Gap-K%: Measuring Top-1 Prediction Gap for Detecting Pretraining Data

Minseo Kwak **Jaehyung Kim**
Yonsei University
{rhkralstj103,jaehyungk}@yonsei.ac.kr

Abstract

The opacity of massive pretraining corpora in Large Language Models (LLMs) raises significant privacy and copyright concerns, making pretraining data detection a critical challenge. Existing state-of-the-art methods typically rely on token likelihoods, yet they often overlook the gap between the target token and the model's top-1 prediction, as well as local correlations between adjacent tokens. In this work, we propose Gap-K%, a novel pretraining data detection method grounded in the optimization dynamics of LLM pretraining. By analyzing the next-token prediction objective, we observe that discrepancies between the model's top-1 prediction and the target token induce strong gradient signals, which are explicitly penalized during training. Motivated by this, Gap-K% leverages the log probability gap between the top-1 predicted token and the target token, incorporating a sliding window strategy to capture local correlations and mitigate token-level fluctuations. Extensive experiments on the WikiMIA and MIMIR benchmarks demonstrate that Gap-K% achieves state-of-the-art performance, consistently outperforming prior baselines across various model sizes and input lengths[1].

1 Introduction

Large Language Models (LLMs) have recently demonstrated remarkable capabilities, including understanding, reasoning, and generation tasks (OpenAI, 2024; Comanici et al., 2025). A key to this success is pretraining on massive-scale data, mostly collected through web crawling. However, the details of these pretraining corpora remain largely undisclosed for many state-of-the-art models (Yang et al., 2025; Grattafiori et al., 2024). This lack of transparency raises significant concerns, as pretraining data may contain Personally Identifiable Information (PII) (Lukas et al., 2023) or copyrighted content (Rahman and Santacana, 2023), leading to ethical and legal issues. Moreover, if benchmark datasets are unknowingly included in the pretraining data, model performance would be inflated, leading to unfair comparisons (Oren et al., 2023).

For these reasons, *pretraining data detection* (Shi et al., 2024) has emerged as a critical problem; this problem is an instance of membership inference attack (MIA) (Shokri et al., 2017), which aims to determine whether a data point was in the training dataset. Existing state-of-the-art methods typically address this problem by exploiting token-level likelihoods. For instance, Min-K% (Shi et al., 2024) assumes that non-training samples contain outlier tokens with low log probabilities, thus utilizing the average log probability of the bottom K% tokens as a detection score. Min-K%++ (Zhang et al., 2025) further improves this method by normalizing token log probabilities relative to the mean and standard deviation of the next-token distribution. However, these approaches treat each token independently, failing to exploit the local correlations between adjacent tokens. Moreover, existing methods overlook the gap between the target token and the model's top-1 prediction, thereby missing informative signals induced by the training process.

In this work, we propose **Gap-K%**, a novel pretraining data detection method grounded in the optimization dynamics of LLM pretraining. We leverage the insight that the next-token prediction objective explicitly forces the model to align its top-1 prediction with the ground truth. Consequently, while training samples exhibit minimal difference from the top-1 prediction, unseen data frequently triggers *confident mispredictions*, cases where the model strongly favors a plausible candidate other than the target. Building on this observation, Gap-K% quantifies the normalized gap between the log probabilities of the top-1 predicted token and the target token, effectively capturing such confident

[1]Code: https://github.com/meaoww/gap-k.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 23391–23405

mispredictions. Furthermore, to exploit the local correlations within the token sequence, we apply a sliding window to the scores over adjacent tokens, thereby mitigating token-level fluctuations.

We validate the effectiveness of Gap-K% through experiments on two representative benchmarks, WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024). On WikiMIA, Gap-K% consistently outperforms prior state-of-the-art methods across five evaluated models. In the original setting, averaged across models, Gap-K% achieves absolute AUROC improvements of 9.7% over the average of the existing baselines and 2.4% over the strongest baseline, Min-K%++. In the paraphrased setting, it achieves absolute AUROC gains of 5.7% over the baseline average and 1.7% over Min-K%++. On the more challenging MIMIR benchmark, Gap-K% attains the highest average performance across Pythia models ranging from 1.4B to 12B parameters.

In summary, our contributions are as follows:

- We identify that the gaps between top-1 predictions and target tokens as an effective detection signal, grounded in the optimization dynamics of next-token prediction.
- We propose Gap-K%, a novel pretraining data detection method that leverages top-1 prediction gaps and aggregates token-level signals to capture local correlations between adjacent tokens.
- We achieve state-of-the-art performance on WikiMIA and MIMIR benchmarks across various model sizes and input lengths.

2 Related Works

Membership inference attacks. Membership Inference Attacks (MIAs) aim to determine whether a given data sample was included in the training set. Samples included in the training data are referred to as members, while samples not included are non-members. In the context of LLMs, MIAs have been widely used for quantifying memorization (Carlini et al., 2022) and privacy risks (Mireshghallah et al., 2022; Steinke et al., 2023), as well as for detecting data contamination (Oren et al., 2023) and exposure of copyrighted content (Duarte et al., 2024; Meeus et al., 2024). MIAs can be categorized into reference-based and reference-free methods. Reference-based methods (Carlini et al., 2021; Mireshghallah et al., 2022; Ye et al., 2022) use additional reference models that are trained on data from a similar distribution. However, obtaining reference models is costly and often impractical, especially for pretrained LLMs. Therefore, reference-free methods have gained attention recently. Reference-free methods typically rely on loss (Yeom et al., 2018), and are further improved by comparing it with losses on perturbed samples (Mattern et al., 2023) or calibrating using compression-based entropy (Carlini et al., 2021).

Pretraining data detection. Pretraining data detection is a specific instance of MIAs, where the objective is to infer whether a given text is included in the pretraining corpus of an LLM. Compared to standard MIAs, detecting pretraining data is more challenging because the data is seen only a few times within a massive corpus, resulting in weak memorization signals. Moreover, the lack of access to the pretraining data distribution makes it difficult to employ reference models (Shi et al., 2024). Prior work (Shi et al., 2024; Zhang et al., 2025) has focused on token-level probabilities and treats tokens independently. In contrast, we consider sequential dependencies in text and leverage top-1 predictions, which have received little attention in prior work.

3 Method

3.1 Preliminary

Problem definition. Let $\mathcal{M}$ be an autoregressive language model trained on an unknown dataset $\mathcal{D}$. Given a text sequence $\mathbf{x} = [x_1, \ldots, x_N]$, the goal is to determine whether $\mathbf{x}$ is a member of the training set ($\mathbf{x} \in \mathcal{D}$) or not ($\mathbf{x} \notin \mathcal{D}$). Specifically, a detection method computes a membership score $s(\mathbf{x}; \mathcal{M})$, predicting $\mathbf{x} \in \mathcal{D}$ if $s(\mathbf{x}; \mathcal{M})$ exceeds a threshold λ. We adopt a gray-box setting, where the model's output logits and token probabilities can be accessed, but the model parameters and gradients are not available.

Detection with token probabilities. Existing state-of-the-art methods, such as Min-K% (Shi et al., 2024) and Min-K%++ (Zhang et al., 2025), rely on the likelihood of tokens. Min-K% computes a membership score by averaging log probabilities of the lowest $k\%$ tokens:

$$\text{Min-K}(\mathbf{x}) = \frac{1}{|\mathcal{I}_k(\mathbf{x})|} \sum_{t \in \mathcal{I}_k(\mathbf{x})} \log p(x_t \mid x_{<t}), \quad (1)$$

where $\mathcal{I}_k(\mathbf{x})$ denotes the set of indices of the lowest $k\%$ probability tokens. Min-K%++ builds on the intuition that training tokens are located near local maxima of the likelihood landscape, and thus

have higher probabilities relative to other candidates in the vocabulary. Min-K%++ formulates this by normalizing log-likelihoods:

$$\text{Min-K++}(\mathbf{x}) = \frac{1}{|\mathcal{I}'_k(\mathbf{x})|} \sum_{t \in \mathcal{I}'_k(\mathbf{x})} z_t, \quad (2)$$

$$z_t = \frac{\log p(x_t \mid x_{<t}) - \mu_t}{\sigma_t}, \quad (3)$$

where $\mu_t = \mathrm{E}_{z \sim p(\cdot|x_{<t})}[\log p(z \mid x_{<t})]$ is the mean of the next-token log probability and $\sigma_t = \sqrt{\mathrm{E}_{z \sim p(\cdot|x_{<t})}[(\log p(z \mid x_{<t}) - \mu_t)^2]}$ is the standard deviation. Here, $\mathcal{I}'_k(\mathbf{x})$ denotes the set of indices corresponding to the lowest $k\%$ values of $\{z_t\}$.

