

PDR: A Plug-and-Play Positional Decay Framework for LLM Pre-training Data Detection

Jinhan Liu¹, Yibo Yang^{2,3}, Ruiying Lu⁴, Piotr Piękos²,
Yimeng Chen², Peng Wang¹, Dandan Guo^{1,2}

¹School of Artificial Intelligence, Jilin University

²King Abdullah University of Science and Technology

³University of Oxford

⁴School of Cyber Engineering, Xidian University

jinhan23@mails.jlu.edu.cn, guodandan@jlu.edu.cn

Abstract

Detecting pre-training data in Large Language Models (LLMs) is crucial for auditing data privacy and copyright compliance, yet it remains challenging in black-box, zero-shot settings where computational resources and training data are scarce. While existing likelihood-based methods have shown promise, they typically aggregate token-level scores using uniform weights, thereby neglecting the inherent information-theoretic dynamics of autoregressive generation. In this paper, we hypothesize and empirically validate that memorization signals are heavily skewed towards the high-entropy initial tokens, where model uncertainty is highest, and decay as context accumulates. To leverage this linguistic property, we introduce Positional Decay Reweighting (PDR), a training-free and plug-and-play framework. PDR explicitly reweights token-level scores to amplify distinct signals from early positions while suppressing noise from later ones. Extensive experiments show that PDR acts as a robust prior and can usually enhance a wide range of advanced methods across multiple benchmarks.

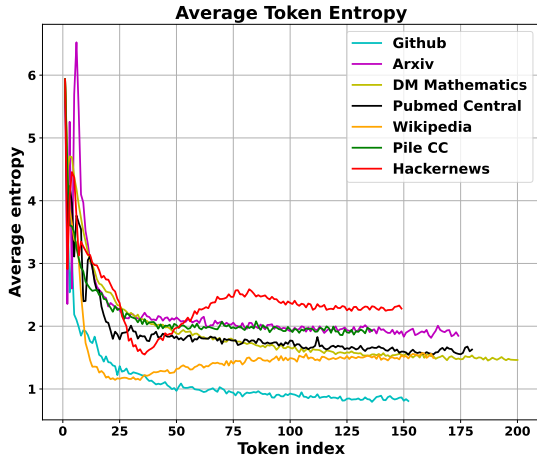
1 Introduction

As Large Language Models (LLMs) are trained on vast and diverse corpora from the internet (Achiam et al., 2023; Touvron et al., 2023b), there exists a non-negligible risk that sensitive or personally identifiable information may be memorized and unintentionally exposed through model outputs (Grynbau and Mac, 2023; Mozes et al., 2023). Pre-training data detection as a special case of Membership Inference Attack (MIA) aims to determine whether a sample was part of a model’s training set (Hu et al., 2022b; Wu and Cao, 2025), which has become increasingly critical in scenarios such as training data auditing, copyright infringement detection, and test set contamination analysis (Bertran et al., 2023; Zhang et al., 2025b), where identifying

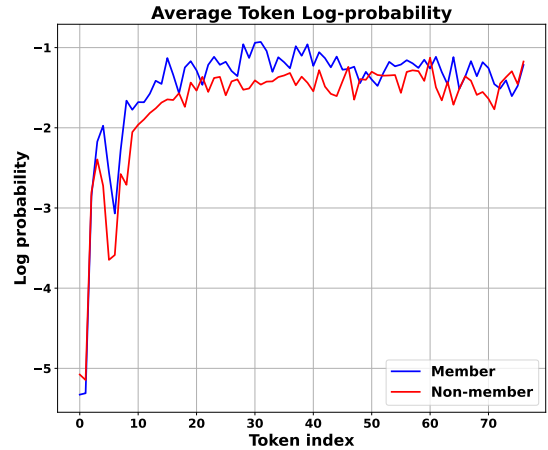
memorized content is essential for ensuring data integrity and compliance.

Existing detection methods for LLMs can be broadly categorized into likelihood-based and non-likelihood-based approaches. Among dominant likelihood-based methods, Loss (Yeom et al., 2018) averages log-likelihoods across all tokens in the test sequence to serve as the detection score, while Min-k% (Shi et al., 2024) and Min-k%++ (Zhang et al., 2025b) select some of the tokens with the lowest-probability from a sequence to compute its detection score. Methods like ReCaLL (Xie et al., 2024), and Ref (Carlini et al., 2021) introduce a reference point to calibrate likelihood-based scores, either prefixing target data points with non-member context or using a smaller auxiliary LLM. FSD (Zhang et al., 2025a) fine-tunes the target LLM on some non-member samples before computing the likelihood-based score (Zhang et al., 2025a). While varied in their specific strategies, these methods share a fundamental, unaddressed limitation: they typically aggregate token-level scores using uniform weights. Whether aggregating scores from all tokens or a selected subset in the sequence, they assign equal weight to each included token’s contribution to the final detection score, failing to explicitly account for the positional decay of memorization signals.

Our work is motivated by a key insight from information theory: conditioning on more information cannot increase entropy (Shannon, 1948). In autoregressive models, this implies that as more context accumulates, the model’s predictive uncertainty for the same token should not increase. This motivated us to hypothesize an empirical trend: in typical language generation, token-level entropy usually tends to decrease as a sequence progresses. We empirically investigate this hypothesis in Figure 1 (a). The results reveal a dominant, albeit sometimes noisy, downward trend across diverse datasets. While corpora with heterogeneous struc-



(a) Entropy trends



(b) Log-probability trends

Figure 1: Visualization of (a) token-level entropy on subsets of the challenging Mimir dataset and (b) the average token-level log-probability for members and non-members on WikiMIA dataset for LLaMA-13B model.

tures like ArXiv and HackerNews show volatility, all datasets share a crucial characteristic: a high-entropy initial region that drops sharply. Consequently, an unusually confident prediction (high probability) for an early, high-entropy token is far more surprising—and thus more indicative of memorization—than comparable confidence later in the sequence. This is because in later positions, the abundance of context makes predictions easier for both member and non-member samples, thus shrinking the discriminative gap between them.

This leads to our core hypothesis: *the memorization signal is not uniformly distributed, but is concentrated in the initial stages of a sequence, with its discriminative power generally decaying with token position*. However, existing likelihood-based methods, by utilizing uniform score aggregation, dilute this skewed and powerful signal with less informative signals from later positions.

Capitalizing on this key observation, we introduce Positional Decay Reweighting (PDR), a simple, effective, and “plug-and-play” method designed to align the scoring process with this positional signal decay. By applying monotonic decay functions (e.g., linear, exponential, polynomial), PDR systematically amplifies the high-value signals from early tokens while attenuating potential noise from later ones, thereby focusing the inference on the most informative parts of the sequence. Its key advantage is versatility: PDR can be seamlessly integrated into existing likelihood-based scoring functions. Extensive experiments validate that this straightforward modification yields substantial and consistent performance gains, improving upon advanced Min- $k\%$ ++ by up to 4.7 AUROC points

on the WikiMIA benchmark of 128 length.

Our main contributions can be summarized as follows: (1) We are the first to systematically demonstrate and analyze the positional decay of memorization signals from the view of token-level entropy, exposing the “uniform score aggregation” limitation inherent in prior methods. (2) We propose Positional Decay Reweighting (PDR), a training-free and plug-and-play framework that reweights token scores to amplify early signals. Crucially, PDR incurs zero additional computational overhead, while requiring no access to training data. (3) Our results across diverse LLMs and benchmarks establish PDR as an effective plug-and-play method, yielding notable performance gains especially for Min- $k\%$ ++.

2 Related Work

2.1 Membership Inference Attacks.

Membership Inference Attacks (MIA) have been a key topic in security and privacy (Shokri et al., 2017; Yeom et al., 2018), aiming to determine if a data point was part of a model’s training dataset. Research in both vision (Dubiński et al., 2024) and language (Watson et al., 2022; Mattern et al., 2023) has led to advanced attack techniques, including LiRA (Carlini et al., 2022) using shadow models to estimate logit distributions for likelihood-ratio tests, RMIA (Zarifzadeh et al., 2023) that constructs robust pairwise likelihood-ratio tests using a population of reference models, and RaMIA (Tao and Shokri, 2025), which extends the scope by testing if the model was trained on any data within a specified semantic range, capturing privacy risks from similar or partially overlapping data. These meth-

ods offer insights into privacy risks (Miresghallah et al., 2022), test-set contamination (Oren et al., 2023), and copyright protection (Meeus et al., 2023; Duarte et al., 2024).

2.2 Pre-training Data Detection.

Detecting training data in LLMs is challenging due to the scale of pre-training and the impracticality of shadow models (Shi et al., 2024). While some frameworks focus on prompt tuning or distribution-free attacks (Huang et al., 2025; Fu et al., 2025), **likelihood-based methods** remain dominant. Foundational approaches like **Loss** (Yeom et al., 2018) utilize average log-likelihood, which is often calibrated using reference models **Ref** (Carlini et al., 2021) or synthetic neighbors **Neighbor** (Mattern et al., 2023). To better capture memorization, outlier-based methods such as **Min- $k\%$** (Shi et al., 2024) and its normalized version **Min- $k\%++$** (Zhang et al., 2025b) focus on the lowest-probability tokens. More recent variants incorporate conditional contexts **ReCaLL** (Xie et al., 2024), fine-tuning-based amplification (Zhang et al., 2025a), or multi dynamic signals like loss change rates **CAMIA** (Chang et al., 2025). Distinct from these scoring functions, our work identifies the **positional decay of memorization signals** through token entropy and provides a training-free and plug-and-play enhancement applicable to various likelihood-based baselines.

3 Background

We first formalize the problem of pre-training data detection as defined in prior studies (Shokri et al., 2017; Shi et al., 2024; Duan et al., 2024), and then the likelihood-based scoring functions for Pre-training Data Detection methods in LLMs.

3.1 Problem Statement

Pre-training data detection is cast as a membership inference attack (MIA) (Shokri et al., 2017). Denote a pre-trained auto-regressive LLM as M and its unknown training corpus as \mathcal{D} . For an arbitrary text sample x , MIA aims to infer whether $x \in \mathcal{D}$ (member sample) or $x \notin \mathcal{D}$ (non-member sample). Let $s(x; M)$ represent the scoring function that assigns a real-valued “membership” score to x based on M ’s outputs. We make a binary decision via

$$\hat{y} = \mathbb{I}(s(x; M) \geq \epsilon), \quad (1)$$

where ϵ is a case-specific threshold and $\mathbb{I}(\cdot)$ is the indicator function. Consistent with the grey-box

setting (Shi et al., 2024; Duan et al., 2024; Zhang et al., 2025b), we assume that only M ’s output statistics (logits, token probabilities, loss values) are accessible; internal weights and gradients remain hidden. Designing an effective $s(x; M)$ to maximize the separation between member and non-member distributions is the core of detection task.

3.2 Likelihood-based Score functions

Modern LLMs are trained by maximizing the likelihood of training token sequences (Radford et al., 2019; Brown et al., 2020). Concretely, given a sequence $\mathbf{x} = (x_1, \dots, x_T)$, an auto-regressive LLM factorizes its joint probability using the chain rule: $p(\mathbf{x}) = \prod_{t=1}^T p(x_t | x_{<t})$, where $x_{<t} = (x_1, \dots, x_{t-1})$ is the prefix context. At inference time, the model generates text token by sampling from the conditional distribution $p(\cdot | x_{<t})$. In light of this, researchers usually design likelihood-based scoring functions to detect pre-training data in LLMs (Yeom et al., 2018). For example, based on the observation that members tend to have higher log-likelihood than non-members, the loss-based score (Yeom et al., 2018) is defined as the (negative) log-likelihood of the input sequence,

$$s_{\text{loss}}(x) = \frac{1}{T} \sum_{t=1}^T \log p(x_t | x_{<t}), \quad (2)$$

where we flip the sign of the conventional loss-based score so that, consistent with other methods, higher scores indicate stronger membership. Instead of using the likelihood of all tokens, **Min- $k\%$** (Shi et al., 2024) selects the $k\%$ tokens with the smallest log-probabilities and averages them:

$$s_{\text{Min-}k\%}(x) = \frac{1}{|\mathcal{S}_k|} \sum_{x_t \in \text{Min-}k\%(x)} \log p(x_t | x_{<t}), \quad (3)$$

where \mathcal{S}_k represents the set of token positions corresponding to the smallest $k\%$ log-probabilities in the sequence. The intuition is that a non-member example is more likely to include a few outlier words with low likelihoods than members. Other methods are deferred to Appendix A.

4 Methodology

We first present the core motivation based on an empirical observation about token entropy. We then introduce the general, plug-and-play method, Positional Decay Reweighting (PDR), and demonstrate how to apply it to enhance existing likelihood-based Pre-training Data Detection scores.

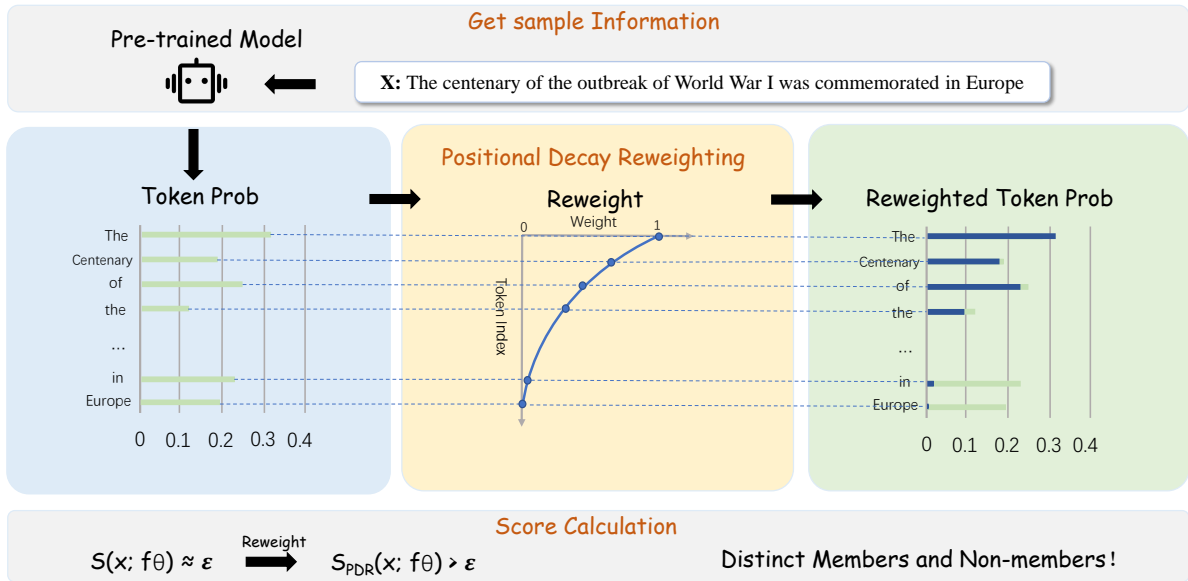


Figure 2: **Overview of Positional Decay Reweighting (PDR).** Our method reweights the predictive probabilities of input samples based on token positions, emphasizing early tokens with higher weights. This reweighting enhances the distinction between member and non-member samples by amplifying critical signals in the score S , making it more effective for MIA. The framework is training-free, also plug-and-play for likelihood-based scoring methods.

4.1 Motivation

Our methodology is built on a key insight into how autoregressive language models process information. From an information-theoretic perspective, a fundamental principle is that conditioning on more information cannot increase entropy, i.e., $H(z|x, y) \leq H(z|y)$. For autoregressive models, the uncertainty at each step can be quantified by the conditional entropy of the next token over the vocabulary V , given the prefix context $x_{<t}$:

$$H(p(\cdot|x_{<t})) = - \sum_{v \in V} p(v|x_{<t}) \log p(v|x_{<t}). \quad (4)$$

Although the classic principle compares the entropy of the same random variable, whereas here we are comparing the entropy for different variables (x_t and x_{t+1}), it is a widely observed empirical phenomenon that the entropy at position t is frequently greater than at position $t + 1$.

To empirically investigate this phenomenon, we visualized the average token-level entropy across multiple diverse datasets, as shown in Figure 1 (a). The visualization reveals two crucial findings. First, despite the varied nature of the corpora, they all exhibit a **dominant trend**: a high-entropy initial region followed by a general downward trend as the sequence progresses. Second, it highlights key differences in these trends; while datasets like Github and Wikipedia show a relatively smooth decay, corpora with more heterogeneous structures—such as

ArXiv (with section headings and equations) and HackerNews (mixing prose, code, and quotes)—display significantly more volatility.

These empirical findings have direct implications for membership inference, as the initial high-entropy region provides a unique setting to distinguish memorization from generalization. An unusually confident (i.e., high-probability) prediction for a token in such a position strongly suggests that this confidence does not stem from contextual generalization, but rather from rote memorization of specific sequences in its training set. Conversely, in later positions (low-entropy regions), the abundance of context makes predictions easier for **both** member and non-member samples, thus shrinking the discriminative gap between them. This is directly confirmed by Figure 1 (b), which shows that this gap is largest at the beginning of the sequence and diminishes over time.

This analysis leads to our refined core hypothesis: *the memorization signal is not uniformly distributed* but is heavily skewed towards the beginning of a sequence, with its strength generally decaying with token position. This crucial insight reveals a limitation in existing likelihood-based methods. Whether they use scores from all tokens (like Loss) or from a subset of low-likelihood tokens (e.g., Min- $k\%$), **they overwhelmingly rely on uniform weighting schemes**. By treating the scores from different positions as equally impor-

tant, they dilute the potent, high-fidelity signals concentrated in the early positions with noisy, less informative signals from the end. This oversight prevents them from fully exploiting the powerful evidence of memorization. Therefore, a principled, position-aware approach is not merely an incremental improvement, but a necessary step to enhance performance.

4.2 Plug-and-Play Positional Decay Reweighting (PDR)

Based on our core hypothesis established above that memorization signals are heavily skewed towards the beginning of a sequence, we argue that the performance of likelihood-based Detection methods is fundamentally limited by their uniform scoring mechanism. To rectify this, we propose Positional Decay Reweighting (PDR): a simple, effective, and “plug-and-play” framework designed to inject this crucial positional prior into existing methods. Our overview is illustrated in Figure 2.

