

Black-Box Membership Inference via Word-Level Probability Estimation

Shengjie Niu¹, Yeheng Ge¹, and Jian Huang^{1,2*}

¹Department of Data Science and Artificial Intelligence
The Hong Kong Polytechnic University, Hong Kong SAR, China

²Department of Applied Mathematics
The Hong Kong Polytechnic University, Hong Kong SAR, China
shengjie.niu@connect.polyu.hk, {yeheng.ge, j.huang}@polyu.edu.hk

Abstract

Membership inference attacks (MIAs) have emerged as critical tools for auditing privacy risks in large language models (LLMs), aiming to determine whether a given text was included in a model’s training corpus. However, most existing MIAs require access to per-token logits or probabilities, making them inapplicable in practice to proprietary LLMs that expose only textual continuations. To address this underexplored setting, we propose Word-level Probability MIA (WPMIA), a statistically principled MIA for strict black-box privacy auditing. WPMIA estimates word-level generation probabilities via Monte Carlo sampling with local kernel smoothing, then aggregates these estimates into a sequence-level likelihood estimator. Furthermore, WPMIA constructs the likelihood conditioned on different prefixes, thereby amplifying the distributional differences between members and non-members. We evaluate WPMIA across various open-source LLMs and find that it consistently outperforms existing black-box baselines. Importantly, we also evaluate WPMIA on modern proprietary LLMs, including GPT-5-Chat, Gemini-2.5-Flash, and Claude-4.5-Haiku, achieving an average TPR@5%FPR of 42.0 across these models. These results offer a sound foundation for future research on strict black-box membership inference. Code is available at <https://github.com/niusj03/WPMIA>.

1 Introduction

As the pretraining corpora of modern large language models (LLMs) continue to grow in scale (Raffel et al., 2020; Brown et al., 2020), they may still contain sensitive or restricted content despite data filtering (Almazrouei et al., 2023). This raises concerns about the inclusion of copyrighted content (Chang et al., 2023; Duarte et al., 2024) and personally identifiable information (Mozes et al., 2023; Tang et al., 2024). To audit these

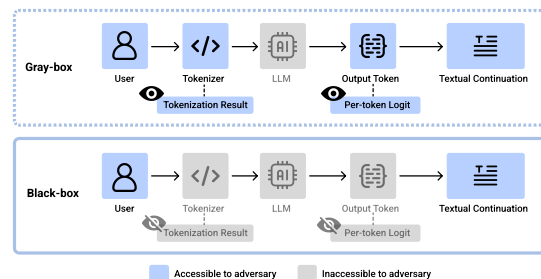


Figure 1: **Adversary access levels.** *Gray-box*: the adversary has access to tokenization results and per-token logits, but not to the model’s internal weights. *Black-box (of interest)*: the adversary has no access to model weights, tokenization results, or token logits, and can only observe textual continuations.

risks, pretraining data detection (Shi et al., 2024) aims to determine whether a given text was included in the training corpus of a target LLM, typically through membership inference attacks (MIAs) (Shokri et al., 2017). Prior studies have developed effective MIA methods for LLMs (Carlini et al., 2021; Mattern et al., 2023; Shi et al., 2024; Zhang et al., 2025). However, most of these methods require *gray-box* access, where the adversary can obtain tokenization results and per-token logits, as shown in Fig. 1. In practical deployment, proprietary LLMs such as ChatGPT and Claude are typically accessed through APIs or subscriptions, where users can only observe textual continuations. Consequently, these gray-box methods are inapplicable to *black-box* proprietary LLMs that do not disclose per-token logits. Despite its practical relevance, black-box pretraining data detection remains underexplored.

MIA methods generally rely on the overfitting effect that a text included in the training data (*member*) tends to have a higher likelihood than a text not seen during training (*non-member*) (Yeom et al., 2018). Given a target text split into a sequence of tokens, gray-box MIA methods can directly compute sequence-level likelihoods from token-

*Corresponding author.

level logits (Chen et al., 2025). Based on this likelihood signal, they further improve detection by focusing on low-probability tokens (Shi et al., 2024; Zhang et al., 2025), measuring likelihood changes under textual perturbations (Mattern et al., 2023; Xie et al., 2024; Wang et al., 2025). In contrast, recent black-box methods (He et al., 2025; Kaneko et al., 2025; Yi and Li, 2026) cannot directly compute likelihoods and instead attempt to recover membership signals from generated continuations alone. However, these methods are developed largely through heuristics rather than a principled statistical formulation of the membership signal. Therefore, constructing a reliable and theoretically grounded black-box MIA remains an open problem.

We propose Word-level Probability MIA (WPMIA), a statistically principled MIA for the black-box setting. The key insight is to recover the likelihood signal directly from textual continuations. Given a target text represented as a sequence of words, WPMIA estimates the generation probability of each word from Monte Carlo samples (Robert and Casella, 2004) of the target model’s continuations, and aggregates these word-level probabilities into a sequence-level likelihood estimator. Moreover, WPMIA constructs both member and non-member prefixes and estimates the corresponding prefix-conditioned sequence-level likelihoods. It then contrasts the likelihoods conditioned on different prefixes, thereby amplifying the membership signal. Compared with existing black-box MIAs, WPMIA directly estimates the sequence-level likelihood signal, providing a statistically sound alternative to heuristic black-box scoring. Importantly, when applying prefix-based enhancement, WPMIA first aggregates word-level probabilities into sequence-level likelihoods and then contrasts the conditional likelihood, avoiding the unstable word-level ratios.

We conduct extensive experiments on three standard benchmarks, WikiMIA (Shi et al., 2024), MIMIR (Duan et al., 2024), and WikiMIA-25 (Yi and Li, 2026), covering a broad range of open-source LLM families. Importantly, we also evaluate recent proprietary LLMs, including GPT-5-Chat (OpenAI, 2025), Gemini-2.5-Flash (Google DeepMind, 2025), and Claude-4.5-Haiku (Anthropic, 2025). (1) On open-source LLMs, WPMIA achieves the strongest black-box MIA performance and matches or exceeds many gray-box MIAs. (2) On proprietary LLMs, WPMIA remains

effective, achieving an average TPR@5%FPR of 42.0 across commercial models, which validates the practical feasibility of black-box privacy auditing for commercial LLMs without access to the target tokenizer or per-token logits.

2 Method

2.1 Preliminary

Membership Inference against LLMs. Let f_θ denote an autoregressive LLM parameterized by θ and trained on a corpus D . Denote a text as a sequence of tokens $x = (x_1, \dots, x_L)$, where $x_{<i} := (x_1, \dots, x_{i-1})$ denotes the preceding tokens before x_i . Given $x_{<i}$, the model defines a conditional distribution $f_\theta(\cdot \mid x_{<i})$ over the vocabulary, and $f_\theta(x_i \mid x_{<i})$ denotes the probability assigned to the next token x_i . Membership inference attacks (MIAs) aim to determine whether a text x was included in D . By the chain rule, the LLM assigns x the following probability:

$$p_\theta(x) = \prod_{i=1}^L f_\theta(x_i \mid x_{<i}). \quad (1)$$

In practice, the average log-likelihood (LL) is used for numerical stability:

$$LL(x) = \frac{1}{L} \sum_{t=1}^L \log f_\theta(x_t \mid x_{<t}). \quad (2)$$

LL reflects the probability that the target model generates x token-by-token, and is used as a membership signal by many MIA methods. Generally, an MIA method assigns each text x a membership score $S(\theta, x)$ based on the log-likelihood $LL(x)$ and decides the membership bit by thresholding this score with a decision threshold κ :

$$\mathcal{A}(\theta, x) = \mathbb{I}[S(\theta, x) > \kappa], \quad (3)$$

where $\mathbb{I}[\cdot]$ is the indicator function. Despite their effectiveness, these likelihood-based MIAs require access to the token-level logits or probabilities when querying the target model f_θ , which are unavailable for many proprietary LLMs.

Threat Model. The threat model specifies the information available to the adversary when querying the target model. Most existing MIAs assume a *gray-box* setting (Carlini et al., 2021; Shi et al., 2024; Zhang et al., 2025; Xie et al., 2024; Wang et al., 2025), where the adversary has access to tokenization results and per-token logits, but not to

the model’s internal weights. We consider a more realistic *black-box* setting, where the adversary has no access to model weights, tokenization results, or logits, and can only observe generated textual continuations (Kaneko et al., 2025). This setting aligns with API interactions for proprietary LLMs, such as ChatGPT, Gemini, and Claude.

2.2 Estimating Sequence-level Likelihood

The ground-truth likelihood of the target text is unavailable in the black-box setting, so we first aim to estimate its sequence-level likelihood. Moreover, since the target model’s tokenizer is unavailable, we consider words rather than model-specific tokens as the basic units, and split the text into a word sequence $x = (x_1, \dots, x_L)$ using a standard word tokenizer (Bird et al., 2009). We start from word-level primitives and work on the log scale, so that the resulting score preserves the additive structure of log-likelihoods in Eq. (2). For each word x_i , the word-level log-probability is

$$\widehat{\ell}(x_i|c_i) = \log(\epsilon + \widehat{p}(x_i|c_i)), \quad (4)$$

where $\widehat{p}(x_i|c_i)$ denotes the estimated probability of generating x_i conditioned on the context c_i , and $\epsilon > 0$ prevents zero-valued estimates.

