ReCall: Membership Inference via Relative Conditional Log-Likelihoods

Roy Xie **Junlin Wang** **Ruomin Huang** **Minxing Zhang** **Rong Ge**
Jian Pei **Neil Zhenqiang Gong** **Bhuwan Dhingra**
Duke University
royxie.com/recall-project-page

Abstract

The rapid scaling of large language models (LLMs) has raised concerns about the transparency and fair use of the data used in their pretraining. Detecting such content is challenging due to the scale of the data and limited exposure of each instance during training. We propose ReCall, (**Re**lative **C**onditional Log-Likelihood), a novel membership inference attack (MIA) to detect LLMs' pretraining data by leveraging their conditional language modeling capabilities. ReCall examines the relative change in conditional log-likelihoods when prefixing target data points with non-member context. Our empirical findings show that conditioning member data on non-member prefixes induces a larger decrease in log-likelihood compared to non-member data. We conduct comprehensive experiments and show that ReCall achieves state-of-the-art performance on WikiMIA dataset, even with random and synthetic prefixes, and can be further improved using an ensemble approach. Moreover, we conduct an in-depth analysis of LLMs' behavior with different membership contexts, providing insights into how LLMs leverage membership information for effective inference at both the sequence and token level.

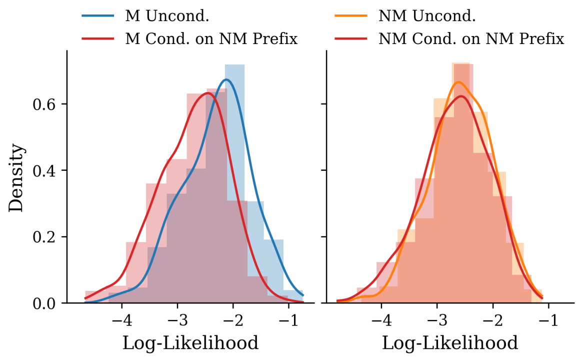


Figure 1: Log-Likelihood comparison between members (M) and non-members (NM). Members experience a higher likelihood reduction than non-members when conditioned with non-member context.

1 Introduction

The amount of pretraining data used to train large language models (LLMs) has quickly expanded in recent years, comprising of trillions of tokens sourced from a vast array of sources (Raffel et al., 2020; Brown et al., 2020). While such diversity and volume allow for a comprehensive language understanding, it also raises the concerns of including sensitive or unintended content such as copyrighted materials (Meeus et al., 2023; Duarte et al., 2024), personally identifiable information (Tang et al., 2023), or test data from benchmarks (Oren et al., 2023; Deng et al., 2024). Additionally, the lack of transparency regarding the composition of pretraining datasets exacerbates these concerns, as many developers are reluctant to disclose full details due to proprietary reasons or the sheer volume of data involved.

To address these concerns, many works have proposed to detect pretraining data in LLMs (Shi et al., 2024; Zhang et al., 2024; Duan et al., 2024), which involve using Membership Inference Attacks (MIAs) to infer whether a given data point was part of the training set. The basic MIA leverages a simple fact that member data are trained and memorized by the model, which leaves footprints in the model, resulting in higher log-likelihood (LL) than non-member data (Yeom et al., 2018). However, the massive scale of pretraining data means that LLMs are typically trained for only a single epoch, making this problem particularly challenging (Duan et al., 2024; Shi et al., 2024), as each instance is only exposed once to the model, leading to limited memorization (Kandpal et al., 2022; Leino and Fredrikson, 2020).

In this work, we propose ReCall (**Re**lative **C**onditional **L**og **L**ikelihood), an efficient MIA detects LLMs pretraining data. ReCall leverages the conditional language modeling capabilities of

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 8671–8689
November 12-16, 2024 ©2024 Association for Computational Linguistics

LLMs by examining the relative change in conditional LLs when prefixing data with non-member context. Our key empirical finding, as illustrated in Figure 1,[1] is that conditioning member data on non-member prefix induces a larger decrease in LL compared to conditioning non-member data on other non-members. This observation forms the basis of RECALL. We leverage a few non-members from the target domain to construct the non-member prefix. While this might seem like a limitation, in practice, it is not a hard constraint. For most real-world applications, we can easily obtain non-member data points by selecting recent data that postdates the model's training data or by creating new, synthetic data points (Shi et al., 2024; Cheng et al., 2024; Duarte et al., 2024).

One interpretation of this empirical finding comes from prior work on in-context learning (ICL), which suggests that it has an effect similar to fine-tuning (Akyürek et al., 2022). By filling the context with non-members, we are essentially changing the predictive distribution of the language model. This change has a larger detrimental effect on members, which are already memorized by the model, compared to non-members, which the model is unfamiliar with regardless of the context. We further discuss this in §5.3.

We evaluate RECALL on two existing benchmarks, WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024). WikiMIA provides different data length for fine-grained evaluation, while MIMIR presents a more challenging setting with minimal distribution shifts between members and non-members. Our comprehensive experiments demonstrate that RECALL achieves state-of-the-art performance on WikiMIA, outperforming existing MIA methods by a large margin, and obtains competitive results on MIMIR (§4.2). We show that using random and synthetic prefixes achieves comparable performance to using real and optimal non-member data (§5.1). We propose an ensemble approach to further enhance the performance of RECALL and mitigate the limitations imposed by the fixed context window size of LLMs (§5.2). Lastly, we conduct an in-depth empirical investigation on how LLMs behave with different membership contexts at both the sequence and token level, providing insights into how LLMs leverage membership information for effective inference (§5.3).

[1] Plot result is from 5-shot setting using the Pythia-6.9B model on the WikiMIA-32 dataset. Additional visualizations are presented in Appendix A.

2 Related Work

Membership Inference Attacks Membership Inference Attacks (MIAs) aim to determine whether a given data sample was part of a model's training set. It was initially proposed by Shokri et al. (2017) and has significant implication in tasks such as measuring memorization and privacy risk (Carlini et al., 2022b; Mireshghallah et al., 2022; Steinke et al., 2024), serving as basis of advance attacks (Carlini et al., 2021; Nasr et al., 2023), and detection on test-set contamination (Oren et al., 2023), copyrighted content (Meeus et al., 2023; Duarte et al., 2024) and knowledge cutoff (Cheng et al., 2024) for LLMs. Research in MIA has been explored in natural language domain for both finetuning (Watson et al., 2021; Mireshghallah et al., 2022; Fu et al., 2023; Mattern et al., 2023) and pretraining settings (Shi et al., 2024; Duan et al., 2024; Zhang et al., 2024). Current LLMs typically train on the massive data for only a single epoch, which makes MIA more challenging compared to the multi-epoch finetuning setting (Carlini et al., 2022b; Shi et al., 2024).

In-context Learning as Attack Vectors Transformer-based (Vaswani et al., 2017) LLMs, pretrained on vast amounts of data, have demonstrated a striking ability known as in-context learning (ICL) (Brown et al., 2020). Specifically, after pretraining, these models can learn and complete new tasks during inference without updating their parameters. In ICL, the model takes in a short sequence of supervised examples (prefix) from the task and then generates a prediction for a query example. Recently, ICL has been used as attack vectors for LLMs, such as jailbreaks (Wei et al., 2023b; Anil et al., 2024), sensitive information extraction (Tang et al., 2023), and backdoor attacks (Kandpal et al., 2023). In this work, we leverage a similar notion from ICL in an *unsupervised* manner to conduct MIA by prefixing the target data points with non-member context. To the best of our knowledge, this is the first study to undertake such a task.

3 RECALL: Relative Conditional Log-Likelihood

Problem Definition Given that M is an autoregressive language model that outputs a probability distribution over the next token given a prefix, let D be a dataset used to train M. The goal of a membership inference attack is to determine, for a target data point $\mathbf{x}$, whether $\mathbf{x} \in D$ or $\mathbf{x} \notin D$. A membership score $S(\mathbf{x}; M)$ is calculated and thresholded to classify whether $\mathbf{x}$ is a member or non-member of the training dataset D.

Proposed Method The key idea behind our proposed method is measuring the behavior of M when conditioning the target data point with a non-member context (prefix). The ReCall score, which is the ratio of the conditional LL to the unconditional LL, is used to quantify this change. To begin, we select a prefix P, which is a sequence of non-member data points p_i concatenated together:

$$P = p_1 \oplus p_2 \oplus ... \oplus p_n. \tag{1}$$

The non-member data points are *known* to be non-members of the model M. Non-member data can typically be obtained for LLMs based on knowledge cutoff time, or by using user-generated or machine-generated synthetic data (Shi et al., 2024; Cheng et al., 2024; Duarte et al., 2024), and we discuss the process of selecting prefixes in detail in §5. For a given dataset D, the prefix P is fixed. For each target data point $\mathbf{x}$, we calculate two LLs from M: (1) the unconditional LL of $\mathbf{x}$ itself, $LL(\mathbf{x})$, and (2) the LL of $\mathbf{x}$ conditioned on the prefix P, denoted as $LL(\mathbf{x}|P)$.[2] The ReCall score for target data point $\mathbf{x}$ is then calculated as:

$$\text{ReCall}(\mathbf{x}) = \frac{LL(\mathbf{x}|P)}{LL(\mathbf{x})}. \tag{2}$$

By providing P to the model, we introduce additional unseen context and new knowledge without explicitly fine-tuning or updating the model parameters (Liu et al., 2023; Brown et al., 2020). For member data points, denoted as $\mathbf{x_m}$, which have already been learned and memorized by the model, the introduction of unseen text may perturb the model's existing confidence to a larger scale compared to non-member data points, denoted as $\mathbf{x_{nm}}$. Consequently, as illustrated in Figure 2,[3] we expect member data points to have *higher* ReCall scores than non-member data points:

$$\mathbb{E}\left[\text{ReCall}(\mathbf{x}_\text{m})\right] > \mathbb{E}\left[\text{ReCall}(\mathbf{x}_\text{nm})\right]. \tag{3}$$

We provide a detailed discussion on the relationship between LL and ReCall scores in Appendix B.2. As an inference-time algorithm, ReCall does not

[2]Note that LLs are *negative* values. More information about LL can be found in Appendix B.1.