PDR operates by re-weighting token-level scores using monotonically decreasing functions based on a token’s position t in a sequence of length T . This systematically assigns higher importance to earlier tokens, where the signal is strongest, and lower importance to later ones. We explore three simple, standardized, and effective families of decay functions:

1. **Linear Decay:** This function linearly decreases the weight from 1. The rate of decay is controlled by a single hyperparameter $\alpha \in [0, 1]$:

$$w_{\text{linear}}(t) = 1 - \alpha \left(\frac{t-1}{T-1} \right), \quad (5)$$

where T is the total sequence length. When $\alpha = 0$, all tokens are weighted equally, reducing to the original unweighted score.

2. **Exponential Decay:** This function applies a sharper, non-linear decay, placing a much stronger emphasis on the initial tokens:

$$w_{\text{exp}}(t) = \exp(-\alpha \cdot (t-1)). \quad (6)$$

The hyperparameter $\alpha \geq 0$ controls the steepness of the decay.

3. **Polynomial Decay:** This function provides a flexible decay curve whose shape is controlled by the exponent α . The hyperparameter $\alpha > 0$ determines the curvature of the decay. Values of

$0 < \alpha < 1$ result in a slower initial decay, while values $\alpha > 1$ lead to a faster initial decay:

$$w_{\text{poly}}(t) = \left(1 - \frac{t-1}{T-1} \right)^\alpha. \quad (7)$$

We defer the visualization of three weight decay functions into Figure 7 of Appendix G.

4.3 Applying PDR to Detection Scoring Functions

A key advantage of PDR is its “plug-and-play” nature. It operates as a lightweight wrapper designed to correct existing likelihood-based methods, requiring no modification to the target model’s architecture or training process. This makes it a broadly applicable technique. We now show how PDR integrates with two representative scoring functions.

For methods that aggregate scores across the entire sequence, such as standard Loss score in Eq.(2), PDR injects the positional prior by applying weights to each token’s log-probability before aggregation. The resulting PDR-Loss score is defined as:

$$s_{\text{PDR-Loss}}(x) = \frac{1}{T} \sum_{t=1}^T w(t) \cdot \log P(x_t | x_{<t}). \quad (8)$$

The integration is more nuanced for outlier-based methods like Min- $k\%$ in Eq.(3). Here, a crucial detail is the order of operations. To preserve the integrity of the outlier selection process, PDR is applied *after* the tokens have been selected based on their original, unweighted scores. The re-weighting then uses the *original position* of these selected tokens, ensuring that we are amplifying the most informative signals as identified by the baseline method. The PDR-Min- $k\%$ score is thus:

$$s_{\text{PDR-Min-}k\%}(x) = \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot \log P(x_t | x_{<t}), \quad (9)$$

where \mathcal{S}_k is the set of token positions with the smallest $k\%$ log-probabilities.

PDR can also be combined with other scoring functions, such as the reference-based method (Ref), the normalized outlier method (Min- $k\%+$), and finetuned-based FSD . The full set of PDR-enhanced scoring functions and algorithm are detailed in Appendix B and Algorithm 1. By systematically amplifying the signal from critical early tokens, PDR aims to widen the score distribution gap between member and non-member samples, thereby enhancing overall detection performance.

Table 1: AUROC results on WikiMIA benchmark (Shi et al., 2024). *Ori.* and *Para.* denote the original and paraphrased settings. [†]Neighbor results are from Zhang et al. (2025b). For method pairs, the format is: baseline / **w/ LPDR** (our Linear PDR). The higher score in a pair is in **bold**, and the performance gains of our method on the average results are highlighted in **purple**.

Model	Len	Lowercase		Zlib		Neighbor		Loss		Ref		Min- <i>k</i> %		Min- <i>k</i> %++	
		<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>	<i>Ori.</i>	<i>Para.</i>
Mamba-1.4B	32	60.9	60.6	61.9	62.3	64.1	63.6	61.0/61.5	61.3/61.8	62.2/62.2	62.3/62.3	63.3/63.5	62.9/63.1	66.4/67.4	65.7/66.3
	64	57.0	57.0	60.4	59.1	60.6	60.6	58.2/59.7	56.4/59.5	60.6/61.1	59.6/60.8	61.7/62.9	58.0/61.8	67.2/68.2	62.2/65.5
	128	58.5	57.7	65.6	65.3	64.8	62.6	63.3/63.6	62.7/64.1	62.0/64.1	61.1/64.6	66.8/65.5	64.4/65.8	67.7/70.2	63.3/68.2
Pythia-6.9B	32	62.2	61.7	64.4	64.2	65.8	65.5	63.8/64.0	64.1/64.2	63.6/63.5	63.5/63.5	66.3/66.3	65.1/65.1	70.3/70.8	67.6/67.7
	64	58.2	57.7	62.6	61.6	63.2	63.1	60.7/62.6	59.3/62.4	62.4/63.3	62.9/64.0	65.0/66.7	61.1/65.1	71.6/72.1	64.2/68.3
	128	60.5	59.9	67.6	67.4	67.5	64.3	65.1/65.6	64.7/66.6	63.3/65.1	62.9/65.9	69.5/67.8	67.0/68.9	69.8/72.4	65.9/72.2
LLaMA-13B	32	64.0	63.2	67.8	68.3	65.8	65.0	67.5/67.7	68.0/68.2	57.9/57.8	56.2/56.1	66.8/66.8	66.2/66.2	84.4/85.9	82.7/84.1
	64	62.0	61.0	65.3	65.3	64.1	64.7	63.6/65.0	63.1/66.2	63.4/59.8	60.9/57.9	66.0/67.4	63.5/67.2	84.3/87.2	78.8/84.3
	128	60.6	56.3	69.7	69.6	68.3	64.0	67.8/68.7	67.2/69.1	62.6/64.9	59.7/61.7	71.5/71.2	68.6/71.0	83.8/88.4	76.2/84.3
GPT-NeoX-20B	32	68.3	66.9	69.3	68.5	70.2	68.3	69.1/68.9	68.6/68.3	67.6/67.4	66.7/66.6	72.2/72.0	69.6/69.4	75.1/75.2	69.7/69.5
	64	66.3	65.6	68.1	66.5	67.1	67.4	66.6/67.6	64.4/67.2	66.0/66.8	66.0/67.2	72.2/70.8	66.1/68.7	76.5/76.4	66.2/68.2
	128	68.0	67.6	72.3	72.0	71.6	69.6	70.7/70.7	69.7/71.4	68.3/69.5	68.4/70.2	75.6/74.5	73.0/75.2	75.4/75.7	70.6/72.6
OPT-66B	32	62.8	62.3	65.8	65.3	68.2	66.7	65.6/65.6	65.3/65.0	68.6/68.6	67.9/68.0	67.5/67.7	65.8/65.8	69.7/70.2	67.0/67.1
	64	61.1	60.0	63.9	62.2	64.1	64.6	62.3/64.2	60.3/63.1	66.9/68.2	67.8/69.1	66.5/68.1	62.5/66.0	69.8/70.1	63.3/66.6
	128	58.9	57.6	67.3	66.9	67.7	63.4	65.5/66.7	64.5/66.9	66.9/68.6	67.0/69.5	70.6/70.6	67.2/69.9	71.1/72.9	67.0/69.5
Average	32	63.7	63.0	65.8	65.7	66.8	65.8	65.4/65.5 ^{+0.1}	65.5/65.5	64.0/63.9	63.3/63.3	67.2/67.3 ^{+0.1}	65.9/65.9	73.2/73.9 ^{+0.7}	70.5/70.9 ^{+0.4}
	64	60.9	60.3	64.1	62.9	63.8	64.1	62.3/63.8 ^{+1.5}	60.7/63.7 ^{+3.0}	63.9/63.9	63.5/63.8 ^{+0.3}	66.3/67.2 ^{+0.9}	62.2/65.8 ^{+3.6}	73.9/74.8 ^{+0.9}	66.9/70.6 ^{+3.7}
	128	61.3	59.8	68.5	68.2	68.0	64.8	66.5/67.1 ^{+0.6}	65.7/67.6 ^{+1.9}	64.6/66.4 ^{+1.8}	63.8/66.4 ^{+2.6}	70.8/69.9	68.0/70.2 ^{+2.2}	73.6/75.9 ^{+2.3}	68.6/73.3 ^{+4.7}

5 Experiments

5.1 Setup

Benchmarks. We evaluate our method on two commonly-used benchmarks for LLM pre-training data detection. (1) **WikiMIA** (Shi et al., 2024) uses Wikipedia texts, distinguishing members by timestamps, and includes different length text for both *original* and *paraphrased* settings. (2) **MIMIR** (Duan et al., 2024), built on the Pile dataset (Gao et al., 2020), is more challenging as it minimizes distributional and temporal shifts between member and non-member data.

Baselines. We consider several representative and advanced methods as our baselines. A fundamental approach is **Loss** (Yeom et al., 2018). Reference-based methods include **Ref** (Carlini et al., 2021), **Zlib**, and **Lowercase** (Carlini et al., 2021). **Neighbor** (Mattern et al., 2023) compares the sample’s score against its neighbors’ scores. **Min-*k*%** (Shi et al., 2024) averages the lowest *k*% of token scores, and **Min-*k*%++** (Zhang et al., 2025b) adds score normalization. We also include **FSD** (Zhang et al., 2025a), which leverages score differences after fine-tuning the model on non-member data.

Models. For WikiMIA, we use Pythia (Biderman et al., 2023) (2.8B, 6.9B, 12B), LLaMA (Touvron et al., 2023a) (13B, 30B), GPT-NeoX (Black et al., 2022) (20B), OPT (Zhang et al., 2022) (66B), and Mamba (Gu and Dao, 2023) (1.4B, 2.8B). For MIMIR, we follow (Duan et al., 2024) and use the Pythia model series (160M, 1.4B, 2.8B, 6.9B, 12B). For FSD, we follow Zhang et al. (2025a) and use

GPT-J-6B, OPT-6.7B, Pythia-6.9B, LLaMA-7B, and GPT-NeoX-20B.

Metrics and Settings. Following standard practice (Carlini et al., 2021; Shi et al., 2024), we use AUROC as the primary metric and also report True Positive Rate (TPR) at low False Positive Rates. For brevity, we use LPDR, EPDR, and PPDR to denote our PDR with Linear, Exponential, and Polynomial decay. We use a commonly-used $k = 20$ for Min-*k*% and Min-*k*%++. In the main body, we primarily report results for LPDR. To demonstrate the general effectiveness of a simple and strong positional prior, we use a fixed $\alpha = 1$ for all our experiments with LPDR except for very short sequences (WikiMIA, $T = 32$), where such a sharp decay is suboptimal. More details about datasets, baselines and settings are deferred to Appendix C.

5.2 Main Results.

Results on WikiMIA. As shown in Table 1, we report the AUROC results of different methods on WikiMIA with varying sequence lengths of {32, 64, 128} on different backbones; please see Appendix D.1 for overall results on more methods (including our EPDR, PPDR), backbones and other metrics. Besides, we also plot the ROC curves in Appendix D.2 to demonstrate the consistent superiority of our method across various False Positive Rate (FPR) thresholds.

We observe that introducing our proposed linear positional decay reweighting strategy generally enhances the performance of existing likelihood-based MIA methods. This improvement is especially evident on the advanced Min-*k*%++. For

Table 2: AUROC scores of various MIA methods over five Pythia models on the Mimir dataset. Pub and Wiki denote Pubmed Central and Wikipedia (en). Avg* scores are computed by excluding Arxiv and HackerNews. †Neighbor results are from Zhang et al. (2025b), induces significant extra computational cost than others ($25\times$ in this case), for which reason we don’t run on the 12B model.

Method	Wiki	Pile-CC	Pub	Math	GitHub	Avg*
Lowercase	52.2	49.3	51.1	48.9	71.1	54.5
Zlib	52.7	50.4	50.4	48.1	71.9	54.7
†Neighbor	51.9	50.1	49.2	47.4	69.3	53.6
Loss	51.9	50.3	50.3	48.5	70.8	54.4
w/ LPDR	52.8	50.7	50.3	48.6	70.9	54.6
Min- $k\%$	51.8	50.7	50.9	49.2	70.9	54.7
w/ LPDR	54.2	51.2	51.0	49.5	71.0	55.4
Min- $k\%++$	54.0	50.5	51.9	50.3	70.4	55.4
w/ LPDR	55.5	50.8	52.4	50.3	70.1	55.8

instance, when combined with the Min- $k\%++$ method, the performance gains from our LPDR become more pronounced as sequence length increases. Our LPDR improves the average AUROC by 0.7 (*Ori.*) and 0.4 (*Para.*) for length 32, by 0.9 (*Ori.*) and 3.7 (*Para.*) for length 64, and achieves the most significant gains of 2.3 (*Ori.*) and 4.7 (*Para.*) for length 128. These results prove the effectiveness of the our designed weight decay method in re-weighting token-level scores.

Combination with FSD on WikiMIA. Since ours is a plug-and-play reweighting method, it can also be used to enhance the finetune-based FSD. We perform the experiments on WikiMIA dataset with different LLMs, where we first finetune LLMs with non-member samples following its official code, then use Min- $k\%$ and Min- $k\%++$ as its score functions. To combine ours with FSD, we apply our LPDR to reweight the score functions. We show the results in Figure 3 and details in Appendix E. We can find that our LPDR provides consistent improvements on FSD no matter with Min- $k\%++$ or Min- $k\%$ as the score functions. It demonstrates that ours is also beneficial for finetune-based methods by reweighting its score functions.

MIMIR Results. As noted by prior work, MIMIR is particularly difficult because its training and non-training texts are sourced from the same sources, minimizing distributional shifts. Furthermore, we identify that the sub-datasets within MIMIR exhibit notable differences in structural composition. As visually confirmed by their volatile entropy profiles in Figure 1 (a), corpora like *ArXiv* and *HackerNews* are structurally heterogeneous. This distinguishes them from more homogeneous corpora like Wikipedia or GitHub. We treat the heterogeneous datasets as stress tests and compute an Avg* score on the five sub-datasets that

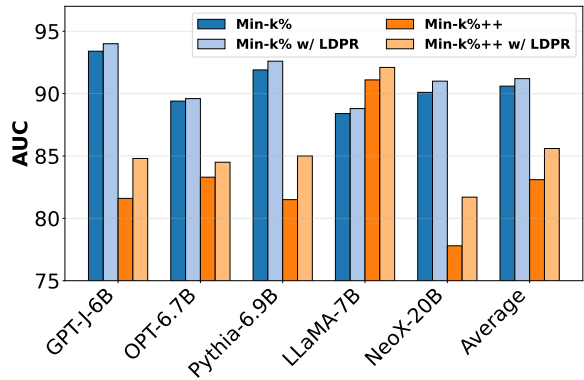


Figure 3: AUROC comparison of our LPDR method when integrated with Min- $k\%$ and Min- $k\%++$ across various LLMs on WikiMIA dataset within the FSD framework.

align with our method’s positional prior. As listed in Table 2, on this benchmark, baselines themselves perform close to random guess, underscoring its difficulty. We can find that introducing ours can improve the Loss, Min- $k\%$, and Min- $k\%++$. It validates the efficacy of PDR’s positional prior in structurally homogeneous text datasets. The complete averaged results for all sub-datasets and the detailed results are deferred to Appendix F.

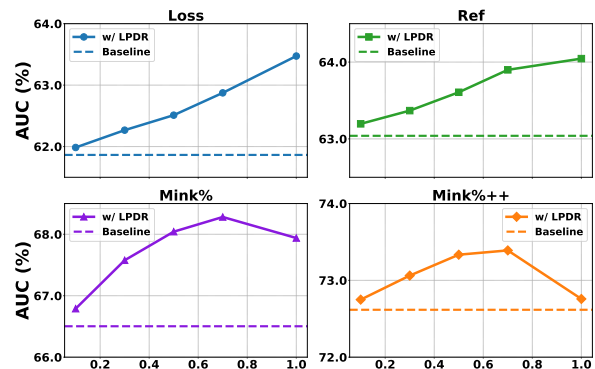


Figure 4: AUROC comparison of different α for LPDR on log-likelihood methods (Loss, Ref, Min- $k\%$, Min- $k\%++$), results from Pythia-12B model at WikiMIA *ori.* of 64 length.

5.3 Further Analysis

Ablation Study about Weight Design. We conduct an ablation study with several alternative weighting schemes. We consider applying PDR weights in different orders: *Random* (a random shuffled sequence) and *Reverse* (a monotonically increasing sequence). We also evaluate token-level entropy weights derived from a single *Sample* or the entire *Dataset*. Additionally, following CAMIA (Chang et al., 2025), we explore a dynamic strategy using a *Fitted Slope* from each sample’s loss sequence as the decay parameter α (details in Appendix C.2). For Min- $k\%$ methods, we compare our standard reweighting *after* selection against reweighting the full sequence *Before* selection.

Table 3: Ablation study about weighting schemes. We evaluate slope(CAMIA sample loss slope), different weight orderings (Random, Reverse), entropy-based weights (Sample, Dataset), and the reweighting for K-select in Min- $k\%$ and Min- $k\%++$. Results are reported on the WikiMIA dataset using the Pythia-6.9B model with a sequence length of $T = 128$.