We use Monte Carlo sampling (Robert and Casella, 2004) to estimate $\widehat{p}(x_i|c_i)$. Specifically, for a context c_i , we query f_θ M times and collect the *first* generated word from each response. Let $\widehat{x}_{i,j}^{c_i}$ denote the first word generated in the j -th query conditioned on c_i . We estimate the word-level probability mass assigned to x_i by

$$\widehat{p}(x_i|c_i) = \frac{1}{M} \sum_{j=1}^M K(x_i, \widehat{x}_{i,j}^{c_i}), \quad (5)$$

where M is the number of Monte Carlo samples and K is a kernel function that assigns probability mass to x_i according to the sampled word $\widehat{x}_{i,j}^{c_i}$.

A straightforward choice is the exact-match kernel: $K_{\text{em}}(x_i, \widehat{x}) = \mathbb{I}[x_i = \widehat{x}]$, under which Eq. (5) becomes the empirical probability of sampling x_i as the next word. However, with a limited sampling budget and a large vocabulary, exact matching can lead to sparse or near-zero estimates. We therefore use a semantic kernel

$$K_\tau(x_i, \widehat{x}) = \exp\left(\frac{\cos(e(x_i), e(\widehat{x})) - 1}{\tau}\right), \quad (6)$$

where $e(\cdot)$ denotes a word embedding function and τ is a temperature parameter that controls the

smoothness of the semantic kernel. This kernel estimates the probability of x_i by locally averaging over generated words in the embedding space: words that are semantically closer to x_i receive larger weights, while distant words contribute less. This local smoothing reduces the sparsity of exact matching and produces more stable probability estimates under a limited sampling budget (Takezawa, 2005).

This formulation has two key advantages. First, the semantic kernel provides semantic robustness: not only exact matches but also semantically related words can provide evidence for the target word. Second, the log-scale estimator in Eq. (4) enables additive aggregation over words, yielding the sequence-level log-likelihood estimator

$$\widehat{LL}^0(x) = \frac{1}{L} \sum_{i=1}^L \widehat{\ell}(x_i|c_i^0), \quad c_i^0 = x_{<i}, \quad (7)$$

where $\widehat{\ell}(x_i|c_i^0)$ is the word-level log-probability in Eq. (4). This form aligns with the token-sequence likelihood formulation in Eq. (2).

2.3 Enhancing with Contrastive Prefixes

Motivated by ReCaLL (Xie et al., 2024) and ConReCall (Wang et al., 2025), we view prefix conditioning as a test-time intervention that induces distinct log-likelihood shifts for member and non-member texts. We use these shifts to strengthen the membership signal through relative comparison. Specifically, we concatenate either a non-member prefix or a member prefix with the preceding words of the target text. For $q \in \{nm, m\}$, where nm and m denote non-member and member prefix types, respectively, we define

$$c_i^q = P_q \oplus x_{<i}, \quad (8)$$

where P_{nm} and P_m denote non-member and member prefixes, respectively. Each prefix is constructed by concatenating T texts: $P_q = p_1^q \oplus p_2^q \oplus \dots \oplus p_T^q$ for $q \in \{nm, m\}$.

Given the prefix-conditioned context c_i^q , we define the prefix-conditioned sequence-level LL estimator as $\widehat{LL}^q(x) = \frac{1}{L} \sum_{i=1}^L \widehat{\ell}(x_i|c_i^q)$. We then use the normalized contrastive LL as the membership score:

$$\mathcal{S}(\theta, x) = \frac{\widehat{LL}^{nm}(x) - \widehat{LL}^m(x)}{\widehat{LL}^0(x)}, \quad (9)$$

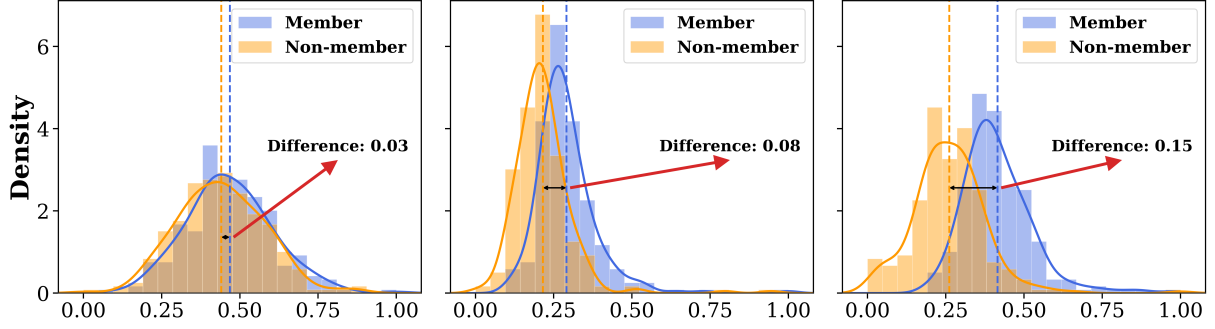


Figure 2: **Visualization of normalized membership-score distributions.** From left to right, we show min-max-normalized distributions of the unprefixed score $\mathcal{S}_0 = \widehat{LL}^0(x)$, the non-member-prefix relative score $\mathcal{S}_{nm} = \widehat{LL}^{nm}(x)/\widehat{LL}^0(x)$, and the contrastive-prefix score $\mathcal{S}_{ctr} = (\widehat{LL}^{nm}(x) - \widehat{LL}^m(x))/\widehat{LL}^0(x)$. The contrastive-prefix score yields the largest separation between members and non-members.

where $\widehat{LL}^0(x)$ is the unprefixed sequence-level log-likelihood estimator in Eq. (7). A larger value of $\mathcal{S}(\theta, x)$ indicates stronger evidence of membership.

Fig. 2 illustrates how the membership score distribution changes as prefix conditioning and contrastive comparison are introduced. Compared with the unprefixed score $\mathcal{S}_0$, the non-member-prefix relative score $\mathcal{S}_{nm}$ increases the separation between members and non-members, while the contrastive-prefix score $\mathcal{S}_{ctr}$ achieves the largest separation. This pattern is consistent with the prefix-induced likelihood shifts observed in prior gray-box settings (Wang et al., 2025). When conditioned on non-member prefixes, member texts tend to experience a larger likelihood reduction than non-member texts. In contrast, member-prefix conditioning induces asymmetric likelihood shifts between member and non-member texts. Therefore, Eq. (9) contrasts the likelihoods induced by non-member and member prefixes to enlarge the membership signal, and normalizes this difference by the unprefixed likelihood to account for the base likelihood of the target text.

Remark. Recent work (Yi and Li, 2026) computes prefix-conditional scores at the word level and then aggregates these word-level scores into a sequence-level score. In our notation, this strategy takes the form

$$\frac{1}{L} \sum_{i=1}^L \frac{\widehat{\ell}(x_i|c_i^{nm}) - \widehat{\ell}(x_i|c_i^m)}{\widehat{\ell}(x_i|c_i^0)}. \quad (10)$$

However, we argue that the word-level log-probabilities should first be aggregated into sequence-level LL, and the relative conditional LL should then be computed at the sequence

level. This is because individual word-level log-probabilities can be close to zero, causing unstable ratios, and such per-word normalization breaks the additive structure of sequence-level log-likelihoods. We empirically validate this design choice in Tab. 3. Algorithm 1 presents the full pipeline of the proposed method.

3 Comparison with Heuristic Methods

In this section, we compare our likelihood-oriented formulation with existing black-box MIAs that rely on heuristic scoring rather than principled log-likelihood estimation.

SaMIA. SaMIA (Kaneko et al., 2025) approximates a continuation-level pseudo-probability mass. It divides each target text x into a prefix x_{prefix} and a reference suffix x_{suffix} . The LLM then generates M candidate continuations $\{x_{\text{cand}}^{(j)}\}_{j=1}^M$ conditioned on x_{prefix} , and SaMIA computes the average ROUGE-N overlap (Lin, 2004) between the continuations and the reference suffix:

$$\mathcal{S}_{\text{SaMIA}}(x) = \frac{1}{M} \sum_{j=1}^M \text{ROUGE}_N(x_{\text{cand}}^{(j)}, x_{\text{suffix}}).$$

This score estimates the expected surface overlap between sampled continuations and the reference suffix. It can be viewed as a continuation-level pseudo-probability mass, but it is not a log-likelihood estimator and does not decompose into additive word-level log-probabilities. Moreover, the continuation-level score can suffer from *distribution shift* (Yi and Li, 2026): as generation proceeds, later words depend on model-generated history rather than the original target context.

SimMIA. SimMIA (Yi and Li, 2026) constructs word-level scores and aggregates them into an overall membership score:

$$S_{\text{SimMIA}}(x) = -\frac{1}{L} \sum_{i=1}^L \frac{s(x_i | c_i^{nm})}{s(x_i | c_i^0)}. \quad (11)$$

SimMIA considers both hard and soft word-level scores. The hard score is defined as

$$s^*(x_i | c_i) = \frac{1}{M} \sum_{j=1}^M \mathbb{I}[x_i = \hat{x}_i^{(j)}], \quad (12)$$

which approximates the empirical word-level probability, but is not on a log scale. The soft score incorporates semantic similarity:

$$s(x_i | c_i) = \frac{1}{M} \sum_{j=1}^M \text{sim}(x_i, \hat{x}_i^{(j)}), \quad (13)$$

where $\text{sim}(\cdot, \cdot)$ is the cosine similarity between word embeddings. Because s is either an empirical frequency or an embedding-based similarity score, the aggregation in Eq. (11) operates on raw score ratios rather than log-probabilities. Consequently, the resulting average is a heuristic relative score, rather than a sequence-level log-likelihood estimator.