[3]Plot result is from the same setting as Figure 1.

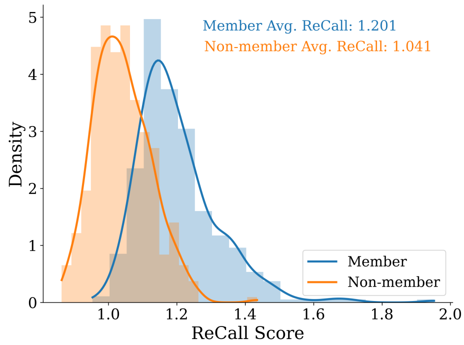


Figure 2: Distribution of ReCall scores for members and non-members. Values close to 1 indicate changes are minimal. Overall, members tend to have higher ReCall scores compared to non-members. More visualizations can be found in Appendix F.

rely on access to the pretraining data distribution or a reference model, which are assumptions made by previous membership inference attacks (Carlini et al., 2022a; Watson et al., 2021).

Prefix Selection The number of non-member data points n used in P (referred to as "shots") is the only hyperparameter in our method. The optimal number of n may vary for different models since models have varying context window lengths (Jin et al., 2024). Additionally, varying lengths of the target data points in D can also affect the n being used (§5.2). Generally, longer data points lead to better MIA performance, as they contain more information that can be memorized by the target model (Shi et al., 2024). We will demonstrate in §5.1 that it is possible to use a random or synthetic prefix generated by LLMs to achieve high performance. In §5.2, we will show that only *one* shot is needed for ReCall to outperform baselines.

4 Experiments and Results

In this section we conduct comprehensive experiments to demonstrate the effectiveness of ReCall.

4.1 Experimental Setup

Benchmarks We focus on WikiMIA (Shi et al., 2024) and MIMIR (Duan et al., 2024) benchmarks. WikiMIA consists of text from Wikipedia, with member and non-member samples determined based on model's knowledge cutoff time. The dataset is grouped into splits based on sentence length (32, 64, 128) to enable fine-grained evaluation. MIMIR is derived from the Pile dataset (Gao et al., 2020) and covers various domains. To create a challenging setting for membership inference, MIMIR employs n-gram filtering to select member and non-member samples from the same dataset, maximizing their similarity (Duan et al., 2024). While this deviates from the standard MIA setting, we report results from both 13-gram and 7-gram MIMIR versions for a rigorous evaluation.

Baselines We compare ReCall against 6 state-of-the-art baselines: ***Loss*** (Yeom et al., 2018) simplely uses input loss as the membership score; ***Reference*** (Carlini et al., 2022a) calibrates input loss using a reference model; ***Zlib*** (Carlini et al., 2021) compresses input loss using Zlib entropy; ***Neighbor*** (Mattern et al., 2023) compares input loss to the average loss of similar tokens; ***Min-K%*** (Shi et al., 2024) averages top-k% minimum token probabilities from the input; and ***Min-K%++*** (Zhang et al., 2024) extends Min-K% with normalization factors. More details can be found in Appendix C.

Models For WikiMIA, we experiment with a diverse set of transformer-based LLMs, including the Pythia 6.9B (Biderman et al., 2023), GPT-NeoX 20B (Black et al., 2022), LLaMA 30B (Touvron et al., 2023), and OPT 66B (Zhang et al., 2022). We also include the Mamba model which uses a state space-based architecture (Gu and Dao, 2023). For MIMIR, we focus on the Pythia model family (160M, 1.4B, 2.8B, 6.9B, 12B parameters), consistent with (Duan et al., 2024; Zhang et al., 2024). Following Shi et al. (2024) and Duan et al. (2024), we use the smallest model version and the best performing reference model for Reference method.

Metrics Following the standard evaluation procedure for MIAs (Shi et al., 2024; Duan et al., 2024; Zhang et al., 2024; Mattern et al., 2023), we use the area under the ROC curve (AUC) as our main evaluation metric, along with the true positive rate at a one percent false positive rate (TPR@1%FPR) (Carlini et al., 2022a). More details about the evaluation metrics can be found in Appendix D.

Implementation Details As all benchmarks do not provide validation set, and also following Zhang et al. (2024), we sweep over 1 to 12 for the number of shots and report the best result. We randomly select 12 data points as prefix candidates from the test set and exclude them from evaluation. We will show in §5.1 that ReCall is robust to random selection. In §5.2, we will demonstrate that ReCall significantly outperforms all baselines even with just one shot. We also compare the *best* possible performance of other methods for a fair comparison. More implementation details can be found in the Appendix E.

4.2 Main Results

WikiMIA Results Table 1 shows that ReCall achieves state-of-the-art performance on the WikiMIA benchmark. Our method consistently outperforms all existing baseline methods in all settings by a large margin. On average, ReCall surpasses the runner-up Min-K%++ by 14.8%, 15.4%, 14.8% in terms of AUC scores for input lengths of 32, 64, 128, respectively. Moreover, ReCall's superior performance is consistent across different model architectures. The improvement is particularly significant for shorter inputs and smaller models, which are known to be more challenging for MIAs (Shi et al., 2024), which demonstrates the effectiveness of ReCall in capturing membership signals even in a challenging setting. We also report the TPR@1%FPR results in Appendix I, which again shows the significant improvements and highlights the effectiveness of our approach in detecting pretraining data with high precision.

MIMIR Results On the more challenging MIMIR benchmark, ReCall achieves competitive performance compared to state-of-the-art methods, as shown in Table 2 (13-gram). It is important to note that MIMIR presents a much more challenging scenario as it deviates from the standard membership inference setting by minimizing the distribution shift between members and non-members (Maini et al., 2024; Duan et al., 2024). The AUC scores for the 13-gram setting for all MIAs are close to *random guessing*, indicating the difficulty for MIAs when members and non-members are very similar (Duan et al., 2024). Despite this, ReCall on average outperforms all baselines on 160M and 1.4B models. For other models, the Reference method dominates, but it is important to note that exhaustively searching for the best reference model among different candidate models is not only computationally expensive but also may not be feasible in practice (Duan et al., 2024). In contrast, ReCall does not rely on any reference models, yet still provides competitive performance. The 7-gram results in Appendix G show better performance than the 13-gram setting, with ReCall achieving the highest AUC on 1.4B, 2.8B, 6.9B, and 12B models. We also report the TPR@1%FPR results for both settings in Appendix J.

Len.	Method	Mamba-1.4B	Pythia-6.9B	LLaMA-13B	NeoX-20B	LLaMA-30B	OPT-66B	Average
32	Loss	60.7	63.6	67.6	68.7	69.5	65.4	65.9
	Ref	60.9	63.7	67.7	68.9	70.0	65.8	66.2
	Zlib	61.6	64.1	67.8	68.9	69.9	65.5	66.3
	Neighbor	64.1	65.8	65.8	70.2	67.6	68.2	66.9
	Min-K%	63.2	66.3	68.0	71.8	70.1	67.4	67.8
	Min-K%++	66.8	70.3	84.8	75.1	84.3	70.3	75.3
	ReCall	**90.2**	**91.6**	**92.2**	**90.5**	**90.7**	**85.1**	**90.1**
64	Loss	59.2	61.7	64.4	67.4	66.7	63.1	63.8
	Ref	59.5	61.8	64.9	67.7	67.6	63.6	64.2
	Zlib	61.4	63.3	65.9	68.7	68.0	64.6	65.3
	Neighbor	60.6	63.2	64.1	67.1	67.1	64.1	64.4
	Min-K%	63.2	65.0	66.9	73.5	69.1	67.9	67.6
	Min-K%++	67.2	71.7	85.6	76.0	84.8	70.2	75.9
	ReCall	**91.4**	**93.0**	**95.2**	**93.2**	**94.9**	**79.9**	**91.3**
128	Loss	63.1	65.0	69.1	70.6	72.0	65.3	67.5
	Ref	63.0	65.1	69.3	70.8	73.0	65.5	67.8
	Zlib	65.5	67.8	71.5	72.6	73.6	67.6	69.8
	Neighbor	64.8	67.5	68.3	71.6	72.2	67.7	68.7
	Min-K%	66.8	69.5	71.5	75.0	74.2	70.2	71.2
	Min-K%++	66.8	69.7	83.9	75.8	82.9	72.1	75.2
	ReCall	**91.2**	**92.6**	**92.5**	**91.7**	**91.2**	**81.0**	**90.0**

Table 1: AUC results on WikiMIA benchmark. **Bolded** number shows the best result within each column for the given length. ReCall achieves significant improvements over all existing baseline methods in all settings.

