CON-RECALL: Detecting Pre-training Data in LLMs via Contrastive Decoding

Cheng Wang[†], Yiwei Wang[§], Bryan Hooi[†], Yujun Cai[‡], Nanyun Peng[§], Kai-Wei Chang[§]
[†] National University of Singapore
[§] University of California, Los Angeles   [‡] University of Queensland
wcheng@comp.nus.edu.sg

Abstract

The training data in large language models is key to their success, but it also presents privacy and security risks, as it may contain sensitive information. Detecting pre-training data is crucial for mitigating these concerns. Existing methods typically analyze target text in isolation or solely with non-member contexts, overlooking potential insights from simultaneously considering both member and non-member contexts. While previous work suggested that member contexts provide little information due to the minor distributional shift they induce, our analysis reveals that these subtle shifts can be effectively leveraged when contrasted with non-member contexts. In this paper, we propose CON-RECALL, a novel approach that leverages the asymmetric distributional shifts induced by member and non-member contexts through contrastive decoding, amplifying subtle differences to enhance membership inference. Extensive empirical evaluations demonstrate that CON-RECALL achieves state-of-the-art performance on the WikiMIA benchmark and is robust against various text manipulation techniques. [1]

1 Introduction

Large Language Models (LLMs) (OpenAI, 2024a; Touvron et al., 2023b) have revolutionized natural language processing by achieving remarkable performance across a wide range of language tasks. These models owe their success to extensive training datasets, often encompassing trillions of tokens. However, the sheer volume of these datasets makes it practically infeasible to meticulously filter out all inappropriate data points. Consequently, LLMs may unintentionally memorize sensitive information, raising significant privacy and security concerns. This memorization can include test data from benchmarks (Sainz et al., 2023; Oren et al.,
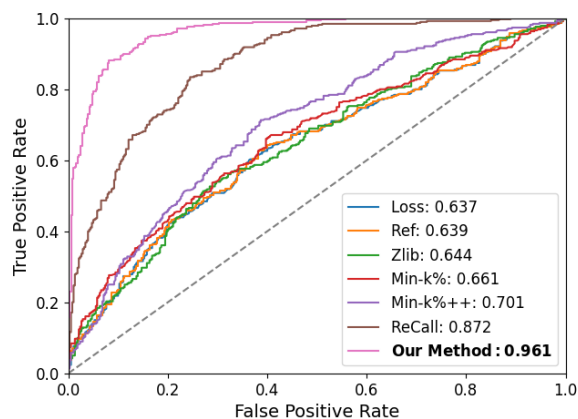

Figure 1: **AUC performance on WikiMIA-32 dataset.** Our CON-RECALL significantly outperforms the current state-of-the-art baselines.

2023), copyrighted materials (Meeus et al., 2023; Duarte et al., 2024; Chang et al., 2023), and personally identifiable information (Mozes et al., 2023; Tang et al., 2024), leading to practical issues such as skewed evaluation results, potential legal ramifications, and severe privacy breaches. Therefore, developing effective techniques to detect unintended memorization in LLMs is crucial.

Existing methods for detecting pre-training data (Yeom et al., 2018; Zhang et al., 2024; Xie et al., 2024) typically analyze target text either in isolation or alongside with non-member contexts, while commonly neglecting member contexts. This omission is based on the belief that member contexts induce only minor distributional shifts, offering limited additional value (Xie et al., 2024).

However, our analysis reveals that these subtle shifts in member contexts, though often dismissed, hold valuable information that has been underexploited. The central insight of our work is that information derived from member contexts gains significant importance when contrasted with non-member contexts. This observation led to the development of CON-RECALL, a novel approach that harnesses the contrastive power of prefixing

---

[1] Our code is available at the following repo: https://github.com/WangCheng0116/CON-RECALL

Method	Formula	Reference Based
Loss (Yeom et al., 2018)	$\mathcal{L}(x, \mathcal{M})$	✗
Ref (Carlini et al., 2022)	$\mathcal{L}(x, \mathcal{M}) - \mathcal{L}(x, \mathcal{M}_{ref})$	✓
Zlib (Carlini et al., 2021)	$\frac{\mathcal{L}(x,\mathcal{M})}{zlib(x)}$	✗
Neighborhood Attack (Mattern et al., 2023)	$\mathcal{L}(x; \mathcal{M}) - \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(\tilde{x}_i; \mathcal{M})$	✗
Min-K% (Shi et al., 2024a)	$\frac{1}{\lvert\text{min-}k(x)\rvert} \sum_{x_i \in \text{min-}k(x)} -\log(p(x_i \mid x_1,\ldots,x_{i-1}))$	✗
Min-K%++ (Zhang et al., 2024)	$\text{Min-K\%++}_{\text{token}}(x_t) = \frac{\log p(x_t \mid x_{<t}) - \mu_{x_{<t}}}{\sigma_{x_{<t}}},$ $\text{Min-K\%++}(x) = \frac{1}{\lvert\text{min-k\%}\rvert} \sum_{x_t \in \text{min-k\%}} \text{Min-K\%++}_{\text{token}}(x_t)$	✗
ReCall (Xie et al., 2024)	$\frac{LL(x\mid P_{\text{non-member}})}{LL(x)}$	✗

Table 1: **Comparison of baseline methods.** This table provides an overview of different membership inference methods, their mathematical formulations, and whether they require a reference model.

target text with both member and non-member contexts. By exploiting the asymmetric distributional shifts induced by these different prefixes, CON-RECALL provides more nuanced and reliable signals for membership inference. This contrastive strategy not only uncovers previously overlooked information but also enhances the accuracy and robustness of pre-training data detection, offering a more comprehensive solution than existing methods.

To demonstrate the effectiveness of CON-RECALL, we conduct extensive empirical evaluations on the method across a variety of models of different sizes. Our experiments show that CON-RECALL outperforms the current state-of-the-art method by a significant margin, as shown in Figure 1. Notably, CON-RECALL only requires a gray-box access to LLMs, i.e., token probabilities, and does not necessitate a reference model, enhancing its applicability in real-world scenarios.

2 Related Work

Detecting Pre-training Data in LLMs. While membership inference attacks (MIA) have been extensively studied in various domains (Shokri et al., 2017; Carlini et al., 2023a; Watson et al., 2022), detecting pre-training data in LLMs presents unique challenges. Unlike classical MIA, LLM developers rarely release full training data (OpenAI, 2024a; Touvron et al., 2023b), and single-epoch training on vast datasets makes memorization detection difficult (Carlini et al., 2023b; Shi et al., 2024a). Shi et al. (2024a) pioneered this research with the WikiMIA benchmark and Min-K% baseline method. Zhang et al. (2024) improved Min-K% through token log-probability normalization, while the ReCall method (Xie et al., 2024) currently

achieves state-of-the-art performance using relative conditional log-likelihoods. These methods contribute to the broader application of MIA in detecting copyrighted materials, personally identifiable information, and test-set contamination (Meeus et al., 2023; Mozes et al., 2023; Sainz et al., 2023).