In contrast, our method estimates word-level probability mass and aggregates these word-level estimates into a sequence-level log-likelihood before applying contrastive prefix normalization. This preserves the additive property of log-likelihoods and avoids the numerical instability of directly averaging word-level scores, where the denominator in Eq. (11) may approach zero.

4 Experiments

4.1 Setup

Benchmark. We evaluate WPMIA on three widely used benchmarks: WikiMIA (Shi et al., 2024), MIMIR (Duan et al., 2024), and WikiMIA-25 (Yi and Li, 2026). For WikiMIA, we consider the 32-, 64-, and 128-token sequence-length subsets. For MIMIR, we use the 7-gram setting and report results across all seven domains: Wikipedia, GitHub, Pile-CC, PubMed Central, ArXiv, DM Mathematics, and HackerNews. For WikiMIA-25, we follow the cost-efficient evaluation setting of Yi and Li (2026) by using the length-32 split and sampling a balanced subset of 120 members and 120 non-members. See detailed dataset statistics in App. A.

Evaluation Metrics. We evaluate MIA methods using the true positive rate (TPR) and false positive rate (FPR). We sweep the decision threshold to construct the receiver operating characteristic (ROC) curve, which captures the trade-off between TPR and FPR, and report the area under the ROC curve (AUC) as the primary metric. We also report TPR at 5% FPR to measure attack performance in the low-false-positive regime. Unless otherwise specified, averaged results are reported as macro-averages over the corresponding benchmark splits and target models.

Models. For WikiMIA, we evaluate four open-source models: OPT-6.7B (Zhang et al., 2022), Pythia-6.9B (Biderman et al., 2023), LLaMA-13B (Touvron et al., 2023), and GPT-NeoX-20B (Black et al., 2022). For MIMIR, we evaluate the open-source Pythia-dedup family, with model sizes of 160M, 1.4B, 2.8B, and 6.9B. For WikiMIA-25, we evaluate the advanced open-source model Qwen3-8B-Base (Yang et al., 2025), along with proprietary LLMs such as Claude-4.5-Haiku, Gemini-2.5-Flash, and GPT-5-Chat. All open-source models are accessed via Hugging Face, while proprietary models are accessed via their official APIs. We follow Yi and Li (2026) and use the prompt template for next-word sampling on proprietary models.

MIA Baselines. We compare WPMIA against representative MIA baselines under both gray-box and black-box access settings. The gray-box baselines consist of Loss, Reference, Lowercase, Zlib, Neighbor, Min-K%, Min-K%++, and ReCaLL. Since these methods rely on access to the target model’s tokenizer and per-token logits, we include their results in the Appendix for reference. For black-box baselines, we compare against PETAL (He et al., 2025), SaMIA (Kaneko et al., 2025), and SimMIA (Yi and Li, 2026).

Implementation Details. We reproduce all baselines using their official hyperparameter configurations. For WPMIA, we split each target text into a word sequence using the standard word tokenizer (Bird et al., 2009). We set the number of Monte Carlo samples to $M = 100$ for WikiMIA and MIMIR, and reduce it to $M = 10$ for WikiMIA-25 to control the API cost. The number of prefix shots is set to $T = 7$ for WikiMIA and WikiMIA-25, and $T = 10$ for MIMIR. For each benchmark, prefix examples are held out from the

Len	Method	OPT-6.7B		Pythia-6.9B		LLaMA-13B		GPT-NeoX-20B		Average	
		AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR
32	PETAL	61.4	10.5	62.8	15.3	<u>63.2</u>	11.1	67.6	21.6	63.8	14.6
	SaMIA	55.4	5.3	55.2	8.7	59.4	10.0	60.1	10.8	57.5	8.7
	SimMIA*	<u>85.1</u>	<u>28.9</u>	<u>82.8</u>	<u>27.4</u>	60.0	8.9	<u>85.9</u>	<u>34.5</u>	<u>78.5</u>	<u>24.9</u>
	SimMIA	78.8	20.8	71.4	12.9	62.9	<u>11.6</u>	76.3	18.7	72.4	16.0
	WPMIA	89.1	38.7	84.8	27.9	75.9	21.6	89.5	52.6	84.8	35.2
64	PETAL	59.8	12.7	60.7	15.5	62.0	10.4	65.8	15.1	62.1	13.4
	SaMIA	65.1	10.4	61.2	8.8	63.1	<u>10.8</u>	67.1	11.6	64.1	10.4
	SimMIA*	<u>85.7</u>	<u>28.3</u>	<u>86.2</u>	39.4	61.8	10.0	<u>84.9</u>	<u>28.7</u>	<u>79.7</u>	<u>26.6</u>
	SimMIA	79.8	22.3	74.4	14.3	<u>67.1</u>	6.4	70.4	15.9	72.9	14.7
	WPMIA	90.2	57.8	86.5	<u>35.9</u>	74.3	23.9	86.9	45.4	84.5	40.8
128	PETAL	64.8	12.5	65.2	<u>19.2</u>	64.3	<u>13.5</u>	71.5	34.6	66.5	20.0
	SaMIA	67.2	11.5	62.4	<u>7.7</u>	<u>65.2</u>	<u>7.7</u>	67.5	16.3	65.6	10.8
	SimMIA*	<u>85.5</u>	<u>14.4</u>	<u>82.8</u>	16.3	61.7	9.6	<u>86.2</u>	<u>40.4</u>	<u>79.0</u>	<u>20.1</u>
	SimMIA	78.9	13.5	73.4	9.6	60.0	3.8	71.3	20.2	70.9	11.8
	WPMIA	87.2	49.0	84.7	39.4	67.4	15.4	87.5	48.1	81.7	38.0

Table 1: Black-box results on the WikiMIA benchmark. Complete results including gray-box baselines are provided in Tab. 15. **Bold** and underlined values indicate the best and second-best results in each column, respectively. * denotes scoring with Eq. (12) instead of Eq. (13).

Method	Pythia-6.9B		Qwen3-8B-Base		Claude-4.5-Haiku		Gemini-2.5-Flash		GPT-5-Chat	
	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR
PETAL	72.2	30.1	62.8	<u>18.6</u>	-	-	67.1	23.9	78.0	35.4
SaMIA	57.8	8.8	50.3	2.7	62.6	<u>8.0</u>	57.0	15.0	65.7	13.3
SimMIA*	88.1	37.2	63.2	2.7	<u>65.7</u>	6.2	<u>75.7</u>	<u>30.2</u>	<u>90.6</u>	73.3
SimMIA	88.8	47.8	<u>63.0</u>	6.2	60.5	6.2	74.1	28.5	83.3	37.2
WPMIA	89.0	56.6	65.9	23.0	66.3	22.1	78.2	36.5	91.0	<u>67.3</u>

Table 2: Black-box results on the WikiMIA-25 benchmark. Complete results including gray-box baselines are provided in Tab. 17. **Bold** and underlined values indicate the best and second-best results in each column, respectively. * denotes scoring with Eq. (12) instead of Eq. (13).

evaluation set for all methods. We use all-MiniLM-L6-v2 (Sentence-Transformers, 2021) as the embedding model for the semantic kernel. For the DM Mathematics split in MIMIR, we use exact matching for numerical tokens rather than semantic similarity, since numerical proximity does not necessarily imply semantic equivalence. Detailed data handling and hyperparameter settings are reported in App. B. All open-source experiments are conducted on 8 NVIDIA L40 GPUs, whereas proprietary models are evaluated via their official APIs.

4.2 Main Results

Due to space constraints, we report black-box comparisons on WikiMIA and WikiMIA-25 in the main manuscript and defer complete results to App. D.

WPMIA achieves the strongest black-box performance and is competitive with gray-box methods on open-source LLMs. As shown in Tabs. 1 and 16, WPMIA consistently outperforms existing black-box baselines on both WikiMIA and MIMIR. On WikiMIA, WPMIA improves over the best black-box baseline by 4.6 AUC points and 14.1

TPR@5%FPR points on average across the three lengths. Notably, although WPMIA observes only textual continuations, it also surpasses the strongest gray-box method in several settings, including OPT-6.7B across all WikiMIA lengths, as shown in the complete results in Tab. 15. On MIMIR, WPMIA remains the best-performing black-box method across most domain-model-size pairs, showing that its effectiveness generalizes beyond WikiMIA-style data.

