Membership Inference Attacks on Tokenizers of Large Language Models

Meng Tong[1] Yuntao Du[2] Kejiang Chen[1]
Weiming Zhang[1] Ninghui Li[2]

[1]University of Science and Technology of China [2] Purdue University

Abstract

Membership inference attacks (MIAs) are widely used to assess the privacy risks associated with machine learning models. However, when these attacks are applied to pre-trained large language models (LLMs), they encounter significant challenges, including mislabeled samples, distribution shifts, and discrepancies in model size between experimental and real-world settings. To address these limitations, we introduce tokenizers as a new attack vector for membership inference. Specifically, a tokenizer converts raw text into tokens for LLMs. Unlike full models, tokenizers can be efficiently trained from scratch, thereby avoiding the aforementioned challenges. In addition, the tokenizer's training data is typically representative of the data used to pre-train LLMs. Despite these advantages, the potential of tokenizers as an attack vector remains unexplored. To this end, we present the first study on membership leakage through tokenizers and explore five attack methods to infer dataset membership. Extensive experiments on millions of Internet samples reveal the vulnerabilities in the tokenizers of state-of-the-art LLMs. To mitigate this emerging risk, we further propose an adaptive defense. Our findings highlight tokenizers as an overlooked yet critical privacy threat, underscoring the urgent need for privacy-preserving mechanisms specifically designed for them.[1]

1 Introduction

Scaling up the pre-training data for large language models (LLMs) has been shown to improve performance [10, 43, 44, 52, 71]. Nevertheless, the rapid expansion of pre-training data has also raised concerns about whether these commercial models are trained on sensitive or copyrighted information [18, 68]. For instance, on June 4, 2025, Reddit filed a lawsuit against Anthropic, alleging the unlawful use of data from its 100 million daily users to train LLMs [73]. Furthermore, an increasing body of research [12, 41, 42] has documented instances in which LLMs memorize and leak private information.

[1]Code is available at: https://github.com/mengtong0110/Tokenizer-MIA.

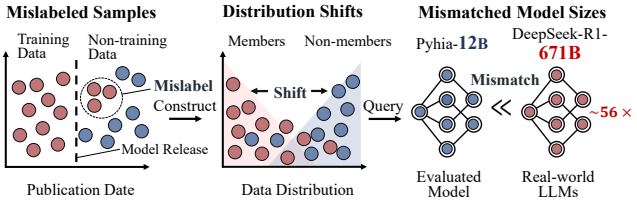
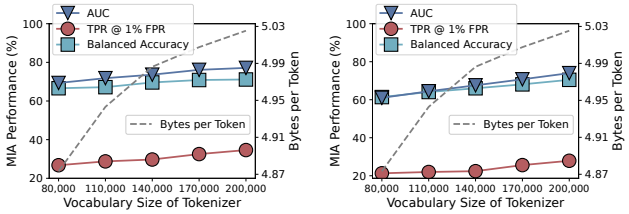


Figure 1: Evaluation challenges in MIAs against LLMs.

To assess potential data misuse, extensive research has explored the membership inference attacks (MIAs) in LLMs [54, 88, 94, 106]. In particular, an MIA aims to determine whether a specific data sample or dataset was used to train the target model (i.e., *member*) or not (i.e., *non-member*). To achieve this, existing MIAs primarily rely on the model's output as the attack vector [26, 41, 107]. Although this vector is widely adopted, these attacks face significant challenges in reliably demonstrating their effectiveness for LLMs [39, 68], as shown in Figure 1. The primary obstacle is that faithful evaluation [44] requires an evaluator to pre-train an LLM from scratch [103], which incurs significant computational costs. As a result, existing MIAs are typically evaluated using LLMs that have already been pre-trained by others. Nevertheless, this may lead to MIA evaluation exhibiting *distribution shifts* [26, 63] or containing *mislabeled samples* [68, 92]. Furthermore, many of the evaluated models (e.g., Pythia-12B [8]) are much smaller than practical deployed LLMs (e.g., DeepSeek-R1-671B [35]), limiting the ability to assess current MIAs in real-world conditions. Given these challenges, a natural question arises: *Can we exploit an attack vector for MIAs against LLMs that avoids these limitations?*