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	50.2	51.3	51.8	52.8	53.5	65.6	69.5	71.0	72.8	73.8	49.6	50.0	50.1	50.7	51.1	50.0	50.0	50.2	50.9	51.5
Ref	50.9	**54.7**	**57.6**	**60.3**	**61.7**	63.9	67.0	65.3	64.3	63.1	48.8	**52.3**	**53.7**	**54.6**	**56.4**	**51.0**	52.1	**53.6**	**55.9**	**58.1**
Zlib	51.1	52.1	52.5	53.6	54.4	**67.4**	**70.8**	**72.1**	73.8	74.7	49.5	50.0	50.2	50.7	51.1	50.1	50.2	50.3	50.9	51.4
Neighbor	50.7	51.7	52.2	53.2	/	65.3	69.4	70.5	72.1	/	49.6	50.0	50.1	50.8	/	47.9	49.1	49.7	50.1	/
Min-K%	49.8	51.3	51.5	53.2	54.3	64.4	68.8	70.3	72.2	73.4	50.2	51.0	50.5	51.3	51.4	50.4	49.9	50.5	51.0	52.4
Min-K%++	49.5	53.4	54.9	57.6	61.2	64.7	69.3	70.2	72.9	73.4	50.0	50.8	50.6	52.6	53.4	50.4	50.7	52.1	54.2	54.8
ReCall	**51.3**	52.3	52.3	54.0	54.6	64.9	70.0	71.7	**74.2**	**74.8**	**50.9**	51.7	50.2	51.6	51.1	49.9	**52.3**	50.0	51.4	53.4

	ArXiv					DM Mathematics					HackerNews					Average				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	51.0	51.5	51.9	52.9	53.4	48.9	48.5	48.4	48.5	48.5	49.4	50.4	51.2	51.9	52.6	52.1	53.0	53.5	54.4	54.9
Zlib	50.0	50.8	51.2	52.2	52.6	48.2	48.2	48.1	48.1	48.1	49.7	50.2	50.7	51.1	51.6	52.3	53.2	53.6	54.3	54.8
Ref	50.0	51.6	53.5	**56.0**	**57.8**	**51.4**	51.4	50.7	51.6	51.3	49.5	52.3	**55.6**	**57.9**	**60.9**	52.2	54.5	**55.7**	**57.2**	**58.5**
Neighbor	50.7	51.4	51.8	52.2	/	49.0	47.0	46.8	46.6	/	50.9	51.7	51.5	51.9	/	52.0	52.9	53.2	53.8	/
Min-K%	51.7	52.0	53.1	53.7	55.2	50.3	50.0	50.0	49.7	50.2	50.9	51.9	52.4	53.6	54.7	52.5	53.6	54.0	55.0	55.9
Min-K%++	50.7	51.0	**53.9**	55.5	58.4	50.9	49.8	**51.8**	**52.0**	**52.1**	50.7	51.2	52.9	54.6	56.8	52.4	53.7	55.3	57.0	**58.5**
ReCall	**52.5**	**52.8**	52.7	54.6	55.9	50.9	**52.8**	51.3	50.8	50.8	**52.4**	**53.0**	53.2	54.2	54.7	**53.3**	**54.6**	54.5	55.8	56.5

Table 2: AUC results on MIMIR benchmark. ReCall achieves competitive performance compared to state-of-the-art methods, especially on smaller models (160M and 1.4B), while not relying on any reference models. The best results for each dataset and model size are highlighted in **bold**.

5 Analysis

In this section, we conduct a series of investigations to better understand ReCall. Following Zhang et al. (2024), we focus on the WikiMIA using the Pythia-12B model for our analysis.

5.1 Prefix Selection

Dynamic Prefix with Different Similarities We investigate if using prefixes that are similar to the target data points results in better performance. For each data point, we search the entire dataset and create prefixes based on the Term Frequency-Inverse Document Frequency (TF-IDF) similarity scores (Sparck Jones, 1972): (i) most similar (highest scores), (ii) moderately similar (middle scores), (iii) least similar (lowest scores), and (iv) random selection (random scores). In this setting, each target data point has its own prefix.

We compare the results with the original fixed-prefix setting[4] in Table 3. The results indicate that using the most similar prefix yields the best performance, followed by random selection, moderate similarity, and least similar prefix. This suggests

[4]Using one fixed prefix for all target data points.

Similarity	Len. 32	Len. 64	Len. 128
Random	69.0	71.9	74.2
Least	57.2	69.6	61.0
Moderate	66.9	71.6	70.2
Most	74.1	76.1	77.6
Fixed	88.2	88.8	87.8

Table 3: RECALL perform better with fixed prefix than dynamic prefix. Similar prefix results best performance, followed by random selection.

that the most effective prefixes are those similar to the target data point, with random selection providing the next best performance. We also observe that dynamic prefix selection does not perform as well as using a fixed prefix, likely because each data point creates a different threshold when using a dynamic prefix, leading to inconsistencies.

Randomly Selected Prefix We investigate the impact of randomly selecting non-member prefixes on the performance of RECALL. We randomly select 12 non-member data points from the test set, divide them into 3 sets, and compare the results with the best-performing baselines, Min-K% and Min-K++%, in Table 4. The results show that the performance of RECALL across all groups is similar, with an average difference of 2.5% between the top and lowest prefix, while significantly outperforming the baselines. This finding suggests that RECALL is robust to random prefix selection, as long as the prefixes are indeed non-members. In §5.3, we will show that while their similarity to the target data point is preferred, the effectiveness of RECALL appears to be largely dependent on the non-member status of the prefixes.

Prefix Set	Len. 32	Len. 64	Len. 128
Set 1	88.2	88.8	87.8
Set 2	90.4	91.4	90.5
Set 3	87.7	89.4	89.2
Min-K%	67.7	67.9	70.2
Min-K%++	72.4	72.5	72.7

Table 4: AUC scores of RECALL with randomly selected prefixes divided into three sets, compared to the best-performing baselines, Min-K% and Min-K++.

Varying Domain Prefix We explore the impact of prefix similarity on RECALL's at a domain-level. The WikiMIA dataset was constructed using knowledge cutoff, with Wikipedia articles published after the model's training data used as non-members. We randomly select prefixes from Wikipedia (most similar), arXiv (moderately similar), and GitHub (least similar) and present the results in Table 5. We observe that when the prefix domain is significantly different from the target data (GitHub), RECALL's effectiveness is reduced. However, the Wikipedia prefix achieves performance close to the original dataset, indicating that RECALL is not overfitting to the peculiarity of the original dataset. This suggests that selecting prefixes from the same or similar domains as the target data is preferred. While the original WikiMIA dataset and the Wikipedia prefix setting both use knowledge cutoff, there might be differences in the specific passages selected. However, the strong performance of the Wikipedia prefix demonstrates that RECALL can generalize well to other non-member data points from the same domain, providing further evidence of its effectiveness.

Prefix Domain	Len. 32	Len. 64	Len. 128
GitHub	72.4	69.5	68.8
arXiv	73.4	72.0	72.0
Wikipedia	83.3	87.1	86.5
Original	88.2	88.8	87.8
Min-K%	67.7	67.9	70.2
Min-K%++	72.4	72.5	72.7

Table 5: AUC scores of RECALL when using prefixes from different domains obtained based on model knowledge cutoff time. Similar domains are preferred.

GPT-4o Generated Prefix The assumption of accessing a small number of non-member data points as prefix might not always be feasible as the membership of the selected prefix *itself* could be mixed or unknown in a real-world scenario. Therefore, we explore the possibility using synthetic prefixes generated by LLMs from a mix of members and non-members. We randomly select 6 members and nonmembers and generate a synthetic prefix using GPT-4o (OpenAI, 2024) based on them.[5] We compare results to the baselines and using ground-truth non-member data point as prefix in Table 6. We observe that even with the synthetic prefix from a mix of members and nonmembers, the performance is still close to the original setting, where the ground-truth non-members are used. This highlights the potential of using synthetic prefixes in situations where access to ground-truth non-member data is limited or unavailable, expanding the applicability of RECALL to a practical scenario. More synthetic prefixes results can be found in Appendix K.

[5]Prompt can be found in Appendix H.

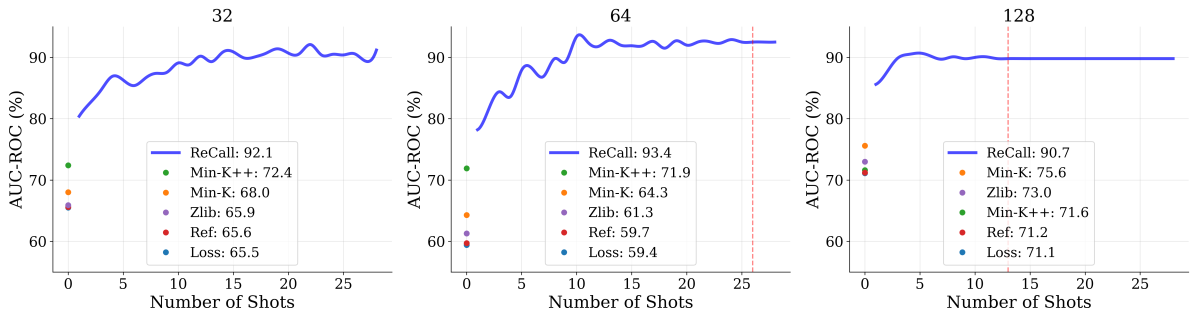


Figure 3: ReCall performance up to 28 shots. Red dash line represents the LLMs' context window limit. ReCall consistently outperforms baselines across all settings, even with just *one* shot.