WPMIA remains effective on advanced open-source and proprietary LLMs. As shown in Tab. 2, WPMIA achieves strong results on both Pythia-6.9B and the more recent Qwen3-8B-Base, reaching 89.0 and 65.9 AUC, respectively. More importantly, WPMIA also succeeds on proprietary LLMs where gray-box attacks are inapplicable. For example, WPMIA achieves AUCs of 66.3, 78.2, and 91.0 on Claude-4.5-Haiku, Gemini-2.5-Flash, and GPT-5-Chat, respectively, with corresponding TPR@5%FPR scores of 22.1, 36.5, and 67.3. The complete results, including gray-box baselines for open-source models, are reported in Tab. 17. These

results demonstrate the practical feasibility of auditing commercial LLMs under strict black-box access, without relying on the target tokenizer or per-token logits.

4.3 Analysis

We present a detailed analysis of WPMIA in this section. Unless otherwise specified, we report AUC results on WikiMIA with Pythia-6.9B.

Component-Wise Analysis. We analyze the contribution of each component of WPMIA and report the results in Tab. 3. The baseline uses the unprefix sequence-level LL estimator as a membership score $\mathcal{S}_0 = \widehat{LL}^0(x)$. The +NP rows introduce the non-member prefix and use the relative score $\mathcal{S}_{nm} = \widehat{LL}^{nm}(x)/\widehat{LL}^0(x)$. The +MP rows further introduce the member prefix and use the contrastive score $\mathcal{S}_{ctr} = (\widehat{LL}^{nm}(x) - \widehat{LL}^m(x))/\widehat{LL}^0(x)$. For both +NP and +MP, we compare sequence-level normalization with the corresponding word-level normalization described in Eq. (10).

As shown in Tab. 3, we have two observations. (1) **Prefix conditioning amplifies the membership signal beyond the unprefix likelihood.** The unprefix score $\mathcal{S}_0$ yields limited AUC scores, e.g., 55.4 at Len. 64. Introducing the non-member prefix improves the AUC from 55.4 to 77.2, and adding the member prefix further improves it to 86.5, showing that non-member and member prefixes provide complementary likelihood shifts that strengthen the membership signal. (2) **Relative conditional likelihood is more effective when computed at the sequence level.** Although word-level probabilities are useful primitives for likelihood estimation, applying relative normalization before sequence-level likelihood aggregation severely hurts performance, with AUC scores lower than those of the unprefix LL across all lengths.

Method	Len. 32	Len. 64	Len. 128
LL ($\mathcal{S}_0$)	58.8	55.4	62.0
+NP, sequence ($\mathcal{S}_{nm}$)	75.5	77.2	76.2
+NP, word	52.0	46.9	54.7
+MP, word	51.4	48.8	48.8
+MP, sequence ($\mathcal{S}_{ctr}$)	84.8	86.5	84.7

Table 3: Component-wise ablation of WPMIA.

Crossed Ablation of Estimator and Prefix Conditioning. To separate the contributions of word-

level estimation and prefix conditioning, Tab. 4 evaluates all combinations of three estimators, namely exact matching (Eq. (12)), raw embedding similarity (Eq. (13)), and the semantic kernel (Eq. (6)), with three prefix settings. Two consistent patterns emerge. First, prefix conditioning improves all three estimators: averaged over estimators and sequence lengths, non-member-prefix conditioning raises AUC from 59.1 to 73.3, while introducing member-prefix contrast further improves it to 81.8. Second, the estimator remains critical under the full contrastive setting. The semantic kernel achieves an average AUC of 85.3, outperforming raw embedding similarity and exact matching by 4.1 and 6.3 points, respectively. These results show that contrastive prefix conditioning and semantic-kernel estimation provide complementary gains.

Estimator	Prefix	Len. 32	Len. 64	Len. 128
Exact match	No prefix	59.87	58.01	65.08
	+NP	73.88	72.63	68.81
	+MP	81.00	80.58	75.50
Raw similarity	No prefix	57.93	54.86	59.86
	+NP	70.57	72.91	72.21
	+MP	79.80	82.65	81.12
Semantic kernel	No prefix	58.75	55.40	61.99
	+NP	75.47	77.19	76.16
	+MP	84.78	86.48	84.67

Table 4: Crossed ablation of word-level estimators and prefix-conditioning strategies. +NP denotes non-member-prefix conditioning, whereas +MP additionally introduces member-prefix contrast.

Impact of Continuation Ratio. We vary the continuation ratio, the proportion of target text used for membership inference, and examine its effect on attack performance. As shown in Fig. 3a, the AUC of all methods increases as the continuation ratio grows from 0.1 to 0.9. This trend is expected, since a larger continuation ratio provides a longer target segment and thus more exploitable membership evidence. WPMIA consistently outperforms SimMIA* and SimMIA across all continuation ratios, demonstrating its stronger ability to extract membership signals from black-box continuations.

Impact of Sample Size. We vary the number of Monte Carlo samples M to study the trade-off between the sampling budget and attack performance. As shown in Fig. 3b, all methods benefit from increasing the number of samples, since more samples reduce the estimation variance of the word-level probability in Eq. (5). WPMIA consistently

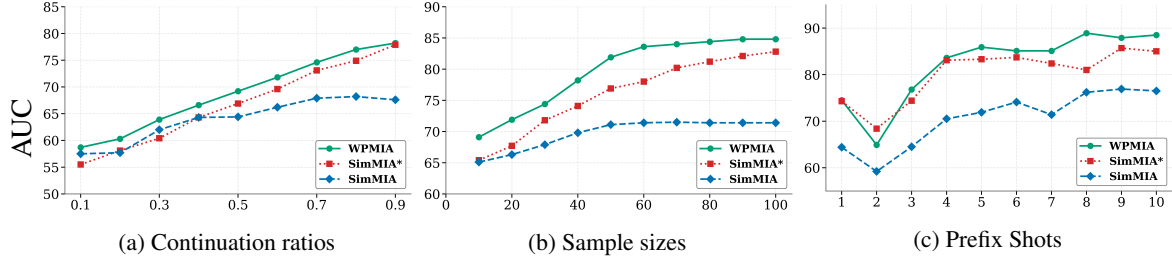


Figure 3: Attack performance under different continuation ratios, sample sizes, and prefix shots at Len. 32.

outperforms both SimMIA* and SimMIA across all sampling budgets. Moreover, WPMIA shows better scalability with respect to the sampling budget. SimMIA saturates early, around $M = 50$, whereas WPMIA continues to improve as M increases and stabilizes after $M = 70$, reaching an AUC of around 85. SimMIA* improves with more samples but remains consistently below WPMIA.

Impact of Prefix Shots. We vary the number of shots T used to construct the member and non-member prefixes. As shown in Fig. 3c, the performance is unstable when only a few shots are used, since 1- or 2-shot prefixes cannot reliably characterize the prefix-conditioned likelihood shifts. From $T = 3$ onward, increasing the number of shots generally improves attack performance, and the results stabilize around $T = 8$. In this range, WPMIA achieves the highest AUC and consistently outperforms SimMIA* and SimMIA. These results indicate that a moderate number of prefix shots is sufficient for reliable estimation, and that WPMIA benefits more from additional prefix examples than the baselines.

Embedding Model	Len. 32	Len. 64	Len. 128
Word2Vec	61.5	58.2	62.2
fastText	80.9	82.1	82.2
bge-large-en-v1.5	86.3	88.0	84.9
all-MiniLM-L6-v2	84.8	86.5	84.7
UAE-Large-V1	86.5	88.3	85.1
mxbai-embed-large-v1	86.3	88.0	84.8

Table 5: Results with different embedding models.

Impact of Embedding Model. To study WPMIA’s sensitivity to the choice of embedding model, we evaluate it with two static embedding models, Word2Vec (Mikolov et al., 2013) and fastText (Bojanowski et al., 2017), and four dense embedding models: bge-large-en-v1.5 (Xiao et al., 2024), all-MiniLM-L6-v2, UAE-Large-V1 (Li and

Li, 2024), and mxbai-embed-large-v1 (Lee et al., 2024). As shown in Tab. 5, Word2Vec yields substantially lower AUC, whereas fastText provides a clear improvement across all lengths. Modern dense embedding models perform strongly and consistently, with average AUCs ranging only from 85.3 to 86.6. This suggests that high-quality semantic matching is important for WPMIA, but its performance is not overly sensitive to the choice of a specific dense embedding model.

In terms of encoder scale, Word2Vec and fastText use 300-dimensional word embeddings. Among the dense encoders, all-MiniLM-L6-v2 produces 384-dimensional embeddings with 22.7M parameters, whereas the other three produce 1,024-dimensional embeddings with approximately 335M parameters each. Despite using roughly one-fifteenth as many parameters, all-MiniLM-L6-v2 achieves an average AUC of 85.3, only 1.3 points below the best-performing dense encoder. This result suggests that the default encoder offers a favorable trade-off between attack performance and model scale.

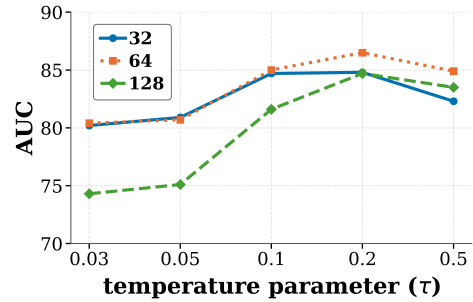


Figure 4: Attack performance under different semantic-kernel temperature parameters τ .