Method	Base	Slope	Weights Order		Entropy		K select	w/ PDR(Ours)		
			Random	Reverse	Sample	Dataset	Before	LPDR	EPDR	PPDR
Loss	65.1	65.1	64.5	63.4	63.2	66.4	-	<u>65.6</u>	64.8	<u>66.1</u>
Ref	63.3	63.5	61.8	58.0	62.1	<u>63.8</u>	-	65.1	67.5	65.7
Min- $k\%$	69.5	69.5	64.3	59.8	64.1	70.6	68.1	67.8	69.2	66.7
Min- $k\%++$	69.8	69.8	66.5	61.0	68.7	70.5	70.3	72.4	71.2	72.7

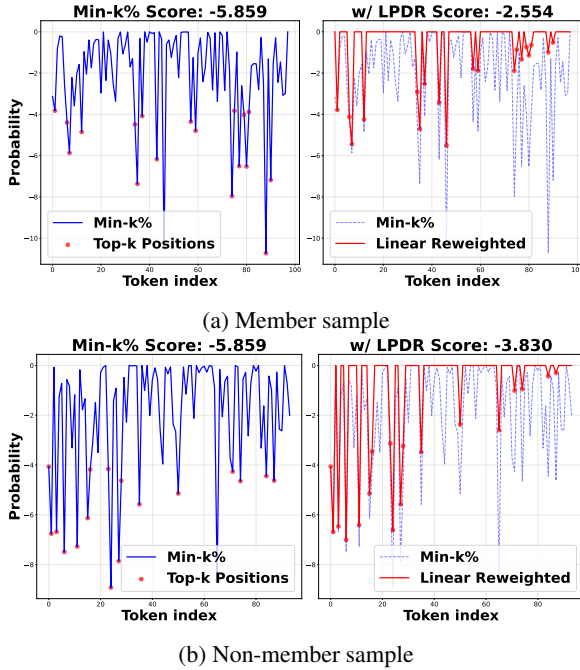


Figure 5: Token-level score changes for (a) member sample and (b) non-member sample after applying LPDR to Min- $k\%$. The red dot means the selected token by Min- $k\%$, blue and red line denote the original and reweighted token-level score.

As shown in Table 3, *Random* and *Reverse* orders degrade performance, confirming the necessity of monotonically decreasing weights. For Min- $k\%$, reweighting *after* selection proves superior by amplifying isolated signals. Regarding alternatives, *Dataset*-level entropy is effective but impractical, while *Sample*-level entropy and the *Fitted Slope* suffer from high volatility, yielding marginal gains. Thus, PDR strikes a critical balance as a simple, robust, and effective prior.

The Effect of Sharpness in Weight Decay. As shown in Figure 4, LPDR consistently outperforms baselines across various decay rates α . While full-sequence methods (Loss, Ref) peak with sharp decay ($\alpha = 1$) and outlier-based methods (Min- $k\%$, Min- $k\%++$) favor smoother decay, $\alpha = 1$ remains a robust default delivering substantial gains without method-specific tuning. Analyses of weight functions and truncation ratios in Appendix G and H further confirm PDR’s robustness across settings.

Score Changes when Using PDR. To further illustrate the mechanism of PDR, we selected a pair of member and non-member samples that are indistinguishable within Min- $k\%$ for sharing the same score. Figure 5 visualizes how our LPDR method resolves this ambiguity. By applying a monotonically decreasing weight, PDR enhances the importance of tokens at earlier positions and decreases the importance of tokens at later positions. For the member sample (a), whose memorization signals (high-probability tokens) are concentrated at the beginning, this reweighting process significantly amplifies its final score. In contrast, the non-member sample (b) is less affected. Thus, PDR effectively breaks the tie, creating a clear distinction between the two samples and enhancing overall detection accuracy. For additional examples, detailed sample analysis, and score distribution visualizations, refer to Appendices I, J, and K.

6 Conclusion

In this paper, we address a critical oversight in likelihood-based MIA methods: the failure to account for the positional nature of memorization signals. Based on the observation that a model’s predictive uncertainty decreases as a sequence progresses, we argue that membership signals are strongest in early tokens. Existing methods dilute these signals by treating all positions equally. We introduce Positional Decay Reweighting (PDR), a simple, plug-and-play framework that applies a monotonically decreasing weight to token scores, enhancing existing MIA methods without requiring model changes. Extensive experiments validate that PDR significantly improves the performance of strong baselines like Min- $k\%$ and Min- $k\%++$, both standalone and within advanced frameworks like FSD. Our work provides a more robust approach to membership inference by systematically prioritizing more reliable positional signals, contributing to a deeper understanding of privacy risks in LLMs.

7 Limitations

We acknowledge several limitations in our study that could influence the generalization and applicability of our findings.

Evaluation Language Limitation. Our evaluations primarily focus on English-language data, following established benchmarks like WikiMIA and MIMIR. While the principle of positional entropy decay is likely universal across auto regressive models regardless of language, its practical effectiveness may vary. For instance, tokenization granularity differs significantly across languages, which could alter positional information distribution and the ideal decay shape. Furthermore, unique linguistic structures or cultural contexts might create different memorization patterns that are less susceptible to a uniform decay model. Assessing how PDR generalizes to multilingual settings, code-switched data, or low-resource languages remains an important direction for future research to confirm its universality.

Black-box Constraints and Dataset Complexity. We primarily operate in a black-box scenario where only the logits corresponding to the input text are available (Shi et al., 2024; Zhang et al., 2025b). This setting, while realistic for external auditors, is highly restrictive compared to white-box or gray-box access where gradients or model parameters could be used for more powerful attacks. This restriction limits the information leverageable for attacks, making it difficult to train effective auxiliary models without access to additional in-distribution samples. Furthermore, on complex benchmarks like MIMIR (Duan et al., 2024), the high degree of data homogeneity between members and non-members means the statistical signals for membership are extremely subtle. For black-box methods that rely on summary statistics like loss, these signals are often too weak to achieve strong discrimination, posing a significant challenge for detection.

8 Acknowledgement

The authors would like to thank the anonymous referees for their valuable comments. In this work, Jinhan Liu, Peng Wang and Dandan Guo are supported by the National Natural Science Foundation of China (No.62306125). This work was carried out during Dandan Guo’s visit to King Abdullah University of Science and Technology (KAUST).

9 Ethical Considerations

We undertake this study with a strong commitment to ethical research practices and responsible disclosure. By transparently communicating our methodology, findings, and limitations, we aim to raise awareness about privacy vulnerabilities associated with Large Language Models (LLMs).

Privacy Research Purpose. Our goal is to contribute constructively to the broader community by highlighting how easily memorized data can be exposed, even by simple reweighting strategies. Understanding these vulnerabilities is a prerequisite for developing stronger defenses, such as differential privacy training or advanced unlearning techniques, aligning our efforts with ongoing global initiatives for AI safety and privacy.

Data and Model Usage. Our study exclusively utilizes publicly available datasets (WikiMIA, MIMIR) and open-weights models (Pythia, LLaMA, OPT, etc.). These resources were employed strictly within their intended research purposes and license terms. Our research did not involve the collection of private user data or human subjects. To the best of our knowledge, the artifacts employed do not include personally identifiable information (PII) beyond what is already publicly present in the training corpora of these open models.

Potential for Misuse. We recognize that Membership Inference Attacks (MIAs) could be misused in ways that compromise data privacy. It is important to emphasize, however, that the techniques examined in this work are intended strictly as post hoc auditing mechanisms to assess the security and privacy safeguards of machine learning models. We maintain that the advantages of openly examining these vulnerabilities, which accelerates the development of more robust defensive measures within the research community, justify their disclosure, as the associated benefits outweigh the potential risks.

Tools and AI Assistance. For data preprocessing, modeling, and evaluation tasks, we employed widely-used, open-source software packages including Hugging Face Transformers and standard Python libraries such as NumPy and SciPy. Model-specific tokenizers and default parameter configurations were used unless explicitly stated otherwise. Lastly, AI assistants (e.g., ChatGPT, Gemini) were utilized to revise this manuscript. All generated content was rigorously reviewed and finalized by the authors.

References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Martin Bertran, Shuai Tang, Aaron Roth, Michael Kearns, Jamie H Morgenstern, and Steven Z Wu. 2023. Scalable membership inference attacks via quantile regression. *Advances in Neural Information Processing Systems*, 36:314–330.
- 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 *International Conference on Machine Learning*, pages 2397–2430. PMLR.
- Sidney Black, Stella Biderman, Eric Hallahan, Quentin Gregory Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Martin Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. 2022. GPT-neox-20b: An open-source autoregressive language model. In *Challenges & Perspectives in Creating Large Language Models*.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.
- Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE.
- 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.
- Hongyan Chang, Ali Shahin Shamsabadi, Kleomenis Katevas, Hamed Haddadi, and Reza Shokri. 2025. Context-aware membership inference attacks against pre-trained large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7299–7321.
- 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? *arXiv preprint arXiv:2402.07841*.
- André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. 2024. DE-COP: Detecting Copyrighted Content in Language Models Training Data. *Preprint*, arXiv:2402.09910.
- Jan Dubiński, Antoni Kowalczyk, Stanisław Pawlak, Przemysław Rokita, Tomasz Trzciniński, and Paweł Morawiecki. 2024. Towards more realistic membership inference attacks on large diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 4860–4869.
- Wenjie Fu, Xuandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2025. MIA-tuner: Adapting large language models as pre-training text detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Philadelphia, Pennsylvania, USA.
- 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*.
- Michael M Grynbaum and Ryan Mac. 2023. The times sues openai and microsoft over ai use of copyrighted work. *The New York Times*, 27(1).
- Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.
- Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022a. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.
- Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. 2022b. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37.
- Zhiheng Huang, Yannan Liu, Daojing He, and Yu Li. 2025. Df-mia: A distribution-free membership inference attack on fine-tuned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 343–351.
- 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, Toronto, Canada. Association for Computational Linguistics.

- Matthieu Meeus, Shubham Jain, Marek Rei, and Yves-Alexandre de Montjoye. 2023. Did the neurons read your book? document-level membership inference for large language models. *arXiv preprint arXiv:2310.15007*.
- 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 *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- 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*.
- Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B Hashimoto. 2023. Proving test set contamination in black box language models. *arXiv preprint arXiv:2310.17623*.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.
- 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.
- Jiashu Tao and Reza Shokri. 2025. Range membership inference attacks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 346–361. IEEE.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2022. On the importance of difficulty calibration in membership inference attacks. In *International Conference on Learning Representations*.
- Hengyu Wu and Yang Cao. 2025. Membership inference attacks on large-scale models: A survey. *arXiv preprint arXiv:2503.19338*.
- 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, Miami, Florida, USA. Association for Computational Linguistics.
- 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.
- Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. 2023. Low-cost high-power membership inference attacks. *arXiv preprint arXiv:2312.03262*.
- Hengxiang Zhang, Songxin Zhang, Bingyi Jing, and Hongxin Wei. 2025a. Fine-tuning can help detect pretraining data from large language models. In *The Thirteenth International Conference on Learning Representations*.
- Jingyang Zhang, Jingwei Sun, Eric Yeats, Yang Ouyang, Martin Kuo, Jianyi Zhang, Hao Frank Yang, and Hai Li. 2025b. 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 Diab, Xian Li, Xi Victoria Lin, and 1 others. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Likelihood-Based Score functions

This section provides the formal definitions for the baseline likelihood-based MIA score functions discussed in the main paper. For a given input sequence $\mathbf{x} = \{x_1, \dots, x_T\}$ of length T , these methods compute a score based on the token-level log-probabilities produced by the target model P .

Loss. (Yeom et al., 2018) The standard Loss-based method, also known as Negative Log-Likelihood (NLL), uses the average log-probability of a sequence as its score. A higher score (lower loss) is indicative of membership. The score is defined as:

$$S_{\text{Loss}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T \log P(x_t|x_{<t}) \quad (10)$$

Ref. (Carlini et al., 2021) The Reference-based method (Ref) calibrates the target model’s likelihood by subtracting the log-likelihood from a smaller reference model (P_{ref}). This helps to normalize for tokens that are inherently common or easy to predict. The score is:

$$S_{\text{Ref}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T (\log P(x_t|x_{<t}) - \log P_{\text{ref}}(x_t|x_{<t})) \quad (11)$$

Min- $k\%$. (Shi et al., 2024) The Min- $k\%$ method operates on the assumption that member samples have fewer "outlier" tokens with very low probabilities. It computes the score by averaging only the lowest $k\%$ log-probabilities in the sequence. Let S_k be the set of token positions corresponding to the lowest $k\%$ log-probabilities. The score is:

$$S_{\text{Min-}k\%}(\mathbf{x}) = \frac{1}{|S_k|} \sum_{t \in S_k} \log P(x_t|x_{<t}) \quad (12)$$

Min- $k\%++$. (Zhang et al., 2025b) The Min- $k\%++$ method enhances Min- $k\%$ by first normalizing the log-probability at each position t using pre-computed mean (μ_t) and standard deviation (σ_t) statistics for that position. This accounts for positional biases in the model’s predictions. Let the normalized score be $z_t = (\log P(x_t|x_{<t}) - \mu_t) / \sigma_t$. Let S_k be the set of positions corresponding to the lowest $k\%$ normalized scores z_t . The final score is:

$$S_{\text{Min-}k\%++}(\mathbf{x}) = \frac{1}{|S_k|} \sum_{t \in S_k} z_t \quad (13)$$

FSD. (Zhang et al., 2025a) Finetuning-based Score Difference (FSD) is a framework that enhances any base scoring function $S(\cdot)$. It computes

the difference between the score from the original model (M) and the score from a model fine-tuned on non-member data (M'). A larger difference suggests membership.

$$S_{\text{FSD}}(\mathbf{x}) = S(\mathbf{x}; M) - S(\mathbf{x}; M') \quad (14)$$

B Likelihood-Based Score Functions with PDR

This section details how our Position Difference Reweighting (PDR) method is integrated with various likelihood-based MIA score functions. For a given input sequence $\mathbf{x} = \{x_1, \dots, x_T\}$ of length T , PDR introduces a positional weight $w(t)$ for each token x_t at position t . The final score is then computed based on the weighted combination of token-level scores.

PDR-Loss. The standard Loss-based method, often conceptualized as Negative Log-Likelihood (NLL), uses the average log-probability of a sequence as its score. With PDR, we apply positional weights to the log-probabilities of each token before averaging. A lower weighted loss (which corresponds to a higher score) suggests the sequence is a member. The PDR-enhanced score is:

$$S_{\text{PDR-Loss}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T w(t) \cdot \log P(x_t|x_{<t}) \quad (15)$$

PDR-Ref. The Reference-based method (Ref) calibrates the target model’s likelihood by subtracting the log-likelihood from a smaller reference model (P_{ref}). PDR is applied to the resulting difference at each position. The score is defined as:

$$S_{\text{PDR-Ref}}(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^T w(t) \cdot \left(\log P(x_t|x_{<t}) - \log P_{\text{ref}}(x_t|x_{<t}) \right) \quad (16)$$

PDR-Min- $k\%$. Following the standard Min- $k\%$ procedure, we first identify the token positions corresponding to the lowest $k\%$ log-probabilities. Then compute the final score by taking a weighted average of the log-probabilities at only these selected positions, where each score is multiplied by its corresponding positional weight $w(t)$. Let S_k be the set of token positions corresponding to the lowest $k\%$ values of $\{\log P(x_t|x_{<t})\}_{t=1}^T$. The score is:

$$S_{\text{PDR-Min-}k\%}(\mathbf{x}) = \frac{1}{|S_k|} \sum_{t \in S_k} w(t) \cdot \log P(x_t|x_{<t}) \quad (17)$$

Algorithm 1 Overall algorithm for applying PDR to different logit-based MIA methods

1: **Input:**Test dataset $\mathcal{D} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$;Target model's predictive distribution $P(x_t|x_{<t})$;A chosen decay function $f_{\text{decay}} \in \{\text{Linear}, \text{Exponential}, \text{Polynomial}\}$;Decay hyperparameter α or p ;A base MIA scoring method $\mathcal{M} \in \{\text{Loss}, \text{Ref}, \text{Min-}k\%, \text{Min-}k\%++\}$;*(Optional)* Reference model $P_{\text{ref}}(x_t|x_{<t})$;*(Optional)* Positional normalization stats $\{\mu_t, \sigma_t\}_{t=1}^T$.2: **Output:** A list of PDR-enhanced scores $\mathcal{S}_{\text{PDR}} = \{s_1, \dots, s_N\}$.3: Initialize an empty list \mathcal{S}_{PDR} .4: **for** $i = 1$ to N **do** ▷ Iterate over all sequences in the dataset5: $\mathbf{x}^i = \{\mathbf{x}_1^i, \dots, \mathbf{x}_T^i\}$ *// Step 1: Compute Positional Weights for the current sequence*6: **for** $t = 1$ to T **do** ▷ Iterate over all positions in the sequence7: **if** f_{decay} is Linear **then**8: $w(t) \leftarrow 1 - \alpha \cdot \frac{t-1}{T-1}$ 9: **else if** f_{decay} is Exponential **then**10: $w(t) \leftarrow \exp(-\alpha \cdot (t-1))$ 11: **else if** f_{decay} is Polynomial **then**12: $w(t) \leftarrow \left(1 - \frac{t-1}{T-1}\right)^p$ 13: **end if**14: **end for** *// Step 2: Apply PDR to the chosen base MIA method*15: Initialize current score $s^i \leftarrow 0$.16: **if** \mathcal{M} is Loss **then**17: $s^i \leftarrow -\frac{1}{T} \sum_{t=1}^T w(t) \cdot \log P(x_t^i|x_{<t}^i)$ 18: **else if** \mathcal{M} is Ref **then**19: $s^i \leftarrow \frac{1}{T} \sum_{t=1}^T w(t) \cdot (\log P(x_t^i|x_{<t}^i) - \log P_{\text{ref}}(x_t^i|x_{<t}^i))$ 20: **else if** \mathcal{M} is Min- $k\%$ **then**21: Let \mathcal{S}_k be the set of token positions with the smallest $k\%$ log-probabilities.22: $s^i \leftarrow \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot \log P(x_t^i|x_{<t}^i)$ 23: **else if** \mathcal{M} is Min- $k\%++$ **then**24: For each t , compute normalized score $z_t = \frac{\log P(x_t^i|x_{<t}^i) - \mu_t}{\sigma_t}$.25: Let \mathcal{S}_k be the set of token positions with the smallest $k\%$ normalized scores z_t .26: $s^i \leftarrow \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot z_t$ 27: **end if**28: Append s^i to \mathcal{S}_{PDR} .29: **end for**30: **return** \mathcal{S}_{PDR}

PDR-Min- $k\%++$. Similarly, for Min- $k\%++$, we first identify the positions of the lowest $k\%$ normalized z-scores. The PDR-enhanced score is then the weighted average of these selected z-scores, with positional weights applied before averaging. Let $z_t = (\log P(x_t|x_{<t}) - \mu_t) / \sigma_t$, and let \mathcal{S}_k be the set of positions for the lowest $k\%$ values of $\{z_t\}_{t=1}^T$. The score is:

$$S_{\text{PDR-Min-}k\%++}(\mathbf{x}) = \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot z_t \quad (18)$$

PDR-FSD. Finetuning-based Score Difference (FSD) calculates the difference between a score function $S(\cdot)$ computed before and after finetuning the model on non-member data. Our PDR method can be applied to the base score function $S(\cdot)$ used within the FSD framework. If we denote the fine-tuned model as M' , the FSD score using a PDR-enhanced base method $S_{+\text{PDR}}$ is:

$$S_{\text{PDR-FSD}}(\mathbf{x}) = S_{+\text{PDR}}(\mathbf{x}; M) - S_{+\text{PDR}}(\mathbf{x}; M') \quad (19)$$

where $S_{+\text{PDR}}(\mathbf{x}; M)$ and $S_{+\text{PDR}}(\mathbf{x}; M')$ are the PDR-enhanced scores computed using the original model M and the fine-tuned model M' , respectively.