Prefix Setting	Len. 32	Len. 64	Len. 128
Synthetic	85.4	90.3	86.4
Original	88.2	88.8	87.8
Min-K%	67.7	67.9	70.2
Min-K%++	72.4	72.5	72.7

Table 6: AUC scores of ReCall with synthetic prefixes generated from GPT-4o, compared to the prefix from original dataset and two best-performing baselines.

Prefix Setting	Len. 32	Len. 64	Len. 128
28-shot	92.1	93.4	90.7
Ensemble	93.0	94.9	91.5
Min-K%	67.7	67.9	70.2
Min-K%++	72.4	72.5	72.7

Table 7: Performance comparison of the ensemble method, 28-shot method, and baselines. By taking the average of the ReCall scores using ensemble method, we can make a more robust prediction.

5.2 Shots vs. Performance

Impact of Number of Shots and Context Window Size In general, increasing the context length improves the performance of MIAs, as the model can leverage more information to distinguish between members and non-members (Shi et al., 2024). However, LLMs have fixed context window sizes that limit the amount of text they can process in a single input. Exceeding the context length resultsing in remaining at the maximum number of shots which can fit in the context, so the performance should plateau after a while. We evaluate the performance of ReCall with up to 28 shots, intentionally exceeding the context window to probe the limitation. We compare the results to baselines in Figure 3 and observe that ReCall consistently outperforms all baselines by a significant margin, even with just *one* shot. As the number of shots increases, ReCall's performance improves across all settings. As expected, the performance plateau when the length of the input exceeds the context window limit, which can be observed in the 64 and 128 length settings. This is because the longer the data length, the fewer shots are needed to exceed the context window.

Further Improvement with Ensemble Method When dealing with a large number of shots, the context window limit of LLMs can be a bottleneck for the performance of ReCall. To circumvent this problem, we propose an ensemble method. Instead of using all 28 shots at once, we divide them into smaller sets as prefixes and calculate the ReCall score for the target data point under each set. We then take the mean of these independent ReCall scores to obtain the final score. The intuition behind this approach is that each group provides an independent ReCall score on the membership status of the input text, and by averaging them, we can reduce the variance and obtain a more robust estimate. We compare the performance of the ensemble method with the 28-shot method and the baselines in Table 7. The results show that the ensemble method provides further improvement over the 28-shot method for all settings, demonstrating ReCall's utility in leveraging a large number of shots while respecting the context window limit.

5.3 Discussions

Why Non-member Prefixes? While non-member data can be easily accessed based on the data characteristics such as knowledge cutoff time, access to member data is much harder to assume, as LLMs training data are usually not disclosed (Shi et al., 2024). We empirically show that using member prefix is not only an unrealistic assumption but also does not yield the desired

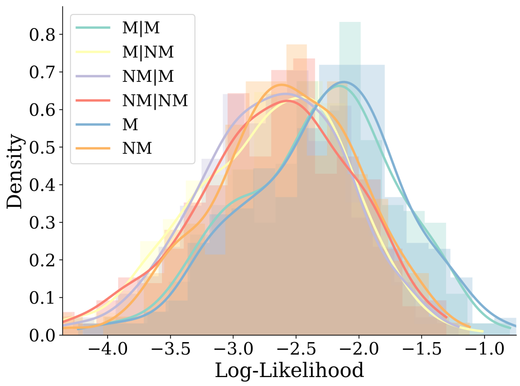


Figure 4: Conditioning both member and non-member with member prefix do not yield significant changes in LL compare to non-member prefix. More visualization can be found in Appendix L.

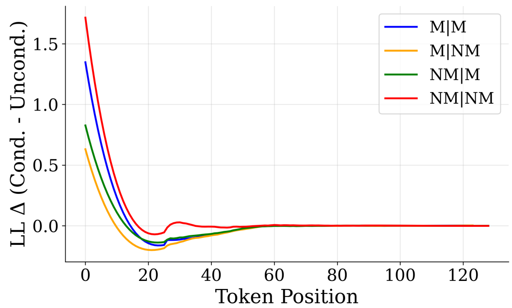


Figure 5: Token-level LL changes for members and non-members with different membership prefix. The largest changes occur in the beginning tokens. Member and non-member data are most different when prefixed with non-member context.

effect for detecting pretraining data. Following the same setting as Figure 1, we prefix the target data points with both member and non-members and present the results in Figure 4. We observe that conditioning both member and non-member data with a member prefix does not result in significant changes in LL compared to their unconditional LL. This suggests that using member data as context does not induce the distribution shift necessary for ReCall to effectively distinguish between members and non-members. We hypothesize that prefixing with additional member data does not significantly alter the model's predictive distribution because the model has already memorized the member data during pretraining and is familiar with its distribution. In contrast, prefixing with non-member data introduces a distribution shift that has a more pronounced effect on the LL of member data compared to non-member data. These findings demonstrate that ReCall is indeed leveraging the membership information of the prefix data to make prediction.

Token-level Analysis Previous works, such as Shi et al. (2024) and Zhang et al. (2024), have leveraged token-level signals for MIAs. Similarly, we investigate ReCall at the token level with both member and non-member prefixes. We examine where the changes are occurring and how member and non-member prefixes impact token-level LL. For each token position, we take the average from all data points and present the LL change in Figure 5 and more in Appendix M. We observe three interesting points: (i) Most changes occur in the beginning tokens for all settings, especially the first few tokens. This is because the model becomes more confident about predicting the next token as it approaches the end of the sequence, given that it has already seen the preceding context. (ii) The changes are most dominant when a data point is prefixed with context from the same membership (e.g., M|M and NM|NM). This means the model has a stronger preference to continue with text from the same membership status. (iii) The differences between NM|NM and M|NM are more pronounced than those between M|M and NM|M, which further supports our finding that using non-member prefixes is effective for ReCall to distinguish between members and non-members, while member prefixes do not yield desired performance.

MIA Evaluation Recently, the MIA community has been discussing the effectiveness of MIA for LLMs (Duan et al., 2024; Das et al., 2024). Two key challenges complicate MIA evaluations: the vast scale of pretraining data and the temporal distribution shift between members and non-members. While Das et al. (2024) demonstrates that simple text classifiers trained *directly* on the dataset can achieve superior performance, the equivalence of using model internal logits versus direct text classification remains an open question. We conduct additional experiments on two applicable datasets used by Das et al. (2024): Temporal Wiki and Temporal arXiv (Duan et al., 2024). We present results in Table 8 and observe that ReCall surpasses text classifier performance on both Temporal Wiki and Temporal arXiv datasets, while the text classifier shows superior results on WikiMIA. These results indicate that ReCall effectively discriminates between members and non-members, particularly for datasets structured based on temporal distributions.

Dataset	Text Classifier	RECALL
WikiMIA	98.7	95.2
Temporal Wiki	79.9	81.2
Temporal arXiv	75.6	76.0

Table 8: AUC score comparison between text classifier and RECALL for three different datasets.

6 Conclusion

We introduced RECALL, a novel MIA for detecting pretraining data in LLMs by leveraging their conditional language modeling capabilities. RECALL captures the relative change in conditional log-likelihoods when prefixing target data points with non-member context. Through extensive experiments on WikiMIA and MIMIR benchmarks, we demonstrated RECALL's state-of-the-art performance, outperforming existing MIA methods on WikiMIA. We showed that random and synthetic prefixes achieve comparable performance to real non-member data, enhancing RECALL's practicality. RECALL consistently outperforms baselines and can be further improved using an ensemble method. Our in-depth analysis revealed valuable insights into LLMs' behavior under different membership contexts. As future work, we plan to investigate the theoretical aspects of RECALL and explore more efficient MIA methods.

7 Limitations

While RECALL demonstrates strong empirical performance, the theoretical analysis of why it works is limited in this work. We provide some hypotheses based on the connections between ICL and conditional language modeling in LLM, but a more rigorous and in-depth theoretical investigation is needed to fully understand the underlying mechanisms. This is particularly important given that ICL itself is an *understudied* area, and the research community is still actively exploring how and why it works (Wei et al., 2023a; Liu et al., 2023; Anil et al., 2024). A better understanding of ICL could provide valuable insights into our method. We believe that further theoretical analysis of RECALL and its interplay with ICL is an important direction for future research. Our method assumes gray-box access to the target model, which requires access to its output probabilities. However, it is important to note that this limitation is shared by existing pretraining data detection methods (Shi et al., 2024; Zhang et al., 2024; Duan et al., 2024; Mattern et al., 2023; Carlini et al., 2022a; Yeom et al., 2018). In the future, we plan to explore methods that require less access to the target model.