3.2 Motivation and Insight

To distinguish training data from non-training data effectively, we focus on the fundamental behavior of the next-token prediction objective; we hypothesize that *the primary fingerprint of training is in the alignment between the model's top prediction and the ground truth.*

Gradient-level analysis. Consider the cross-entropy loss at step t, $\ell_t = -\log p(y_t \mid x_{<t})$, where y_t denotes the next token in the pretraining sequence. The gradient of the loss with respect to logit $s_t(v)$ for any token $v \in V$ is:

$$\frac{\partial \ell_t}{\partial s_t(v)} = p(v \mid x_{<t}) - \mathbb{1}[v = y_t]. \quad (4)$$

Crucially, the magnitude of the gradient for non-target tokens ($v \neq y_t$) is directly proportional to their probability $p(v \mid x_{<t})$. The token with the highest probability, $v_t^{\max} = \arg\max_v p(v \mid x_{<t})$, exerts the strongest gradient signal if it does not match the target y_t. Therefore, during the training process, the optimization algorithm aggressively penalizes cases where $v_t^{\max} \neq y_t$.

As a consequence of this optimization, for samples within the training set $\mathcal{D}$, the model learns to align its top-1 prediction with the target token by minimizing the gap between $\log p(v_t^{\max})$ and $\log p(y_t)$. In contrast, for unseen data $\mathbf{x} \notin \mathcal{D}$, the model cannot rely on memorization and instead predicts the next token based on learned syntactic and semantic correlations. As a result, the target token (y_t) may not coincide with the top-1 prediction ($v_t^{\max}$), leading to a gap between their log probabilities. Since such mismatches are explicitly penalized during training, these gaps tend to be smaller for training data and larger for unseen data. Thus, we hypothesize that this top-1 prediction gap serves as an informative signal for inferring membership.

Figure 1: Illustration of token-level scores before and after sequential smoothing. The gray curve shows raw token-level scores, while the green curve shows the smoothed scores.

3.3 Proposed Method: Gap-K%

Building on the insight that training data exhibits minimal top-1 prediction gaps, we propose **Gap-K%**, a method that quantifies this gap while accounting for sequential dependencies.

Top-1 gap scoring. First, we measure the token-level gap. For each token x_t, we compute the difference between its log probability and the maximum log probability over the vocabulary (*i.e.*, the top-1 prediction). To account for the varying sharpness of the output distribution (*e.g.*, flat vs. peaked distributions), we normalize this difference by the standard deviation σ_t of the log probabilities, similar to Min-K%++:

$$g_t = \frac{\log p(x_t \mid x_{<t}) - \max_{v \in V} \log p(v \mid x_{<t})}{\sigma_t}. \quad (5)$$

Here, g_t is always *non-positive*; a value close to 0 indicates that the target token x_t has a log probability close to that of the top-1 prediction, while a large negative value indicates a large gap between the target token and the top-1 prediction (likely for non-training data).

Sequential smoothing. Membership signals in LLMs are rarely isolated to a single token; they often span phrases or sentences, exhibiting sequential consistency due to the memorization of continuous text segments. To capture this local correlation and mitigate token-level fluctuations, we apply a

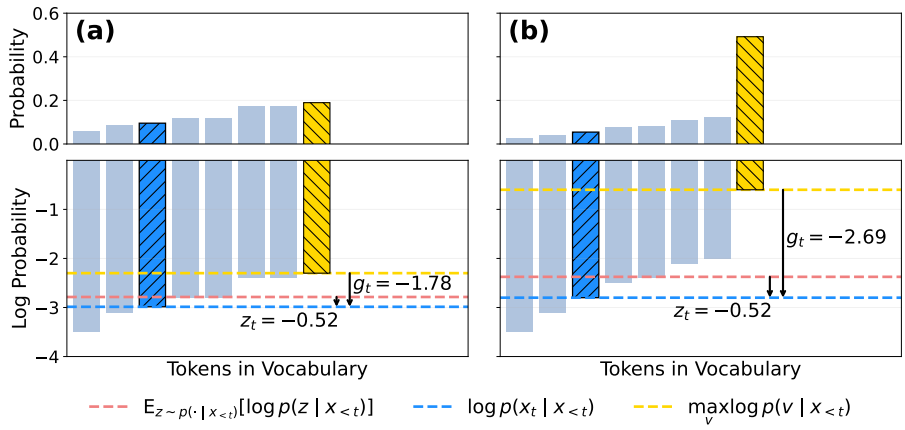


Figure 2: Conceptual comparison between Min-K%++ and Gap-K% using a toy example with a small vocabulary size of 8. Inspired by the illustrative analysis in Zhang et al. (2025), we compare the token-level scores of Min-K%++ (z_t) and Gap-K% (g_t) under two different next-token probability distributions. The z_t and g_t values annotated in the figure are normalized quantities. The blue hatched bar denotes the observed token x_t, while the yellow hatched bar indicates the top-1 token. In (a), the distribution is relatively flat, resulting in low-confidence incorrect predictions, whereas in (b) the model assigns high confidence to an incorrect top-1 token. While the Min-K%++ score z_t is identical in both cases, the Gap-K% score g_t distinguishes confident mispredictions from uncertain predictions by capturing the gap between the observed token and the top-1 prediction.

sliding window of size w over the gap scores:

$$\bar{g}_t^{(w)} = \frac{1}{w}\sum_{i=0}^{w-1} g_{t+i}. \tag{6}$$

As illustrated in Fig. 1, this smoothing step effectively highlights regions where the model consistently aligns (or fails to align) with the input text.

Gap-K% metric. Finally, as in Min-K%, we focus on the *worst-case segments* that provide the strongest counter-evidence for membership. We define the Gap-K% score as the average of the lowest $k\%$ smoothed gap scores in the sequence:

$$\text{Gap-K}(\mathbf{x}) = \frac{1}{|\tilde{\mathcal{I}}_k(\mathbf{x})|}\sum_{t\in\tilde{\mathcal{I}}_k(\mathbf{x})} \bar{g}_t^{(w)}, \tag{7}$$

where $\tilde{\mathcal{I}}_k(\mathbf{x})$ denotes the set of token indices corresponding to the bottom $k\%$ values of the smoothed scores $\{\bar{g}_t^{(w)}\}_{t=1}^{N-w+1}$. A higher Gap-K% score (closer to 0) implies that even the hardest-to-predict segments of the text exhibit a small top-1 prediction gap, indicating membership in the training data. We provide empirical validation of this design choice in Appendix E.

3.4 Comparison with Min-K%++

To validate the advantage of Gap-K% over Min-K%++, we analyze the relationship between our scoring function g_t and the Min-K%++ score z_t. By expanding Eq. 5 using the definition of z_t (Eq. 3), we can express g_t as:

$$g_t = z_t - \underbrace{\frac{\max_{v\in V}\log p(v\mid x_{<t}) - \mu_t}{\sigma_t}}_{\Delta_t}. \tag{8}$$

This formulation shows that our score differs from Min-K%++ by an additional term Δ_t, capturing the normalized gap between the top-1 log probability and the mean. It provides two critical insights.

Penalizing confident errors. Both methods focus on outlier tokens (lowest $k\%$ scores), which typically occur when the target token x_t differs from the top-1 prediction. In such scenarios, Min-K%++ (z_t) measures deviation solely from the mean, treating all low-probability tokens similarly regardless of the distribution shape. In contrast, Gap-K% (g_t) incorporates the confidence term Δ_t. Crucially, this allows our method to distinguish between *uncertain predictions* where the distribution is flat, and *confident mispredictions* where the model assigns high probability to an incorrect token, as illustrated in Fig. 2. By imposing a severe penalty on the latter, Gap-K% leverages these confident mispredictions as strong counter-evidence against membership. Since training samples are optimized to minimize such gaps, detecting these gaps provides a more robust signal for identifying non-training data than relying on likelihood alone.