The detailed process of applying our PDR method to various likelihood-based MIA methods is outlined in Algorithm 1. This algorithm specifically illustrates the computation for combining PDR with Loss, Ref, Min- $k\%$, and Min- $k\%++$.

PDR-Min- $k\%$. Following the standard Min- $k\%$ procedure, we first identifies the token positions corresponding to the lowest $k\%$ log-probabilities. Then computes the final score by taking a weighted average of the log-probabilities at only these selected positions, where each score is multiplied by its corresponding positional weight $w(t)$. Let \mathcal{S}_k be the set of token positions corresponding to the lowest $k\%$ values of $\{\log P(x_t|x_{<t})\}_{t=1}^T$. The score is:

$$S_{\text{PDR-Min-}k\%}(\mathbf{x}) = \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot \log P(x_t|x_{<t}) \quad (20)$$

PDR-Min- $k\%++$. Similarly, for Min- $k\%++$, we first identify the positions of the lowest $k\%$ normalized z-scores. The PDR-enhanced score is then the weighted average of these selected z-scores, with positional weights applied before averaging. Let $z_t = (\log P(x_t|x_{<t}) - \mu_t) / \sigma_t$, and let \mathcal{S}_k be the set

of positions for the lowest $k\%$ values of $\{z_t\}_{t=1}^T$. The score is:

$$S_{\text{PDR-Min-}k\%++}(\mathbf{x}) = \frac{1}{|\mathcal{S}_k|} \sum_{t \in \mathcal{S}_k} w(t) \cdot z_t \quad (21)$$

PDR-FSD. Finetuning-based Score Difference (FSD) calculates the difference between a score function $S(\cdot)$ computed before and after finetuning the model on non-member data. Our PDR method can be applied to the base score function $S(\cdot)$ used within the FSD framework. If we denote the fine-tuned model as M' , the FSD score using a PDR-enhanced base method $S_{+\text{PDR}}$ is:

$$S_{\text{PDR-FSD}}(\mathbf{x}) = S_{+\text{PDR}}(\mathbf{x}; M) - S_{+\text{PDR}}(\mathbf{x}; M') \quad (22)$$

where $S_{+\text{PDR}}(\mathbf{x}; M)$ and $S_{+\text{PDR}}(\mathbf{x}; M')$ are the PDR-enhanced scores computed using the original model M and the fine-tuned model M' , respectively.

The detailed process of applying our PDR method to various likelihood-based MIA methods is outlined in Algorithm 1. This algorithm specifically illustrates the computation for combining PDR with Loss, Ref, Min- $k\%$, and Min- $k\%++$.

C Experiments Setting details

C.1 Benchmarks

We focus on two commonly-used benchmarks for pre-training data detection: (1) **WikiMIA** (Shi et al., 2024) is the first benchmark for pre-training data detection, comprising texts from Wikipedia events. The distinction between training and non-training data is established based on temporal timestamps. To enable fine-grained evaluation, WikiMIA organizes data into splits according to sentence length. It also includes two evaluation settings: the *original* setting evaluates the detection of verbatim training texts, while the *paraphrased* setting uses ChatGPT to paraphrase training texts and evaluates on paraphrased inputs.

(2) **MIMIR** (Duan et al., 2024) is built upon the Pile dataset (Gao et al., 2020). This benchmark poses greater challenges compared to WikiMIA, as the shared dataset origin between training and non-training texts eliminates substantial distribution shifts and temporal discrepancies (Duan et al., 2024).

C.2 Baselines

We consider several representative methods as our baselines:

- **Loss** (Yeom et al., 2018) is a general technique that directly uses the loss of the model as the detection score.
- **Ref** (Carlini et al., 2021) employs an additional, typically smaller, language model as a reference to calibrate the likelihood of the input text.
- **zlib** and **lowercase** (Carlini et al., 2021) use the compression entropy of zlib and the likelihood of the lowercase text as references to calibrate the likelihood.
- **Min- $k\%$** (Shi et al., 2024) examines the exact probabilities of the token and averages a subset of the lowest token scores from the input sequence.
- **Min- $k\%++$** (Zhang et al., 2025b) extends Min- $k\%$ by standardizing the log-probability of each token using the mean and standard deviation of log-probabilities at that specific position, making scores more comparable across different positions before applying the Min- $k\%$ selection.
- **FSD** (Zhang et al., 2025a) involves fine-tuning the model on non-member samples and using the difference in logit-based scores before and after fine-tuning for detection.
- **CAMIA** (Chang et al., 2025) esigns multiple dynamic signals for detection, we explore a variant based on one of its key signals: the loss decreasing rate. Specifically, for each sample \mathbf{x} , we compute a slope by performing a linear regression of its token-level losses $L_t(x_t)$ against their positions t . The slope is calculated as:

$$f_{\text{Slope}}(\mathbf{x}) = \frac{\sum_{t=1}^T (t - \bar{t})(L_t(x_t) - \bar{L})}{\sum_{t=1}^T (t - \bar{t})^2} \quad (23)$$

with $\bar{t} = \frac{T+1}{2}$ and $\bar{L} = \frac{1}{T} \sum_{t=1}^T L_t(x_t)$. We then use this dynamically computed current sample’s slope sample as the decay parameter α in our reweighting scheme, referring to this method a w/ LDPR (CAMIA slope).

C.3 Environment

All experiments were conducted on the Ubuntu 20.04.4 LTS operating system, Intel(R) Xeon(R) Gold 5220 CPU @ 2.20GHz with a single NVIDIA

A40 48GB GPU and 512GB of RAM. The framework is implemented with Python 3.9.0 and PyTorch 2.6.0. Other key packages include transformer 4.40.1, numpy 1.24 and accelerate 0.26.0.

C.4 Models

This section details the specific models used in our experiments. For the Ref method, the choice of reference model depends on the dataset following (Carlini et al., 2021; Shi et al., 2024; Zhang et al., 2025b). On the **WikiMIA** dataset, we used the following reference models for different model families:

- For the **Pythia** family, we used Pythia-70M.
- For the **Llama** family, we used Llama-7B.
- For the **GPT** family, we used GPT-Neo-125M.
- For the **Mamba** family, we used Mamba-130M.
- For the **OPT** family, we used OPT-350M.

C.5 Hyperparameter

For Min- $k\%$ and Min- $k\%++$, we consistently use $k = 20\%$ following common practice. In our experiments, settings are as follows:

- **LPDR**: on the WikiMIA dataset, we generally set $\alpha = 1.0$ for sequence lengths of 64 and 128. For the shorter length of 32, a smaller weight was preferred like 0.1 or 0.5. On the Mimir dataset, we set $\alpha = 1.0$.
- **EPDR**: on the WikiMIA dataset, Ref and Min- $k\%++$ set $\alpha = 0.02$, where as Loss and Min- $k\%$ set $\alpha = 0.002$. On Mimir, $\alpha = 0.002$ was used.
- **PPDR**: on the WikiMIA dataset, we used $p = 0.1$ for length 32, but a much steeper decay of $p = 2.0$ for lengths 64 and 128. On Mimir, we set a gentler $p = 0.1$.

A clear trend emerges from these results. On the WikiMIA dataset, longer sequences tend to benefit from more aggressive, steeper weight functions, while shorter sequences and the more challenging Mimir dataset favor gentler, more gradual decay.

C.6 FSD Settings

For the Finetuning-based Score Difference (FSD) experiments, we follow the implementation details from the original paper. To construct the non-member dataset for fine-tuning, we first randomly

sample 30% of the entire dataset. All non-member samples within this subset are then used as the fine-tuning dataset. The remaining 70% of the data is reserved for testing. We use LoRA (Hu et al., 2022a) to fine-tune the base model for 3 epochs with a batch size of 8. The initial learning rate is set to 0.001 and is adjusted using a cosine scheduling strategy.

Table 4: Performance comparison on Pythia-6.9B across different sequence lengths. We report AUROC and TPR@0.5%FPR (mean \pm std). **w/ LPDR** denotes our method using linear weights. The p-value indicates the statistical significance of the improvement of LPDR over the corresponding baseline.

Length	Method	AUROC	AUROC Std	TPR@0.5%FPR	TPR Std	p-value	
32	Zlib	64.2	2.0	12.7	2.6	-	
	Lowercase	61.7	2.1	11.9	3.2	-	
	Loss	64.1	2.0	15.0	2.8	-	
	w/ LPDR	64.2	2.0	15.0	2.8	0.275	
	Ref	63.5	2.0	6.2	3.0	-	
	w/ LPDR	63.5	2.0	5.7	3.2	0.785	
	Min- k %	65.1	2.0	21.7	3.5	-	
	w/ LPDR	65.1	2.0	22.0	4.0	0.342	
	Min- k %++	67.6	2.0	14.5	3.2	-	
	w/ LPDR	67.8	2.0	14.5	2.6	0.218	
	64	Zlib	61.6	2.5	15.8	3.5	-
		Lowercase	57.7	2.5	11.3	2.2	-
Loss		59.3	2.5	10.6	3.7	-	
w/ LPDR		62.4	2.5	13.0	5.4	0.001	
Ref		62.9	2.4	16.2	2.8	-	
w/ LPDR		64.0	2.3	7.7	3.6	0.152	
Min- k %		61.1	2.5	12.7	3.0	-	
w/ LPDR		65.1	2.4	18.3	5.2	0.019	
Min- k %++		64.2	2.4	10.2	3.2	-	
w/ LPDR		68.3	2.4	12.7	4.6	0.006	
128		Zlib	67.4	3.4	20.9	5.8	-
		Lowercase	59.9	3.6	11.5	4.2	-
	Loss	64.7	3.5	16.5	6.8	-	
	w/ LPDR	66.6	3.4	15.1	4.9	0.061	
	Ref	62.9	3.5	8.6	5.7	-	
	w/ LPDR	65.9	3.4	10.8	5.0	0.016	
	Min- k %	67.0	3.4	16.5	6.1	-	
	w/ LPDR	68.9	3.3	23.0	9.4	0.270	
	Min- k %++	65.9	3.5	18.0	7.1	-	
	w/ LPDR	72.2	3.3	18.7	6.9	0.001	

D WikiMIA Results

D.1 WikiMIA Results on fixed α

This section presents the comprehensive results on the WikiMIA benchmark. We report detailed AUROC and TPR scores across all models and sequence lengths under both the original and paraphrased settings. The AUROC results are shown in Table 5 and Table 6, respectively, while the corresponding TPR results are reported in Table 7 and Table 8. Additionally, we supplement our main results with Accuracy and F1-score in Table 9 and Table 10. Since these metrics require a specific decision threshold to distinguish member from non-member samples, we determined the optimal threshold using the Youden’s J statistic.

D.2 ROC curve Visualization

This section provides ROC curve visualizations to offer a more detailed view of our method’s performance. Figure 6 plots the ROC curves for several baseline methods and their LPDR-enhanced counterparts on the WikiMIA benchmark (length 128). Specifically, we show results for (a) Llama-13B on the paraphrased setting and (b) Pythia-6.9B on the original setting. As illustrated, the PDR-enhanced methods consistently offer a more favorable trade-off, achieving a higher True Positive Rate (TPR) for any given False Positive Rate (FPR). This enhanced discriminative capability helps to explain the AUROC gains reported throughout the paper.

D.3 Static Analysis

To rigorously assess the statistical significance of our method’s improvements, we performed a non-parametric bootstrap analysis based on the model prediction scores and ground-truth labels. Our procedure is as follows:

- **Bootstrap Resampling:** For each method, we generated $N = 1000$ bootstrap replicates. Each replicate was created by sampling indices from the original test set with replacement, forming a new dataset of the same size. A random seed was fixed to ensure reproducibility. If a replicate happened to contain samples from only one class, it was discarded for that specific calculation.
- **Metrics and Confidence Intervals:** For each of the N bootstrap replicates, we calculated the AUROC and TPR@0.5%FPR. After generating all replicates, we computed the mean and standard deviation of these metrics. The 95% confidence intervals were then derived empirically from the 2.5th and 97.5th percentiles of the resulting distribution of 1000 metric values.
- **Paired Significance Testing:** To evaluate if the improvements of our PDR-enhanced methods over their respective baselines are statistically significant, we conducted a paired bootstrap test. This approach is crucial for reducing variance caused by the sampling process. For each of the $N = 1000$ replicates, we used the *exact same set of resampled indices* to evaluate both the baseline method and the PDR-enhanced method. We then calculated the performance difference for that

Table 5: AUC-ROC on WikiMIA benchmark under original setting. w/ LPDR utilizes linear weights for reweighting, w/ EPDR utilizes exponential weights for reweighting, w/ PPDR utilizes polynomial weights for reweighting. †Neighbor results are from Zhang et al. (2025b).

Length	Models	Lowerrace	Zib	†Neighbor	Loss	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Ref	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Min-f%	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Min-f% +†	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR
32	Mamba-1.4B	60.9	61.9	64.1	61.0	61.1	61.5	60.9	61.2	62.2	62.0	62.2	62.7	62.2	63.3	63.2	63.5	63.2	63.8	66.4	66.6	67.4	66.6	67.1
	Mamba-2.8B	63.6	64.7	67.0	64.1	64.2	64.5	64.0	64.2	67.0	66.6	66.6	67.1	66.7	66.1	66.1	66.2	66.0	66.1	69.0	69.1	69.4	68.9	69.1
	Pythia-2.8B	60.9	62.1	64.2	61.4	61.4	61.7	61.2	61.4	61.3	61.2	61.3	62.4	61.0	61.7	61.6	61.9	61.6	61.9	64.0	64.1	64.7	64.2	64.4
	Pythia-6.9B	62.2	64.4	65.8	63.8	63.8	64.0	63.7	63.8	63.6	63.4	63.5	64.6	63.2	66.3	66.2	66.3	66.1	65.4	70.3	70.4	70.8	70.1	70.5
	Pythia-12B	64.8	65.8	66.6	65.4	65.3	65.4	65.2	65.4	65.1	65.0	65.1	66.1	64.7	68.1	67.9	68.0	67.8	67.4	72.2	72.3	72.3	71.3	72.0
	Llama-1.8B	64.0	67.8	65.8	67.5	67.5	67.7	67.5	67.5	57.9	57.8	57.8	57.2	57.6	66.8	66.8	66.8	66.8	66.6	84.4	84.4	85.9	86.2	85.6
	Llama-3.0B	64.1	69.8	67.6	69.4	69.4	69.6	69.4	69.5	63.5	63.5	63.5	62.8	63.2	69.3	69.3	69.3	69.2	84.4	84.3	85.4	85.5	84.0	
	OPT-66B	62.8	65.8	68.2	65.6	65.5	65.6	65.5	65.7	68.6	68.5	68.6	69.3	68.4	67.5	67.4	67.7	67.4	67.5	69.7	69.8	70.2	69.4	70.0
	GPT-NeoX-20B	68.3	69.3	70.2	69.1	68.9	68.9	68.9	69.0	67.4	67.0	67.4	66.7	67.0	72.2	71.9	72.0	71.8	71.1	75.1	75.2	75.2	74.4	75.0
	Average	65.5	65.7	66.9	65.3	65.2	65.4	65.1	65.3	64.1	63.9	64.0	64.3	63.8	66.8	66.7	66.9	66.7	66.5	72.8	72.9	73.5	72.9	73.1
64	Mamba-1.4B	57.0	60.4	60.6	58.2	58.4	59.7	58.2	60.7	60.6	60.7	61.1	61.9	60.3	61.7	62.0	62.9	62.0	62.8	67.2	67.4	68.2	67.9	68.2
	Mamba-2.8B	61.7	63.0	63.6	61.2	61.3	63.0	61.2	63.6	64.3	64.5	66.1	66.6	65.5	65.1	65.4	66.2	65.4	65.3	70.6	70.8	70.4	70.0	69.6
	Pythia-2.8B	57.8	60.6	61.3	58.4	58.6	60.1	58.4	60.8	59.6	59.8	60.5	62.2	60.0	61.2	61.4	63.3	61.4	62.8	64.8	64.9	65.9	65.1	65.6
	Pythia-6.9B	58.2	62.6	63.2	60.7	60.9	62.6	60.6	63.0	62.4	62.5	63.5	60.0	62.7	65.0	65.0	65.2	66.3	65.3	71.6	71.8	72.1	71.2	71.2
	Pythia-12B	59.6	63.5	62.6	61.9	62.0	63.5	61.8	63.9	63.0	63.2	64.0	65.9	63.4	66.5	66.7	67.9	66.7	67.2	72.6	72.8	72.8	71.8	71.5
	Llama-1.8B	62.0	65.3	64.1	63.6	63.7	65.0	63.7	65.7	63.4	63.4	63.4	63.4	63.4	66.0	66.1	67.4	66.1	66.7	84.3	84.4	87.2	87.0	87.4
	Llama-3.0B	61.9	67.5	67.1	66.1	66.2	67.5	66.2	67.9	65.9	66.8	66.8	65.5	65.4	68.4	68.5	69.6	68.6	68.9	84.3	84.3	87.7	87.0	87.4
	OPT-66B	61.1	63.9	64.1	62.3	62.4	64.2	62.2	64.4	66.9	67.1	68.2	69.0	68.0	66.5	66.7	68.1	66.7	66.8	69.8	69.9	70.1	69.5	69.0
	GPT-NeoX-20B	66.3	68.1	67.1	66.6	66.7	67.6	66.5	67.6	66.0	66.1	66.8	67.1	65.4	72.2	72.3	70.8	72.2	68.6	76.5	76.6	76.4	76.0	75.0
	Average	60.6	63.9	63.7	62.1	62.2	63.7	62.1	64.2	63.9	64.0	63.9	64.8	62.8	65.8	66.0	67.0	66.0	66.0	73.5	73.7	74.5	73.9	73.9
128	Mamba-1.4B	58.5	65.6	64.8	63.3	63.3	63.6	62.9	64.2	62.0	62.1	64.1	66.7	65.4	66.8	66.9	65.5	66.9	64.8	67.7	67.8	70.2	69.8	70.9
	Mamba-2.8B	62.4	68.5	67.7	66.3	66.2	66.7	65.9	66.9	66.9	67.1	69.4	71.7	70.2	70.3	70.5	69.3	70.4	68.0	71.9	72.0	73.4	71.1	72.3
	Pythia-2.8B	59.5	65.0	65.2	62.8	62.9	63.1	62.6	63.3	59.6	59.8	61.1	63.5	62.2	60.9	60.9	64.3	66.6	63.2	66.3	66.4	66.8	66.4	66.9
	Pythia-6.9B	60.5	67.6	67.5	65.1	65.1	65.6	64.8	66.1	63.1	63.5	65.1	67.5	65.7	69.8	69.8	69.5	69.2	69.2	69.8	69.8	72.4	71.2	72.7
	Pythia-12B	61.4	67.8	67.1	65.8	65.8	66.2	65.6	67.7	63.9	64.0	65.1	67.2	66.0	70.7	70.7	70.8	70.5	68.9	71.8	72.0	73.5	71.9	73.5
	Llama-1.8B	60.6	69.7	68.3	67.8	67.7	68.7	67.8	69.1	62.6	62.6	64.9	62.2	64.1	71.5	71.6	71.2	71.1	70.9	83.8	83.9	88.4	89.1	89.1
	Llama-3.0B	59.0	71.8	72.2	70.3	70.3	71.0	70.3	71.2	71.9	72.0	70.4	66.0	68.0	73.7	73.7	72.5	73.8	71.6	82.7	82.8	85.9	87.4	87.4
	OPT-66B	58.9	67.3	67.7	65.5	65.5	66.7	65.6	67.6	66.9	67.1	68.6	70.3	69.0	70.6	70.7	70.6	71.3	70.3	71.1	71.3	72.9	71.9	72.0
	GPT-NeoX-20B	68.0	72.3	71.6	70.7	70.8	70.7	70.4	71.0	68.3	68.6	69.5	69.3	69.6	75.6	75.8	74.5	75.7	72.6	75.4	75.5	75.7	73.9	75.1
	Average	61.0	68.4	68.0	66.4	66.4	66.9	66.2	67.4	65.0	65.2	66.5	67.2	66.7	70.6	70.7	69.5	70.7	68.5	73.4	73.5	75.8	74.7	75.8