Contrastive Decoding. Contrastive decoding is primarily a method for text generation. Depending on the elements being contrasted, it serves different purposes. For example, DExperts (Liu et al., 2021) use outputs from a model exposed to toxicity to guide the target model away from undesirable outputs. Context-aware decoding (Shi et al., 2024b) contrasts model outputs given a query with and without relevant context. Zhao et al. (2024) further enhance context-aware decoding by providing irrelevant context in addition to relevant context. In this paper, we adapt the idea of contrastive decoding to MIA, where the contrast occurs between target data prefixed with member and non-member contexts.

3 CON-RECALL

3.1 Problem Formulation

Consider a model $\mathcal{M}$ trained on dataset $\mathcal{D}$. The objective of a membership inference attack is to ascertain whether a data point x belongs to $\mathcal{D}$ (i.e., $x \in \mathcal{D}$) or not (i.e., $x \notin \mathcal{D}$). Formally, we aim to develop a scoring function $s(x, \mathcal{M}) \to \mathbb{R}$, where the membership prediction is determined by a threshold τ:

$$\begin{cases} x \in \mathcal{D} & \text{if } s(x, \mathcal{M}) \geq \tau \\ x \notin \mathcal{D} & \text{if } s(x, \mathcal{M}) < \tau \end{cases}.$$

3.2 Motivation

Our key insight is that prefixing target text with contextually similar content increases its log-
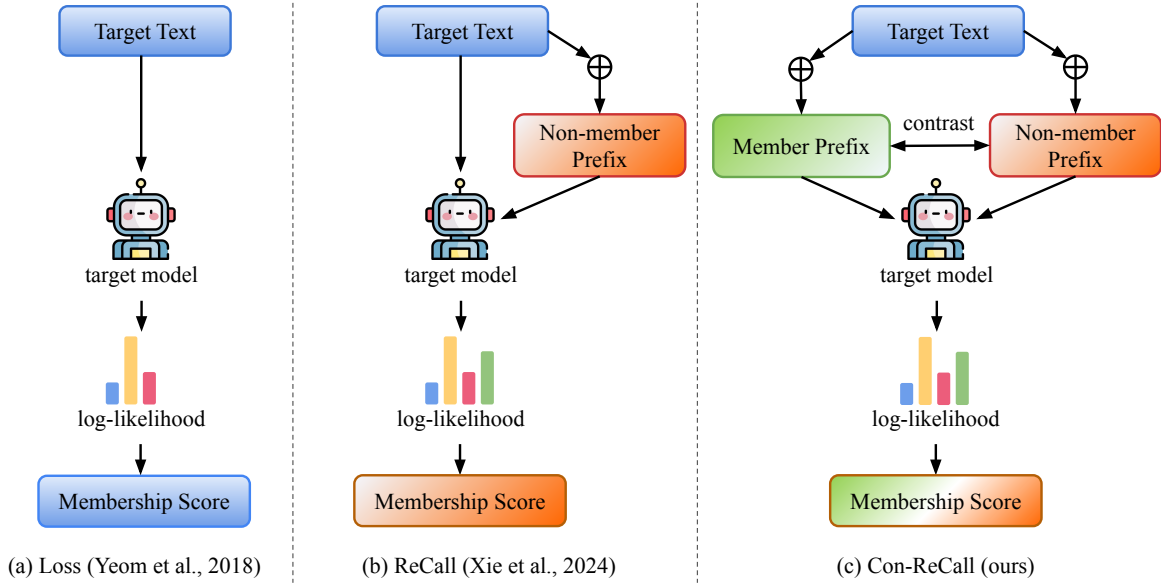
Figure 2: **Overview of three MIA methods**. Our method refines the previous membership score by incorporating contrastive information when prefixing target text with members and non-members.
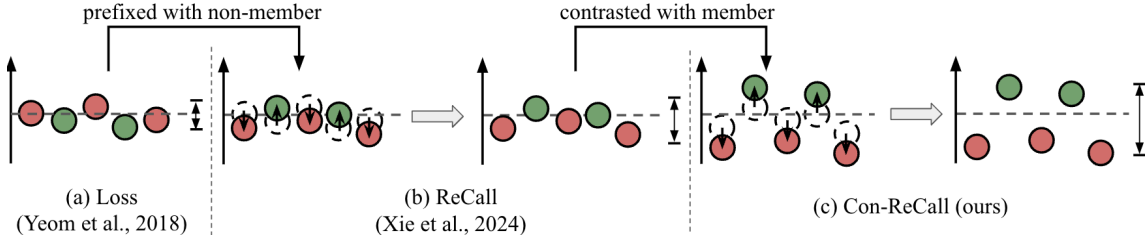

Figure 3: **Distribution shifts induced by three methods.** (a) Loss directly uses log-likelihoods, resulting in no shift. (b) ReCall examines the shift caused by non-member prefixes. (c) Our CON-RECALL enhances the distinction by contrasting with both member and non-member prefixes.

likelihood, while dissimilar content decreases it. Member prefixes boost log-likelihoods for member data but reduce them for non-member data, with non-member prefixes having the opposite effect. This principle stems from language models' fundamental tendency to generate contextually consistent text.

To quantify the impact of different prefixes, we use the Wasserstein distance to measure the distributional shifts these prefixes induce. For discrete probability distributions P and Q defined on a finite set X, the Wasserstein distance W is given by:

$$W(P,Q) = \sum_{x \in X} |F_P(x) - F_Q(x)|,$$

where F_P and F_Q are the cumulative distribution functions of P and Q respectively. To capture the directionality of the shift, we introduce a signed variant of this metric:

$$W_{\text{signed}}(P,Q) = \text{sign}(\mathbb{E}_Q[X] - \mathbb{E}_P[X]) \cdot W(P,Q).$$

Our experiments reveal striking asymmetries in how member and non-member data respond to different prefixes. Figure 5 illustrates these asymmetries, showing the signed Wasserstein distances between original and prefixed distributions across varying numbers of shots, where shots refer to the number of non-member data points used in the prefix.

We observe two key phenomena:

1. **Asymmetric Shift Direction**: Member data exhibits minimal shift when prefixed with other member contexts, indicating a degree of distributional stability. However, when prefixed with non-member contexts, it undergoes a significant negative shift. In contrast, non-member data displays a negative shift when prefixed with member contexts and a positive shift with non-member prefixes.
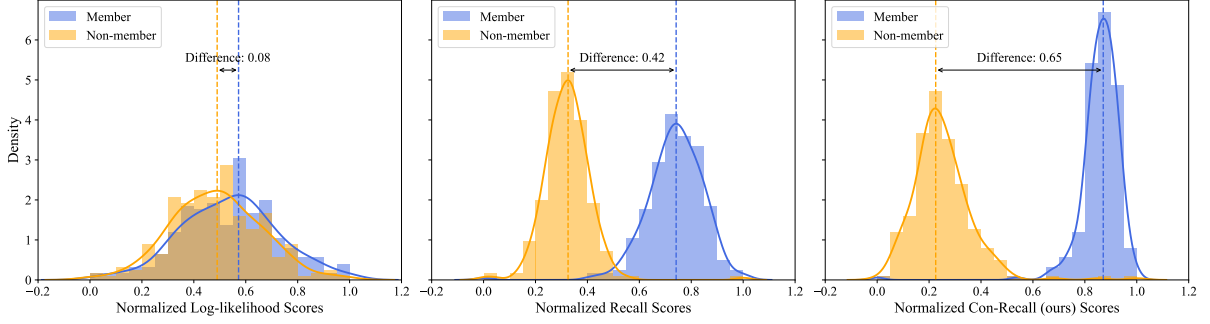
1015