8 Ethics Statement

Our primary intention is to advance the detection of sensitive content in LLMs, which is important for protecting privacy and intellectual property. However, we acknowledge that, like any tool, it could be misused to extract private information. Protecting user privacy should be a key priority as LLMs become increasingly ubiquitous and powerful. We call for further research on privacy-preserving LLM development and strategies to prevent misuse.

9 Acknowledgment

We thank Jack Parker and Jabari Kwesi for helpful discussions. We also thank anonymous reviewers for their insightful feedback. This work was supported in part by the NSF Graduate Research Fellowship and NSF grants No. 2112562, 1937787, 2131859, 2125977, 2331065.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-shot jailbreaking. *Anthropic, April*.

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

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

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

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022a. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914. IEEE.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2022b. Quantifying memorization across neural language models. *arXiv preprint arXiv:2202.07646*.

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

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. *arXiv preprint arXiv:2403.12958*.

Debeshee Das, Jie Zhang, and Florian Tramèr. 2024. Blind baselines beat membership inference attacks for foundation models. *arXiv preprint arXiv:2406.16201*.

Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. 2024. Investigating data contamination in modern benchmarks for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8698–8711, Mexico City, Mexico. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2402.09910*.

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

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

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Michael M. Grynbaum and Ryan Mac. 2023. The Times Sues Openai and Microsoft Over A.I. Use of Copyrighted Work. *The New York Times*.

Albert Gu and Tri Dao. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Hongye Jin, Xiaotian Han, Jingfeng Yang, Zhimeng Jiang, Zirui Liu, Chia-Yuan Chang, Huiyuan Chen, and Xia Hu. 2024. Llm maybe longlm: Self-extend llm context window without tuning. *arXiv preprint arXiv:2401.01325*.

Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. In *International Conference on Machine Learning*, pages 10697–10707. PMLR.

Klas Leino and Matt Fredrikson. 2020. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. 2024. Llm dataset inference: Did you train on my dataset? *arXiv preprint arXiv:2406.06443*.

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. *arXiv preprint arXiv:2203.03929*.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*.

OpenAI. 2024. GPT-4o. https://openai.com/index/hello-gpt-4o/.

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*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

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.

Karen Sparck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Thomas Steinke, Milad Nasr, and Matthew Jagielski. 2024. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems*, 36.

Xinyu Tang, Richard Shin, Huseyin A Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2023. Privacy-preserving in-context learning with differentially private few-shot generation. *arXiv preprint arXiv:2309.11765*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. 2021. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, et al. 2023a. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Zeming Wei, Yifei Wang, and Yisen Wang. 2023b. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*.

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.

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

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

A Conditional and Unconditional LL Visualizations

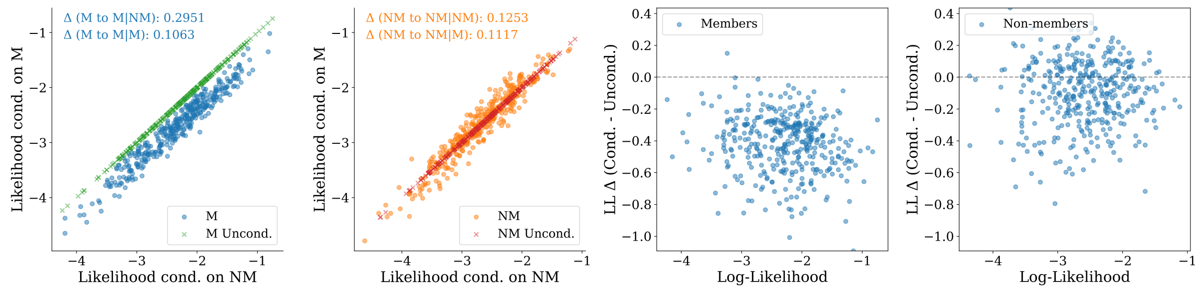


Figure 6: Log-likelihood comparison with 5-shot non-member prefix for Pythia-6.9B model on WIKIMIA dataset. **Left Two**: shows distribution the conditional and unconditional version of both members and non-members. **Right Two**: shows the LL difference between the conditioned and unconditioned version of the data. Member data is experiencing much higher distributional shift than non-member data.

B Additional Log-Likelihoods and RECALL Scores Details

B.1 Log-Likelihood

Log-likelihood (LL) can be used to measure of how likely a given text is trained under a specific language model. A higher LL indicates that the model is more confident in predicting the text, suggesting that the model might have been trained such data. Conversely, a lower LL implies that the model is less familiar with the text. LL is closely related to other metrics such as loss and perplexity, where a higher LL corresponds to a lower loss and perplexity. While metrics like perplexity or probability can be used for membership inference, we use LL in this work since it is more numerically stable and mitigates underflow problems (Goodfellow et al., 2016).

B.2 Relationship between LL and RECALL Scores

We show that conditioning member data on non-member prefixes induces a larger decrease in LL compared to non-member data, as suggested by Figure 1. We introduce RECALL score as our membership score to quantify this change. For a member data point x_m, the RECALL score is typically greater than 1, as the conditional LL is generally lower than the unconditional LL:

$$\text{RECALL}(x_m) = \frac{LL(x_m|P)}{LL(x_m)} > 1, \text{ since } LL(x_m|P) < LL(x_m) < 0. \tag{4}$$

Note that LLs are **negative** values. For a non-member data point x_{nm}, the RECALL score can be either greater than, equal to, or less than 1. In some cases, prefixing a non-member data point with a non-member prefix might increase the model's confidence, resulting in a RECALL score less than 1. However, the core idea of RECALL, as illustrated in Figure 2 and Appendix L, is that for member data points, the RECALL score is *consistently higher* than non-member data points regardless the LL changes (increase or decrease) in non-members (Equation (3)). To better illustrate this, consider a member data point x_m with $LL(x_m|P) = -4$ and $LL(x_m) = -3$. The RECALL score for x_m is calculated as:

$$\text{RECALL}(x_m) = \frac{LL(x_m|P)}{LL(x_m)} = \frac{-4}{-3} = 1.3 \tag{5}$$

Note that $LL(x_m|P) < LL(x_m)$, indicating a decrease in LL when conditioning on the non-member prefix P. Now, consider a non-member data point x_{nm} with $LL(x_{nm}|P) = -3.3$ and $LL(x_{nm}) = -3$. The RECALL score for x_{nm} is:

$$\text{RECALL}(x_{nm}) = \frac{LL(x_{nm}|P)}{LL(x_{nm})} = \frac{-3.3}{-3} = 1.1 \tag{6}$$

Comparing the member and non-member data points, we observe that $LL(x_m|P) < LL(x_{nm}|P)$, indicating a larger decrease in LL for the member data point when conditioned on the non-member prefix. However, the ReCall score for the member data point is higher than that of the non-member data point: $\text{ReCall}(x_m) > \text{ReCall}(x_{nm})$.

C Additional Baseline Details

Given target data point $\mathbf{x}$, a MIA aims to determine if $\mathbf{x}$ was part of the training dataset D used to train a model M by computing a membership score $S(\mathbf{x}; M)$. We provide a detailed description of baseline MIA methods used in our experiments. For each method, we explain how the membership score is calculated and the intuition behind the approach.

C.1 LOSS

The LOSS baseline (Yeom et al., 2018) uses the model's computed loss over the target sample as the membership score. The intuition behind this approach is that the model will have lower loss values for data points it has seen during training (members) compared to unseen data points (non-members).

$$S(\mathbf{x}; M) = Loss(\mathbf{x}; M) \tag{7}$$

C.2 Reference-based

The Reference-based baseline (Carlini et al., 2022a) extends the LOSS attack by calibrating the target model's loss with respect to a reference model trained on similar data but not necessarily the same data points. This helps to account for the intrinsic complexity of the target sample and reduces false negatives.

$$S(\mathbf{x}; M) = Loss(\mathbf{x}; M) - Loss(\mathbf{x}; M_{\text{ref}}) \tag{8}$$

C.3 Zlib Entropy

The Zlib Entropy baseline (Carlini et al., 2021) normalizes the target model's loss using the zlib compression size of the input sample. The idea is that the loss of member samples will have lower entropy and thus a smaller compression size compared to non-members.

$$S(\mathbf{x}; M) = \frac{Loss(\mathbf{x}; M)}{\text{zlib}(\mathbf{x})} \tag{9}$$

C.4 Neighborhood Attack

The Neighborhood Attack baseline (Mattern et al., 2023) estimates the curvature of the loss function around the target sample by comparing its loss to the average loss of its perturbed neighbors. The intuition is that member samples will have a lower loss compared to their neighbors, resulting in a larger difference.

$$S(\mathbf{x}; M) = Loss(\mathbf{x}; M) - \frac{1}{n}\sum_{i=1}^{n} Loss(\tilde{\mathbf{x}}_i; M) \tag{10}$$

C.5 Min-K%

The Min-K% baseline (Shi et al., 2024) computes the membership score using the average log-likelihood of the $k\%$ of tokens with the lowest probabilities. This focuses on the least likely tokens, which are expected to have higher probabilities for member samples compared to non-members.

$$S(\mathbf{x}; M) = \frac{1}{|\text{min-}k(\mathbf{x})|} \sum_{x_i \in \text{min-}k(\mathbf{x})} -\log(p(x_i \mid x_1, ..., x_{i-1})) \tag{11}$$

C.6 Min-K%++

The Min-K%++ baseline (Zhang et al., 2024) is an extension of the Min-K% that calibrates the next token log-likelihood with two factors: the mean ($\mu_{x_{<t}}$) and standard deviation ($\sigma_{x_{<t}}$) of the log-likelihood over all candidate tokens in the vocabulary.

$$S_{\text{token}}(x_{<t}, x_t; M) = \frac{\log p(x_t|x_{<t}; M) - \mu_{x_{<t}}}{\sigma_{x_{<t}}}, \tag{12}$$

$$S(\mathbf{x}; M) = \frac{1}{|\text{min-}k\%|} \sum_{(x_{<t}, x_t) \in \text{min-}k\%} f_{\text{token}}(x_{<t}, x_t; M). \tag{13}$$

$\mu_{x_{<t}}$ and $\sigma_{x_{<t}}$ are the mean and standard deviation of the log-likelihoods over the model's vocabulary distribution given the prefix $x_{<t}$, respectively. The final membership score is obtained by averaging the normalized log-likelihoods of the $k\%$ of token sequences with the lowest scores (Equation 13).