Table 6: AUC-ROC on WikiMIA benchmark under paraphrased setting. w/ LPDR utilizes linear weights for reweighting, w/ EPDR utilizes exponential weights for reweighting, w/ PPDR utilizes polynomial weights for reweighting. †Neighbor results are from Zhang et al. (2025b).

Length	Models	Lowerrace	Zib	†Neighbor	Loss	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Ref	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Min-f%	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Min-f% +†	w/ LPDR (CAMA slope)	w/ LPDR	w/ EPDR	w/ PPDR
32	Mamba-1.4B	60.6	62.3	63.6	61.3	61.5	61.8	61.3	61.5	62.3	62.2	62.3	62.7	62.4	62.9	62.9	63.4	63.4	65.7	65.9	66.3	65.3	66.2	
	Mamba-2.8B	63.5	64.8	66.3	64.5	64.5	64.8	64.3	64.6	66.6	66.2	66.5	66.7	66.2	65.3	65.2	65.4	65.2	64.9	67.3	67.5	67.5	66.6	67.4
	Pythia-2.8B	60.3	62.5	64.5	61.6	61.6	61.8	61.4	61.5	61.2	61.1	61.2	62.3	60.9	60.9	60.8	61.1	60.8	60.6	61.0	61.3	61.5	61.7	61.0
	Pythia-6.9B	61.7	64.2	65.5	64.1	64.1	64.2	63.9	64.1	63.5	63.3	63.5	64.4	63.1	65.1	65.0	65.1	64.9	64.4	67.6	67.7	67.7	66.6	67.3
	Pythia-12B	64.5	65.9	66.6	65.4	65.4	65.6	64.7	64.9	66.0	64.5	64.5	64.5	64.5	67.2	67.0	67.2	66.9	66.2	69.4	69.6	69.4	68.1	69.1
	Llama-1.8B	63.2	68.3	65.0	68.0	68.0	68.2	68.0	68.2	56.2	56.0	56.1	54.9	55.9	66.2	66.2	66.3	65.6	62.7	82.8	84.1	84.3	83.3	83.7
	Llama-3.0B	61.3	70.4	66.3	70.2	70.2	70.3	70.1	70.2	62.4	62.5	62.4	61.3	62.2	68.5	68.5	68.4	68.3	67.7	81.2	81.2	82.4	82.6	81.6
	OPT-66B	62.3	65.3	66.7	65.3	65.1	65.0	65.1	65.3	68.0	68.0	68.0	68.9	68.1	65.8	65.6	65.8	65.5	64.9	67.0	67.0	67.1	66.0	66.9
	GPT-NeoX-20B	66.9	68.5	68.3	68.3	68.4	68.4	68.5	66.7	66.6	66.1	66.0	66.1	69.6	69.4	69.5	69.3	68.0	69.7	69.5	69.7	68.4	69.2	69.6
	Average	62.7	65.8	65.9	65.5	65.4	65.6	65.3	65.5	63.5	63.3	63.5	63.7	63.3	65.7	65.6	65.6	65.6	65.1	70.3	70.6	69.9	70.3	
64	Mamba-1.4B	57.0	59.1	60.6	56.4	56.7	59.5	56.4	60.5	59.6	59.7	60.8	61.1	60.2	58.0	58.3	61.8	58.4	61.6	62.2	62.4	65.5	63.5	65.8
	Mamba-2.8B	62.0	61.9	63.7	59.8	60.1	62.9	59.7	63.6	64.5	64.7	66.3	66.6	65.2	62.4	62.7	65.1	62.7	64.0	64.9	65.1	67.9	66.0	67.5
	Pythia-2.8B	56.1	59.0	59.6	56.5	56.8	59.3	56.5	60.3	59.2	59.3	60.8	62.0	60.1	56.7	57.0	61.6	57.0	61.3	57.7	57.9	62.0	59.8	62.1
	Pythia-6.9B	57.7	61.6	63.1	59.3	59.6	62.4	59.3	62.9	62.9	63.1	64.0	65.6	63.1	61.1	61.4	65.1	61.5	63.7	64.2	64.5	68.3	65.6	68.0
	Pythia-12B	59.2	62.1	62.8	60.0	60.2	63.0	60.0	63.6	63.2	63.3	64.5	66.1	63.7	62.5	62.8	66.1	62.7	64.8	65.1	65.3	68.2	65.5	67.5
	Llama-1.8B	61.0	65.3	64.7	63.1	63.3	66.2	63.4	66.9	60.9	60.8	57.9	57.7	55.2	63.5	63.7	67.2	64.1	66.2	78.8	78.9	84.3	83.5	85.0
	Llama-3.0B	59.8	67.4	66.7	65.5	65.7	68.4	65.7	69.9	65														

Table 8: TPR on WikiMIA benchmark under paraphrased setting. w/ LPDR utilizes linear weights for reweighting, w/ EPDR utilizes exponential weights for reweighting, w/ PPDR utilizes polynomial weights for reweighting. †Neighbor results are from Zhang et al. (2025b).

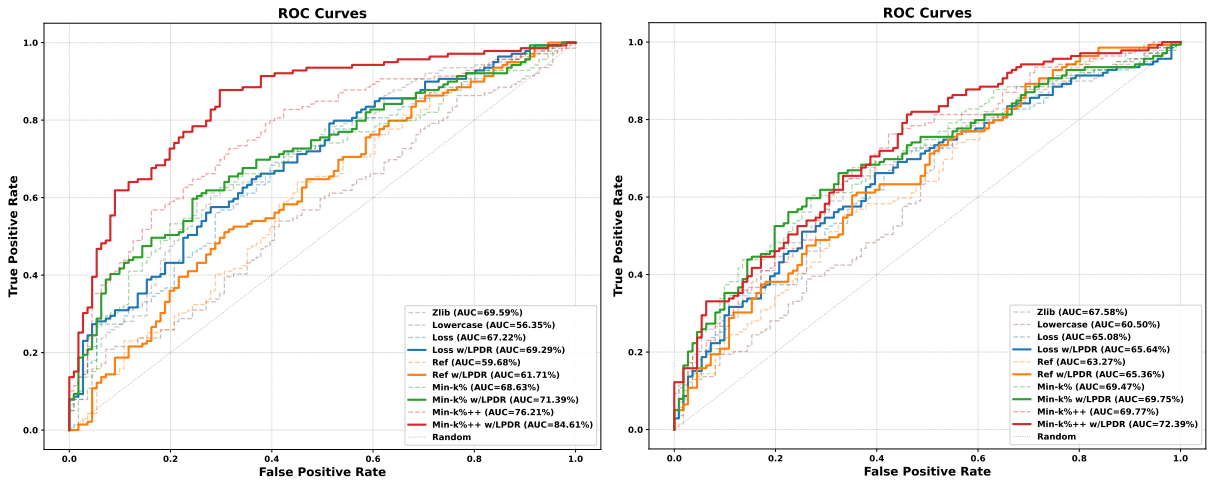
Length	Models	Neighbor			Loss			Ref			Min-k%			Min-k% ++										
		Lowercase	Zlib	Neighbor	w/ LPDR (CAMIA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Ref	w/ LPDR (CAMIA slope)	w/ LPDR	w/ EPDR	w/ PPDR	Min-k%	w/ LPDR (CAMIA slope)	w/ LPDR	w/ EPDR	w/ PPDR						
32	Mamba-1.4B	13.2	13.2	7.2	14.2	15.0	16.0	12.9	15.5	5.9	6.5	5.9	9.6	6.2	11.9	13.7	14.0	12.4	15.0	7.8	7.5	11.4	11.6	9.8
	Mamba-2.8B	15.0	12.7	9.3	16.5	16.3	17.8	16.3	17.3	10.1	9.3	9.3	11.9	10.6	19.9	20.4	20.2	20.4	15.2	11.1	11.6	12.9	11.1	9.8
	Pythia-2.8B	11.6	14.5	8.5	14.2	15.0	15.5	15.0	14.2	7.2	6.7	7.2	12.9	7.5	16.3	16.8	16.8	15.2	15.8	10.9	10.6	10.6	10.9	11.4
	Pythia-6.9B	11.9	12.7	9.6	15.0	15.0	14.0	14.2	14.5	6.2	5.9	5.7	12.9	5.7	21.7	23.0	22.0	22.7	18.1	14.5	15.0	16.5	15.5	15.0
	Pythia-12B	16.5	15.5	9.8	17.3	17.6	16.8	17.8	18.9	8.0	9.3	8.3	10.3	8.3	19.9	20.7	20.4	20.7	18.3	15.5	15.0	13.4	14.0	13.4
	Llama-13B	9.6	15.0	8.5	16.3	16.3	16.5	16.0	16.8	5.4	5.9	5.9	4.4	6.2	14.2	14.2	15.2	14.5	15.5	33.9	34.4	37.7	33.9	35.9
	Llama-30B	13.2	15.2	9.3	14.7	15.2	17.6	15.5	16.8	8.3	8.0	7.5	5.9	7.8	18.1	18.3	17.8	17.8	19.4	25.8	26.6	29.2	31.8	24.8
	OPT-66B	12.9	16.8	12.1	15.2	15.2	17.6	15.5	15.5	9.3	9.3	9.6	10.9	8.5	16.8	16.0	18.3	18.1	14.7	16.3	16.0	16.3	17.1	13.4
	GPT-NeoX-20B	14.5	19.6	15.2	17.6	19.4	19.4	18.9	19.9	15.2	14.2	15.0	14.2	14.0	19.1	18.6	19.4	19.4	19.6	13.4	13.2	13.2	11.9	11.1
	Average	13.1	15.0	9.9	15.7	16.8	15.8	16.5	15.1	15.3	10.3	10.3	11.5	18.2	17.9	18.9	16.6	17.9	17.5	16.1	16.1	16.1	16.1	16.1
64	Mamba-1.4B	9.5	15.1	9.5	8.1	8.1	12.0	9.9	12.7	8.1	7.4	9.2	10.9	6.3	7.7	8.1	14.8	9.2	16.2	7.0	7.0	8.5	6.3	10.2
	Mamba-2.8B	14.8	14.8	18.3	12.3	12.3	13.0	13.0	15.8	11.3	10.9	12.3	14.8	9.5	11.6	11.6	19.7	12.3	18.3	9.5	9.5	14.1	12.7	16.5
	Pythia-2.8B	11.3	16.5	11.3	9.5	11.3	13.4	12.3	15.1	13.0	12.7	9.9	12.7	7.4	9.9	9.9	18.3	11.3	18.7	8.1	8.1	9.9	7.4	13.4
	Pythia-6.9B	11.3	15.8	12.7	10.6	10.9	13.0	12.7	14.4	16.2	14.4	7.7	15.5	5.6	12.7	12.0	18.3	13.0	17.3	10.2	10.6	12.7	10.9	16.5
	Pythia-12B	13.4	16.2	10.6	11.6	11.3	14.1	11.3	12.0	14.4	14.4	10.2	14.4	9.2	12.7	13.7	20.8	16.9	17.6	10.9	11.6	18.7	15.8	16.9
	Llama-13B	13.7	13.4	14.4	12.0	12.0	14.1	12.3	14.8	4.6	4.6	6.0	6.0	5.6	13.4	13.0	21.1	12.3	14.4	23.2	23.6	31.3	31.3	29.2
	Llama-30B	7.7	16.9	11.4	13.4	13.4	13.0	13.7	15.8	8.5	7.7	7.4	6.7	6.7	14.8	14.1	17.3	10.6	16.5	20.8	21.5	35.6	30.6	35.6
	OPT-66B	12.3	14.8	13.7	13.4	13.4	14.8	15.1	14.1	13.7	13.4	8.5	12.7	6.7	13.0	13.7	22.2	14.8	19.7	11.6	12.3	15.5	11.3	13.4
	GPT-NeoX-20B	13.4	19.0	18.3	14.8	15.8	19.4	15.5	18.7	15.5	15.5	14.4	15.5	12.0	15.1	15.1	20.8	14.4	17.6	10.6	10.6	13.4	15.1	13.0
	Average	11.9	15.8	13.4	11.7	12.1	14.1	12.9	14.8	11.7	11.1	9.5	12.1	7.7	12.3	12.4	19.2	12.8	17.4	12.4	12.8	17.7	15.7	18.3
128	Mamba-1.4B	11.5	17.3	13.7	13.7	13.7	18.7	10.1	14.4	11.5	11.5	12.2	16.5	12.2	5.0	5.8	20.9	10.8	23.7	6.5	6.5	9.4	13.7	8.6
	Mamba-2.8B	15.1	20.1	17.3	16.5	15.1	22.3	16.5	15.8	10.8	10.1	15.1	18.0	20.9	12.2	11.5	30.2	18.7	20.1	13.0	12.2	16.5	15.1	15.1
	Pythia-2.8B	8.6	16.5	12.2	14.4	14.4	11.5	7.2	13.0	7.2	7.2	10.1	11.5	5.8	12.2	12.2	21.6	15.8	18.0	7.9	8.6	10.1	13.0	8.6
	Pythia-6.9B	11.5	20.9	17.3	16.5	16.5	15.1	18.0	13.0	8.6	9.4	10.8	18.0	8.6	16.5	16.5	23.0	16.5	21.6	18.0	18.0	18.7	14.4	17.3
	Pythia-12B	12.2	19.4	10.1	19.4	19.4	14.4	16.5	12.2	8.6	9.4	13.0	10.1	13.0	18.7	18.0	33.1	10.8	29.5	9.4	10.1	15.1	20.1	18.7
	Llama-13B	15.8	21.6	13.7	18.0	18.0	23.7	21.6	25.9	4.3	4.3	7.9	6.5	7.9	15.1	14.4	30.9	20.9	33.1	35.3	35.3	46.8	45.3	49.6
	Llama-30B	11.5	19.4	14.4	18.0	17.3	25.9	15.1	23.0	17.3	17.3	13.0	11.5	10.1	17.3	17.3	33.8	16.5	37.4	21.6	22.3	34.5	49.6	46.8
	OPT-66B	11.5	18.7	12.9	18.7	18.7	15.8	18.0	15.8	16.5	17.3	17.3	13.0	12.2	15.1	15.8	31.7	21.6	23.7	13.7	13.7	15.8	22.3	18.0
	GPT-NeoX-20B	15.1	22.3	18.7	20.1	20.0	25.9	20.1	24.5	18.0	18.0	12.2	11.5	13.0	23.0	23.0	35.4	23.7	31.7	20.0	20.0	9.4	20.9	12.2
	Average	12.6	19.6	14.5	17.3	17.1	19.9	15.9	17.5	11.4	11.6	12.4	12.9	11.5	15.0	14.9	28.6	17.3	26.5	16.2	16.4	19.6	23.8	21.7

Table 9: Accuracy on WikiMIA benchmark. Ori. is original setting, Para. is paraphrased setting. Results are shown as 'baseline / w/ LPDR'.