Sensitivity to Temperature. We analyze the sensitivity of WPMIA to the temperature parameter τ , which controls the smoothness of the semantic kernel in Eq. (6). As shown in Fig. 4, very small temperatures lead to weaker performance, especially at Len. 128. This is because an overly sharp kernel may discard useful semantic evidence from

Prefix condition	Construction	Mislabeled prefixes	AUC	TPR@5%FPR
Clean	Correct WikiMIA member/non-member prefix pools	0/14	84.78	27.89
Noisy-light	Swap one prefix from each pool	2/14	79.76	15.79
Noisy-heavy	Swap three prefixes from each pool	6/14	48.73	6.05
Fully shuffled	Randomly repartition all prefixes with a fixed score direction	8/14	48.14	5.08
GitHub OOD	Correctly labeled GitHub subset prefixes from MIMIR	N/A	56.54	8.11
Synthetic	Synthetic non-member and member-like prefixes	N/A	75.27	14.84

Table 6: Robustness to prefix-pool corruption and distribution shift at Len. 32. Each condition uses seven member and seven non-member prefixes.

generated words that are not exact matches but remain semantically close to the target word. Increasing τ improves AUC across all lengths, with the near-best performance obtained around $\tau = 0.2$. When τ further increases to 0.5, the performance slightly drops, suggesting that excessive smoothing weakens the distinction between informative and uninformative matches.

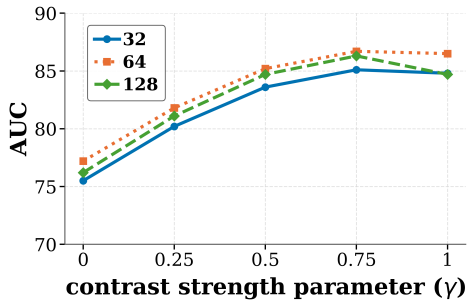


Figure 5: Results under different contrast strengths γ .

Sensitivity to Contrast Strength. Following Wang et al. (2025), we analyze a generalized form of the contrastive score in Eq. (9): $S_\gamma(x) = (\widehat{LL}_{nm}(x) - \gamma \widehat{LL}_m(x)) / \widehat{LL}_0(x)$, where γ controls the relative contribution of the member-prefix likelihood. When $\gamma = 1$, this generalized score reduces to the full contrastive score used by WPMIA. As shown in Fig. 5, setting $\gamma = 0$ removes the member-prefix contrast and gives the weakest performance. As γ increases, the AUC improves consistently across all lengths, confirming the importance of contrasting the non-member- and member-prefix likelihoods. The performance plateaus when γ is between 0.75 and 1.0. Although $\gamma = 0.75$ is slightly better for some lengths, $\gamma = 1.0$ remains highly competitive and corresponds to the full contrastive formulation. Therefore, we set $\gamma = 1.0$ as the default contrast strength in all experiments.

Robustness to Prefix Construction. We examine whether WPMIA depends on a particular con-

struction of the prefix pools by considering label corruption, domain shift, and synthetic alternatives, as shown in Tab. 6. Light label corruption reduces AUC by only 5.0 points relative to the clean setting, whereas severe corruption and complete shuffling yield a mean AUC of 48.4. This degradation indicates that WPMIA relies on a meaningful, label-aligned contrast between the two prefix pools rather than arbitrary contextual calibration. Correctly labeled but domain-mismatched GitHub prefixes from MIMIR achieve only an AUC of 56.5, further showing that label correctness alone is insufficient without domain compatibility. In contrast, synthetic prefixes constructed following ReCaLL (Xie et al., 2024) and Con-ReCall (Wang et al., 2025) achieve 75.3 AUC, remaining within 9.5 points of the clean setting. Thus, WPMIA is not tied to the exact held-out benchmark examples, and synthetic prefixes provide a practical alternative when verified prefixes are unavailable.

5 Conclusion

This work presents WPMIA, a strict black-box membership inference attack for auditing the privacy risks of LLMs. Unlike existing gray-box MIAs that require access to the target tokenizer and per-token logits, WPMIA recovers likelihood-based membership signals directly from textual continuations. It estimates word-level generation probabilities through Monte Carlo sampling with semantic kernel smoothing, aggregates these estimates into sequence-level likelihoods, and further strengthens the membership signal through contrastive prefix conditioning. Extensive experiments on WikiMIA, MIMIR, and WikiMIA-25 show that WPMIA achieves the strongest black-box performance on open-source LLMs and remains effective on recent proprietary LLMs, demonstrating the practical feasibility of strict black-box privacy auditing.

Limitations

Despite its effectiveness, WPMIA has a limitation common to sampling-based black-box MIAs for LLMs: it requires repeated sampling to recover membership signals. Unlike gray-box methods, which can obtain sequence-level likelihoods with a single forward pass, WPMIA must repeatedly query the target model and estimate word-level probabilities from sampled textual continuations. In addition, its contrastive prefix mechanism requires estimating conditional likelihoods under both member and non-member prefixes, thereby increasing the overall query budget. This cost becomes more substantial for long target texts and proprietary LLM APIs, where repeated queries incur additional computation, latency, and monetary expense. An important direction for future work is therefore to improve the sample efficiency of WPMIA, such as through adaptive sampling, variance-reduced estimation, or early-stopping strategies that reduce the number of required continuations while preserving reliable detection performance.

Another limitation is that our evaluation is restricted to English text. The effectiveness of WPMIA in multilingual settings may be affected by language-specific tokenization, semantic embeddings, and prefix construction. Future work should therefore evaluate WPMIA on multilingual benchmarks and investigate language-specific adaptations.

Ethical Considerations

This work studies membership inference attacks for privacy auditing of LLMs and is intended to help researchers and model providers assess potential training-data leakage under strict black-box access. However, membership inference is inherently dual-use and could be misused to probe for the presence of sensitive or private texts in training data; therefore, WPMIA should be applied only in authorized auditing, controlled research, and privacy-risk evaluation settings.

Acknowledgments

Jian Huang acknowledges support from The Hong Kong Polytechnic University (Research Grant P0046811). We used ChatGPT solely for language polishing and refinement of the manuscript. It was not used to generate research ideas, design experiments, analyze results, or formulate conclusions.

All final text was reviewed and verified by the authors.

References

- Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. [The Falcon series of open language models](#). *arXiv preprint arXiv:2311.16867*.
- Anthropic. 2025. [System card: Claude Haiku 4.5](#).
- Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O’Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, Usven Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 2397–2430. PMLR.
- Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. O’Reilly Media, Inc.
- Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, Usven Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. [GPT-NeoX-20B: An open-source autoregressive language model](#). In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.
- Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. [Enriching word vectors with subword information](#). *Transactions of the Association for Computational Linguistics*, 5:135–146.
- Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.
- Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. [Extracting training data from large language models](#). In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.
- Kent K. Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. [Speak, memory: An archaeology of books known to ChatGPT/GPT-4](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.
- Bowen Chen, Namgi Han, and Yusuke Miyao. 2025. A statistical and multi-perspective revisiting of the membership inference attack in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22854–22874.
- Michael Duan, Anshuman Suri, Niloofar Miresghallah, 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 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11940–11956. PMLR.
- Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. [The Pile: An 800GB dataset of diverse text for language modeling](#). *ArXiv*, abs/2101.00027.
- Google DeepMind. 2025. [Gemini 2.5 Flash - Model Card](#).
- Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. 2025. Towards Label-Only membership inference attack against pre-trained large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 1609–1628.
- Masahiro Kaneko, Youmi Ma, Yuki Wata, and Naoaki Okazaki. 2025. Sampling-based pseudo-likelihood for membership inference attacks. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8894–8907.
- Sean Lee, Aamir Shakir, Darius Koenig, and Julius Lipp. 2024. [Open source strikes bread - new fluffy embedding model](#).
- Xianming Li and Jing Li. 2024. [AoE: Angle-optimized embeddings for semantic textual similarity](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1825–1839. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.
- Justus Mattern, Fatemehsadat Miresghallah, Zhijing Jin, Bernhard Schölkopf, 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.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D Griffin. 2023. Use of LLMs for illicit purposes: Threats, prevention measures, and vulnerabilities. *arXiv preprint arXiv:2308.12833*.
- OpenAI. 2025. [GPT-5 system card](#).
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.
- Christian Robert and George Casella. 2004. *Monte Carlo Statistical Methods*, 2nd edition. Springer Texts in Statistics. Springer-Verlag, New York.
- Sentence-Transformers. 2021. [all-MiniLM-L6-v2](#). Hugging Face model card. Accessed 2026-08-30.
- 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 *The Twelfth International Conference on Learning Representations*.
- 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.
- Kunio Takezawa. 2005. *Introduction to nonparametric regression*. John Wiley & Sons.
- Xinyu Tang, Richard Shin, Huseyin Inan, Andre Manoel, Fatemehsadat Miresghallah, Zinan Lin, Sivakanth Gopi, Janardhan (Jana) Kulkarni, and Robert Sim. 2024. Privacy-preserving in-context learning with differentially private few-shot generation. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. [LLaMA: Open and efficient foundation language models](#). *ArXiv*, abs/2302.13971.

Cheng Wang, Yiwei Wang, Bryan Hooi, Yujun Cai, Nanyun Peng, and Kai-Wei Chang. 2025. ConReCall: Detecting pre-training data in LLMs via contrastive decoding. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1013–1026.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muenighoff, Defu Lian, and Jian-Yun Nie. 2024. C-Pack: Packed resources for general Chinese embeddings. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, pages 641–649.