Figure 4: **Visualization of membership score distributions.** Min-max normalized distributions are shown for log-likelihood (left), ReCall (middle), and CON-RECALL (right). CON-RECALL achieves the largest separation between members and non-members.
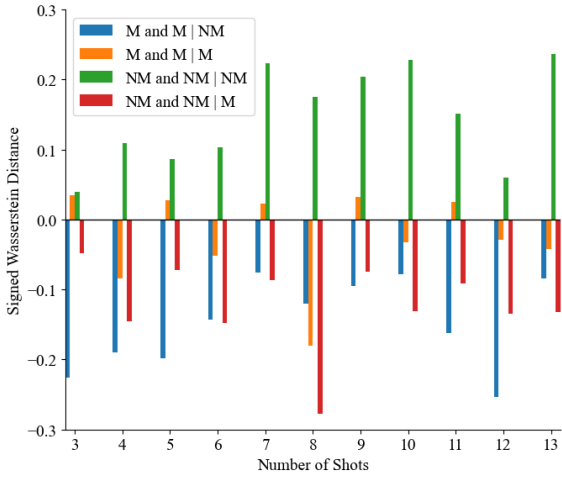

Figure 5: **Signed Wasserstein distances between original and prefixed distributions across varying shot numbers.** The plot illustrates how the distributional shift, measured by signed Wasserstein distance, changes for member and non-member data when prefixed with different contexts (M: member, NM: non-member).

2. **Asymmetric Shift Intensity**: Non-member data demonstrated heightened sensitivity to contextual modifications, manifesting as larger magnitude shifts in the probability distribution, regardless of the prefix type. Member data, while generally more stable, still exhibited notable sensitivity, particularly to non-member prefixes.

These results corroborate our initial analysis and establish a robust basis for our contrastive approach. The asymmetric shifts in both direction and intensity provide crucial insights for developing a membership inference technique that leverages these distributional differences effectively.

3.3 Contrastive Decoding with Member and Non-member Prefixes

Building on the insights from our analysis, we propose CON-RECALL, a method that exploits the contrastive information between member and non-member prefixes to enhance membership inference through contrastive decoding. Our approach is directly motivated by the two key observations from the previous section:

1. The asymmetric shift direction suggests that comparing the effects of member and non-member prefixes could provide a strong signal for membership inference.

2. The asymmetric shift intensity indicates the need for a mechanism to control the relative importance of these effects in the decoding process.

These insights lead us to formulate the membership score $s(x, M)$ for a target text x and model M as follows:

$$\frac{LL(x|P_{\text{non-member}}) - \gamma \cdot LL(x|P_{\text{member}})}{LL(x)},$$

where $LL(\cdot)$ denotes the log-likelihood, P_{member} and $P_{non-member}$ are prefixes composed of member and non-member contexts respectively, and γ is a parameter controlling the strength of the contrast.

This formulation provides a robust signal for membership inference by leveraging the distributional differences revealed in our analysis. Figure 3 illustrates how our contrastive approach amplifies the distributional differences

Importantly, CON-RECALL requires only gray-box access to the model, utilizing solely token probabilities. This characteristic enhances its practical

utility in real-world applications where full model access may not be available, making it a versatile tool for detecting pre-training data in large language models.

4 Experiments

In this section, we will evaluate the effectiveness of CON-RECALL across various experimental settings, demonstrating its superior performance compared to existing methods.

4.1 Setup

Baselines. In our experiment, we evaluate CON-RECALL against seven baseline methods. *Loss* (Yeom et al., 2018) directly uses the loss of the input as the membership score. *Ref* (Carlini et al., 2022) requires another reference model, which is trained on a dataset with a distribution similar to $\mathcal{D}$, to calibrate the loss calculated in the *Loss* method. *Zlib* (Carlini et al., 2021) instead calibrates the loss by using the input's Zlib entropy. *Neighbor* (Mattern et al., 2023) perturbs the input sequence to generate n neighbor data points, and the loss of x is compared with the average loss of the n neighbors. *Min-K%* (Shi et al., 2024a) is based on the intuition that a member sequence should have few outlier words with low probability; hence, the top-k% words having the minimum probability are averaged as the membership score. *Min-K%++* (Zhang et al., 2024) is a normalized version of *Min-K%* with some improvements. *ReCall* (Xie et al., 2024) calculates the relative conditional log-likelihood between x and x prefixed with a non-member contexts $P_{\text{non-member}}$. More details can be found in Table 1.

Datasets. We primarily use WikiMIA (Shi et al., 2024a) as our benchmark. WikiMIA consists of texts from Wikipedia, with members and non-members determined using the knowledge cutoff time, meaning that texts released after the knowledge cutoff time of the model are naturally non-members. WikiMIA is divided into three subsets based on text length, denoted as WikiMIA-32, WikiMIA-64, and WikiMIA-128.

Another more challenging benchmark is MIMIR (Duan et al., 2024), which is derived from the Pile (Gao et al., 2020) dataset. The benchmark is constructed using a train-test split, effectively minimizing the temporal shift present in WikiMIA, thereby ensuring a more similar distribution between members and non-members. More details

about these two benchmarks are presented in Appendix A.

Models. For the WikiMIA benchmark, we use Mamba-1.4B (Gu and Dao, 2024), Pythia-6.9B (Biderman et al., 2023), GPT-NeoX-20B (Black et al., 2022), and LLaMA-30B (Touvron et al., 2023a), consistent with Xie et al. (2024). For the MIMIR benchmark, we use models from the Pythia family, specifically 2.8B, 6.9B, and 12B. Since Ref (Carlini et al., 2022) requires a reference model, we use the smallest version of the model from that series as the reference model, for example, Pythia-70M for Pythia models, consistent with previous works (Shi et al., 2024a; Zhang et al., 2024; Xie et al., 2024).

Metrics. Following the standard evaluation metrics (Shi et al., 2024a; Zhang et al., 2024; Xie et al., 2024), we report the AUC (area under the ROC curve) to measure the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). We also include TPR at low FPRs (TPR@5%FPR) as an additional metrics.

Implementation Details. For Min-K% and Min-K%++, we vary the hyperparameter k from 10 to 100 in steps of 10. For CON-RECALL, we optimize γ from 0.1 to 1.0 in steps of 0.1. Following Xie et al. (2024), we use seven shots for both ReCall and CON-RECALL on WikiMIA. For MIMIR, due to its increased difficulty, we vary the number of shots from 1 to 10. In all cases, we report the best performance. For more details, see Appendix B.