Length	Models	Loss		Ref		Min-k%		Min-k% ++	
		Ori. / w/ LPDR	Para. / w/ LPDR	Ori. / w/ LPDR	Para. / w/ LPDR	Ori. / w/ LPDR	Para. / w/ LPDR	Ori. / w/ LPDR	Para. / w/ LPDR
32	Mamba-1.4B	59.0 / 59.3	59.5 / 59.7	59.9 / 59.9	60.1 / 60.3	60.7 / 61.9	62.2 / 62.5	63.1 / 64.8	62.6 / 63.1
	Mamba-2.8B	61.3 / 61.7	61.5 / 61.3	64.8 / 64.6	63.0 / 63.1	62.4 / 63.3	62.5 / 62.5	65.1 / 65.3	64.8 / 65.1
	Pythia-2.8B	60.2 / 60.3	59.3 / 59.8	58.8 / 59.0	59.0 / 59.0	59.4 / 60.1	58.6 / 59.5	60.4 / 61.9	59.1 / 59.3
	Pythia-6.9B	61.7 / 61.7	61.3 / 61.9	60.6 / 60.4	60.1 / 60.1	63.1 / 63.1	62.5 / 62.4	66.6 / 66.8	63.8 / 63.9
	Pythia-12B	63.1 / 63.1	62.8 / 63.4	61.1 / 61.6	61.2 / 61.9	64.9 / 64.9	64.8 / 65.5	67.9 / 68.0	65.2 / 65.3
	Llama-13B	64.3 / 64.0	64.8 / 64.9	57.1 / 56.8	56.6 / 57.0	63.0 / 63.4	63.8 / 63.0	77.4 / 78.0	76.2 / 77.2
	Llama-30B	65.3 / 65.6	65.7 / 65.9	60.2 / 61.0	59.1 / 59.8	64.4 / 65.7	65.1 / 65.3	77.2 / 77.7	75.6 / 76.9
	OPT-66B	62.8 / 63.3	62.6 / 63.1	65.2 / 64.9	64.8 / 64.7	65.2 / 64.9	62.6 / 62.8	66.5 / 66.8	64.0 / 64.3
	GPT-NeoX-20B	65.6 / 65.6	64.7 / 65.3	65.2 / 65.1	63.4 / 63.3	66.6 / 66.5	64.7 / 64.6	69.7 / 70.2	65.7 / 67.0
64	Mamba-1.4B	56.3 / 58.3	55.2 / 57.6	60.3 / 60.0	57.6 / 59.4	59.0 / 61.1	59.0 / 61.1	64.8 / 65.7	62.0 / 63.8
	Mamba-2.8B	59.6 / 61.4	57.4 / 60.1	62.0 / 64.0	62.0 / 64.4	61.8 / 63.7	61.1 / 61.3	66.1 / 66.4	64.4 / 66.1
	Pythia-2.8B	56.5 / 59.2	55.4 / 59.0	57.0 / 58.7	58.9 / 60.5	58.1 / 60.7	56.1 / 61.8	61.3 / 61.4	58.9 / 58.5
	Pythia-6.9B	58.9 / 61.3	57.6 / 61.3	58.9 / 61.8	60.7 / 62.5	62.7 / 63.7	59.2 / 62.5	67.2 / 68.3	61.3 / 63.1
	Pythia-12B	60.5 / 61.8	58.1 / 60.1	60.3 / 61.1	60.9 / 62.5	62.5 / 64.0	59.8 / 63.8	67.9 / 67.0	62.4 / 64.0
	Llama-13B	60.9 / 63.3	61.3 / 64.0	62.7 / 58.5	60.1 / 59.0	62.5 / 63.8	61.6 / 64.4	78.2 / 79.0	73.4 / 78.0
	Llama-30B	62.2 / 64.6	62.7 / 65.5	65.9 / 62.9	63.7 / 60.9	64.9 / 64.4	62.5 / 64.8	77.9 / 79.7	71.0 / 76.2
	OPT-66B	60.3 / 62.4	59.0 / 61.8	63.3 / 65.7	64.2 / 65.7	62.4 / 64.9	60.7 / 63.7	65.5 / 67.0	60.9 / 64.2
	GPT-NeoX-20B	63.1 / 64.9	60.9 / 63.3	64.6 / 65.3	62.7 / 64.0	69.4 / 66.8	61.8 / 64.2	71.8 / 70.8	64.4 / 64.8
128	Mamba-1.4B	61.2 / 60.0	62.4 / 63.6	61.2 / 64.4	59.2 / 64.4	65.2 / 62.8	64.0 / 60.8	64.4 / 70.0	62.8 / 66.0
	Mamba-2.8B	63.6 / 64.4	64.0 / 64.4	64.8 / 65.6	63.2 / 65.6	67.6 / 64.4	65.2 / 66.4	66.4 / 69.2	66.4 / 70.4
	Pythia-2.8B	62.4 / 59.6	62.4 / 63.6	57.6 / 58.8	56.0 / 58.0	64.4 / 61.2	63.2 / 59.6	61.2 / 65.2	61.2 / 65.6
	Pythia-6.9B	64.0 / 63.6	62.8 / 67.2	60.4 / 61.6	62.0 / 60.4	66.0 / 64.8	63.6 / 64.8	68.0 / 69.2	62.0 / 69.6
	Pythia-12B	64.0 / 64.8	63.2 / 64.8	60.4 / 61.2	62.0 / 63.6	67.2 / 66.0	67.2 / 64.0	65.6 / 70.0	64.4 / 68.0
	Llama-13B	64.4 / 65.2	65.2 / 66.4	59.2 / 63.2	60.0 / 59.2	67.6 / 67.2	66.4 / 67.2	76.8 / 81.6	70.8 / 79.6
	Llama-30B	66.8 / 68.0	67.2 / 68.8	70.0 / 67.6	66.4 / 64.8	68.8 / 66.0	67.2 / 67.6	78.0 / 78.0	70.8 / 74.0
	OPT-66B	63.2 / 62.8	60.8 / 63.6	66.0 / 62.4	66.0 / 67.6	67.2 / 68.4	64.8 / 61.6	65.6 / 70.0	64.0 / 66.4
	GPT-NeoX-20B	68.8 / 68.4	66.4 / 68.8	66.8 / 68.4	66.4 / 68.8	70.8 / 69.6	68.0 / 68.4	69.6 / 72.0	65.6 / 70.4

Table 10: F1-Score on WikiMIA benchmark. *Ori.* is original setting, *Para.* is paraphrased setting. Results are shown as ‘baseline / w/ LPDR’.

Length	Models	Loss		Ref		Min-k%		Min-k% ++	
		<i>Ori.</i> / w/ LPDR	<i>Para.</i> / w/ LPDR	<i>Ori.</i> / w/ LPDR	<i>Para.</i> / w/ LPDR	<i>Ori.</i> / w/ LPDR	<i>Para.</i> / w/ LPDR	<i>Ori.</i> / w/ LPDR	<i>Para.</i> / w/ LPDR
32	Mamba-1.4B	49.7 / 51.5	50.5 / 50.6	60.9 / 60.8	64.2 / 64.3	55.9 / 55.8	60.4 / 60.5	65.8 / 64.1	68.0 / 66.0
	Mamba-2.8B	64.5 / 64.7	63.8 / 52.8	65.1 / 65.2	69.2 / 69.6	63.4 / 56.2	60.8 / 62.5	70.4 / 70.4	66.4 / 67.3
	Pythia-2.8B	57.7 / 56.1	61.4 / 51.4	63.7 / 63.9	64.4 / 63.7	58.4 / 60.6	59.0 / 41.2	65.5 / 63.8	61.1 / 61.2
	Pythia-6.9B	62.4 / 62.0	54.8 / 56.7	63.9 / 59.1	64.6 / 64.7	64.7 / 57.6	59.3 / 56.4	67.7 / 67.2	62.5 / 63.4
	Pythia-12B	57.8 / 62.8	59.2 / 57.9	66.9 / 61.7	63.9 / 67.1	63.9 / 65.0	62.4 / 62.9	68.0 / 67.6	62.5 / 61.6
	Llama-13B	66.1 / 65.2	65.3 / 64.3	64.7 / 64.9	65.8 / 66.1	59.2 / 60.6	64.6 / 63.8	79.8 / 80.6	77.9 / 80.4
	Llama-30B	67.9 / 67.9	67.3 / 67.1	64.4 / 58.9	61.7 / 59.7	65.6 / 66.0	62.7 / 63.2	76.8 / 78.4	76.6 / 78.3
	OPT-66B	59.4 / 58.6	59.8 / 60.1	68.5 / 69.4	67.4 / 67.4	60.1 / 60.6	61.2 / 61.6	67.0 / 66.2	60.5 / 60.0
	GPT-NeoX-20B	63.0 / 62.7	59.2 / 58.3	67.9 / 67.9	67.5 / 67.0	68.8 / 63.9	64.2 / 60.9	68.5 / 69.8	61.9 / 64.8
64	Mamba-1.4B	52.5 / 54.4	38.2 / 47.0	69.4 / 66.7	59.4 / 63.6	55.1 / 53.2	56.5 / 58.4	67.1 / 69.7	68.0 / 69.2
	Mamba-2.8B	63.3 / 61.2	44.6 / 55.4	64.4 / 71.0	64.1 / 69.6	61.0 / 63.2	55.6 / 52.5	68.4 / 70.5	70.3 / 66.9
	Pythia-2.8B	44.6 / 58.8	36.6 / 54.7	50.3 / 63.3	66.3 / 64.6	47.8 / 55.2	48.7 / 60.0	60.4 / 62.7	69.7 / 56.5
	Pythia-6.9B	61.9 / 59.9	47.2 / 60.7	54.0 / 68.3	64.2 / 65.5	56.7 / 58.0	48.7 / 55.4	68.9 / 69.3	60.5 / 59.2
	Pythia-12B	65.1 / 62.6	48.1 / 48.8	67.0 / 63.6	68.8 / 67.1	58.8 / 58.1	50.5 / 60.3	69.1 / 65.8	63.8 / 66.0
	Llama-13B	56.9 / 63.5	58.2 / 61.2	66.4 / 61.5	65.9 / 61.6	60.4 / 62.5	55.0 / 61.2	78.4 / 80.8	76.0 / 79.7
	Llama-30B	56.1 / 67.0	59.3 / 65.7	68.5 / 67.6	64.6 / 64.2	63.5 / 58.8	56.5 / 61.1	78.9 / 80.8	73.3 / 79.0
	OPT-66B	64.1 / 60.2	49.1 / 56.4	59.6 / 65.9	65.8 / 66.7	56.4 / 64.3	53.0 / 60.0	67.0 / 63.2	65.4 / 66.0
	GPT-NeoX-20B	61.4 / 65.2	56.4 / 55.1	68.0 / 69.5	62.3 / 63.8	68.8 / 68.9	54.7 / 65.1	73.8 / 74.7	69.3 / 67.7
128	Mamba-1.4B	62.8 / 54.5	67.1 / 65.9	63.7 / 70.2	62.2 / 72.9	64.8 / 65.9	62.2 / 55.9	64.0 / 74.6	71.0 / 68.4
	Mamba-2.8B	65.7 / 62.8	65.4 / 61.1	68.3 / 68.8	64.1 / 68.4	71.2 / 57.8	63.6 / 66.1	68.4 / 72.2	67.9 / 75.8
	Pythia-2.8B	66.2 / 54.3	65.9 / 68.1	57.9 / 61.1	52.2 / 52.1	62.1 / 54.9	64.3 / 55.1	54.5 / 71.1	63.1 / 72.4
	Pythia-6.9B	68.5 / 66.9	64.4 / 71.3	60.9 / 63.9	67.8 / 56.8	66.4 / 62.7	65.7 / 62.4	72.6 / 74.6	63.3 / 74.5
	Pythia-12B	69.0 / 67.4	65.7 / 67.2	59.3 / 63.7	67.6 / 67.1	68.0 / 65.3	72.1 / 60.5	65.0 / 76.5	65.6 / 73.5
	Llama-13B	66.7 / 69.5	69.5 / 72.0	54.5 / 66.2	64.0 / 58.9	66.7 / 65.0	66.4 / 69.6	81.8 / 84.8	73.5 / 81.9
	Llama-30B	70.3 / 72.2	70.7 / 73.1	72.7 / 73.1	66.7 / 66.4	68.3 / 60.8	67.2 / 70.1	82.8 / 80.3	73.8 / 73.7
	OPT-66B	66.7 / 61.1	58.8 / 60.9	70.6 / 59.8	70.0 / 71.4	70.3 / 72.5	64.8 / 50.5	63.9 / 74.0	64.3 / 72.0
	GPT-NeoX-20B	71.1 / 71.1	69.1 / 70.9	70.9 / 74.3	70.8 / 71.5	72.0 / 69.1	67.2 / 63.9	70.5 / 74.8	67.9 / 75.5



(a) Llama-13B (WikiMIA-128 Para.)

(b) Pythia-6.9B (WikiMIA-128 Ori.)

Figure 6: ROC curve comparison for various baseline methods and their PDR-enhanced versions on the WikiMIA-128 benchmark.

Table 11: AUROC results on WikiMIA benchmark, compare with FSD

Dataset	Method	GPT-J-6B		OPT-6.7B		Pythia-6.9B		LLaMA-7B		NeoX-20B		Average	
		Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD
WikiMIA	Min-k%	67.9	93.4	62.5	89.4	66.7	91.9	65.4	88.4	73.4	90.1	67.2	90.6
	w/ LPDR	68.1	94.0	62.9	89.6	67.0	92.6	66.7	88.8	73.5	91.0	67.6	91.2
	w/ EPDR	67.1	94.1	62.1	90.0	66.0	92.8	65.8	89.1	72.1	91.3	66.6	91.4
	w/ PPDR	67.9	93.8	62.9	89.5	66.9	92.2	66.6	88.9	73.3	90.7	67.5	91.0
	Min-k%++	67.6	81.6	63.0	83.3	68.1	81.5	79.9	91.1	74.4	77.8	70.6	83.1
	w/ LPDR	68.3	84.8	63.6	84.5	69.4	85.0	80.8	92.1	74.8	81.7	71.4	85.6
	w/ EPDR	68.2	83.7	63.3	84.5	69.2	83.6	80.8	91.8	74.8	81.2	71.2	85.0
	w/ PPDR	68.3	85.5	63.7	84.6	69.5	85.8	80.7	92.1	74.6	82.4	71.4	86.1

Table 12: TPR results on WikiMIA benchmark, compare with FSD

Dataset	Method	GPT-J-6B		OPT-6.7B		Pythia-6.9B		LLaMA-7B		NeoX-20B		Average	
		Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD	Base	w/ FSD
WikiMIA	Min-k%	17.2	55.6	13.9	40.6	17.2	57.6	14.7	32.6	24.7	36.2	17.5	44.5
	w/ LPDR	18.0	61.9	16.4	41.9	18.2	61.1	16.4	33.7	27.7	52.9	19.3	50.3
	w/ EPDR	16.5	60.1	13.5	45.4	16.5	59.8	17.0	33.6	21.5	50.8	17.0	49.9
	w/ PPDR	17.9	59.9	15.4	39.9	19.9	61.9	18.5	31.6	27.0	50.3	19.7	48.7
	Min-k%++	15.9	24.2	11.7	29.0	19.0	25.0	20.4	39.1	17.5	10.5	16.9	25.6
	w/ LPDR	18.9	32.2	12.9	35.7	19.0	37.4	22.2	48.4	19.7	15.5	18.5	33.9
	w/ EPDR	19.0	39.6	12.2	32.4	19.4	34.9	20.2	46.4	19.0	18.7	18.0	34.4
	w/ PPDR	18.9	34.9	12.9	36.7	17.0	40.4	24.0	51.6	19.7	17.4	18.5	36.2

replicate: $\delta = \text{AUROC}_{\text{PDR}} - \text{AUROC}_{\text{Baseline}}$. A one-sided P-value was subsequently derived by calculating the proportion of replicates where this difference was not positive ($\delta \leq 0$). This P-value directly tests the null hypothesis $H_0 : \text{AUROC}_{\text{PDR}} \leq \text{AUROC}_{\text{Baseline}}$. A P-value less than 0.05 is considered to indicate a statistically significant improvement.

The analysis reveals a clear trend: the effectiveness of PDR is strongly correlated with sequence length. For short sequences (e.g., 32 tokens), the performance gains are marginal and not always statistically significant. However, as the sequence length increases to 64 and 128, PDR’s improvements become both substantial and statistically significant (p-value < 0.05) for most baselines. For instance, on Pythia-6.9B with length 128, PDR boosts the AUROC of Min-k%++ from 65.9 to 72.2 (p-value < 0.001). This demonstrates that the positional prior becomes a more robust and discriminative signal as more context becomes available in longer sequences, confirming that the observed gains are not due to random noise.

E FSD Results

This section presents the detailed results of combining our method with the Finetuning-based Score Difference (FSD) framework on the WikiMIA dataset. Please refer to Table 11 and Table 12 for the full experimental results.

F Mimir Results

This section presents the complete AUROC and TPR results on the challenging MIMIR benchmark, which is known for its minimal distribution shift and increased difficulty compared to WikiMIA. The overall average results for five Pythia models are summarized in Table 13. The results in Table 14 and Table 15 comprehensively demonstrate the performance of our proposed PDR methods (LPDR,

EPDR, PPDR) and all baselines on MIMIR, across different models and sub datasets.

Table 13: AUROC scores of various MIA methods over five Pythia models on the Mimir dataset. Pub, Wiki, and Hack denote Pubmed Central, Wikipedia (en) and HackerNews. Avg* scores are computed by excluding Arxiv and HackerNews. †Neighbor results are from Zhang et al. (2025b), induces significant extra computational cost than others (25× in this case), for which reason we don’t run on the 12B model.