Table 1: AUROC results on WikiMIA. *Ori.* and *Para.* indicate the original and paraphrased settings, respectively. **Bold** numbers denote the best performance.

Len.	Method	Mamba-1.4B		Pythia-6.9B		Pythia-12B		LLaMA-13B		LLaMA-65B		Average	
		Ori.	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*
32	Loss	61.0	61.3	63.8	64.1	65.4	65.6	67.5	68.0	70.8	71.9	65.7	66.2
	Zlib	61.9	62.3	64.4	64.2	65.8	65.9	67.8	68.3	71.2	72.1	66.2	66.6
	Neighbor	64.1	63.6	65.8	65.5	66.6	66.8	65.8	65.0	69.6	68.7	66.4	65.9
	Min-K%	63.3	62.9	66.3	65.1	68.1	67.2	66.8	66.2	70.6	70.1	67.0	66.3
	Min-K%++	66.4	65.7	70.3	67.6	72.2	69.4	84.4	82.7	85.3	81.6	75.7	73.4
	Gap-K%	**69.2**	**67.2**	**71.4**	**68.1**	**73.7**	**70.2**	**86.8**	**83.7**	**88.0**	**82.5**	**77.8**	**74.3**
64	Loss	58.2	56.4	60.7	59.3	61.9	60.0	63.6	63.1	67.9	67.9	62.5	61.3
	Zlib	60.4	59.1	62.6	61.6	63.5	62.1	65.3	65.3	69.1	69.5	64.2	63.5
	Neighbor	60.6	60.6	63.2	63.1	62.6	62.8	64.1	64.7	69.6	69.5	64.0	64.1
	Min-K%	61.7	58.0	65.0	61.1	66.5	62.5	66.0	63.5	69.9	66.8	65.8	62.4
	Min-K%++	67.2	62.2	71.6	64.2	72.6	65.1	84.3	78.8	83.5	74.3	75.8	68.9
	Gap-K%	**69.9**	**64.3**	**73.3**	**66.7**	**74.8**	**67.1**	**87.2**	**81.2**	**86.7**	**76.7**	**78.4**	**71.2**
128	Loss	63.3	62.7	65.1	64.7	65.8	65.4	67.8	67.2	70.8	70.2	66.6	66.0
	Zlib	65.6	65.3	67.6	**67.4**	67.8	67.9	69.7	69.6	72.2	**72.2**	68.6	68.5
	Neighbor	64.8	62.6	67.5	64.3	67.1	64.3	68.3	64.0	73.7	70.3	68.3	65.1
	Min-K%	66.8	64.4	69.5	67.0	70.7	68.5	71.5	68.6	73.8	70.5	70.5	67.8
	Min-K%++	67.7	63.3	69.8	65.9	71.8	67.7	83.8	76.2	80.8	70.0	74.8	68.6
	Gap-K%	**71.2**	**67.4**	**71.5**	66.2	**75.0**	**69.5**	**85.7**	**79.4**	**83.6**	70.7	**77.4**	**70.6**

Mode vs. Mean. From a statistical perspective, Min-K%++ assumes that training tokens are located near local maxima of the likelihood landscape. However, it adopts an *indirect* approach to capture this by measuring deviations from the *mean* of the vocabulary distribution. In contrast, Gap-K% aligns the metric with this assumption by directly measuring the gap between the target token and the mode of the distribution.

4 Experiments

In this section, we conduct comprehensive experiments to evaluate the proposed Gap-K%, designed to address the following research questions:

- **RQ1:** How effectively can Gap-K% detect pretraining data compared to existing baselines? (Table 1, 2 and Fig. 3)
- **RQ2:** Is the top-1 prediction gap empirically supported as a detection signal? (Table 3)
- **RQ3:** How do the components and hyperparameters of Gap-K% affect detection performance? (Table 4,5 and Fig. 4,5)
- **RQ4:** Does Gap-K% generalize to recent LLMs and remain robust under adversarial paraphrasing? (Table 6,7)
- **RQ5:** How does Gap-K% work differently from Min-K%++? (Fig. 6)

4.1 Setup

Benchmarks. We evaluate our method on WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024), two representative benchmarks for pretraining data detection. WikiMIA is constructed based on Wikipedia event pages and assigns training and non-training labels based on timestamps. WikiMIA provides both original and paraphrased versions. Since detection difficulty may vary with input length, WikiMIA includes length-based splits. In our experiments, we evaluate the length splits of 32, 64, and 128 for both the original and paraphrased settings, following Zhang et al. (2025). MIMIR is a more challenging benchmark, designed to have minimal distributional differences between training and non-training samples. MIMIR is constructed from the Pile dataset (Gao et al., 2020) and consists of seven domains: English Wikipedia, GitHub, Pile-CC, PubMed Central, arXiv, DM Mathematics, and HackerNews.

Models. Since WikiMIA is constructed from Wikipedia, which is included in the pretraining data of many LLMs, we evaluate Mamba-1.4B (Gu and Dao, 2024), Pythia-6.9B and Pythia-12B (Biderman et al., 2023), LLaMA-13B and LLaMA-65B (Touvron et al., 2023). For MIMIR, which is designed for models trained on the Pile dataset, we follow Duan et al. (2024) and evaluate Pythia-160M, 1.4B, 2.8B, 6.9B, and 12B.

Metrics. We evaluate the performance using the Area Under the ROC curve (AUROC) as our primary evaluation metric. AUROC captures the overall trade-off between the true positive rate (TPR) and false positive rate (FPR) across different thresh-

Table 2: AUROC results on MIMIR under the 13-gram 0.8 overlap setting (Duan et al., 2024). The best and second-best scores are highlighted in **bold** and underlined, respectively. For the Pythia-12B model, results for the Neighbor method are not reported due to computational constraints, consistent with Zhang et al. (2025).

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	50.2	51.3	51.8	52.8	53.5	65.7	69.8	71.3	73.0	71.0	49.6	50.0	50.1	50.7	51.1	49.9	49.8	49.9	50.6	51.3
Zlib	**51.1**	52.0	52.4	53.5	54.3	**67.5**	**71.0**	**72.3**	**73.9**	72.2	49.6	50.1	50.3	50.8	51.1	49.9	50.0	50.1	50.6	51.2
Neighbor	50.7	51.7	52.2	53.2	/	65.3	69.4	70.5	72.1	/	49.6	50.0	50.1	50.8	/	47.9	49.1	49.7	50.1	/
Min-K%	48.8	51.0	51.7	53.1	54.2	65.7	70.0	71.4	73.3	72.2	50.1	50.5	**50.5**	51.2	51.5	50.3	50.3	50.5	51.2	52.3
Min-K%++	49.2	53.1	53.8	56.1	57.9	64.7	69.6	70.9	72.8	73.8	49.7	50.0	49.8	51.2	51.7	50.2	50.8	51.5	52.8	54.0
Gap-K%	48.9	**53.5**	**54.1**	**56.7**	**58.6**	64.0	69.6	71.0	73.1	**74.1**	**50.5**	**50.6**	50.3	**51.6**	**52.0**	**50.9**	**51.3**	**51.7**	**53.2**	**54.3**

	ArXiv					DM Mathematics					HackerNews					Average				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	**51.0**	**51.5**	51.9	52.9	53.4	48.8	48.5	48.4	48.5	48.5	49.4	50.5	51.3	52.1	52.8	52.1	53.1	53.5	54.4	54.5
Zlib	50.1	50.9	51.3	52.2	52.7	48.1	48.2	48.0	48.1	48.1	49.7	50.3	50.8	51.2	51.7	**52.3**	53.2	53.6	54.3	54.5
Neighbor	50.7	51.4	51.8	52.2	/	49.0	47.0	46.8	46.6	/	**50.9**	**51.7**	51.5	51.9	/	52.0	52.9	53.2	53.8	/
Min-K%	50.4	51.4	52.1	53.4	54.3	49.3	49.3	49.1	49.2	49.2	50.6	51.2	52.4	53.5	54.5	52.2	53.4	54.0	55.0	55.5
Min-K%++	49.3	50.9	53.0	53.6	**56.2**	**50.1**	**50.2**	**50.2**	**50.5**	**50.4**	50.7	51.3	**52.6**	54.1	**55.8**	52.0	53.7	54.5	55.9	57.1
Gap-K%	49.9	**51.5**	**53.3**	**53.8**	**56.2**	50.0	49.9	49.9	50.2	50.1	50.2	51.2	52.2	**54.2**	55.6	52.1	**53.9**	**54.6**	**56.1**	**57.3**

olds. We also report the true positive rate at a low false positive rate (TPR@5%FPR).