D Additional Metrics Details

D.1 Area Under the ROC Curve (AUC)

The area under the ROC curve (AUC) is a widely used metric for evaluating the performance of binary classification models, including MIAs. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various decision thresholds. The TPR, also known as sensitivity or recall, is the proportion of actual positive samples (i.e., member samples) that are correctly identified as such. The FPR, on the other hand, is the proportion of actual negative samples (i.e., non-member samples) that are incorrectly identified as positive.

The AUC ranges from 0 to 1, with a value of 0.5 indicating a random classifier and a value of 1 indicating a perfect classifier. In the context of MIAs, a higher AUC value indicates that the attack is better at distinguishing between member and non-member samples across *all possible* decision thresholds.

D.2 True Positive Rate at a Low False Positive Rate (TPR@low%FPR)

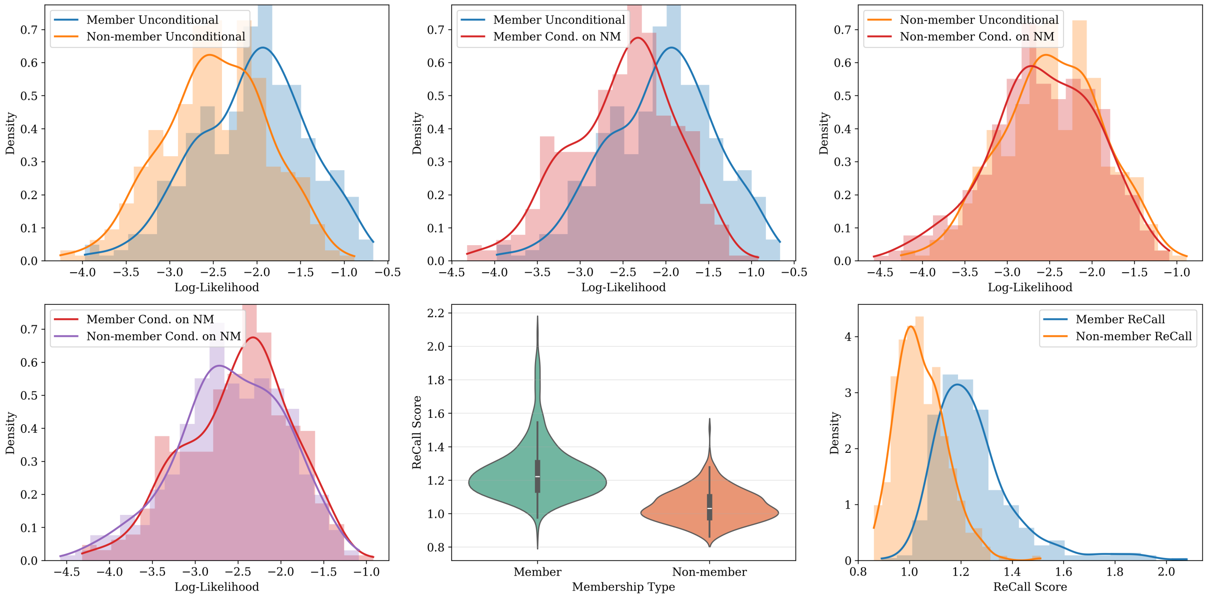
While the AUC provides an overall measure of an MIA's performance, it may not always be the most appropriate metric for practical applications. In many cases, the cost of false positives (i.e., incorrectly identifying a non-member sample as a member) can be much higher than the cost of false negatives (i.e., incorrectly identifying a member sample as a non-member). For example, in privacy-sensitive applications, falsely accusing an individual of being a member of a sensitive dataset can have severe consequences (Grynbaum and Mac, 2023).

To address this concern, we report the true positive rate at a low false positive rate (TPR@low%FPR) (Carlini et al., 2022a). In our experiments, we set the false positive rate threshold to 1%, which means that we measure the proportion of member samples that are correctly identified as such while allowing only 1% of non-member samples to be incorrectly identified as members. This metric provides a more stringent evaluation of an MIA's performance, focusing on its ability to correctly identify member samples while maintaining a low false positive rate.

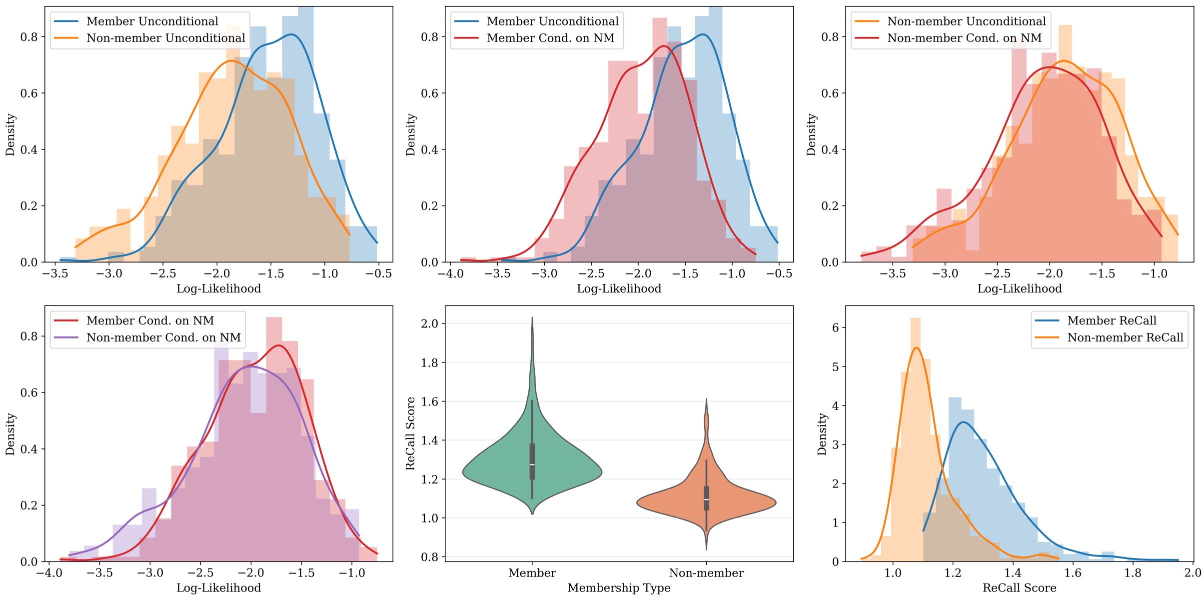
E Additional Implementation Details

We use 16-bit floating-point precision for models larger than 60B to reduce computational requirement, and experiments are all conducted on 4 NVIDIA A6000 GPUs. In some experiments, we intentionally exceed the context window to test the limit, which might result in an out-of-memory (OOM) error. To ensure a fair evaluation, we also remove 12 data points from the member set for data balance, as this is a binary classification task. We report the best number of shot used for the main results in Appendix N from the main results. It's worth noting that the Neighbor attack is significantly more computationally intensive than other methods (Duan et al., 2024; Zhang et al., 2024), as it needs to iterate through the input's neighbor. Therefore, we obtain the Neighbor attack AUC results from Zhang et al. (2024). In contrast, ReCallis computationally efficient, as it only requires two LL calculations per sample, avoiding the need for expensive operations like building reference models (Carlini et al., 2022a; Watson et al., 2021) or exploring neighboring samples (Mattern et al., 2023).

F Additional Model's Visualizations



(a) NeoX-20B



(b) LLaMA-13B

Figure 7: Visualizations for **(a) NeoX-20B** and **(b) LLaMA-13B** for WIKIMIA with 5-shot results. Similar patterns are observed that members tend to be shifted further and have higher RECALL scores compared to non-members.