Method	Wiki	Pile-CC	Pub	Math	GitHub	ArXiv	Hack	Avg*
Lowercase	52.2	49.3	51.1	48.9	71.1	51.5	50.9	54.5
Zlib	52.7	50.4	50.4	48.1	71.9	51.4	50.8	54.7
†Neighbor	51.9	50.1	49.2	47.4	69.3	51.5	51.5	53.6
Loss	51.9	50.3	50.3	48.5	70.8	52.1	51.2	54.4
w/ LPDR	52.8	50.7	50.3	48.6	70.9	51.8	51.3	54.6
Min-k%	51.8	50.7	50.9	49.2	70.9	52.3	52.4	54.7
w/ LPDR	54.2	51.2	51.0	49.5	71.0	51.4	51.5	55.4
Min-k%++	54.0	50.5	51.9	50.3	70.4	52.6	52.9	55.4
w/ LPDR	55.5	50.8	52.4	50.3	70.1	52.7	52.1	55.8

G Visualization of Weight Decay Functions

In this section, we visualize weight decay functions (Linear, Exponential, or Polynomial), with varying α . We use the following ranges:

- For **Linear** decay, the range for the coefficient α is: $\{0.1, 0.3, 0.5, 0.7, 1.0\}$
- For **Exponential** decay, the range for the coefficient α is: $\{0.002, 0.004, 0.006, 0.008, 0.01, 0.02, 0.04, 0.06, 0.08, 0.1\}$.
- For **Polynomial** decay, the range for the coefficient α is: $\{0.1, 0.3, 0.5, 0.7, 1.0, 1.2, 1.5, 1.8, 2.0\}$.

Figure 7 visualizes how the decay function becomes steeper as the hyperparameter (α) increases.

Table 14: AUROC results on the challenging MIMIR benchmark.[†]Neighbor results are from Zhang et al. (2025b), induces significant extra computational cost than others (25× in this case), for which reason we don't run on the 12B model.

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
Lowercase	50.1	51.3	51.7	53.5	54.3	67.2	70.3	71.3	72.9	73.7	47.8	48.6	49.5	50.1	50.6	49.5	50.4	51.5	51.5	52.7
Zlib	51.1	52.0	52.4	53.5	54.3	67.5	71.0	72.3	73.9	74.9	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	/
Loss	50.2	51.3	51.8	52.8	53.5	65.7	69.8	71.3	73.0	74.0	49.6	50.0	50.1	50.7	51.1	49.9	49.8	49.9	50.6	51.3
w/ LPDR (CAMIA slope)	50.3	51.3	51.8	52.8	53.5	65.8	69.8	71.3	73.0	74.0	49.6	50.0	50.1	50.7	51.1	49.9	49.8	49.9	50.6	51.3
w/ LPDR	51.2	52.0	52.6	53.7	54.4	66.0	70.0	71.5	73.0	73.9	49.9	50.4	50.5	51.0	51.4	50.1	49.9	50.0	50.6	51.1
w/ EPDR	50.5	51.5	51.9	53.0	53.7	65.7	69.9	71.3	73.0	74.0	49.6	50.0	50.1	50.7	51.0	49.7	49.6	49.7	50.3	50.9
w/ PPDR	50.4	51.4	51.9	52.9	53.6	66.0	70.0	71.5	73.2	74.2	49.6	50.1	50.2	50.8	51.1	49.9	49.8	49.9	50.6	51.3
Min- <i>k</i> %	48.8	51.0	51.7	53.1	54.2	65.7	70.0	71.4	73.3	74.2	50.1	50.5	50.5	51.2	51.5	50.3	50.3	50.5	51.2	52.3
w/ LPDR (CAMIA slope)	48.9	51.0	51.7	53.2	54.2	65.7	70.0	71.5	73.3	74.2	50.1	50.5	50.5	51.2	51.5	50.3	50.3	50.5	51.2	52.3
w/ LPDR	52.7	53.3	54.1	55.0	56.0	65.9	70.2	71.7	73.2	74.1	50.2	51.0	51.1	51.6	51.9	50.7	50.7	50.6	51.1	52.0
w/ EPDR	50.4	51.7	52.5	53.9	54.9	65.7	70.0	71.5	73.3	74.3	50.4	50.7	50.7	51.4	51.6	50.1	50.1	50.2	50.8	51.7
w/ PPDR	49.2	51.3	52.1	53.5	54.5	66.1	70.3	71.8	73.5	74.5	50.2	50.6	50.6	51.4	51.6	50.4	50.4	50.5	51.2	52.4
Min- <i>k</i> %++	49.2	53.1	53.8	56.1	57.9	64.7	69.6	70.9	72.8	74.2	49.7	50.0	49.8	51.2	51.8	50.2	50.8	51.5	52.8	54.0
w/ LPDR (CAMIA slope)	49.2	53.2	53.8	56.1	57.9	64.7	69.6	70.9	72.8	74.2	49.7	50.0	49.8	51.2	51.8	50.2	50.8	51.5	52.8	54.0
w/ LPDR	51.0	54.8	55.5	57.5	59.0	64.6	69.5	70.6	72.3	73.7	49.7	50.6	49.9	51.8	51.9	50.4	51.8	52.2	53.6	54.4
w/ EPDR	49.9	53.7	54.4	56.7	58.5	64.7	69.7	70.8	72.7	74.1	49.9	50.3	49.9	51.5	51.8	50.0	50.9	51.5	52.8	53.7
w/ PPDR	49.3	53.5	54.2	56.5	58.3	64.8	69.8	71.0	72.9	74.3	49.7	50.2	49.8	51.4	51.9	50.2	51.1	51.7	53.1	54.3
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
Lowercase	50.8	50.7	51.3	51.9	52.8	48.9	49.0	49.0	49.1	48.2	49.0	50.4	51.1	51.6	52.3	52.4	53.4	54.1	54.8	55.4
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	52.7	53.7	54.1	54.9	55.4
[†] 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	/
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	52.5	53.5	53.9	54.7	55.3
w/ LPDR (CAMIA slope)	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	54.9
w/ LPDR	50.4	51.1	51.5	52.7	53.2	48.7	48.6	48.4	48.5	48.6	49.7	50.6	51.3	52.2	52.6	52.7	53.7	54.1	54.9	55.4
w/ EPDR	50.7	51.2	51.6	52.5	53.0	48.5	48.3	48.2	48.3	48.3	49.5	50.3	51.0	51.6	52.1	52.4	53.4	53.8	54.6	55.1
w/ PPDR	50.9	51.4	51.9	52.9	53.4	48.8	48.6	48.4	48.5	48.5	49.4	50.5	51.3	52.1	52.8	52.6	53.6	54.0	54.8	55.4
Min- <i>k</i> %	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	52.4	53.7	54.2	55.2	56.0
w/ LPDR (CAMIA slope)	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	55.7
w/ LPDR	49.2	50.8	51.2	52.5	53.2	49.5	49.7	49.3	49.5	49.5	50.3	50.7	51.3	52.4	52.8	53.0	54.3	54.6	55.5	56.1
w/ EPDR	49.9	51.0	51.5	52.8	53.5	49.3	49.4	49.1	49.2	49.2	50.5	51.0	51.9	52.9	53.3	52.6	53.8	54.3	55.2	55.9
w/ PPDR	50.2	51.4	52.1	53.4	54.3	49.3	49.4	49.2	49.3	49.3	50.5	51.1	52.3	53.5	54.4	52.6	53.9	54.4	55.4	56.1
Min- <i>k</i> %++	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	52.2	54.1	54.9	56.2	57.4
w/ LPDR (CAMIA slope)	49.3	50.8	53.0	53.6	56.3	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	57.2
w/ LPDR	50.0	51.1	52.4	53.9	56.1	50.1	50.3	50.2	50.4	50.5	50.7	50.9	51.7	53.0	54.3	52.6	54.7	55.1	56.6	57.6
w/ EPDR	49.6	50.8	52.6	53.4	55.7	49.8	50.0	49.9	50.2	50.2	50.6	50.9	52.0	53.3	54.7	52.3	54.2	54.9	56.2	57.3
w/ PPDR	49.5	51.0	53.1	53.8	56.3	50.2	50.2	50.2	50.6	50.5	50.5	51.2	52.5	54.0	55.6	52.3	54.3	55.0	56.4	57.6

Table 15: TPR results on the challenging MIMIR benchmark. [†]Neighbor results are from Zhang et al. (2025b), induces significant extra computational cost than others (25× in this case), for which reason we don’t run on the 12B model.

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
Lowercase	4.6	4.5	4.9	5.2	5.6	24.4	32.2	34.3	38.1	38.6	3.4	5.3	5.3	6.2	6.4	3.5	5	5.3	6	5.2
Zlib	4.2	5.7	5.9	6.3	6.8	25.1	32.8	36.2	40.1	40.8	4	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	/
Loss	4.2	4.7	4.7	5.1	5	22.6	32.1	33.6	38.5	40.7	3.1	5	4.8	4.9	5.1	4	4.4	4.3	4.9	5
w/ LPDR (CAMIA slope)	4.5	4.7	4.7	4.9	5	22.6	32.1	33.6	38.5	40.7	3.0	4.9	4.8	4.9	5.3	4.0	4.5	4.3	4.8	5.0
w/ LPDR	4.6	4.6	5.4	5.7	6	21	30.8	32.4	34.7	36.4	2.4	3.4	4	4.8	5.8	4.1	4.6	4.5	4.9	5.5
w/ EPDR	4.5	4.8	5.1	5.5	5.7	21.5	31.6	32.6	36.6	38.7	3.2	4.5	4.6	4.7	5.2	3.7	4.6	4.6	4.8	4.9
w/ PPDR	4.2	4.7	4.9	4.9	5.3	22.7	32.1	35	38.8	39.8	3	4.3	4.7	5.3	6.1	4.1	4.2	4.1	4.4	5.2
Min- <i>k</i> %	4.8	5.6	5	6.1	5.8	22.6	31.5	34	39	40.7	3.5	4.5	4.8	5	4.8	4.7	4.6	4.5	5.1	4.9
w/ LPDR (CAMIA slope)	4.8	5.6	5.0	6.1	5.8	22.5	31.5	34.0	39.0	40.7	3.6	4.4	4.8	5.0	4.8	4.7	4.6	4.5	5.1	5.1
w/ LPDR	5.5	6.4	6.2	6.2	7.2	21.4	30.9	32.7	36	36.7	2.3	2.9	3.1	2.4	3.6	6	5.6	5.6	5.7	4.8
w/ EPDR	5	5.9	5.9	5.8	6.8	20.4	31	32.9	37.1	38.4	3.3	4.4	4.4	4.1	5.2	4	4.6	4.9	5.5	5.8
w/ PPDR	5.3	5.3	5.4	5.6	5.6	22.9	31.9	34.6	39.1	40.2	3.8	4.5	4.2	4.7	5.2	5.3	5.2	4.7	5.2	6.4
Min- <i>k</i> %++	5.2	5.3	5.9	7	7.8	25.2	33	34.2	38.2	40.1	5	3.7	3.7	4.8	4.6	4.8	6.1	4.8	5.6	6.4
w/ LPDR (CAMIA slope)	5.2	5.3	5.9	7.0	7.8	25.2	33.0	34.2	38.1	40.1	5.0	3.7	3.6	4.8	4.6	4.8	5.9	4.8	5.6	6.3
w/ LPDR	4.6	6	6.3	7.5	7.9	22.6	30.6	32.7	35	39.7	4.7	2.9	3.4	4	5	4.2	6.4	5.5	7.8	6.2
w/ EPDR	5.1	5.9	6.6	7.8	8.2	24	33	33.5	37.4	40.7	4.6	3.4	3.6	4.3	4.9	4.9	5.4	5.4	6.8	5.8
w/ PPDR	5.2	5	5.5	7.7	7.5	26.2	34.2	34.9	38.8	40	5.2	3.5	3.3	4.3	4.6	5.1	6.4	5.1	5.8	6.7

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
Lowercase	5.1	4.7	5.4	5.6	5.2	5.6	6.2	5.5	6.8	5.8	5.2	5.2	6.3	6.6	6.4	7.8	9.7	10.1	11.3	11.1
Zlib	2.9	4.3	4.1	4.6	4.7	4.1	5	4.6	4.3	4.3	5	5.5	5.8	5.6	5.8	7.4	9.4	10.0	11.0	11.3
[†] 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	/
Loss	4	4.8	4.6	5.4	5.6	3.8	4.3	4.1	4.1	4	5	4.8	5.5	5.9	6.8	7.0	9.2	9.4	10.5	10.9
w/ LPDR (CAMIA slope)	4.0	4.8	4.6	5.4	5.6	3.8	4.3	4.1	4.1	4.0	4.9	4.8	5.4	5.9	6.7	6.7	8.6	8.8	9.8	10.3
w/ LPDR	3.4	3.8	2.8	4.1	3.8	3.6	3.7	3.7	3.5	3.6	4.8	5.6	5.5	6	5.2	6.5	8.5	8.8	9.6	10.2
w/ EPDR	3.5	3.9	4	4.3	4.6	4.8	4.5	4.7	4.6	4.5	5.8	6.1	6	6	6.2	6.9	9.0	9.3	10.1	10.6
w/ PPDR	4.4	4.2	4.8	5.5	5.3	3.9	4.2	3.8	4.1	3.8	5.5	5.6	5.4	5.9	6.1	7.1	9.0	9.6	10.5	10.9
Min- <i>k</i> %	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	7.3	9.1	9.6	10.8	11.0
w/ LPDR (CAMIA slope)	4.6	4.3	4.4	5.2	5.2	4.0	4.1	4.6	4.2	4.6	4.2	4.7	5.7	6.3	6.0	6.9	8.5	9.0	10.1	10.3
w/ LPDR	4	4.1	3.7	4.6	4.6	4	4.1	4.6	3.9	3.6	4.7	6.3	4.1	6	5.7	7.2	9.0	9.3	9.8	10.1
w/ EPDR	3.5	3.5	4	4.2	5	4.3	4.2	4.5	4.5	4.7	4.8	5.5	5	5.5	5.4	6.8	8.9	9.4	10.2	11.0
w/ PPDR	4.6	3.8	4.4	5	6.1	3.8	3.4	3.9	4.1	3.8	4.5	4.7	5.5	6.8	5.7	7.6	9.0	9.5	10.6	11.2
Min- <i>k</i> %++	5.4	4.7	6.2	6.8	7	4.4	4.8	5.4	4.5	5.4	4.4	3.5	4.6	5.7	5.7	8.3	9.6	10.0	11.2	11.9
w/ LPDR (CAMIA slope)	5.5	4.7	6.2	6.6	6.8	4.4	4.8	5.4	4.5	5.4	4.4	3.3	4.7	5.7	5.9	7.8	8.7	9.3	10.3	11.0
w/ LPDR	4.6	5.2	6.8	7.6	8.6	4.4	4.4	4.4	4.5	4.7	5.3	3.9	4.8	6.8	7.2	7.5	9.3	9.9	11.1	12.0
w/ EPDR	5.1	5	7.1	7.3	6.4	4.6	4.8	4.8	4.5	5	4.3	4.7	5.4	6.2	6.4	8.1	9.6	10.2	11.4	11.8
w/ PPDR	5.5	4.6	6.4	7.3	6.7	4.3	4.8	5.2	4.6	5.1	4.3	3.7	5.1	6.6	5.8	8.6	9.8	10.1	11.4	11.8

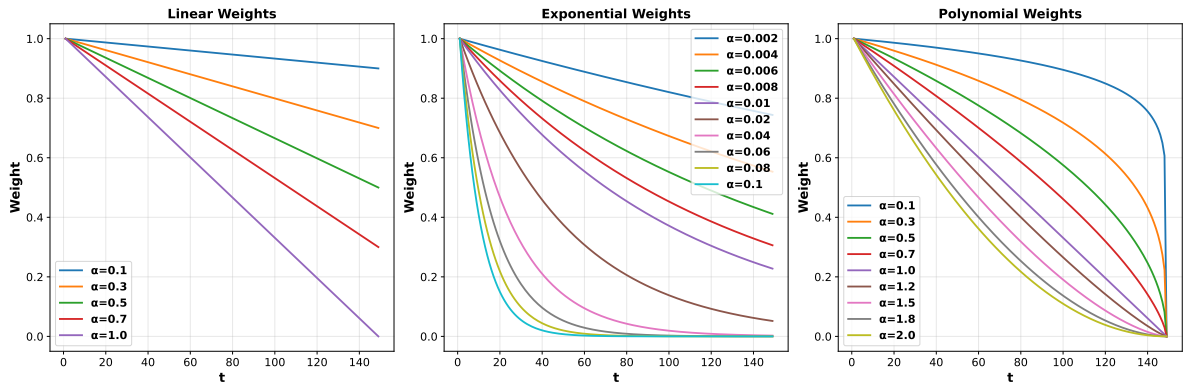


Figure 7: Visualization of different positional weight functions (Linear, Exponential, Polynomial) with various hyperparameters. The x-axis represents the token position in a sequence of length 150, while the y-axis shows the corresponding weight applied to that position.

H Truncation Analysis

We compare PDR to a simple **Truncation** baseline, which discards a suffix of the sequence before scoring. We varied the truncation percentage (the portion of the sequence retained) to find the optimal performance for each base method. As shown in Figure 8, the optimal truncation percentage is highly inconsistent across different methods, making it difficult to find a single best setting. In contrast, our LPDR method (with a fixed $\alpha = 1.0$, shown as dashed lines) robustly outperforms even the best possible truncation result for all base methods. This demonstrates that PDR’s "soft" reweighting is more effective and reliable than the "hard" cutoff of truncation.

I Sample Reweighted Analysis

This section presents additional visualizations of its effect on individual samples. We specifically select pairs of member and non-member samples that are challenging for the baseline **Min- k %** method, meaning their original scores are very close and difficult to distinguish. Figures 9, 10, and 11 illustrate how applying our LPDR, EPDR, and PPDR methods, respectively, alters the token-level scores for these ambiguous pairs. In each case, the reweighting process amplifies the scores of the member samples more significantly than the non-member samples by emphasizing the low-probability tokens that appear early in the sequence. This creates a more distinct separation between them, demonstrating how PDR enhances detection accuracy at the individual sample level.