4.2 Results

Results on WikiMIA. Table 2 summarizes the experimental results on WikiMIA, demonstrating CON-RECALL's significant improvements over baseline methods. In terms of AUC performance, our method improved upon ReCall by 7.4%, 6.6%, and 5.7% on WikiMIA-32, -64, and -128 respectively, achieving an average improvement of 6.6% and state-of-the-art performance. For TPR@5%FPR, CON-RECALL outperformed the runner-up by even larger margins: 30.0%, 34.8%, and 27.6% on WikiMIA-32, -64, and -128 respectively, with an average improvement of 30.8%. Notably, CON-RECALL achieves the best performance across models of different sizes, from Mamba-1.4B to LLaMA-30B, demonstrating its robustness and effectiveness. The consistent performance across varying sequence lengths suggests that CON-RECALL effectively identifies member-

Len.	Method	Mamba-1.4B		Pythia-6.9B		NeoX-20B		LLaMA-30B		Average	
		AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR	AUC	TPR@5%FPR
32	Loss (Yeom et al., 2018)	60.9	13.2	63.7	14.5	68.9	20.8	69.4	18.2	65.7	16.7
	Ref (Carlini et al., 2022)	61.2	13.4	63.9	13.7	69.1	20.3	69.9	18.7	66.0	16.5
	Zlib (Carlini et al., 2021)	62.1	15.0	64.4	16.3	69.3	20.5	69.9	14.7	66.4	16.6
	Neighbor (Mattern et al., 2023)	64.1	11.9	65.8	16.5	70.2	22.2	67.6	9.3	66.9	15.0
	Min-K% (Shi et al., 2024a)	63.2	13.9	66.1	17.1	72.0	28.7	70.1	19.5	67.9	19.8
	Min-K%++ (Zhang et al., 2024)	66.8	12.1	70.0	13.7	75.7	17.9	84.6	27.1	74.3	17.7
	ReCall (Xie et al., 2024)	88.6	43.2	87.0	42.9	86.7	44.7	91.4	49.7	88.4	45.1
	CON-RECALL (ours)	**94.4**	**68.4**	**96.0**	**77.1**	**95.2**	**67.6**	**97.4**	**87.4**	**95.8**	**75.1**
64	Loss (Yeom et al., 2018)	57.8	9.6	60.3	13.1	66.1	16.7	66.1	14.7	62.6	13.5
	Ref (Carlini et al., 2022)	58.1	10.0	60.4	13.5	66.3	15.5	67.0	15.5	63.0	13.6
	Zlib (Carlini et al., 2021)	59.5	12.7	61.6	13.9	67.3	17.5	67.1	16.3	63.9	15.1
	Neighbor (Mattern et al., 2023)	60.6	8.8	63.2	10.9	67.1	13.0	67.1	9.9	64.5	10.7
	Min-K% (Shi et al., 2024a)	61.7	18.7	64.6	17.1	72.5	27.1	68.5	17.1	66.8	20.0
	Min-K%++ (Zhang et al., 2024)	66.9	13.1	71.4	15.1	76.3	23.5	85.3	34.7	75.9	21.6
	ReCall (Xie et al., 2024)	91.0	51.0	90.6	47.4	90.0	45.0	92.7	51.4	91.1	48.7
	CON-RECALL (ours)	**98.6**	**89.2**	**98.2**	**88.8**	**97.0**	**75.7**	**96.9**	**80.5**	**97.7**	**83.5**
128	Loss (Yeom et al., 2018)	63.5	11.5	65.3	14.4	70.3	17.3	70.0	22.1	67.3	16.3
	Ref (Carlini et al., 2022)	63.5	13.5	65.3	15.4	70.5	18.3	70.9	22.1	67.6	17.3
	Zlib (Carlini et al., 2021)	65.3	16.3	67.2	19.2	71.5	19.2	71.2	18.3	68.8	18.3
	Neighbor (Mattern et al., 2023)	64.8	15.8	67.5	10.8	71.6	15.8	72.2	15.1	69.0	14.4
	Min-K% (Shi et al., 2024a)	66.9	8.7	69.6	16.3	76.0	25.0	73.4	23.1	71.5	18.3
	Min-K%++ (Zhang et al., 2024)	67.1	9.6	69.2	17.3	75.2	20.2	83.4	21.2	73.7	17.1
	ReCall (Xie et al., 2024)	88.2	42.3	90.7	55.8	90.0	51.9	91.2	43.3	90.0	48.3
	CON-RECALL (ours)	**94.8**	**77.9**	**96.6**	**84.6**	**95.3**	**67.3**	**96.1**	**74.0**	**95.7**	**75.9**

Table 2: **AUC and TPR@5%FPR results on WikiMIA benchmark. Bolded** number shows the best result within each column for the given length. CON-RECALL achieves significant improvements over all existing baseline methods in all settings.

ship information in both short and long text samples, underlining its potential as a powerful tool for detecting pre-training data in large language models in diverse scenarios.

Results on MIMIR. We summarize the experimental results on MIMIR in Appendix D. The performance of CON-RECALL on the MIMIR benchmark demonstrates its competitive edge across various datasets and model sizes. In the 7-gram setting, CON-RECALL consistently achieved top-tier results, often outperforming baseline methods. Notably, on several datasets, our method frequently secured the highest scores in both AUC and TPR metrics. In the 13-gram setting, CON-RECALL maintained its strong performance, particularly with larger model sizes. While overall performance decreased compared to the 7-gram setting, still held leading positions across multiple datasets. It's worth noting that CON-RECALL exhibited superior performance when dealing with larger models, indicating good scalability for more complex and larger language models. Although other methods occasionally showed slight advantages in certain datasets, CON-RECALL's overall robust performance underscores its potential as an effective method for detecting pre-training data in large language models.
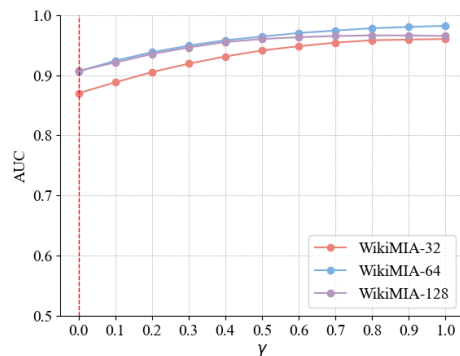
Figure 6: **Ablation on γ.** The plot illustrates the AUC performance across different γ values for the WikiMIA dataset. The red vertical line marks the $\gamma = 0$ case, where the CON-RECALL reverts to the baseline ReCall method. As seen in this figure, CON-RECALL ($\gamma > 0$) consistently outperforms ReCall ($\gamma = 0$).

4.3 Ablation Study

We focus on WikiMIA with the Pythia-6.9B model for ablation study.

Ablation on γ. In CON-RECALL, we introduce a hyperparameter γ, which controls the contrastive strength between member and non-member prefixes. The AUC performance across different γ values for the WikiMIA dataset is depicted in Figure 6. The red vertical lines mark the $\gamma = 0$ case, where CON-RECALL reverts to the baseline ReCall method.