G MIMIR 7-gram Results

G.1 AUC Results

Method	Wikipedia 160M	1.4B	2.8B	6.9B	12B	Github 160M	1.4B	2.8B	6.9B	12B	Pile CC 160M	1.4B	2.8B	6.9B	12B	PubMed Central 160M	1.4B	2.8B	6.9B	12B
Loss	**62.6**	65.7	66.4	68.0	69.0	84.1	87.2	88.0	88.8	89.3	**53.1**	**54.4**	54.8	56.0	56.4	**79.4**	78.7	78.4	78.5	78.4
Ref	62.3	**65.9**	**66.6**	**68.4**	69.5	83.4	87.5	88.3	89.4	90.0	52.9	**54.4**	**54.9**	56.2	**56.7**	**79.4**	78.8	78.4	78.5	78.4
Zlib	57.3	62.0	63.1	65.0	66.2	**87.9**	**89.9**	**90.6**	**91.3**	**91.7**	51.4	53.2	53.7	54.8	55.2	78.0	77.6	77.3	77.5	77.4
Min-K%	60.8	64.8	65.9	68.0	69.6	82.8	87.0	87.9	88.8	89.4	52.6	54.0	54.6	56.0	56.2	77.9	78.6	78.2	78.8	78.9
Min-K%++	62.3	63.6	64.7	68.1	**70.0**	83.1	83.2	84.8	85.5	86.9	51.5	52.6	53.7	55.7	56.1	76.9	66.1	66.6	68.3	68.8
ReCall	61.3	64.9	66.0	67.5	69.0	83.5	87.6	88.6	90.7	91.6	51.8	53.6	54.7	**56.5**	**56.7**	76.4	**78.9**	**78.9**	**81.3**	**79.8**

Method	ArXiv 160M	1.4B	2.8B	6.9B	12B	DM Mathematics 160M	1.4B	2.8B	6.9B	12B	HackerNews 160M	1.4B	2.8B	6.9B	12B	Average 160M	1.4B	2.8B	6.9B	12B
Loss	**75.5**	77.5	78.0	79.0	79.4	93.6	91.7	91.4	91.5	91.4	58.6	59.6	60.5	61.2	62.0	**72.4**	73.5	73.9	74.7	75.1
Ref	**75.5**	**77.8**	**78.2**	**79.3**	**79.8**	93.7	91.4	91.0	91.1	90.9	58.6	59.7	60.6	61.3	62.2	72.3	73.6	74.0	74.9	75.4
Zlib	74.9	76.9	77.3	78.1	78.5	81.8	81.2	81.6	81.4	81.3	57.8	58.9	59.5	59.9	60.6	69.9	71.4	71.9	72.6	73.0
Min-K%	70.6	74.2	75.3	76.7	77.7	92.9	92.5	92.4	92.4	**92.2**	55.5	56.8	58.0	59.0	60.4	70.4	72.6	73.2	74.2	74.9
Min-K%++	70.0	62.4	64.6	67.0	69.0	90.9	67.0	69.1	64.3	66.3	58.8	55.8	57.5	58.8	60.4	70.5	64.4	65.9	66.8	68.2
ReCall	75.4	76.6	77.8	77.0	77.6	**95.3**	**94.3**	**93.9**	**92.9**	92.1	**59.4**	**60.1**	**60.8**	**63.1**	**63.3**	71.9	**73.7**	**74.4**	**75.6**	**75.7**

Table 9: AUC results on the challenging MIMIR benchmark (Duan et al., 2024) in the 7-gram setting. The best result across all methods is **bolded** in each column. ReCall outperforms all baselines on the 1.4B, 2.8B, 6.9B, and 12B models in average, demonstrating its effectiveness in detecting pretraining data even when the distribution shift between members and non-members is minimized.

G.2 TPR@1%FPR Result

Method	Wikipedia 160M	1.4B	2.8B	6.9B	12B	Github 160M	1.4B	2.8B	6.9B	12B	Pile CC 160M	1.4B	2.8B	6.9B	12B	PubMed Central 160M	1.4B	2.8B	6.9B	12B
Loss	**6.9**	11.0	11.5	13.4	13.2	27.0	45.3	47.7	51.2	50.8	2.4	3.9	4.6	5.4	5.7	16.1	16.9	**19.6**	13.6	14.4
Ref	6.2	**12.4**	**12.1**	**14.3**	13.1	20.3	43.8	52.0	55.9	55.5	1.9	3.9	4.5	5.6	6.3	15.0	15.0	17.7	12.3	14.2
Zlib	3.8	7.8	8.9	10.1	10.6	**56.2**	**57.4**	**62.1**	**61.7**	**58.6**	**2.6**	**4.6**	**5.5**	**6.7**	**7.8**	16.9	14.4	14.8	12.7	10.0
Min-K%	7.2	10.3	11.8	13.9	12.6	30.9	46.9	50.0	52.3	52.7	2.4	4.5	5.0	5.4	5.9	**19.6**	**19.2**	19.2	21.5	**22.8**
Min-K%++	5.5	7.1	10.0	11.0	13.2	29.6	30.5	32.4	39.5	38.7	2.1	3.1	3.2	4.5	4.5	15.0	9.8	10.2	8.6	14.0
ReCall	6.7	10.7	11.7	13.6	**15.7**	32.3	48.8	49.6	52.7	56.2	2.1	2.9	4.0	4.9	4.8	16.3	13.8	19.4	**24.4**	22.5

Method	ArXiv 160M	1.4B	2.8B	6.9B	12B	DM Mathematics 160M	1.4B	2.8B	6.9B	12B	HackerNews 160M	1.4B	2.8B	6.9B	12B	Average 160M	1.4B	2.8B	6.9B	12B
Loss	8.6	11.3	16.0	16.6	17.0	67.5	29.9	14.3	14.3	14.3	2.2	1.9	1.9	2.8	1.9	18.7	17.2	16.5	16.8	16.8
Ref	7.4	12.3	17.8	17.8	18.4	71.4	26.0	6.5	5.2	3.9	2.4	1.9	1.9	2.8	2.1	17.8	16.5	16.1	16.3	16.2
Zlib	4.7	9.0	12.3	15.4	16.2	19.5	14.3	7.8	6.5	6.5	**2.8**	**2.4**	**2.7**	2.8	3.3	15.2	15.7	16.3	16.6	16.1
Min-K%	9.4	14.3	**21.3**	**21.1**	**21.3**	66.2	**61.0**	**44.2**	**39.0**	**37.7**	1.9	1.7	1.1	2.4	1.9	19.7	**22.6**	**21.8**	**22.2**	**22.1**
Min-K%++	**10.0**	3.1	4.9	5.5	6.1	20.3	16.9	19.5	10.4	15.6	1.8	1.9	1.4	2.4	2.4	12.0	10.3	11.7	11.7	13.5
ReCall	5.7	**15.6**	15.6	17.4	14.1	**79.2**	44.2	31.2	22.1	15.6	1.1	1.7	**2.7**	**4.9**	**4.6**	**20.5**	19.7	19.2	20.0	19.1

Table 10: TPR@1%FPR results on the challenging MIMIR benchmark (Duan et al., 2024) in 7-gram setting. The best result across all methods is **bolded** in each column.

H Synthetic Prefixes Generation

GPT-4o Prompt Template

Generate a passage that is similar to the given text in length, domain, and style.

Given text: {*a data point (could be member or non-member)*}

New passage:

I WikiMIA TPR@1%FPR Results

Len.	Method	Mamba-1.4B	Pythia-6.9B	LLaMA-13B	NeoX-20B	LLaMA-30B	OPT-66B	Average
32	Loss	4.5	6.1	4.8	10.4	4.3	6.4	6.1
	Ref	4.5	6.9	5.9	10.1	2.7	6.7	6.1
	Zlib	4.0	4.8	5.6	9.1	4.8	5.6	5.7
	Min-K%	6.7	8.8	5.1	10.7	4.5	**9.1**	7.5
	Min-K%++	4.3	5.9	10.4	6.1	9.3	3.7	6.6
	RECALL	**11.2**	**28.5**	**13.3**	**25.3**	**18.4**	8.3	**17.5**
64	Loss	3.3	3.3	4.9	4.5	6.1	4.1	4.4
	Ref	2.8	3.3	4.1	4.9	6.5	4.5	4.4
	Zlib	6.1	6.9	8.9	7.7	10.6	9.8	8.3
	Min-K%	6.9	6.5	6.5	5.7	8.1	10.2	7.3
	Min-K%++	7.3	11.8	15.4	**10.2**	6.9	**11.8**	10.6
	RECALL	**11.0**	**20.7**	**30.1**	6.9	**18.3**	5.3	**15.4**
128	Loss	1.0	3.0	7.1	4.0	1.0	4.0	3.4
	Ref	1.0	3.0	8.1	4.0	0.0	4.0	3.4
	Zlib	**6.1**	6.1	10.1	5.1	**2.0**	**9.1**	6.4
	Min-K%	3.0	4.0	8.1	3.0	**2.0**	4.0	4.0
	Min-K%++	2.0	8.1	8.1	1.0	0.0	0.0	3.2
	RECALL	4.0	**33.3**	**26.3**	**30.3**	1.0	6.1	**16.9**

Table 11: TPR@1%FPR results on WikiMIA benchmark. **Bolded** numbers show the best result within each column. Overall, RECALL consistently achieves the highest average TPR@1%FPR scores across all input lengths, demonstrating its effectiveness in detecting pretraining data with high precision.