Baselines. We compare our method with five state-of-the-art baselines: (1) *Loss* (Yeom et al., 2018) directly uses the input loss. (2) *Zlib* (Carlini et al., 2021) calibrates the loss using Zlib compression entropy. (3) *Neighbor* (Mattern et al., 2023) perturbs the input text using a pretrained masked language model and compares the loss of the original sample with the average loss of perturbed samples. (4) *Min-K%* (Shi et al., 2024) computes the average of the lowest $k\%$ token probabilities. (5) *Min-K%++* (Zhang et al., 2025) extends Min-K% by normalizing token-level log probabilities.

Implementation details. We use $k = 20\%$ for Min-K% following the setting in the original paper (Shi et al., 2024). To ensure a fair comparison across methods, we fix $k = 20\%$ for Min-K%++ and Gap-K%. For Gap-K%, we set the window size to 6 for LLaMA-based models and 3 for other models, as these settings consistently showed stable and near-optimal performance across model sizes within each model family.

4.2 Main Results

WikiMIA results. Table 1 reports the AUROC scores on the WikiMIA benchmark. Overall, Gap-K% consistently outperforms existing methods across diverse settings. Averaged over five models, in the original setting, Gap-K% improves upon Min-K%++ by 2.1%, 2.6%, and 2.6% for input lengths of 32, 64, and 128, respectively. For the paraphrased setting, Gap-K% also outperforms Min-K%++ by 0.9%, 2.3%, and 2.0%. Moreover, the performance gains of Gap-K% generalize across a wide range of model sizes and architectures. We further analyze TPR@5%FPR, with full results reported in Appendix A. Since Min-K% and Min-K%++ achieve the strongest AUROC among the baselines, we focus on these two in Fig. 3, which summarizes the average across input lengths. Gap-K% substantially outperforms Min-K%++ in the original setting, with gains of 7.1%, 7.9%, and 3.0% for input lengths of 32, 64, and 128, respec-

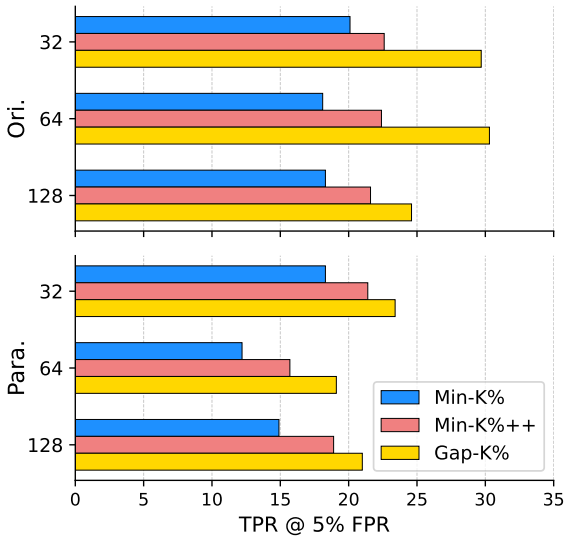


Figure 3: Average TPR@5%FPR across five models on WikiMIA, comparing Min-K%, Min-K%++, and Gap-K%. Results are shown for original and paraphrased inputs at sequence lengths of 32, 64, and 128. Full results are reported in Appendix A.

Table 3: Fraction of tokens whose normalized gap magnitude exceeds threshold τ.

Threshold τ	Train	Non-train
1	0.7244	0.7400
2	0.5063	0.5414
3	0.3553	0.3994
4	0.2478	0.2924
5	0.1638	0.2037
6	0.0998	0.1333

tively. In the paraphrased setting, Gap-K% also achieves consistent gains of 2.0%, 3.4%, and 2.1% for the same input lengths.

MIMIR results. Table 2 reports the AUROC results on the MIMIR benchmark. MIMIR is a particularly challenging setting, as member and non-member samples are drawn from nearly identical distributions, resulting in performance close to random guessing (0.5) for most methods. This difficulty is further amplified in the Pile-CC subset, which consists of noisy web-crawled text with high entropy, weakening memorization signals even for training data (see Appendix B for details). Despite these challenges, Gap-K% achieves the strongest average performance on the MIMIR benchmark across 1.4B, 2.8B, 6.9B, and 12B models. Notably, Gap-K% achieves higher average performance than Min-K%++ across the five evaluated model sizes. The TPR@5%FPR results are in Appendix A.

4.3 Additional Analyses

We conduct additional analyses of Gap-K% on the WikiMIA-64 dataset using Pythia-12B.

Empirical evidence of gap signal. To examine how frequently confident mispredictions occur in non-training data compared to training data, we compute the normalized token-level gap g_t (Eq. 5) and measure the fraction of tokens whose gap magnitude $|g_t|$ exceeds a threshold τ, where τ corresponds to a deviation of $\tau \cdot \sigma_t$ from the top-1 prediction. As shown in Table 3, non-training data consistently exhibits a higher proportion of large-gap tokens across a wide range of thresholds. For example, at $\tau = 3$, such deviations occur in 39.9% of non-training tokens, compared to 35.5% in training data. These results suggest that confident mispredictions are reduced during training, supporting the use of large top-1 gaps as a signal for distinguishing training from non-training data.

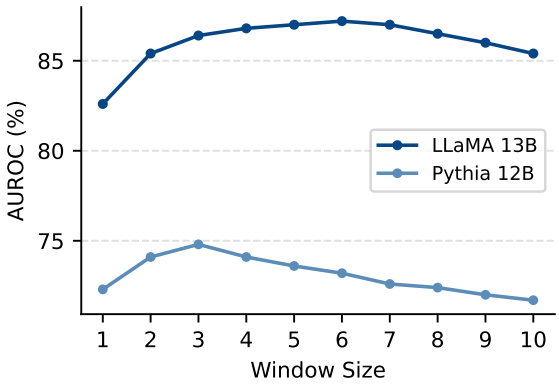


Figure 4: Effect of window size on sequential smoothing for LLaMA-13B and Pythia-12B.

Table 4: Effect of sequential locality. Applying sequential smoothing improves performance only when the original token order is preserved.

Method	AUROC (%)
No smoothing	72.3
Shuffled-order smoothing	72.9
Sequential smoothing (Ours)	74.8

Impact of window size. Fig. 4 presents the AUROC scores for window sizes ranging from 1 to 10 on LLaMA-13B and Pythia-12B. LLaMA-13B and Pythia-12B achieve their best performance at window sizes of 6 and 3, respectively, and the performance degrades for larger window sizes. As the window size increases, smoothing aggregates gap scores over a broader context, which may dilute localized high-gap regions that are indicative of non-training data. We further investigate the difference in optimal window size across model families in Appendix C.

Effect of sequential locality. To examine whether the effectiveness of sequential smoothing stems from local token dependencies, we compare three variants: (1) no smoothing, (2) smoothing applied after randomly shuffling the token order, and (3) sequential smoothing applied to the original token order. As shown in Table 4, sequential smoothing leads to substantial improvements in AUROC only when the original token order is preserved. In contrast, shuffling the token order beforehand yields minimal improvement. This observation indicates that the membership signal exhibits sequential locality, being distributed across contiguous token segments rather than isolated at individual tokens.

Table 5: Ablation of key components relative to Min-K%++. Top-1 denotes replacing the mean in Min-K%++ (Eq. 3) with the top-1 (maximum) log probability, and smoothing denotes applying sequential smoothing.