The performance of CON-RECALL fluctuates as

Len.	Method	Pythia-6.9B								LLaMA-30B							
		Orig.	Random Del.			Synonym Sub.			Para.	Orig.	Random Del.			Synonym Sub.			Para.
			10%	15%	20%	10%	15%	20%			10%	15%	20%	10%	15%	20%	
32	Loss (Yeom et al., 2018)	63.7	60.4	59.6	56.6	61.5	59.6	59.5	63.8	69.4	66.3	67.0	64.5	68.4	66.8	65.8	70.1
	Ref (Carlini et al., 2022)	63.9	60.6	59.7	56.6	61.6	59.7	59.6	63.9	69.9	66.4	67.2	64.7	68.6	66.8	66.1	70.5
	Zlib (Carlini et al., 2021)	64.4	61.2	60.2	58.4	62.2	60.7	60.8	64.0	69.9	66.8	66.9	64.9	68.8	67.2	66.5	70.3
	Min-K% (Shi et al., 2024a)	66.1	60.5	59.6	56.6	61.7	59.9	59.6	64.8	70.1	66.3	67.0	64.6	68.4	66.8	65.8	70.4
	Min-K%++ (Zhang et al., 2024)	70.0	59.0	54.5	51.6	62.5	59.8	60.1	67.6	84.6	71.6	68.2	67.1	76.9	73.5	70.1	81.2
	ReCall (Xie et al., 2024)	87.0	86.2	83.3	75.2	88.5	87.5	83.1	87.8	91.4	88.1	88.3	82.7	87.1	86.4	84.2	91.0
	CON-RECALL (ours)	**96.0**	**92.2**	**94.4**	**90.4**	**96.5**	**94.0**	**90.0**	**97.1**	**97.4**	**97.4**	**95.5**	**94.3**	**97.6**	**95.5**	**90.0**	**97.1**
64	Loss (Yeom et al., 2018)	60.3	58.3	56.4	57.7	59.6	58.1	56.5	58.5	66.1	65.4	61.9	63.4	65.3	63.5	62.3	65.1
	Ref (Carlini et al., 2022)	60.4	58.4	56.5	57.8	59.6	58.2	56.6	58.7	67.0	65.9	62.2	63.7	65.9	64.0	62.7	65.8
	Zlib (Carlini et al., 2021)	61.6	60.9	57.8	60.0	61.8	59.9	58.2	60.5	67.1	67.2	62.6	65.4	67.1	65.0	63.5	66.7
	Min-K% (Shi et al., 2024a)	64.6	59.2	57.4	57.7	61.4	58.5	57.0	60.0	68.5	66.1	62.4	63.4	65.4	63.6	62.3	65.2
	Min-K%++ (Zhang et al., 2024)	71.4	55.9	55.8	52.3	62.8	56.3	59.1	64.4	85.3	69.1	70.4	68.7	72.1	67.1	68.0	75.1
	ReCall (Xie et al., 2024)	90.6	87.5	84.6	84.4	89.2	85.4	87.5	89.7	92.7	89.3	87.5	86.7	91.2	86.5	83.8	94.7
	CON-RECALL (ours)	**98.2**	**96.3**	**94.3**	**96.3**	**97.7**	**95.4**	**96.6**	**97.9**	**96.9**	**96.1**	**97.4**	**96.4**	**97.8**	**97.1**	**95.8**	**97.6**
128	Loss (Yeom et al., 2018)	65.3	64.6	60.4	58.8	63.1	62.4	66.4	65.0	70.0	71.1	65.9	67.3	68.5	67.1	71.4	69.2
	Ref (Carlini et al., 2022)	65.3	64.8	60.5	58.9	63.2	62.4	66.4	65.0	70.9	71.6	66.1	67.5	69.3	67.4	72.0	69.8
	Zlib (Carlini et al., 2021)	67.2	65.9	61.3	62.0	65.1	64.8	67.8	66.9	71.2	71.4	67.5	68.5	70.7	69.0	72.6	71.0
	Min-K% (Shi et al., 2024a)	69.6	65.2	60.8	58.8	65.4	63.0	66.9	67.4	73.4	73.2	67.6	67.9	70.5	67.3	72.2	70.4
	Min-K%++ (Zhang et al., 2024)	69.2	55.2	43.0	45.2	64.8	49.5	48.6	65.1	83.4	68.8	65.5	64.4	72.0	62.3	68.3	73.8
	ReCall (Xie et al., 2024)	90.7	81.8	80.4	80.0	89.6	85.5	84.2	90.4	91.2	83.2	78.2	87.3	81.4	82.4	82.1	90.9
	CON-RECALL (ours)	**96.6**	**94.8**	**93.8**	**93.8**	**97.4**	**93.6**	**96.3**	**95.9**	**96.1**	**95.3**	**91.1**	**99.0**	**95.6**	**92.5**	**94.2**	**95.2**

Table 3: **AUC performance on the WikiMIA benchmark under various text manipulation techniques. Bolded** numbers indicate the best result within each column for the given text length. "Orig." denotes original text without manipulation, "Random Del." refers to random deletion, "Synonym Sub." to synonym substitution, and "Para." to paraphrasing. Our method demonstrates robustness against these manipulations, consistently outperforming other baselines across different text modifications.

γ varies, meaning that there exist an optimal value for γ for us to get the best performance. However, even without any fine-tuning on γ, our method still outperforms ReCall and other baselines.

Ablation on the number of shots. The prefix is derived by concatenating a series of member or non-member strings, i.e., $P = p_1 \oplus p_2 \oplus \cdots \oplus p_n$, and we refer to the number of strings as shots following Xie et al. (2024)'s convention. In this section, we evaluate the relationship between AUC performance and the number of shots. We vary the number of shots on the WikiMIA dataset using the Pythia-6.9B model, and summarize the results in Figure 7.

The general trend shows that increasing the number of shots improves the AUC, as more shots provide more information. Both ReCall and CON-RECALL exhibit this trend, but CON-RECALL significantly enhances the AUC compared to ReCall and outperforms all baseline methods.

5 Analysis

To further evaluate the effectiveness and practicality of CON-RECALL, we conducted additional analyses focusing on its robustness and adaptability in real-world scenarios.

5.1 Robustness of CON-RECALL

As membership inference attacks gain prominence, evaluating their robustness against potential evasion techniques becomes crucial. Real-world data may be altered due to preprocessing, language variations, or intentional obfuscation. Therefore, a robust membership inference method should remain effective when faced with modified target data. To assess CON-RECALL's robustness, we employ three text manipulation techniques. First, Random Deletion, where we randomly remove 10%, 15%, and 20% of words from the original text. Second, Synonym Substitution, replacing 10%, 15%, and 20% of words with their synonyms using WordNet (Miller, 1994). Lastly, we leverage the WikiMIA-paraphrased dataset (Zhang et al., 2024), which offers ChatGPT-generated rephrased versions of the original WikiMIA (Shi et al., 2024a) texts while preserving their meaning.