Roy Xie, Junlin Wang, Ruomin Huang, Minxing Zhang, Rong Ge, Jian Pei, Neil Zhenqiang Gong, and Bhuwan Dhingra. 2024. ReCaLL: Membership inference via relative conditional log-likelihoods. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8671–8689.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

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 *The Thirteenth International Conference on Learning Representations*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [OPT: Open pre-trained transformer language models](#). *ArXiv*, abs/2205.01068.

A Dataset Statistics

Detailed statistics of WikiMIA, MIMIR, and WikiMIA-25 are summarized in Tabs. 7–9.

Statistic	Text Length		
	32	64	128
Total examples	776	542	250
Non-member Ratio	50.1%	47.6%	44.4%
Member Ratio	49.9%	52.4%	55.6%

Table 7: WikiMIA dataset statistics.

Subset	Samples
wikipedia_(en)	2000
github	536
pile_cc	2000
pubmed_central	982
arxiv	1000
dm_mathematics	178
hackernews	1292

Table 8: MIMIR dataset statistics under the 7-gram setting.

Text Length	Members	Non-members	Total Samples
32	120	120	240

Table 9: WikiMIA-25 subset used for cost-efficient evaluation.

B Additional Implementation Details

Benchmark Prefix Construction. WPMIA uses both member-prefix and non-member-prefix conditioning. For each benchmark, we reserve an equal-sized pool of member and non-member examples for prefix construction. The reserved prefix examples are removed from the evaluation set for all methods, including baselines, to ensure a fair comparison. For WikiMIA and WikiMIA-25, we use $T = 7$ prefix shots. For MIMIR, we use $T = 10$ prefix shots.

Hyperparameter Settings. The hyperparameter settings used for WPMIA across different benchmarks are summarized in Tab. 10.

Prompt Template for Proprietary LLMs. For proprietary LLMs, we use the following system prompt based on SimMIA (Yi and Li, 2026). The prompt constrains the model to generate a single next word or punctuation mark for a given [PREFIX].

Setting	WikiMIA	MIMIR	WikiMIA-25
Monte Carlo samples M	100	100	10
Prefix shots T	7	10	7
Smoothing constant ϵ	10^{-8}	10^{-8}	10^{-8}
Kernel temperature τ	0.2	0.2	0.1

Table 10: Hyperparameter settings used for WPMIA.

Prompt
System: Return ONLY the single next token from the text. It can be punctuation or one whole word. No spaces, no quotes, no extra text.
User: Text so far: [PREFIX]

C Practical Prefix Construction

We next discuss how to instantiate the benchmark prefix pools in realistic black-box auditing settings. Depending on the auditor’s access to training-data information, prefix construction can range from verified member/non-member examples to high-confidence or fully synthetic proxies.

Authorized audits with verified prefixes. In authorized audits, a model provider or data curator may have access to examples whose membership status is known. Member prefixes can be drawn directly from the training corpus or from pre-inserted canary documents whose inclusion in training is verifiable (Carlini et al., 2019). Non-member prefixes can be constructed from newly created private documents that were never exposed to the model provider. This setting most closely matches the benchmark construction, where the two prefix pools have reliable membership labels.

External audits with high-confidence proxies. When direct access to the training set is unavailable, an external auditor can instead construct prefix pools using high-confidence membership proxies. For a fixed model snapshot with a known training cutoff, newly created private documents or public documents released after the cutoff can serve as non-member candidates. Member-side proxies can be drawn from pre-cutoff public corpora whose likely inclusion is supported by model documentation or established pretraining practice. For example, Wikipedia, GitHub, and arXiv are standard components of the Pile (Gao et al., 2020), which was used to train the Pythia model family (Biderman et al., 2023). Such examples do not provide verified membership labels, but they offer a practical approximation when the model’s training sources and cutoff are partially known.

External audits with synthetic proxies. When neither verified examples nor sufficiently reliable public proxies are available, both prefix pools can be constructed synthetically. ReCaLL (Xie et al., 2024) uses GPT-4o to generate passages that match the length, domain, and style of the target text, which can serve as approximate non-member prefixes. Con-ReCall (Wang et al., 2025) further introduces a procedure for constructing member-like prefixes without access to the original training set. It first uses GPT-4o to generate descriptions of significant events, partially truncates these descriptions, and then prompts the target model to complete them. The resulting target-model continuations are used as member-like prefixes because they reflect knowledge and language patterns internalized by the target model. Using these synthetic prefixes also achieves appreciable attack performance, as reported in Tab. 6.

D Additional Experiments

Complete Benchmark Results. Tabs. 15–17 report the complete comparisons on WikiMIA, MIMIR, and WikiMIA-25, respectively. These results complement the black-box comparisons in the main manuscript with the corresponding gray-box baselines. Across the three benchmarks, WPMIA is the strongest black-box method in most settings and remains competitive with gray-box.

Budget-Matched Comparison. Beyond comparing sampling-based black-box attacks with the same sample size, we further compare them under approximately matched budgets of generated tokens. Generated output tokens provide a common cost measure that accounts for the long continuations used by SaMIA and the short first-word outputs used by WPMIA. We compare SaMIA, SimMIA, and WPMIA at low, medium, and high budgets, using SimMIA as the cost reference with $M = 30$, $M = 60$, and $M = 100$, respectively. At each level, we select the integer sample sizes for SaMIA and WPMIA whose generated-token counts are closest to the corresponding SimMIA budget. As shown in Tab. 11, WPMIA achieves the best performance across all budget levels. Averaged across the three budgets, it achieves an AUC of 83.1 and a TPR@5%FPR of 27.9, exceeding SimMIA by 12.4 and 15.4 points, respectively. These results show that WPMIA remains the strongest method when the attacks are compared under approximately matched generated-token budgets.

Budget	Method	M	Avg. generated tokens	AUC	TPR@5%FPR
Low	SaMIA	10	6,317.68	51.30	5.53
	SimMIA	30	6,313.98	69.99	11.58
	WPMIA	20	6,313.86	81.25	28.68
Medium	SaMIA	20	12,499.20	51.42	6.58
	SimMIA	60	12,627.62	70.88	13.16
	WPMIA	40	12,628.24	83.40	27.89
High	SaMIA	34	21,106.13	50.69	8.16
	SimMIA	100	21,045.00	71.24	12.63
	WPMIA	67	21,152.33	84.59	27.11

Table 11: Generated-token-budget-matched comparison of sampling-based black-box attacks on WikiMIA with Pythia-6.9B at length 32.

Proprietary API Statistics. We report the request count, runtime, and monetary cost of the proprietary LLM experiments in Tab. 12. For a record with L scored word positions, WPMIA requires approximately $Q = 3ML$ requests because each position is evaluated under the unprefixed, member-prefix, and non-member-prefix conditions. WikiMIA-25 contains 7,898 scored positions in total, resulting in an expected retry-free total of 236,940 requests when $M = 10$. The observed HTTP-response counts in Tab. 12 are slightly higher than this estimate because interrupted or failed requests may trigger retries. Provider-specific throughput and pricing lead to different practical trade-offs: Claude-4.5-Haiku achieves the lowest mean runtime per record, whereas Gemini-2.5-Flash incurs the lowest total monetary cost.

Model	HTTP responses	Runtime/record (s)	Monetary cost (USD)
GPT-5-Chat	241,492	216.43	\$266.8
Gemini-2.5-Flash	238,649	119.81	\$78.0
Claude-Haiku-4.5	239,293	86.25	\$240.7

Table 12: API statistics for WPMIA on WikiMIA-25 with $M = 10$. HTTP response counts include retries; runtime is averaged per record at a concurrency level of 32, and monetary cost is reported in U.S. dollars.

Stability across Sample Sizes. We repeat the evaluation of Pythia-6.9B on WikiMIA using five sampling seeds to assess the stability of WPMIA across different sample sizes. As shown in Tab. 13, WPMIA exhibits moderate variability even at $M = 10$, with AUC standard deviations of 1.10, 2.77, and 1.44 for lengths 32, 64, and 128, respectively. Increasing M consistently improves the mean AUC across all three lengths, while the marginal gains di-

minish around $M = 70$ –80, particularly for lengths 32 and 64. These results reveal a clear trade-off between sampling cost and attack performance.

M	Len. 32	Len. 64	Len. 128
10	68.32 $\pm$ 1.10	71.74 $\pm$ 2.77	67.63 $\pm$ 1.44
20	71.84 $\pm$ 0.96	75.49 $\pm$ 1.82	72.12 $\pm$ 2.30
30	75.11 $\pm$ 0.83	79.00 $\pm$ 0.84	77.80 $\pm$ 1.75
40	78.32 $\pm$ 0.49	80.74 $\pm$ 0.72	80.58 $\pm$ 2.53
50	80.90 $\pm$ 0.64	83.04 $\pm$ 0.70	83.30 $\pm$ 2.12
60	82.31 $\pm$ 0.83	84.57 $\pm$ 0.72	84.66 $\pm$ 1.47
70	83.17 $\pm$ 0.49	85.77 $\pm$ 0.35	86.26 $\pm$ 1.51
80	83.58 $\pm$ 0.55	86.10 $\pm$ 0.22	87.33 $\pm$ 1.80
90	84.13 $\pm$ 0.41	86.41 $\pm$ 0.22	87.95 $\pm$ 1.88
100	84.48 $\pm$ 0.29	86.67 $\pm$ 0.24	87.99 $\pm$ 1.88

Table 13: Mean AUC ($\pm$ standard deviation) over five sampling seeds for WPMIA on WikiMIA with Pythia-6.9B, across sample sizes M and sequence lengths.