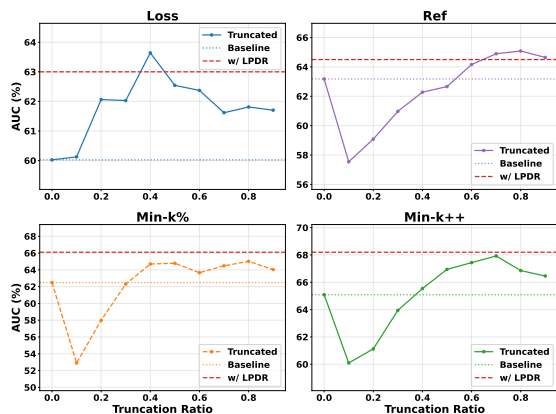


Figure 8: Performance comparison between the Truncation baseline and our LPDR method ($\alpha = 1.0$) on the WikiMIA dataset (length 64, Pythia-12B). The x-axis represents the percentage of the sequence retained for the truncation method. Solid lines indicate results of truncated base methods, whereas dashed lines indicate results with LPDR.

J Analysis of Selected Token Distribution and Case Study

We analyze token-level probability distributions to explain PDR’s effectiveness. Non-member samples often feature high-surprise factual tokens (e.g., dates) early in the sequence, whereas member samples, being memorized, show low surprise on these tokens. Standard methods dilute this early signal by averaging across the sequence. PDR, by assigning higher weights to the prefix, acts as a "matched filter": it amplifies the informative early tokens while suppressing the noise from later function words.

Statistical Evidence (Figure 12): We aggregated the positional frequency of the top- k % lowest-probability tokens across the dataset. The results reveal distinct patterns: for non-members, low-probability tokens are often unmemorized factual details (e.g., dates). In contrast, for memorized members, factual tokens have high probability; the lowest-probability "outliers" are instead common function words (e.g., "the", "of") due to their inherent contextual uncertainty. PDR leverages this by amplifying the early, informative signals while suppressing later noise.

Case Study (Figure 13): Visualizing a specific pair and highlight top k token shows that in the Member sample (a), outlier tokens appear early and are preserved by PDR. In the Non-Member sample (b), outliers are scattered or late, and are effectively suppressed by PDR’s decay. This dual mechanism enhances separability.

Error Study (Figure 14): While PDR shows consistent improvements in most cases, examining failure cases provides valuable insights into its limitations. Figure 14 shows challenging examples: (a) a member sample that remains misclassified after PDR, and (b) a non-member incorrectly pushed towards a higher score. A key observation is that the highlighted Top- k tokens (yellow background) are distributed uniformly across the sequence rather than concentrated at the start. This anomaly suggests **weak memorization**—the model encountered the text but formed no strong memory trace, possibly due to low training frequency, generic content (common function words lacking distinctive features), or **sentence fragmentation** where the dataset’s fixed-length segmentation splits sentences mid-stream, causing the "new sentence start" in the latter half to carry unexpectedly high informativeness and scatter the Top- k tokens. When such positional patterns are absent,

PDR’s monotonic decay assumption becomes less effective or even counterproductive. These cases highlight potential improvements: adaptive weighting that detects weak memorization or sentence boundaries, and sentence-aware segmentation to preserve natural information flow.

K Score distribution.

To visually demonstrate the effectiveness of our method, we analyze the score distributions of member and non-member samples before and after applying PDR. Figure 15 illustrates this comparison for the Min- $k\%$ ++ method on the LLaMA-13B model, using the WikiMIA dataset with a sequence length of 64. For a clearer visualization, the scores are normalized to a range of [0,1]. As the figure shows, the original Min- $k\%$ ++ method already provides some separation between the two distributions. However, after applying our Linear PDR (LPDR), the distributions are pushed further apart. The member sample distribution shifts noticeably towards higher scores, while the non-member distribution remains relatively stable. This increased separation makes it easier to distinguish between member and non-member samples, directly contributing to the improved AUROC performance we observe in our experiments.

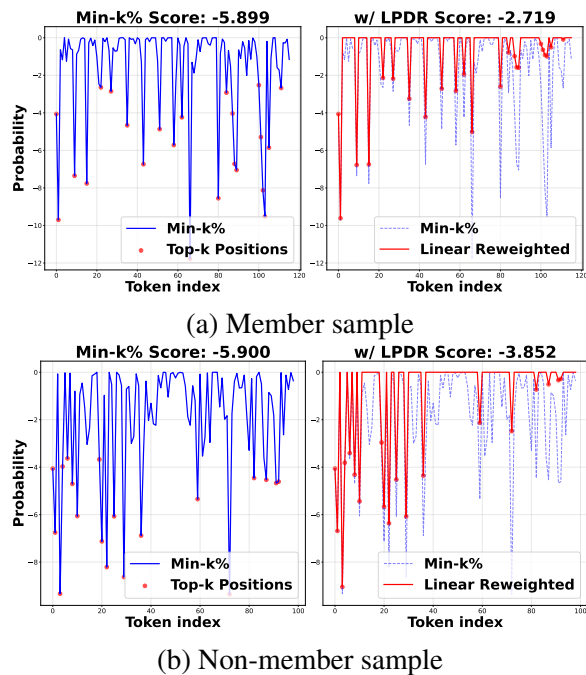


Figure 9: Visualization of token-level score changes for (a) member sample and (b) non-member sample after applying LPDR to the Min- $k\%$ method.

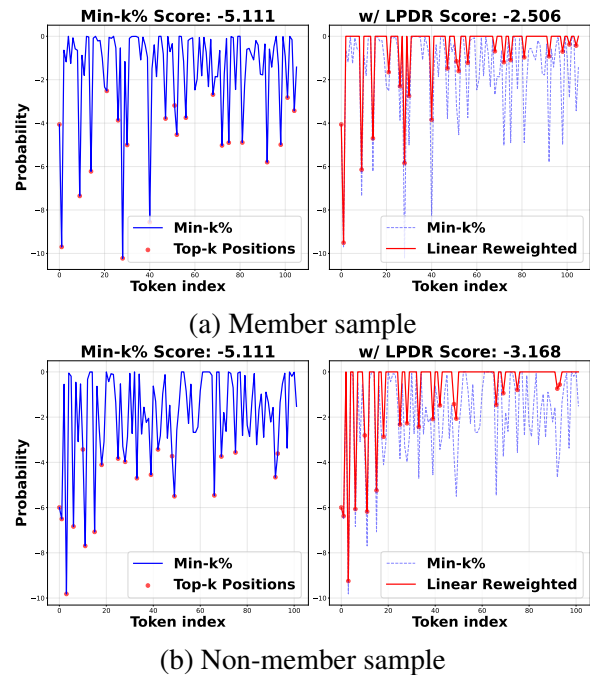


Figure 10: Visualization of token-level score changes for (a) member sample and (b) non-member sample after applying EPDR to the Min- $k\%$ method.

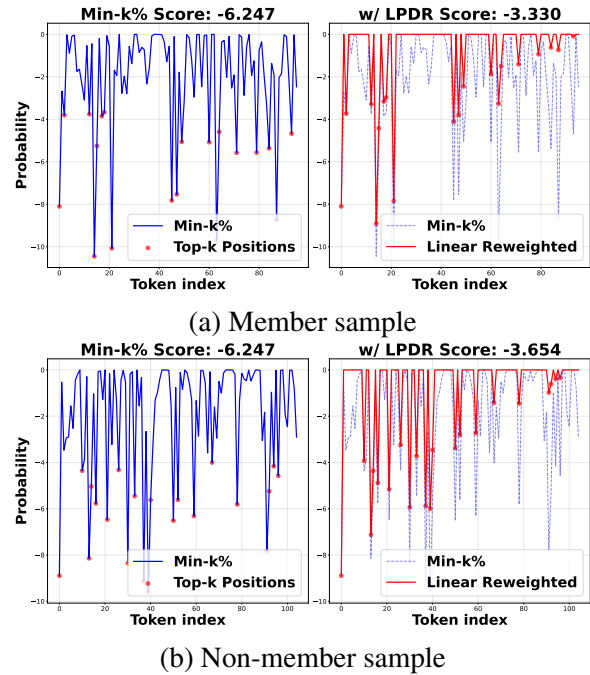


Figure 11: Visualization of token-level score changes for (a) member sample and (b) non-member sample after applying PPDR to the Min- $k\%$ method.

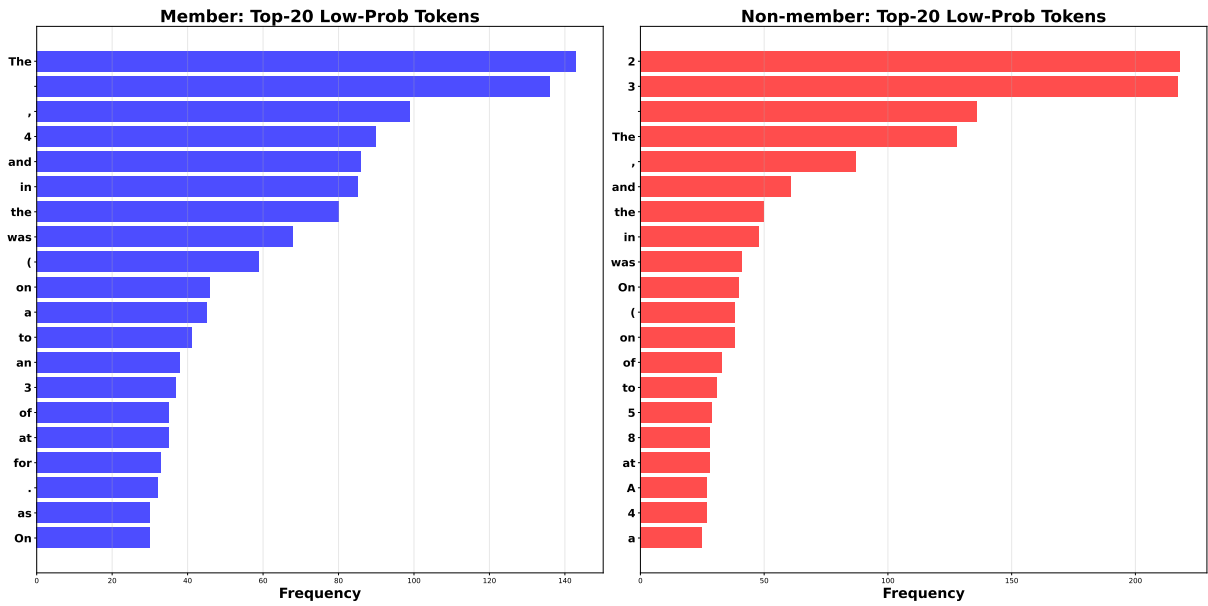
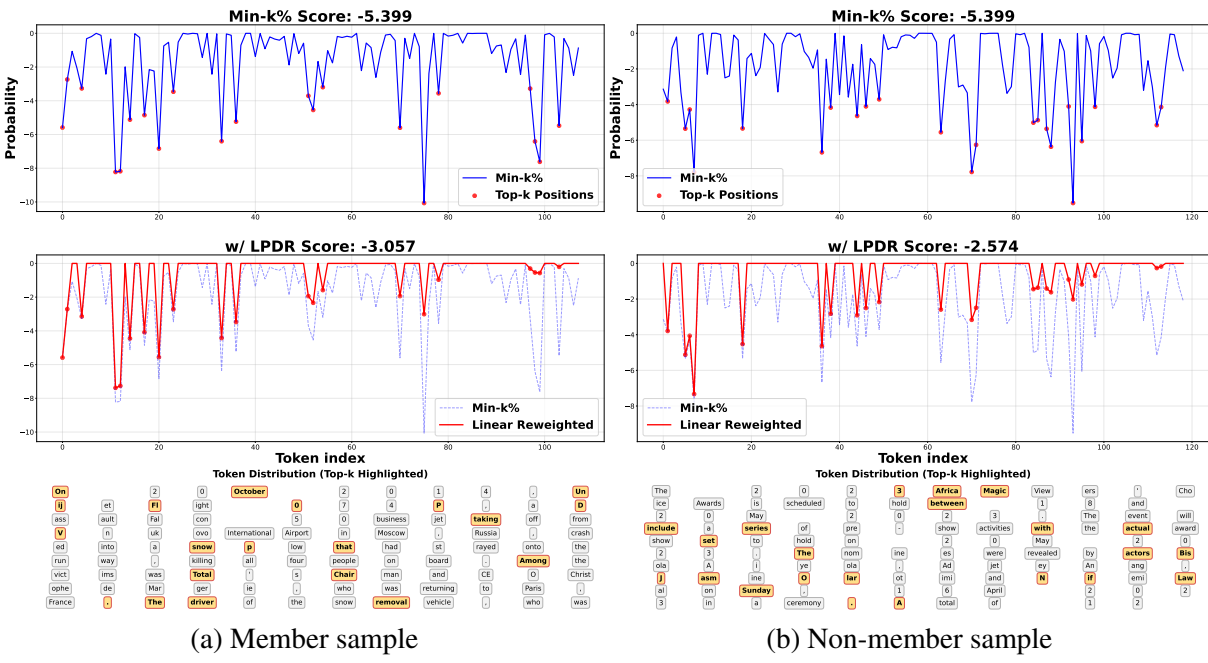


Figure 12: Top k token frequency comparison between member and non-member samples on LLaMA-13B model with 64-token input length on WikiMIA dataset.



(a) Member sample

(b) Non-member sample

Figure 13: Visualization of token-level score changes and highlight top k tokens for (a) member sample and (b) non-member sample after applying PPDR to the Min- k % method.

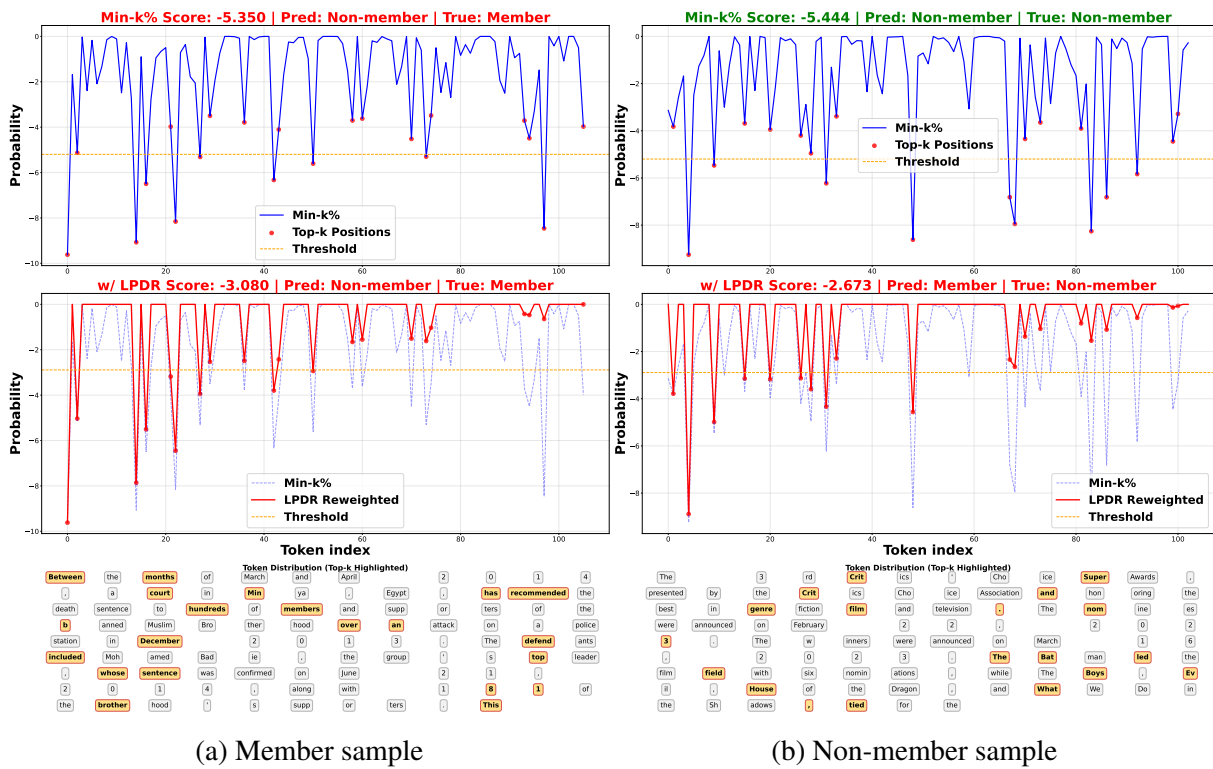


Figure 14: Visualization error case about token-level score changes and highlight top k tokens for (a) member sample and (b) non-member sample.

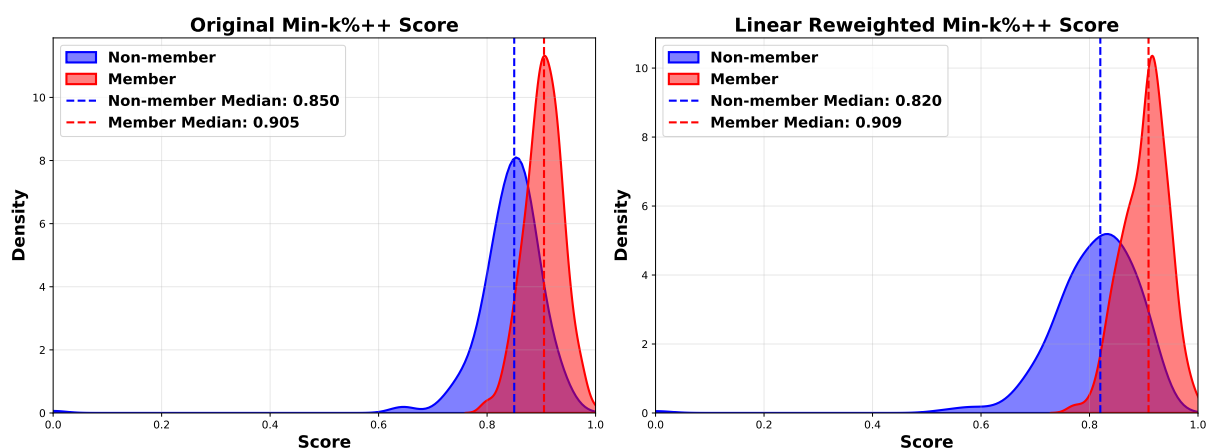


Figure 15: Member and non-member score distribution comparison between Min- $k\%$ and LPDR-Min- $k\%$ on LLaMA-13B model with 64-token input length on WikiMIA dataset. Our PDR method enhances the separation between member and non-member distributions.