Method	Top-1	Smoothing	AUROC (%)
Min-K%++			72.6
+ Top-1	✓		72.3
+ Smoothing		✓	73.8
Gap-K% (Ours)	✓	✓	74.8

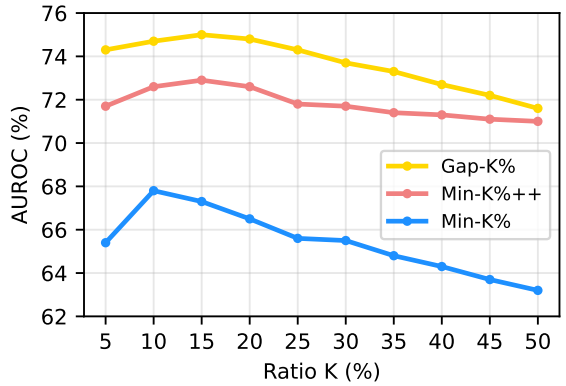


Figure 5: Effect of the $k\%$ ratio on AUROC.

Ablation of key components. We ablate the two key differences from Min-K%++: Top-1 gap score in place of mean-based normalization and sequential smoothing. Table 5 shows that neither modification alone is sufficient to achieve a large performance gain. Replacing the mean-based normalization with the Top-1 gap leads to a marginal drop in performance. On the other hand, applying sequential smoothing to the original Min-K%++ score leads to an improvement in AUROC, indicating that smoothing is generally effective. Importantly, the largest performance gain is achieved when sequential smoothing is applied to the Top-1 gap score. While smoothing alone stabilizes the score, the Top-1 gap provides a more discriminative per-token signal that becomes effective only when combined with sequential smoothing.

Effect of $k\%$. Fig. 5 shows the effect of varying k from 5% to 50%. We observe that the performance peaks around $k = 15\%$, suggesting that focusing on tokens with the largest log probability gaps yields the strongest detection signals. Additionally, Gap-K% consistently outperforms Min-K% and Min-K%++ across the 5%–50% range, regardless of the choice of k.

Generalization to recent models. We further evaluate Gap-K% on recent LLMs, including LLaMA 3.1-8B (Grattafiori et al., 2024) and Gemma2-9B (Team et al., 2024), as well as their instruction-tuned variants. Since the training data of recent models temporally overlaps with the original WikiMIA benchmark, we adopt WikiMIA-25 (Yi and Li, 2026) (length 64 split) for this evaluation. As shown in Table 6, Gap-K% consistently outperforms all baseline methods across both base and instruction-tuned models. These results further demonstrate that the proposed signal is not tied to a specific architecture or training setup and remains effective under modern training regimes, including instruction tuning and reinforcement learning from human feedback (RLHF).

Table 6: AUROC results on recent state-of-the-art LLMs (LLaMA 3.1 and Gemma2) and their instruction-tuned variants, evaluated on WikiMIA-25.

Method	LLaMA 3.1 8B	LLaMA 3.1 8B Instr.	Gemma2 9B	Gemma2 9B Instr.
Loss	66.9	64.9	63.3	61.4
Zlib	67.3	65.3	63.7	61.6
Neighbor	69.2	67.0	65.6	61.8
Min-K%	71.1	68.4	65.1	64.3
Min-K%++	82.7	73.1	75.6	64.5
Gap-K%	**84.1**	**76.6**	**78.3**	**65.8**

Table 7: AUROC results under adversarial paraphrasing using DIPPER.

Method	AUROC (%)
Loss	52.3
Zlib	50.0
Neighbor	60.3
Min-K%	57.9
Min-K%++	65.5
Gap-K%	**66.6**

Robustness to adversarial paraphrases. We evaluate the robustness of Gap-K% under strong adversarial paraphrasing. To the best of our knowledge, there is no established method specifically designed to generate paraphrases for evading pretraining data detection. We adopt DIPPER (Krishna et al., 2023), a paraphrasing framework originally proposed for attacking AI-generated text detectors. We employ the strongest setting reported in the DIPPER paper, with lexical diversity $L = 60$ and order diversity $O = 60$. As shown in Table 7, Gap-K% outperforms baseline methods even under these challenging conditions, demonstrating its robustness to strong adversarial paraphrasing.

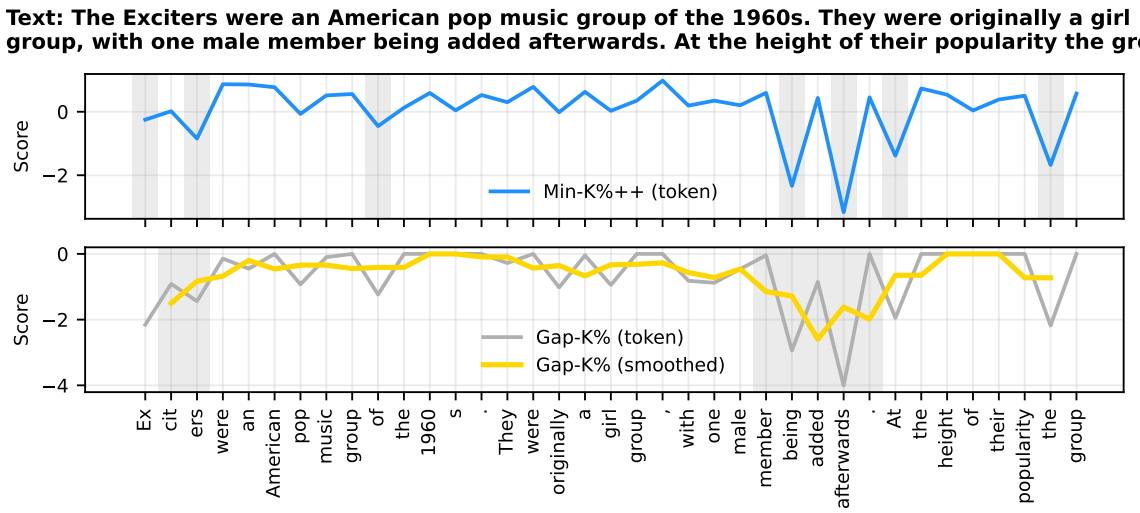


Figure 6: Visualization of token-level scores for Min-K%++ and Gap-K% on a representative WikiMIA-32 sample using Pythia-12B. Min-K%++ (top) selects tokens based on token-level scores, with shaded regions denoting the bottom 20% of tokens. Gap-K% (bottom) applies sequential smoothing to token-level scores and selects the bottom 20% based on the resulting window-averaged scores, which are indicated by shaded regions.

Qualitative example. We analyze the behavior of token-level scores produced by Min-K%++ and Gap-K% to clarify how low-percentage selection mechanisms differ between the two methods. While the main quantitative experiments in prior sections are conducted on WikiMIA-64, we use WikiMIA-32 here for improved clarity in visualization. Fig. 6 visualizes the token-level scores for a representative sample. We observe that Min-K%++ selects multiple low-scoring tokens distributed across the sequence, including both pronounced and minor low-score points. These selected tokens appear at isolated positions rather than forming segments. In contrast, Gap-K% applies sequential smoothing before selection, which reflects score trends over neighboring tokens. As a consequence, the highlighted regions correspond to spans where scores remain relatively low across a short range, rather than isolated tokens. This suggests that smoothing allows Gap-K% to capture membership signals at the level of local patterns, whereas Min-K%++ operates on individual token values.

5 Conclusion

In this work, we studied pretraining data detection through the lens of the optimization dynamics of autoregressive large language models (LLMs). By analyzing the gradient behavior induced by the next-token prediction objective, we identified discrepancies between top-1 predictions and observed tokens as a meaningful signal for membership inference. Based on this insight, we proposed Gap-K%, a simple yet effective detection method that exploits top-1 log probability gaps and incorporates sequential smoothing to capture local correlations. Extensive experiments on the WikiMIA and MIMIR benchmarks demonstrate that Gap-K% consistently outperforms prior state-of-the-art methods. Overall, our findings highlight the importance of top-1 prediction behavior, together with sequential smoothing to capture local correlations. We hope this perspective motivates further exploration for understanding memorization, privacy risks in LLMs.