We evaluate the effectiveness of baselines and CON-RECALL after transforming texts using the above techniques. Our experiments are conducted using Pythia-6.9B (Biderman et al., 2023) and LLaMA-30B (Touvron et al., 2023a) models on the WikiMIA-32 (Shi et al., 2024a) dataset. Table 3 presents the AUC performance for each method under various text manipulation scenarios. The results demonstrate that CON-RECALL consistently
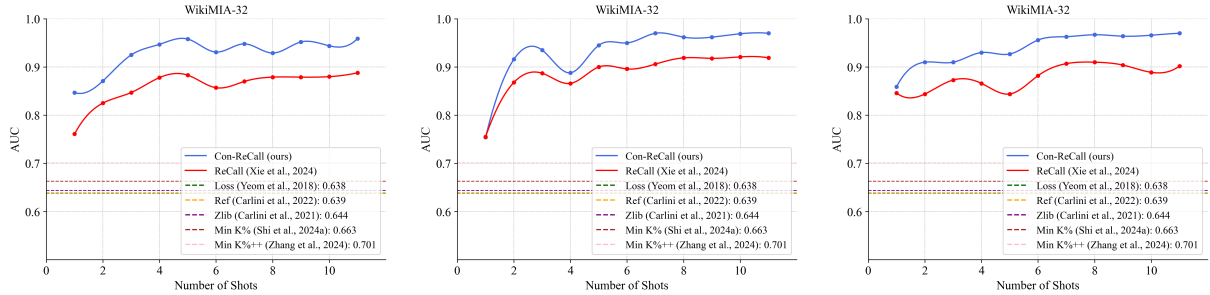
Figure 7: **Ablation on the number of shots.** CON-RECALL consistently outperforms all baseline methods by a great margin on WikiMIA dataset.

outperforms baseline methods across all text manipulation techniques, maintaining its superior performance even when faced with altered versions of the target data. This robustness underscores CON-RECALL's effectiveness in real-world scenarios where data may undergo various transformations.

5.2 Approximation of Members

In real-world scenarios, access to member data may be limited or even impossible. Therefore, it is crucial to develop methods that can approximate member data effectively. Our approach to approximating members is driven by two primary motivations. First, large language models (LLMs) are likely to retain information about significant events that occurred before their knowledge cutoff date. This retention suggests that LLMs have the potential to recall and replicate crucial aspects of such events when prompted. Second, when presented with incomplete information and tasked with its completion, LLMs can effectively leverage their internalized knowledge to generate contextually appropriate continuations. These two motivations underpin our method, where we first utilize an external LLM to enumerate major historical events. We then truncate these events and prompt the target LLM to complete them, hypothesizing that the generated content can serve as an effective approximation of the original data within the training set.

To test this approach, we first employed GPT-4o (OpenAI, 2024b) to generate descriptions of seven major events that occurred before 2020 (the knowledge cutoff date for the Pythia models). We then truncated these descriptions and prompted the target model to complete them. This method allows us to simulate the generation of data resembling the original members without directly accessing the original training set. Details of the prompts and the corresponding responses can be found in

Appendix C.

We evaluated this method using a fixed number of seven shots for consistency with our previous experiments. The results, summarized in Table 4, demonstrate that even without prior knowledge of actual member data, this approximation approach yields competitive results, outperforming several baseline methods.

This finding suggests that when direct access to member data is not feasible, leveraging the model's own knowledge to generate member-like content can be an effective alternative.

Method	WikiMIA-32	WikiMIA-64	WikiMIA-128
Loss (Yeom et al., 2018)	63.7	60.3	65.3
Ref (Carlini et al., 2022)	63.9	60.4	65.3
Zlib (Carlini et al., 2021)	64.4	61.6	67.2
Neighbor (Mattern et al., 2023)	65.8	63.2	67.5
Min-K% (Shi et al., 2024a)	66.1	64.6	69.6
Min-K%++ (Zhang et al., 2024)	70.0	71.4	69.2
ReCall (Xie et al., 2024)	87.0	90.6	90.7
CON-RECALL (zero access)	87.5	91.8	91.2
CON-RECALL (partial access)	**96.1**	**98.2**	**96.6**

Table 4: AUC results on WikiMIA benchmark. Gray rows are our method and **bolded** numbers are the best performance within a column with underline indicating the runner-up.

6 Conclusion

In this paper, we introduced CON-RECALL, a novel contrastive decoding approach for detecting pre-training data in large language models. By leveraging both member and non-member contexts, CON-RECALL significantly enhances the distinction between member and non-member data. Through extensive experiments on multiple benchmarks, we demonstrated that CON-RECALL achieves substantial improvements over existing baselines, highlighting its effectiveness in detecting pre-training data. Moreover, CON-RECALL showed robustness against various text manipulation techniques, including random deletion, syn-

onym substitution, and paraphrasing, maintaining superior performance and resilience to potential evasion strategies. These results underscore CON-RECALL's potential as a powerful tool for addressing privacy and security concerns in large language models, while also opening new avenues for future research in this critical area.

Limitations

The efficacy of CON-RECALL is predicated on gray-box access to the language model, permitting its application to open-source models and those providing token probabilities. However, this prerequisite constrains its utility in black-box scenarios, such as API calls or online chat interfaces. Furthermore, the performance of CON-RECALL is contingent upon the selection of member and non-member prefixes. The development of robust, automated strategies for optimal prefix selection remains an open research question. While our experiments demonstrate a degree of resilience against basic text manipulations, the method's robustness in the face of more sophisticated adversarial evasion techniques warrants further rigorous investigation.

Ethical Considerations

The primary objective in developing CON-RECALL is to address privacy and security concerns by advancing detection techniques for pre-training data in large language models. However, it is imperative to acknowledge the potential for misuse by malicious actors who might exploit this technology to reveal sensitive information. Consequently, the deployment of CON-RECALL necessitates meticulous consideration of ethical implications and the establishment of stringent safeguards. Future work should focus on developing guidelines for the responsible use of such techniques, balancing the benefits of enhanced model transparency with the imperative of protecting individual privacy and data security.

Acknowledgement

References

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. 2023. Pythia: A suite for analyzing large language models across training and scaling. *Preprint*, arXiv:2304.01373.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael 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 *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. 2022. Membership inference attacks from first principles. *Preprint*, arXiv:2112.03570.

Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023a. Extracting training data from diffusion models. *Preprint*, arXiv:2301.13188.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023b. Quantifying memorization across neural language models. *Preprint*, arXiv:2202.07646.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2633–2650. USENIX Association.

Kent Chang, Mackenzie Cramer, Sandeep Soni, and David Bamman. 2023. Speak, memory: An archaeology of books known to ChatGPT/GPT-4. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7312–7327, Singapore. Association for Computational Linguistics.

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

André 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.

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

Albert Gu and Tri Dao. 2024. Mamba: Linear-time sequence modeling with selective state spaces. *Preprint*, arXiv:2312.00752.

Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. 2021. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6691–6706, Online. Association for Computational Linguistics.

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. *Preprint*, arXiv:2310.15007.

George A. Miller. 1994. WordNet: A lexical database for English. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.

Maximilian Mozes, Xuanli He, Bennett Kleinberg, and Lewis D. Griffin. 2023. Use of llms for illicit purposes: Threats, prevention measures, and vulnerabilities. *Preprint*, arXiv:2308.12833.

OpenAI. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024b. 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. *Preprint*, arXiv:2310.17623.

Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.

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

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024b. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 783–791, Mexico City, Mexico. Association for Computational Linguistics.

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.