J MIMIR 13-gram TPR@1%FPR Result

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	0.7	0.8	0.6	0.7	0.9	16.0	19.7	22.2	22.5	23.1	0.4	0.5	0.8	0.8	0.8	**0.8**	0.8	0.8	0.7	0.4
Ref	1.1	0.5	**0.7**	0.9	0.9	17.1	9.2	10.0	11.6	13.1	0.6	**0.6**	0.7	0.9	0.9	**0.8**	**0.9**	0.6	0.6	0.4
Zlib	0.8	0.7	**0.7**	0.9	**1.0**	**17.4**	**23.0**	**24.0**	**26.0**	**25.9**	0.5	**0.6**	0.9	1.1	1.1	0.5	0.5	0.3	0.6	0.5
Min-K%	1.1	0.8	0.6	0.7	0.9	15.2	20.3	21.6	22.7	23.2	0.4	0.5	0.7	0.7	0.9	0.7	0.4	0.6	0.6	0.7
Min-K%++	0.9	0.7	0.6	**1.1**	**1.0**	13.4	18.2	18.8	21.5	23.6	**0.7**	**0.6**	**1.1**	**1.2**	**1.4**	0.6	0.6	**1.0**	**1.1**	**1.2**
RECALL	**1.3**	**0.9**	**0.7**	0.7	0.8	11.6	21.5	23.1	22.5	24.6	**0.7**	0.4	0.5	0.9	1.1	0.4	0.7	0.4	0.3	0.5

	ArXiv					DM Mathematics					HackerNews					Average				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
Loss	0.5	0.3	0.6	0.7	0.7	0.7	0.6	1.0	**1.1**	**1.1**	0.8	0.6	0.6	0.7	0.8	2.8	3.3	3.8	3.9	4.0
Ref	0.6	0.3	0.7	0.8	1.0	0.7	**1.0**	**1.3**	1.0	1.0	**1.1**	0.6	0.6	0.7	1.0	**3.1**	1.9	2.1	2.4	2.6
Zlib	0.5	0.3	0.4	0.4	0.7	**1.1**	0.7	0.9	0.9	0.9	1.0	0.9	1.3	**1.3**	1.1	**3.1**	**3.8**	**4.1**	**4.5**	**4.5**
Min-K%	0.5	0.2	0.5	0.4	0.8	0.7	0.5	0.2	0.4	0.4	0.7	0.8	0.7	0.9	0.9	2.8	3.4	3.6	3.8	4.0
Min-K%++	0.5	**1.3**	**1.4**	1.0	1.8	0.6	**1.0**	1.2	0.4	0.9	0.7	0.4	1.0	**1.3**	0.5	2.5	3.3	3.6	3.9	4.3
RECALL	**1.1**	0.8	0.9	**1.4**	**2.4**	0.3	0.9	0.6	0.3	0.0	1.0	**1.6**	**1.7**	1.1	**1.5**	2.3	**3.8**	4.0	3.9	4.4

Table 12: TPR@1%FPR results on the challenging MIMIR benchmark (Duan et al., 2024) in 13-gram setting. The best result across all methods is **bolded** in each column.

K Additional Synthetic Prefix Results

Model	Len. 32	Len. 64	Len. 128
Pythia-6.9B - Synthetic	83.7	87.1	83.0
Pythia-6.9B - Real	91.6	93.0	92.6
Pythia-12B - Synthetic	85.4	90.3	86.4
Pythia-12B - Real	88.2	88.8	87.8
LLaMA-13B - Synthetic	89.2	93.2	90.5
LLaMA-13B - Real	92.2	95.2	92.5

Table 13: The performance of synthetic and real prefixes across different models (Pythia-6.9B, Pythia-12B, and LLaMA-13B). RECALL achieves comparable performance even with synthetic prefixes generated by GPT-4.

L Additional Prefix Visualizations

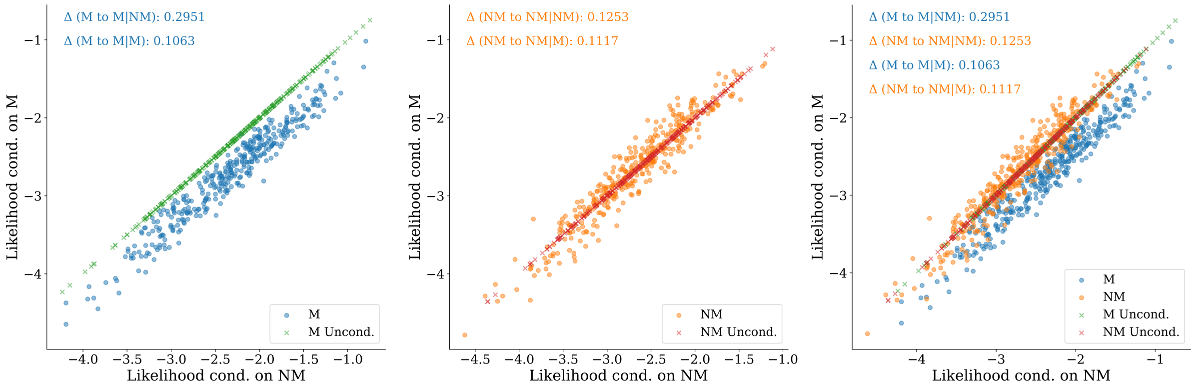


Figure 8: Conditional LL for members and non-members with member and non-member prefix comparison. Their unconditional LL are in the diagonal line. Conditioning both member and non-member data with member prefix do not yield significant changes in LL.

M Additional Token-level Results

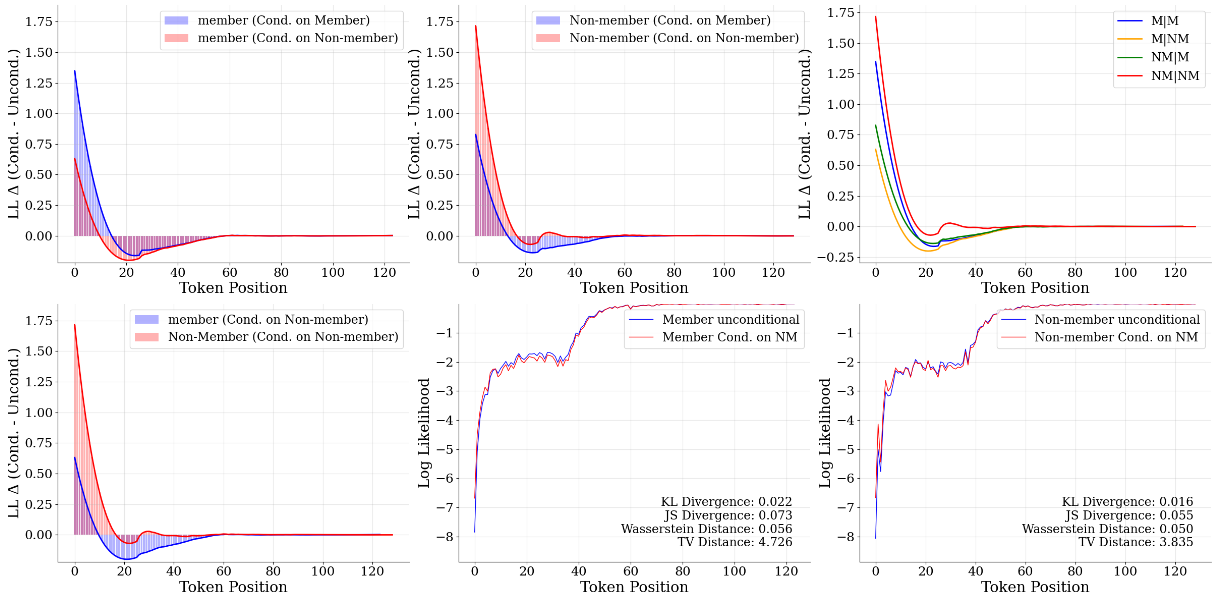


Figure 9: Average token-level log-likelihood changes for member (M) and non-member (NM) data points when prefixed with member and non-member context. The largest changes occur in the beginning tokens, and data points experience the most dominant changes when prefixed with context from the same membership category. Member and non-member data exhibit the largest differences when prefixed with non-member context, consistent with the findings in Figure 5.

N Best Number of Shot

N.1 WikiMIA

Len.	Mamba-1.4B	Pythia-6.9B	LLaMA-13B	NeoX-20B	LLaMA-30B	OPT-66B	Average
32	7	7	6	7	6	6	6.5
64	10	9	8	12	12	9	10
128	11	7	6	6	9	4	7.2

Table 14: Report on the best number of shot is used in WikiMIA main result.

N.2 MIMIR 13-gram

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
RECALL	12	8	1	8	8	10	3	3	7	7	9	9	12	5	11	12	7	11	1	1
	ArXiv					**DM Mathematics**					**HackerNews**					**Average**				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
RECALL	5	6	4	6	6	1	1	1	1	1	5	3	3	5	7	7.7	5.3	5.0	4.7	5.9

Table 15: Report on the best number of shot used in MIMIR 13-gram main result.

N.3 MIMIR 7-gram

	Wikipedia					Github					Pile CC					PubMed Central				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
RECALL	1	2	10	9	9	11	9	7	8	8	12	4	4	4	4	12	7	4	6	1
	ArXiv					**DM Mathematics**					**HackerNews**					**Average**				
Method	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B	160M	1.4B	2.8B	6.9B	12B
RECALL	12	11	12	11	11	12	12	11	10	10	11	10	8	8	8	10.1	7.9	8.0	8.0	7.3

Table 16: Report on the best number of shot used in MIMIR 7-gram main result.