Limitations

Our method operates under a gray-box assumption, requiring access to token-level probability outputs from the target model. While this setting is common in prior work on pretraining data detection and is also assumed by all baselines in our evaluation, it restricts the applicability of our approach to models or APIs that explicitly expose such information. Extending such approaches to fully black-box settings is an important direction for future work.

Although we evaluate Gap-K% on recent model families including LLaMA 3.1 and Gemma2 with their instruction-tuned variants, a number of other recently released LLM families are not covered in our experiments. In addition, our evaluation is limited to models up to tens of billions of parameters, leaving the behavior of Gap-K% at the hundreds-of-

billions scale unverified. A broader investigation across diverse model families and scales would further clarify the generality of our findings.

Finally, while we assess robustness under strong paraphrasing transformations using DIPPER, these evaluations do not fully capture adaptive adversarial scenarios in which paraphrases are explicitly optimized to evade pretraining data detection. In such settings, an adversary may exploit knowledge of the detection mechanism to manipulate token-level statistics, and evaluating robustness against such detection-aware attacks remains an open challenge.

Ethical Statements

This work is conducted with careful attention to ethical research practices and responsible use of data and models. The primary objective of our study is to improve the understanding of model behavior and privacy-related risks in LLMs. All experiments in this study are performed using publicly available datasets and pretrained models that are released for research purposes under permissive open-source licenses. Based on the available documentation provided by the dataset and model developers, the resources used in this work do not contain personally identifiable information (PII) and are commonly adopted benchmarks within the research community.

While our methods aim to enhance transparency and facilitate the detection of potential privacy risks in pretrained models, we acknowledge that techniques could be misused if deployed irresponsibly. In particular, inferring characteristics of training data may pose risks when applied to sensitive or unauthorized data sources. To mitigate such concerns, we emphasize that our approach is intended strictly for non-commercial, academic research. Overall, this work aims to contribute to ongoing discussions on transparency and privacy considerations in LLMs.

Acknowledgments

All authors are affiliated with the Department of Artificial Intelligence at Yonsei University. This research was supported in part by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. RS-2020-II201361, Artificial Intelligence Graduate School Program (Yonsei University); No. RS-2025-02215344, Development of AI Technology with Robust and Flexible Resilience Against Risk Factors.

References

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 2397–2430. PMLR.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, and 1 others. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*.

André Vicente Duarte, Xuandong Zhao, Arlindo L Oliveira, and Lei Li. 2024. De-cop: Detecting copyrighted content in language models training data. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11940–11956. PMLR.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Xinyang Geng and Hao Liu. 2023. Openllama: An open reproduction of llama.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:27469–27500.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schoelkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11330–11343.

Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2024. Did the neurons read your book? document-level membership inference for large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 2369–2385.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022. Quantifying privacy risks of masked language models using membership inference attacks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8332–8347.

OpenAI. 2024. Hello gpt-4o.

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

Noorjahan Rahman and Eduardo Santacana. 2023. Beyond fair use: Legal risk evaluation for training llms on copyrighted text. In *ICML Workshop on Generative AI and Law*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *International Conference on Learning Representations (ICLR)*.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Thomas Steinke, Milad Nasr, and Matthew Jagielski. 2023. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems (NeurIPS)*, 36:49268–49280.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

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*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC conference on computer and communications security*, pages 3093–3106.

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

Jiatong Yi and Yanyang Li. 2026. Membership inference on llms in the wild. *arXiv preprint arXiv:2601.11314*.

Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025. Min-k%++: Improved baseline for pre-training data detection from large language models. In *International Conference on Learning Representations (ICLR)*.

Table 8: TPR@5%FPR results on WikiMIA. The best and second-best scores are highlighted in **bold** and underlined, respectively.

Len.	Method	Mamba-1.4B		Pythia-6.9B		Pythia-12B		LLaMA-13B		LLaMA-65B		Average	
		Ori.	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*	*Ori.*	*Para.*
32	Loss	14.2	**14.2**	14.2	15.0	17.1	17.3	14.0	16.3	22.0	21.7	16.3	16.9
	Zlib	15.5	13.2	16.3	12.7	17.1	15.5	11.6	15.0	19.6	17.3	16.0	14.7
	Neighbor	11.9	7.2	16.5	9.6	19.4	9.8	11.6	8.5	6.5	12.1	13.2	9.4
	Min-K%	14.2	11.9	**17.8**	**21.7**	23.0	**19.9**	18.9	14.2	26.4	23.8	20.1	18.3
	Min-K%++	11.4	7.8	14.5	14.5	16.5	15.5	33.1	**33.9**	37.5	**35.4**	22.6	21.4
	Gap-K%	**17.3**	13.4	16.3	18.9	**24.0**	19.4	**43.4**	**33.9**	**47.3**	31.3	**29.7**	**23.4**
64	Loss	9.5	8.1	13.4	10.6	9.2	11.6	11.3	12.0	15.5	13.4	11.8	11.1
	Zlib	14.1	**15.1**	16.2	**15.8**	11.3	**16.2**	12.7	13.4	17.6	18.3	14.4	15.8
	Neighbor	8.8	9.5	10.9	12.7	11.3	10.6	10.2	14.4	9.9	16.9	10.2	12.8
	Min-K%	15.8	7.7	19.0	12.7	21.5	12.7	17.3	13.4	16.9	14.4	18.1	12.2
	Min-K%++	13.7	7.0	16.2	10.2	16.9	10.9	31.3	23.2	33.8	27.1	22.4	15.7
	Gap-K%	**20.1**	10.9	**23.6**	13.0	**25.4**	13.7	**44.7**	**30.3**	**37.7**	**27.8**	**30.3**	**19.1**
128	Loss	11.5	13.7	14.4	16.5	18.0	**19.4**	21.6	18.0	20.1	24.5	17.1	18.4
	Zlib	**19.4**	**17.3**	**20.9**	**20.9**	**23.7**	**19.4**	18.7	21.6	23.0	22.3	21.1	20.3
	Neighbor	15.8	13.7	10.8	17.3	10.1	10.1	12.9	13.7	15.8	18.7	13.1	14.7
	Min-K%	9.4	5.0	18.0	16.5	20.1	18.7	20.1	15.1	23.7	19.4	18.3	14.9
	Min-K%++	10.1	6.5	20.1	18.0	18.0	9.4	38.1	**35.3**	21.6	25.2	21.6	18.9
	Gap-K%	16.5	13.7	19.4	13.7	21.6	18.0	**39.6**	33.1	**25.9**	**26.6**	**24.6**	**21.0**

Table 9: TPR@5%FPR results on MIMIR. The best and second-best scores are highlighted in **bold** and underlined, respectively.

Method	Wikipedia					Github					Pile CC					PubMed Central				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	4.2	4.7	4.7	5.1	5.0	22.6	32.1	33.6	38.5	30.3	3.1	5.0	4.8	4.9	5.1	4.0	4.4	4.3	4.9	5.0
Zlib	4.2	**5.7**	5.9	6.3	6.8	25.1	32.8	**36.2**	**40.1**	32.9	4.0	**5.1**	**5.4**	**6.2**	**6.6**	3.8	3.6	3.5	4.3	4.4
Neighbor	4.0	4.5	4.9	5.8	/	24.7	31.6	29.8	34.1	/	3.9	3.6	4.0	5.3	/	3.9	3.7	4.5	4.5	/
Min-K%	4.8	5.6	5.0	6.1	5.8	22.6	31.5	34.0	39.0	32.6	3.5	4.5	4.8	5.0	4.8	4.7	4.6	4.5	5.1	4.9
Min-K%++	5.2	5.3	5.9	7.0	**7.8**	**25.2**	**33.0**	34.2	38.2	34.5	**5.0**	3.7	3.7	4.8	4.6	4.8	**6.1**	4.8	5.6	**6.4**
Gap-K%	**5.5**	5.4	**6.0**	**7.5**	7.0	22.9	32.3	34.3	38.5	**34.8**	3.7	4.6	4.3	5.8	5.9	**5.4**	5.2	**5.4**	**5.7**	5.8