Fine-Grained Temperature Sensitivity. We further examine temperature sensitivity by sweeping τ from 0.10 to 0.50 in increments of 0.05. As shown in Tab. 14, the AUC averaged across sequence lengths ranges only from 83.6 to 85.4 over the full sweep. The variation is even smaller within $\tau \in [0.15, 0.30]$, where the average AUC remains between 84.6 and 85.4. The default value $\tau = 0.2$ achieves an average AUC of 85.3, within 0.1 points of the best tested setting. These results indicate that WPMIA is stable over a broad temperature range on WikiMIA and support using $\tau = 0.2$ uniformly for this benchmark. We apply the same value across the MIMIR domains without additional tuning.

τ	Len. 32	Len. 64	Len. 128
0.10	84.69	85.00	81.56
0.15	85.30	86.52	84.26
0.20	84.78	86.48	84.67
0.25	84.16	86.17	84.74
0.30	83.63	85.84	84.44
0.35	83.21	85.58	84.17
0.40	82.87	85.29	83.90
0.45	82.60	85.10	83.69
0.50	82.33	84.93	83.50

Table 14: AUC of WPMIA on WikiMIA with Pythia-6.9B under different semantic-kernel temperatures.

E Pseudocode

We present the pipeline of WPMIA in Algorithm 1.

Algorithm 1 Word-level Probability Membership Inference Attack

```

1: Input: target text  $x = (x_1, x_2, \dots, x_L)$ , language model  $f_\theta$ , number of samples  $M$ , non-member
   prefix pool  $\mathcal{P}_{\text{nm}}$ , member prefix pool  $\mathcal{P}_{\text{m}}$ , number of prefix shots  $T$ , kernel function  $K_\tau$ , smoothing
   constant  $\epsilon$ , decision threshold  $\kappa$ 
2: Output: membership score  $\mathcal{S}(\theta, x)$  and decision on whether  $x$  is included in the training data of  $f_\theta$ 
   (1 or 0)
3:  $P_0 \leftarrow \langle \rangle$ 
4:  $P_{\text{nm}} \leftarrow p_1^{\text{nm}} \oplus p_2^{\text{nm}} \oplus \dots \oplus p_T^{\text{nm}}$ , where  $p_t^{\text{nm}} \in \mathcal{P}_{\text{nm}}$ 
5:  $P_{\text{m}} \leftarrow p_1^{\text{m}} \oplus p_2^{\text{m}} \oplus \dots \oplus p_T^{\text{m}}$ , where  $p_t^{\text{m}} \in \mathcal{P}_{\text{m}}$ 
6: for  $i = 1$  to  $L$  do
7:   for  $q \in \{0, \text{nm}, \text{m}\}$  do
8:      $c_i^q \leftarrow P_q \oplus (x_1, x_2, \dots, x_{i-1})$ 
9:     for  $j = 1$  to  $M$  do
10:       $\hat{x}_{i,j}^{c_i^q} \leftarrow \text{FIRSTWORD}(f_\theta(c_i^q))$   $\triangleright$  sample the first generated word from  $f_\theta$  using  $c_i^q$  as the
      prompt
11:    end for
12:     $\hat{p}(x_i | c_i^q) \leftarrow \frac{1}{M} \sum_{j=1}^M K_\tau(x_i, \hat{x}_{i,j}^{c_i^q})$   $\triangleright$  estimate word-level probability mass
13:     $\hat{\ell}(x_i | c_i^q) \leftarrow \log(\epsilon + \hat{p}(x_i | c_i^q))$ 
14:  end for
15: end for
16: for  $q \in \{0, \text{nm}, \text{m}\}$  do
17:    $\widehat{LL}^q(x) \leftarrow \frac{1}{L} \sum_{i=1}^L \hat{\ell}(x_i | c_i^q)$   $\triangleright$  aggregate word-level log-probabilities into sequence-level
   log-likelihood
18: end for
19:  $\mathcal{S}(\theta, x) \leftarrow \frac{\widehat{LL}^{\text{nm}}(x) - \widehat{LL}^{\text{m}}(x)}{\widehat{LL}^0(x)}$   $\triangleright$  contrast prefix-conditioned log-likelihoods
20: if  $\mathcal{S}(\theta, x) > \kappa$  then
21:   return  $\mathcal{S}(\theta, x), 1$ 
22: else
23:   return  $\mathcal{S}(\theta, x), 0$ 
24: end if

```

Len	Method	OPT-6.7B		Pythia-6.9B		LLaMA-13B		GPT-NeoX-20B		Average	
		AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR
32	Gray-Box										
	Loss	60.3	10.5	63.5	13.7	67.4	13.7	68.7	19.0	65.0	14.2
	Reference	64.0	7.9	63.8	6.3	58.2	4.7	67.9	16.6	63.5	8.9
	Lowercase	57.7	8.9	58.9	11.8	65.3	13.4	61.6	14.5	60.9	12.2
	Zlib	61.3	12.6	64.1	16.8	67.7	11.8	69.0	20.8	65.5	15.5
	Neighbor	64.5	12.1	65.7	<u>17.1</u>	66.2	11.8	70.4	22.9	66.7	16.0
	Min-K%	61.1	<u>16.3</u>	69.5	13.2	66.5	18.4	71.5	<u>28.2</u>	67.2	<u>19.0</u>
	Min-K%++	<u>65.1</u>	8.9	<u>70.0</u>	13.9	<u>84.5</u>	<u>32.4</u>	<u>74.2</u>	19.0	<u>73.5</u>	18.6
	ReCaLL	77.6	25.5	86.1	28.9	89.1	41.8	87.9	34.7	85.2	32.7
	Black-Box										
	PETAL	61.4	10.5	62.8	15.3	<u>63.2</u>	11.1	67.6	21.6	63.8	14.6
	SaMIA	55.4	5.3	55.2	8.7	59.4	10.0	60.1	10.8	57.5	8.7
	SimMIA*	<u>85.1</u>	<u>28.9</u>	<u>82.8</u>	<u>27.4</u>	60.0	8.9	<u>85.9</u>	<u>34.5</u>	<u>78.5</u>	<u>24.9</u>
	SimMIA	78.8	20.8	71.4	12.9	62.9	<u>11.6</u>	76.3	18.7	72.4	16.0
	WPMIA	89.1	38.7	84.8	27.9	75.9	21.6	89.5	52.6	84.8	35.2
64	Gray-Box										
	Loss	56.5	13.9	60.0	13.1	63.4	12.0	65.8	12.0	61.4	12.8
	Reference	62.5	5.2	63.1	12.4	63.8	4.0	66.2	15.9	63.9	9.4
	Lowercase	55.6	11.2	58.7	8.4	61.8	8.4	60.4	12.4	59.1	10.1
	Zlib	59.2	12.0	62.0	15.9	65.2	12.7	67.5	17.1	63.5	14.4
	Neighbor	60.7	12.0	63.6	9.6	64.3	9.6	67.9	12.4	64.1	10.9
	Min-K%	59.8	16.7	64.0	20.3	65.9	18.3	71.7	18.7	65.4	18.5
	Min-K%++	<u>64.9</u>	<u>17.1</u>	<u>71.7</u>	<u>21.9</u>	<u>84.6</u>	<u>29.5</u>	<u>77.2</u>	<u>25.5</u>	<u>74.6</u>	<u>23.5</u>
	ReCaLL	76.3	23.9	89.8	49.8	89.5	48.6	87.1	33.5	85.7	39.0
	Black-Box										
	PETAL	59.8	12.7	60.7	15.5	62.0	10.4	65.8	15.1	62.1	13.4
	SaMIA	65.1	10.4	61.2	8.8	63.1	<u>10.8</u>	67.1	11.6	64.1	10.4
	SimMIA*	<u>85.7</u>	<u>28.3</u>	<u>86.2</u>	39.4	61.8	10.0	<u>84.9</u>	<u>28.7</u>	<u>79.7</u>	<u>26.6</u>
	SimMIA	79.8	22.3	74.4	14.3	<u>67.1</u>	6.4	70.4	15.9	72.9	14.7
	WPMIA	90.2	57.8	86.5	<u>35.9</u>	74.3	23.9	86.9	45.4	84.5	40.8
128	Gray-Box										
	Loss	62.8	13.5	65.2	15.4	68.9	21.2	71.5	24.0	67.1	18.5
	Reference	64.3	12.5	63.8	15.4	63.8	13.5	69.7	23.1	65.4	16.1
	Lowercase	61.2	6.7	64.8	3.8	67.6	20.2	68.0	8.7	65.4	9.9
	Zlib	63.6	16.3	66.9	18.3	70.2	20.2	72.3	25.0	68.3	20.0
	Neighbor	64.0	16.3	69.3	13.5	70.3	22.1	73.6	22.1	69.3	18.5
	Min-K%	67.6	15.4	69.5	<u>20.2</u>	73.3	19.2	75.9	24.0	71.6	19.7
	Min-K%++	71.4	25.0	<u>72.2</u>	19.2	<u>86.4</u>	<u>38.5</u>	<u>76.5</u>	<u>28.9</u>	<u>76.6</u>	<u>27.9</u>
	ReCaLL	<u>71.0</u>	<u>17.3</u>	87.4	41.3	91.3	48.1	88.0	39.4	84.4	36.5
	Black-Box										
	PETAL	64.8	12.5	65.2	<u>19.2</u>	64.3	<u>13.5</u>	71.5	34.6	66.5	20.0
	SaMIA	67.2	11.5	62.4	7.7	<u>65.2</u>	7.7	67.5	16.3	65.6	10.8
	SimMIA*	<u>85.5</u>	14.4	82.8	16.3	61.7	9.6	<u>86.2</u>	<u>40.4</u>	<u>79.0</u>	<u>20.1</u>
	SimMIA	78.9	13.5	73.4	9.6	60.0	3.8	71.3	20.2	70.9	11.8
	WPMIA	87.2	49.0	84.7	39.4	67.4	15.4	87.5	48.1	81.7	38.0

