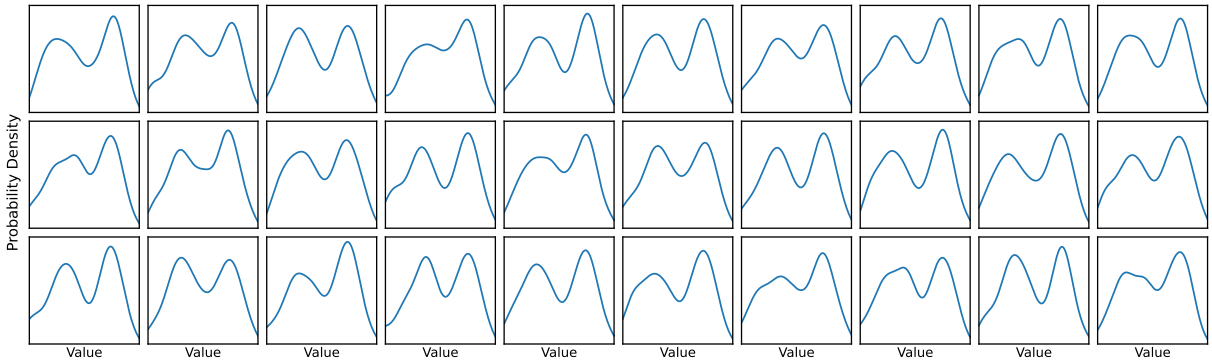

Figure 7: Probability distribution plot of the sum of log probabilities of the min-k tokens sampled from the distribution of contaminated samples

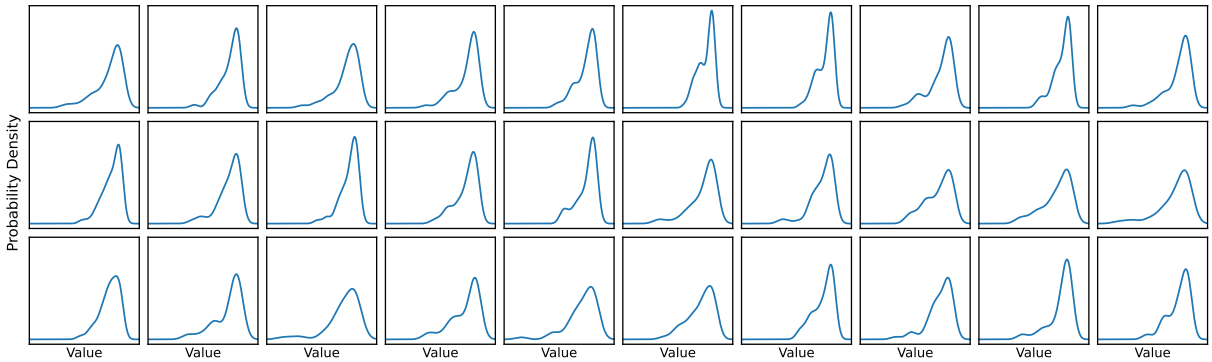

Figure 8: Probability distribution plot of the sum of log probabilities of the min-k tokens sampled from the distribution of uncontaminated samples

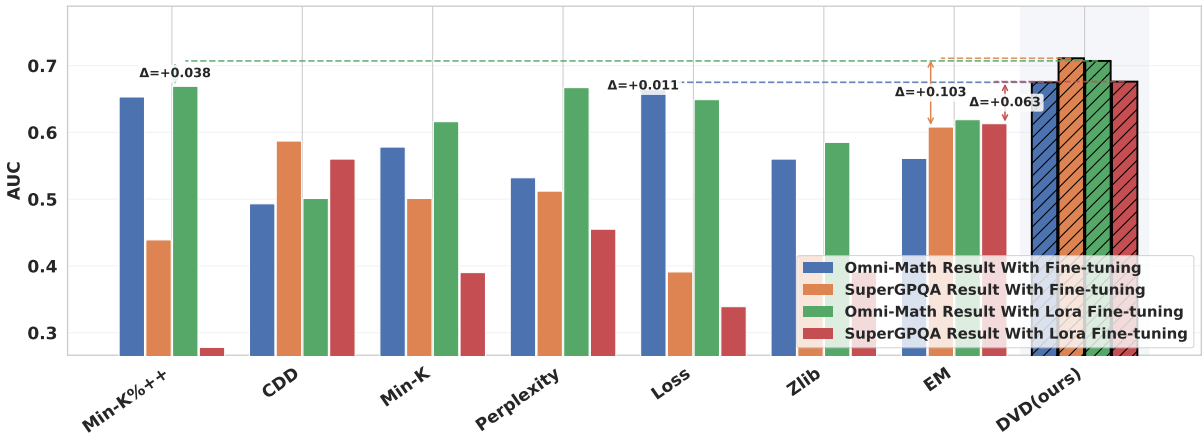

Figure 9: Performance of the DVD method on the Llama architecture compared to other baselines. EM denotes the Embedding-similarity method.

higher log-likelihood region. It corresponds to generation where the model selects high-probability, high-confidence tokens. This behavior indicates that the model is **adhering to answer fragments, linguistic patterns, or templates** encountered during training. We define this as the **Memory Adherence State**.