Method	ArXiv					DM Mathematics					HackerNews					Average				
	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	4.0	**4.8**	4.6	5.4	5.6	3.8	4.3	4.1	4.1	4.0	5.0	4.8	5.5	5.9	**6.8**	6.7	8.6	8.8	9.8	8.8
Zlib	2.9	4.3	4.1	4.6	4.7	4.1	5.0	4.6	4.3	4.3	5.0	**5.5**	**5.8**	5.6	5.8	7.0	**8.9**	**9.4**	10.2	9.4
Neighbor	4.7	**4.8**	4.4	4.1	/	**5.6**	4.4	4.5	4.5	/	**6.5**	5.2	5.3	5.7	/	7.6	8.3	8.2	9.1	/
Min-K%	4.4	4.3	4.5	5.4	5.3	3.9	4.1	4.6	4.3	4.6	4.2	4.6	5.7	**6.3**	6.1	6.9	8.5	9.0	10.2	9.2
Min-K%++	**5.4**	4.7	**6.1**	**6.8**	**7.0**	4.4	4.8	**5.4**	4.5	**5.4**	4.4	3.5	4.6	5.7	5.7	**7.8**	8.7	9.2	10.4	**10.2**
Gap-K%	5.2	4.1	5.3	6.4	6.1	4.2	**5.2**	4.8	**5.3**	**5.4**	3.8	3.9	5.1	6.2	5.1	7.2	8.7	9.3	**10.8**	10.0

A Full TPR@5%FPR Results

We report full TPR@5%FPR results on the WikiMIA and MIMIR benchmarks in Tables 8 and 9, respectively. TPR@5%FPR evaluates performance at the extreme tail of the score distribution, where only a small number of samples determine the outcome. As a result, this metric is inherently sensitive to variance across datasets and models, which can make consistent trends less apparent. Similar fluctuations have been observed in prior work (Zhang et al., 2025), suggesting that this behavior is inherent to the metric. For WikiMIA, Gap-K% consistently outperforms Min-K%++ across different input lengths. In the original setting, averaged over five models, Gap-K% improves upon Min-K%++ by 7.1%, 7.9%, and 3.0% for input lengths of 32, 64, and 128, respectively. In the paraphrased setting, Gap-K% also achieves gains of 2.0%, 3.4%, and 2.1% over Min-K%++ for the same input lengths. These results further demonstrate the robustness of Gap-K% across both original and paraphrased settings. For MIMIR, no single method consistently achieves the best performance across all model sizes. Despite this, Gap-K% attains the highest average performance in the Pythia-6.9B setting and achieves the second-best average scores for the 1.4B, 2.8B, and 12B models.

Table 10: Average entropy of next-token distributions across MIMIR subsets for training and non-training samples. Pile-CC exhibits the highest entropy among all domains.

	ArXiv	DM Mathematics	Github	HackerNews	Pile CC	PubMed Central	Wikipedia
Train	2.05	1.28	0.82	2.51	2.59	2.00	2.01
Not Train	2.05	1.27	1.22	2.51	2.58	1.99	2.01

Table 11: AUROC (%) across different window sizes for models with LLaMA-based architectures. All models consistently achieve their best performance at window size 6.

Model	Window Size						
	2	3	4	5	6	7	8
OpenLLaMA-3B	77.4	78.4	78.7	**78.9**	**78.9**	78.6	77.8
Mistral-7B	86.3	87.8	88.7	89.3	**90.0**	89.3	88.4
LLaMA-13B	81.3	82.2	83.0	83.4	**83.9**	83.8	83.5

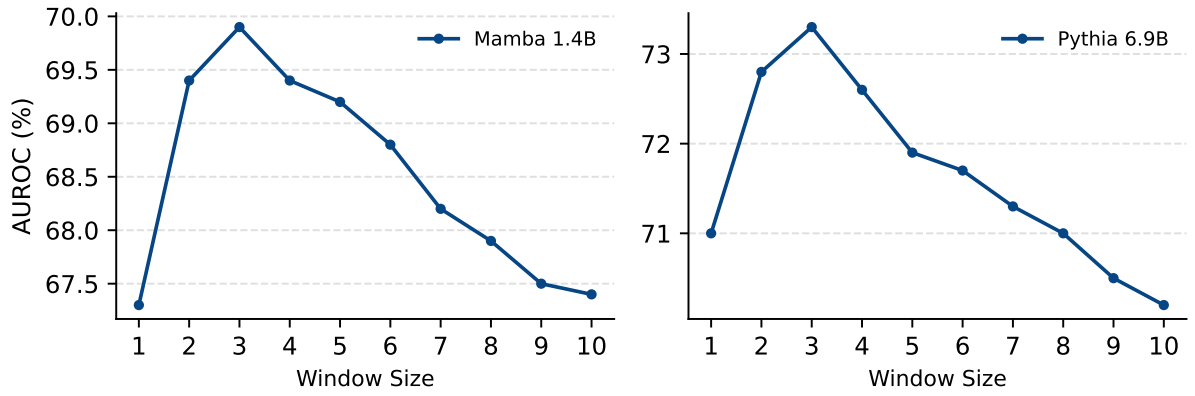


Figure 7: Effect of window size on sequential smoothing for Mamba-1.4B and Pythia-6.9B.

B Understanding Difficulty of Pile-CC

We further investigate why detection methods struggle on the Pile-CC subset of MIMIR. To better understand this behavior, we analyze the entropy of the next-token distribution using Pythia-2.8B. As shown in Table 10, Pile-CC exhibits the highest average entropy among all domains. This suggests that the domain is highly heterogeneous and distributionally broad, with substantial artifacts and formatting noise in pretraining web corpora. As a result, the model struggles to assign high probability to the target token even for training samples, leading to weaker memorization signals and limiting the separability for all detection methods.

C Effect of Window Size across Model Families

We investigate why the optimal window size differs across model families. We hypothesize that this difference may arise from either training procedures or architectural characteristics. To disentangle these factors, we evaluate models that share architectural similarities with LLaMA but differ in training methodology, including Mistral-7B v0.1 (Jiang et al., 2023) and OpenLLaMA-3B (Geng and Liu, 2023). Since the original WikiMIA is not suitable for these newer models due to potential temporal overlap with their training data, we instead evaluate on WikiMIA-25 (length 64 split). As shown in Table 11, all models consistently achieve their best performance at window size 6. We further confirm that LLaMA-13B exhibits the same optimal window size under WikiMIA-25. These results suggest that the observed differences in optimal window size are more likely attributable to architectural factors rather than training procedures.

D Window Size for Other Models

We further evaluate the effect of window size across additional models. Following the experimental

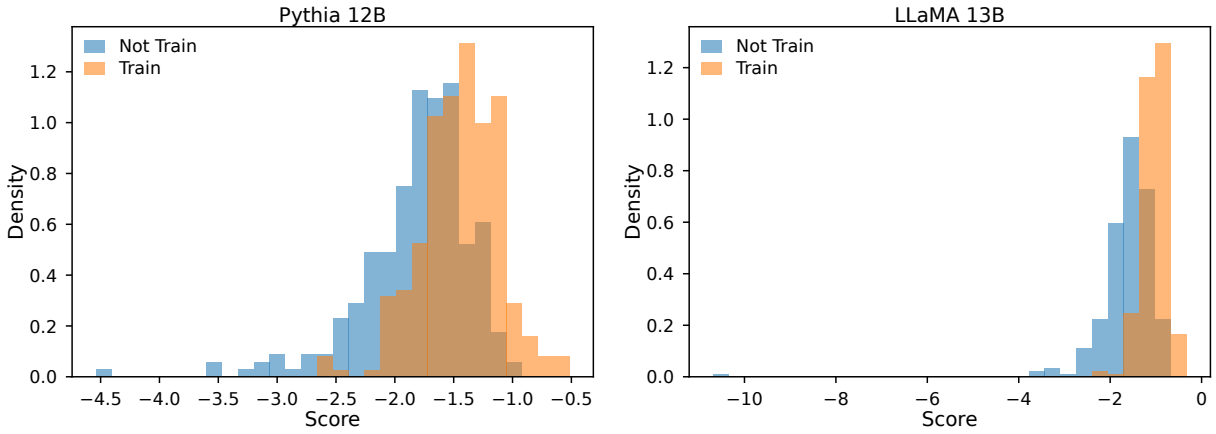


Figure 8: Histogram of the Gap-K% score distributions for trained and not trained samples on the WikiMIA-64 benchmark. The left shows results from Pythia-12B, and the right shows results from LLaMA-13B.