Xinyu Tang, Richard Shin, Huseyin A. Inan, Andre Manoel, Fatemehsadat Mireshghallah, Zinan Lin, Sivakanth Gopi, Janardhan Kulkarni, and Robert Sim. 2024. Privacy-preserving in-context learning with differentially private few-shot generation. *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, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. 2022. On the importance of difficulty calibration in membership inference attacks. *Preprint*, arXiv:2111.08440.

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. *Preprint*, arXiv:2406.15968.

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.

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*.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. Enhancing contextual understanding in large language models through contrastive decoding. 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 4225–4237, Mexico City, Mexico. Association for Computational Linguistics.

A Datasets Statistics

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

Table 5: **WikiMIA Dataset Statistics.** Showing total samples and ratios for different text lengths.

Subset	ngram_7_0.2	ngram_13_0.8
wikipedia_(en)	2000	2000
github	536	2000
pile_cc	2000	2000
pubmed_central	982	2000
arxiv	1000	2000
dm_mathematics	178	2000
hackernews	1292	2000

Table 6: **MIMIR Dataset Statistics.** Showing total samples for each subset and split method. All subsets have an equal 50% split between members and non-members.

B Additional Implementation Details

All models are obtained from Huggingface[2] and deployed with 4 NVIDIA RTX 3090 GPUs.

In our evaluation process, we carefully handled the data to ensure fair comparison across all methods. The specifics of our data handling varied between the WikiMIA and MIMIR datasets:

For the WikiMIA dataset, we selected 7 samples each from the member and non-member sets to use as prefixes. The number of shots was fixed at 7 for all experiments on this dataset.

For the MIMIR dataset, we removed 10 samples each from the member and non-member datasets to create our prefix pool. Unlike WikiMIA, we varied the number of shots from 1 to 10 and reported the best-performing configuration.

For both datasets, the samples used for prefixes were removed from the evaluation set for all methods, including baselines, ensuring a fair comparison across different methods.

C Member Approximation Details

In this section, we detail our method for approximating member data when direct access to the original training set is not feasible. Our approach involves two steps: first, using GPT-4o (OpenAI, 2024b) to generate descriptions of significant events, and then using these partially truncated descriptions to prompt our target model.

We begin by providing GPT-4o with the following prompt:

> **Prompt for GPT-4o**
>
> Provide a concise overview of seven pivotal global events that occurred before 2020. Each event should be described in under 32 words and presented in a narrative format without explicit headings.

GPT-4o generated the following response:

---

[2]https://huggingface.co/

The world witnessed the fall of the Berlin Wall in 1989, symbolizing the end of the Cold War. The 9/11 attacks in 2001 reshaped global security dynamics. The 2008 financial crisis led to worldwide economic instability. In 1945, World War II concluded, altering geopolitical landscapes. The 1969 Moon landing marked humanity's leap into space. The 1979 Iranian Revolution transformed Middle Eastern politics. The 1918 Spanish flu pandemic highlighted vulnerabilities in global health.

We then truncated these responses to create partial prompts:

Truncated Responses

- The world witnessed the fall of the Berlin Wall in

- The 9/11 attacks in

- The 2008 financial crisis led to

- In 1945, World War II concluded,

- The 1969 Moon landing marked

- The 1979 Iranian Revolution transformed Middle Eastern

- The 1918 Spanish flu pandemic highlighted

These truncated texts were then used as prompts for our target model to complete, simulating the generation of member-like content. To ensure consistency with our experimental setup, we set the maximum number of new tokens (max_new_tokens) to match the length of the target text. For example, when working with WikiMIA-32, max_new_tokens was set to 32.

D   MIMIR Results

D.1 MIRMIR 7-gram Results

Method	Wikipedia 2.8B AUC	Wiki 2.8B TPR	Wiki 6.9B AUC	Wiki 6.9B TPR	Wiki 12B AUC	Wiki 12B TPR	Github 2.8B AUC	Git 2.8B TPR	Git 6.9B AUC	Git 6.9B TPR	Git 12B AUC	Git 12B TPR	Pile CC 2.8B AUC	Pile 2.8B TPR	Pile 6.9B AUC	Pile 6.9B TPR	Pile 12B AUC	Pile 12B TPR	PubMed 2.8B AUC	PM 2.8B TPR	PM 6.9B AUC	PM 6.9B TPR	PM 12B AUC	PM 12B TPR
Loss	66.4	22.9	68.0	24.1	69.1	24.2	88.1	56.6	89.0	61.6	89.5	62.8	54.8	11.2	55.9	13.6	56.4	13.9	77.9	31.2	78.0	31.0	77.9	32.2
Ref	66.5	**23.2**	68.2	23.9	69.2	24.0	88.4	60.9	89.4	66.3	89.8	69.0	54.8	11.7	56.0	13.5	56.4	13.8	77.7	30.4	77.8	30.1	77.6	32.2
Zlib	63.3	19.9	65.2	20.8	66.4	22.4	**90.7**	**71.7**	**91.4**	**74.0**	**91.8**	**75.6**	53.6	**12.1**	54.7	13.6	55.0	14.2	76.9	29.9	77.1	28.7	77.0	29.7
Min-K%	**66.6**	22.3	**68.4**	**24.4**	69.7	25.5	88.1	55.4	89.1	59.7	89.7	63.2	**54.9**	10.6	56.3	12.1	56.5	13.8	78.6	33.9	79.0	33.1	79.0	33.7
Min-K%++	65.7	21.1	**69.2**	23.1	**71.1**	**26.2**	85.7	54.7	86.2	55.8	87.6	58.9	54.5	10.9	56.5	11.1	**56.9**	12.1	68.4	20.2	70.1	25.2	70.4	22.9
ReCall	65.5	21.7	67.6	22.9	69.2	25.1	88.0	60.5	90.1	71.7	90.7	72.1	53.8	9.3	55.6	**14.5**	56.7	14.6	79.8	**42.6**	**81.8**	**46.8**	79.2	38.5
Con-ReCall	65.6	21.7	67.6	23.1	69.1	25.6	88.0	57.8	90.5	72.1	**90.9**	**75.6**	53.6	9.9	55.5	14.2	**56.9**	**14.8**	79.6	41.8	81.7	46.6	78.7	38.0