Table 15: Results on the WikiMIA benchmark. **Bold** and underlined values indicate the best and second-best results in each column, respectively. * denotes scoring with Eq. (12) instead of Eq. (13).

Method	Wikipedia				Github				Pile CC				PubMed Central			
	160M	1.4B	2.8B	6.9B	160M	1.4B	2.8B	6.9B	160M	1.4B	2.8B	6.9B	160M	1.4B	2.8B	6.9B
Gray-Box																
Loss	62.3	64.9	<u>66.2</u>	<u>66.4</u>	83.6	86.4	88.4	87.9	53.7	55.2	<u>54.9</u>	57.0	<u>79.1</u>	78.0	77.8	77.6
Reference	51.7	62.9	67.6	67.8	67.7	73.3	73.7	65.7	52.0	58.6	58.2	63.7	68.0	66.8	63.0	60.5
Lowercase	58.6	62.1	63.5	64.1	76.5	81.2	83.5	83.3	<u>53.6</u>	54.3	54.2	55.7	75.1	75.2	75.4	75.2
Zlib	56.5	60.9	63.0	63.0	87.9	89.8	91.2	90.9	51.9	53.9	53.7	55.6	77.8	76.9	<u>76.7</u>	<u>76.7</u>
Neighbor	57.9	60.9	61.8	61.7	<u>86.6</u>	<u>88.8</u>	<u>90.5</u>	<u>89.4</u>	52.7	55.2	54.8	56.8	76.8	74.4	74.0	74.4
Min-K%	59.9	<u>63.8</u>	65.6	65.8	82.9	86.5	88.4	88.3	52.9	<u>55.3</u>	54.7	56.9	77.3	<u>77.4</u>	77.8	77.6
Min-K%++	55.2	62.1	64.2	63.8	72.5	81.6	85.0	84.9	51.6	54.9	54.1	56.3	62.3	63.0	66.0	67.5
ReCaLL	55.9	63.4	65.1	65.2	83.0	86.0	87.9	87.4	51.3	54.7	53.7	<u>57.4</u>	79.5	75.2	76.0	73.7
Black-Box																
PETAL	60.1	62.4	63.7	63.6	68.2	70.7	72.1	71.3	52.8	<u>53.9</u>	54.0	55.4	<u>72.1</u>	<u>72.7</u>	73.3	<u>73.4</u>
SaMIA	49.1	51.3	51.9	52.1	53.6	60.3	61.1	62.7	50.1	48.9	48.3	49.9	53.1	51.2	54.8	52.6
SimMIA*	57.3	62.6	61.1	62.8	79.8	78.3	79.6	76.5	47.8	51.3	53.6	53.3	48.2	52.4	63.7	58.2
SimMIA	61.8	<u>64.3</u>	<u>65.1</u>	<u>65.0</u>	83.5	85.4	88.9	85.1	52.8	55.1	<u>55.3</u>	57.6	71.3	71.7	<u>75.1</u>	65.1
WPMIA	62.4	64.8	65.7	65.5	87.4	88.3	90.6	88.2	<u>52.6</u>	55.1	55.5	<u>56.9</u>	74.5	73.9	77.9	74.2
Gray-Box																
Loss	74.4	77.6	78.0	78.4	<u>94.6</u>	<u>93.3</u>	92.8	<u>92.9</u>	<u>57.9</u>	<u>59.3</u>	60.5	60.7	72.2	73.5	74.1	74.4
Reference	57.6	71.6	71.5	71.9	61.2	48.3	44.9	44.9	51.5	53.7	57.1	57.4	58.5	62.2	62.3	61.7
Lowercase	69.2	74.0	75.5	75.8	80.7	75.9	78.2	72.6	57.7	59.0	59.9	<u>60.2</u>	67.3	68.8	70.0	69.6
Zlib	<u>74.5</u>	<u>77.2</u>	<u>77.7</u>	<u>78.0</u>	80.9	81.0	81.2	81.0	57.5	58.4	59.3	59.4	69.6	71.2	71.8	72.1
Neighbor	70.8	74.9	75.5	76.1	79.7	71.4	73.2	71.5	56.7	57.5	58.4	58.2	68.7	69.0	69.7	69.7
Min-K%	68.6	74.2	65.3	75.6	93.1	<u>93.3</u>	<u>92.9</u>	93.0	55.0	56.9	57.9	58.6	70.0	72.5	71.8	<u>73.7</u>
Min-K%++	52.4	64.1	65.5	65.0	76.9	76.0	70.2	73.5	53.7	56.0	57.4	58.9	60.7	65.4	66.1	67.1
ReCaLL	76.5	76.5	<u>77.9</u>	76.5	96.7	94.4	94.2	91.8	59.1	59.5	<u>60.1</u>	60.7	<u>71.7</u>	<u>72.8</u>	<u>73.6</u>	73.2
Black-Box																
PETAL	68.4	70.9	<u>71.8</u>	72.4	86.5	86.9	86.5	86.8	57.1	58.0	58.5	58.4	66.5	67.9	68.6	68.8
SaMIA	43.6	44.6	45.4	45.9	82.3	85.3	79.9	73.4	49.3	49.6	49.5	50.1	54.4	55.9	55.8	55.2
SimMIA*	57.1	57.9	54.3	61.5	64.4	65.0	67.5	62.4	51.9	54.8	54.4	54.8	58.1	60.3	62.0	61.4
SimMIA	<u>71.7</u>	<u>72.6</u>	66.8	<u>72.5</u>	<u>86.7</u>	<u>89.6</u>	<u>89.3</u>	<u>90.3</u>	53.1	53.3	52.8	53.0	<u>68.7</u>	<u>70.3</u>	<u>70.5</u>	<u>69.8</u>
WPMIA	74.4	75.6	73.6	75.3	91.6	94.9	94.3	95.3	<u>54.7</u>	<u>55.6</u>	<u>56.1</u>	<u>56.0</u>	71.1	72.6	73.4	73.1

Table 16: AUC results on the MIMIR benchmark. **Bold** and underlined values indicate the best and second-best results in each column, respectively. * denotes scoring with Eq. (12) instead of Eq. (13).

Method	Pythia-6.9B		Qwen3-8B-Base		Claude-4.5-Haiku		Gemini-2.5-Flash		GPT-5-Chat	
	AUC	TPR@5FPR	AUC	TPR@5FPR	AUC	TPR@5FPR	AUC	TPR@5FPR	AUC	TPR@5FPR
Gray-Box										
Loss	73.0	36.3	59.1	11.5	-	-	-	-	-	-
Reference	65.2	8.9	53.9	3.5	-	-	-	-	-	-
Lowercase	70.7	23.0	59.4	19.5	-	-	-	-	-	-
Zlib	72.2	31.9	57.5	8.9	-	-	-	-	-	-
Min-K%	<u>77.3</u>	38.9	56.8	14.2	-	-	-	-	-	-
Min-K%++	72.7	<u>41.6</u>	<u>62.7</u>	<u>20.4</u>	-	-	-	-	-	-
ReCaLL	96.1	78.8	91.0	43.4	-	-	-	-	-	-
Black-Box										
PETAL	72.2	30.1	62.8	<u>18.6</u>	-	-	67.1	23.9	78.0	35.4
SaMIA	57.8	8.8	50.3	2.7	62.6	<u>8.0</u>	57.0	15.0	65.7	13.3
SimMIA*	88.1	37.2	63.2	2.7	<u>65.7</u>	6.2	<u>75.7</u>	<u>30.2</u>	<u>90.6</u>	73.3
SimMIA	88.8	<u>47.8</u>	<u>63.0</u>	6.2	60.5	6.2	74.1	28.5	83.3	37.2
WPMIA	89.0	56.6	65.9	23.0	66.3	22.1	78.2	36.5	91.0	<u>67.3</u>

Table 17: Results on the WikiMIA-25 benchmark. **Bold** and underlined values indicate the best and second-best results in each column, respectively. * denotes scoring with Eq. (12) instead of Eq. (13).