Table 12: Effect of token selection strategy.

Token Selection	AUROC (%)
All tokens (no k% selection)	68.6
Random-K%	62.6
Top-K%	54.2
Bottom-K% (Ours)	**74.8**

setup in Section 4.3, we use the WikiMIA-64 dataset. Due to the high computational cost, we do not include results for LLaMA-65B. Fig. 7 shows that both Mamba-1.4B and Pythia-6.9B achieve their best performance at a window size of 3, consistent with the results for Pythia-12B.

E Effect of Token Selection Strategy

We analyze different token selection strategies to validate our choice of selecting low-scoring tokens. As shown in Table 12, selecting the bottom k% tokens (Bottom-K%) achieves the highest AUROC. In contrast, selecting high-scoring tokens (Top-K%) or random tokens leads to substantially worse results. Using all tokens without k% selection also underperforms compared to Bottom-K%, highlighting the importance of focusing on informative tokens. Notably, low-scoring tokens tend to correspond to those with larger gaps between the target and top-1 predictions, making them particularly informative under our gap-based criterion.

F Histogram-Based Comparison of Training and Non-Training Samples

Fig. 8 visualizes the distributions of the proposed Gap-K% scores for member and non-member samples. We plot the results for Pythia-12B and LLaMA-13B on the WikiMIA-64 benchmark. The histograms illustrate the degree of separation induced by the score.

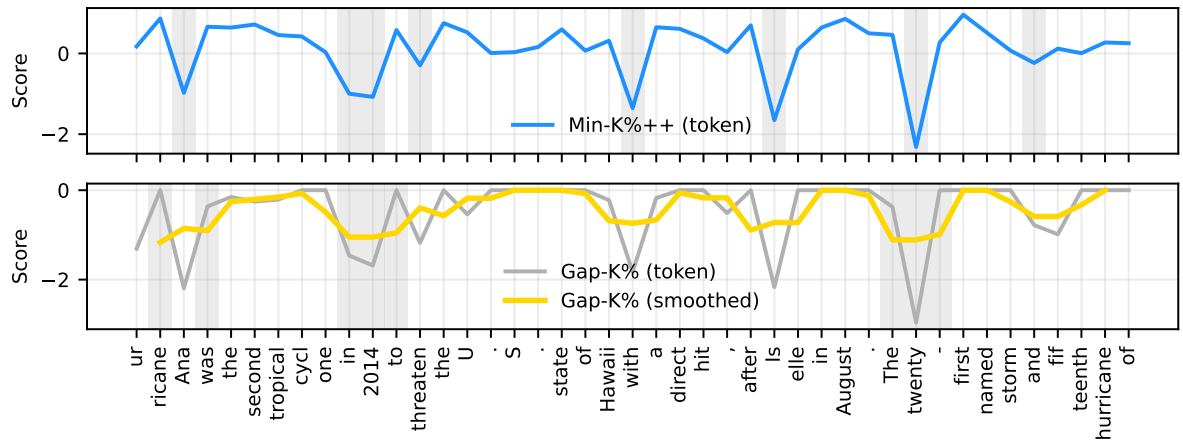
G Additional Token-level Visualization

Fig. 9 presents additional qualitative examples of token-level score visualizations for Min-K%++ and Gap-K%. Consistent with the observation in Fig. 6 in the main text, Min-K%++ selects isolated low-scoring tokens across the sequence, whereas Gap-K% identifies contiguous low-score regions after applying sequential smoothing.

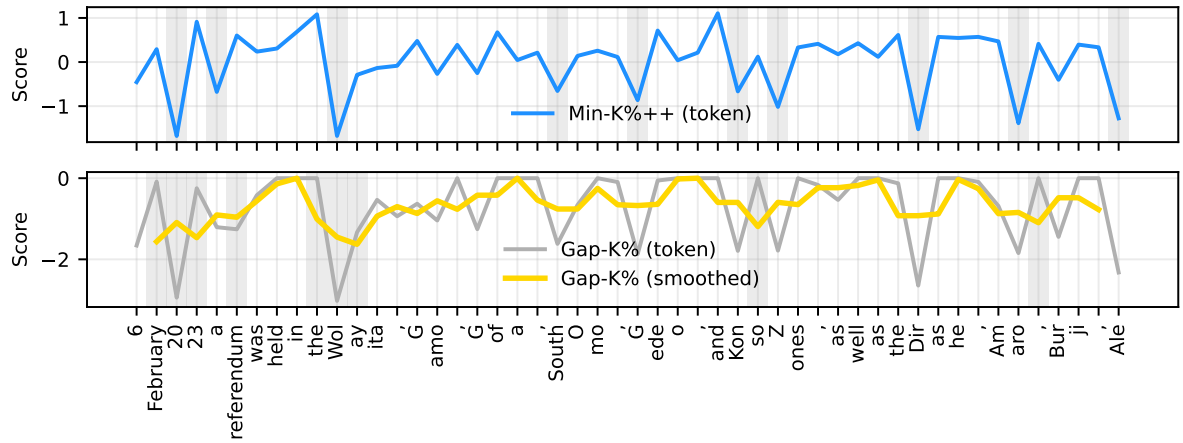
H Usage of AI assistants

During the preparation of this paper, AI-assisted writing tools were used for editorial purposes, including improving readability, coherence, and grammatical accuracy. Their use was restricted to language polishing, without generating or altering any technical content. All aspects of the method and experimental outcomes were developed independently by the authors. The application of AI assistance was limited to surface-level revisions and did not affect the originality or scientific substance of the work.

Text: Hurricane Ana was the second tropical cyclone in 2014 to threaten the U.S. state of Hawaii with a direct hit, after Iselle in August. The twenty-first named storm and fifteenth hurricane of



Text: On 6 February 2023 a referendum was held in the Wolayita, Gamo, Gofa, South Omo, Gedeo, and Konso Zones, as well as the Dirashe, Amaro, Burji, Ale, and Basketo special woredas of



Text: A crisis at the National Public Health Laboratory in Khartoum, Sudan, started after it was seized by armed forces during the Sudan conflict in April 2023. The World Health Organization (WHO) said

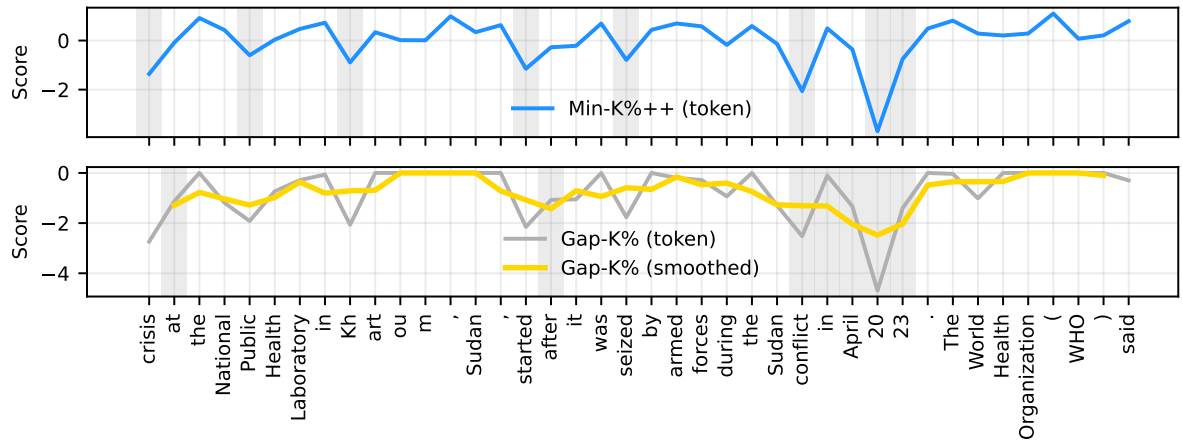


Figure 9: Additional examples of token-level score visualizations for Gap-K% and Min-K%++.