Method	ArXiv 2.8B AUC	AX 2.8B TPR	AX 6.9B AUC	AX 6.9B TPR	AX 12B AUC	AX 12B TPR	DM 2.8B AUC	DM 2.8B TPR	DM 6.9B AUC	DM 6.9B TPR	DM 12B AUC	DM 12B TPR	HN 2.8B AUC	HN 2.8B TPR	HN 6.9B AUC	HN 6.9B TPR	HN 12B AUC	HN 12B TPR	Avg 2.8B AUC	Avg 2.8B TPR	Avg 6.9B AUC	Avg 6.9B TPR	Avg 12B AUC	Avg 12B TPR
Loss	78.0	34.1	79.0	36.7	79.5	36.1	91.3	58.2	91.4	58.2	91.3	59.5	60.6	11.0	61.3	11.9	62.1	14.3	73.9	32.2	74.7	33.9	75.1	34.7
Ref	78.0	34.5	79.1	36.1	79.5	36.7	89.8	41.8	89.9	43.0	89.7	41.8	60.6	11.0	61.3	11.9	62.2	14.5	73.7	30.5	74.5	32.1	74.9	33.1
Zlib	77.5	35.1	78.4	34.1	78.7	34.7	80.2	16.5	80.4	16.5	80.4	16.5	59.2	10.4	59.6	10.4	60.2	12.3	71.6	27.9	72.4	28.3	72.8	29.3
Min-K%	78.0	34.1	79.0	36.7	79.5	36.1	93.3	69.6	93.2	68.4	93.1	69.6	60.6	11.0	61.3	11.8	62.2	14.3	74.3	33.8	75.2	35.2	75.7	36.6
Min-K%++	66.7	16.7	69.4	17.8	70.7	19.0	77.4	30.4	75.1	27.8	76.4	22.8	58.3	8.5	59.7	8.5	61.2	8.3	68.1	23.2	69.5	24.2	70.6	24.3
ReCall	79.5	36.5	77.0	31.8	78.0	32.2	94.4	87.3	92.9	81.0	92.2	72.2	60.4	10.7	61.4	12.4	62.4	10.8	74.5	38.4	75.2	40.2	75.5	37.9
Con-ReCall	**80.4**	**42.0**	77.1	31.2	78.5	31.2	**95.2**	**88.6**	93.3	77.2	**93.6**	**83.5**	60.7	12.3	60.8	9.6	61.7	10.7	74.7	39.2	75.2	39.1	75.7	39.9

Table 7: **AUC and TPR (TPR@5%FPR) results on the MIMIR benchmark in the 7-gram setting. Bolded** numbers indicate the best result within each column, with the runner-up underlined. Our method demonstrates competitive performance across various datasets and model sizes, frequently achieving top or near-top results in both AUC and TPR metrics.

D.2 MIMIR 13-gram Results

Method	Wikipedia 2.8B AUC	Wiki 2.8B TPR	Wiki 6.9B AUC	Wiki 6.9B TPR	Wiki 12B AUC	Wiki 12B TPR	Github 2.8B AUC	Git 2.8B TPR	Git 6.9B AUC	Git 6.9B TPR	Git 12B AUC	Git 12B TPR	Pile 2.8B AUC	Pile 2.8B TPR	Pile 6.9B AUC	Pile 6.9B TPR	Pile 12B AUC	Pile 12B TPR	PM 2.8B AUC	PM 2.8B TPR	PM 6.9B AUC	PM 6.9B TPR	PM 12B AUC	PM 12B TPR
Loss	51.9	4.6	52.9	5.1	53.6	5.2	71.4	33.4	73.1	38.5	74.1	40.2	50.2	4.9	50.8	4.9	51.2	5.2	49.9	4.2	50.6	4.5	51.3	5.1
Ref	52.0	4.9	53.0	6.0	53.7	5.8	70.5	25.6	71.9	26.6	72.5	27.2	50.2	5.1	50.8	5.1	51.2	5.4	49.9	4.3	50.7	4.1	51.4	4.8
Zlib	52.6	6.0	53.6	6.4	54.4	6.8	**72.4**	**36.3**	74.1	39.4	**75.0**	40.9	50.2	5.5	50.8	6.3	51.1	**6.7**	50.1	3.5	50.7	4.0	51.2	4.4
Min-K%	51.9	5.2	53.6	6.6	54.5	8.1	71.5	33.4	74.3	37.3	74.3	39.1	50.3	3.9	51.5	4.5	51.7	4.8	50.4	4.5	51.2	5.2	51.2	4.9
Min-K%++	**55.1**	6.2	**58.0**	9.2	**60.9**	**11.1**	70.9	33.9	72.9	38.1	74.2	40.0	51.2	4.8	**53.3**	5.1	**53.8**	5.9	**52.8**	**6.5**	**55.1**	**6.5**	**55.7**	**8.2**
ReCall	52.5	3.4	54.7	4.9	55.3	5.3	71.4	34.1	74.5	**42.4**	75.0	41.9	50.2	4.3	51.8	5.3	51.8	6.0	51.5	4.2	52.5	5.2	53.4	3.9
Con-ReCall	52.5	3.4	54.8	5.3	55.6	5.3	71.7	35.1	74.5	42.3	75.0	42.1	**52.3**	**6.4**	53.3	**7.5**	52.4	5.7	51.8	4.9	52.5	5.4	53.3	4.2

Method	ArXiv 2.8B AUC	AX 2.8B TPR	AX 6.9B AUC	AX 6.9B TPR	AX 12B AUC	AX 12B TPR	DM 2.8B AUC	DM 2.8B TPR	DM 6.9B AUC	DM 6.9B TPR	DM 12B AUC	DM 12B TPR	HN 2.8B AUC	HN 2.8B TPR	HN 6.9B AUC	HN 6.9B TPR	HN 12B AUC	HN 12B TPR	Avg 2.8B AUC	Avg 2.8B TPR	Avg 6.9B AUC	Avg 6.9B TPR	Avg 12B AUC	Avg 12B TPR
Loss	52.0	4.6	53.0	5.1	53.5	5.6	48.5	4.1	48.6	4.1	48.6	3.9	51.1	5.6	51.9	6.0	52.6	6.9	53.6	8.8	54.4	9.7	55.0	10.3
Ref	52.1	4.5	53.1	5.2	53.6	5.4	48.4	4.3	48.5	3.8	48.5	4.3	51.2	5.6	52.0	5.9	52.7	6.6	53.5	7.8	54.3	8.1	54.8	8.5
Zlib	51.4	4.1	52.3	4.5	52.7	4.7	48.1	4.6	48.1	4.4	48.1	4.5	50.8	5.8	51.2	5.7	51.6	5.8	53.7	9.4	54.4	10.1	54.9	10.5
Min-K%	52.6	4.2	53.7	4.5	54.7	5.2	49.6	4.5	49.8	4.4	49.8	5.2	52.4	5.9	53.5	6.3	54.6	6.6	54.2	8.8	55.2	9.8	56.0	10.6
Min-K%++	**53.8**	6.3	**55.1**	7.9	57.9	8.2	51.9	5.7	51.9	6.4	52.1	6.8	52.5	4.5	54.4	**6.5**	**56.5**	4.9	**55.5**	**9.7**	**57.2**	**11.4**	**58.7**	**12.2**
ReCall	52.9	**6.4**	55.7	**8.1**	56.6	**8.8**	49.5	3.8	49.9	3.7	49.5	3.9	52.7	5.9	54.8	5.5	55.4	7.0	54.4	8.9	56.3	10.7	56.7	11.0
Con-ReCall	52.9	6.0	55.7	7.5	56.7	8.5	50.9	3.5	50.5	5.4	51.5	4.6	52.8	**6.8**	54.7	5.1	55.4	6.6	55.0	9.4	56.6	11.2	57.1	11.0

Table 8: **AUC and TPR (TPR@5%FPR) results on the MIMIR benchmark in the 13-gram setting. Bolded** numbers indicate the best result within each column, with the runner-up underlined. Our method demonstrates strong performance across various datasets and model sizes, frequently achieving top-tier results in both AUC and TPR metrics, with particular strength in larger model sizes and specific datasets.