2. **The Second Peak (Low-Confidence Region):** This peak resides in the **lower** log-likelihood region, corresponding to the selection of low-probability, high-randomness tokens. This mode suggests the model has **deviated from the memory track** and entered a more explorative, lower-confidence generation space, which we term the **Perturbation Drift State**.

The bi-modal distribution directly proves that the model dynamically alternates between leveraging specific, strong memory structures and engaging in randomized, exploratory sampling on the same contaminated inputs.

Unimodal Structure in Uncontaminated Samples

In stark contrast, uncontaminated (clean) samples, used as a control, consistently exhibit a **single, smooth, and approximately Gaussian distribution** (Figures 8).

The unimodal nature confirms that the model is following a **consistent, intrinsic random generation mechanism** without the disruptive influence of strong, pulling memory structures. The absence of a secondary peak supports the hypothesis that state-switching behavior is unique to contaminated data.

A.5 Performance of the DVD method on the Llama architecture compared to other baselines.

Figures 9 reports AUC on the Llama backbone across two benchmarks (Omni-Math and SuperGPQA) under both full fine-tuning and LoRA. Overall, DVD achieves the best or near-best performance in all settings and exhibits the most consistent gains. In contrast, baselines such as perplexity, Min-K%, CDD, and Zlib vary substantially across datasets and fine-tuning regimes, indicating weaker robustness. These results suggest that DVD provides a more reliable detector of variant contamination and generalizes better across training setups.

A.6 Significance testing experiments of the DVD method against other baselines

Method	Omni-MATH	SuperGPQA	Conf
DVD (Ours)	0.731 ± 0.013	0.700 ± 0.016	–
Min-K%++	0.648 ± 0.014	0.364 ± 0.018	99%
CDD	0.513 ± 0.012	0.584 ± 0.011	99%
Min-K	0.531 ± 0.015	0.414 ± 0.016	99%
Perplexity	0.567 ± 0.013	0.404 ± 0.017	99%
Loss	0.567 ± 0.012	0.334 ± 0.019	99%
Zlib	0.550 ± 0.014	0.378 ± 0.018	99%
EM	0.544 ± 0.011	0.599 ± 0.010	99%

Table 4: Significance testing experiments of the DVD method against other baselines on Qwen2.5-7B-Instruct.

Table 4 presents a comparison of the contamination detection performance of the proposed DVD method against various existing baselines on two challenging benchmarks, Omni-MATH and SuperGPQA, using the Qwen2.5-7B-Instruct model. The table also reports the confidence levels (column "Conf") from paired significance tests between DVD and each baseline. The results show that DVD significantly outperforms all competing methods on this model: it achieves an AUC of 0.731 ± 0.013 on Omni-MATH and 0.700 ± 0.0016 on SuperGPQA. The reported margins of error (standard deviations) are computed over 10 independent runs with different random seeds.

A.7 Prompt

Figures 11 and 10 show the prompts used in our data construction pipeline: Figure 11 is the prompt for generating semantic variations, and Figure 10 is the prompt for rejection sampling.

Please determine whether the following variant statement is logically correct and free of obvious errors:

Variant statement: [Variant statement content]

Next, please assess whether the variant statement is logically equivalent to the original statement and whether it shares the same core solution space:

Original statement: [Original statement content]

Variant statement: [Variant statement content]

Finally, respond with either "yes" or "no." If any part of the judgment is "no," the final output should be "no."

Figure 10: The prompt for rejection sampling

You will be provided with a problem in JSON format, with each item separated by a newline.
Please generate 4 different variants of the given problem using the following methods:

1.Entity substitution: Replace referents, variable names, and object categories while maintaining consistency in type and context.

2.Scenario transformation: Alter the background setting and narrative context, preserving logical dependencies and constraint structures.

3.Numerical rewriting: Resample parameters under solvability constraints and update derivations and intermediate values for consistency.

4.Narrative structure transformation: Rearrange syntax or rewrite step-by-step analysis into a paragraph-style narrative while preserving semantic meaning.

Your response should consist only of newline-delimited JSON format text.

Figure 11: The prompt used to generate variations