New Attack Vector. Motivated by this question, we explore a new attack vector that targets other components of LLMs. Typically, an LLM comprises a tokenizer, a transformer network, and an output layer [100]. Among these components, the tokenizer has been open-sourced in commercial LLMs such as OpenAI-o3 [75] and Gemini-1.5 [33] to support transparent billing. Building on this observation, *we propose the*



(a) MIA via Vocabulary Overlap (b) MIA via Frequency Estimation

Figure 2: Performance of our MIAs on tokenizers of LLMs. **Key Finding:** Scaling up LLMs [66] involves expanding the tokenizer's vocabulary [46, 97] and thus improving its compression efficiency (i.e., bytes per token) [59]. Yet, our figures show that it also increases tokenizer's vulnerability to MIAs.

previously overlooked tokenizer as a new attack vector for membership inference. Specifically, a tokenizer [109] is in charge of converting text into tokens for LLMs. Its training data is typically representative of the overall pre-training corpus of the LLM [8,9,100]. Its training process simply involves merging the most frequent strings into a vocabulary using the byte-pair encoding (BPE) algorithm [32]. This straightforward process enables training a tokenizer from scratch, which aligns with the inference game [103] and avoids mislabeled samples or distribution shifts. Furthermore, the simplicity of BPE also makes it feasible to train a tokenizer that matches those used in state-of-the-art LLMs (see Figure 6).

Despite these advantages, the feasibility of using tokenizers as an attack vector has not yet been explored. In this paper, we present the first study to exploit toknizers for MIAs and propose five attack methods for inferring dataset membership:

- **MIA via** Merge Similarity**.** This attack trains shadow tokenizers [94] and compares their token merge orders to that of the target tokenizer. If the merge order of the target tokenizer closely matches that of shadow tokenizers trained on a particular dataset, the dataset is classified as a *member*. However, the effectiveness of this attack is limited. Only a few distinctive tokens display a membership signal in merge order, making it difficult for membership inference.
- **MIA via** Vocabulary Overlap**.** Leveraging these distinctive tokens, we propose a more effective attack, MIA via Vocabulary Overlap. This attack also involves training shadow tokenizers. But instead of comparing merge orders, it classifies a dataset as a *member* if the distinctive tokens in the target tokenizer's vocabulary significantly overlap with those from shadow tokenizers trained on that dataset.
- **MIA via** Frequency Estimation**.** While our results show that MIA via Vocabulary Overlap achieves strong performance, it requires substantial time to train multiple shadow tokenizers. For efficient implementation, we introduce MIA via Frequency Estimation, which trains only a single shadow tokenizer. This attack evaluates whether training the target tokenizer on a dataset is necessary for certain tokens to appear in its vocabulary. If this condition is met, this attack classifies the dataset as a *member*.
- Additionally, as part of our evaluation in Section 5.1, we further explore two attack methods for potential alternatives: MIA via Naive Bayes and MIA via Compression Rate.

We conduct extensive evaluations using millions of Internet data [81]. To match real-world practice, we align the trained tokenizers in evaluations with those used in commercial LLMs [5,8,35,100]. The experimental results indicate that MIAs via Vocabulary Overlap and Frequency Estimation achieve strong performance across various settings. For example, MIA via Vocabulary Overlap achieves an AUC score of 0.771 against a tokenizer with two hundred thousand tokens, whereas MIA via Frequency Estimation achieves an AUC score of 0.740. More importantly, as shown in Figure 2, our experiments show that scaling laws increase tokenizers' vulnerability to MIAs. This finding suggests that MIAs could become more effective on scaled-up tokenizers in the future.

Our Contributions. Our main contributions are as follows:

- We introduce the tokenizer as a new attack vector for membership inference and conduct the first study demonstrating its feasibility in LLMs.
- We explore five attack methods for set-level membership inference against tokenizers, revealing the vulnerabilities in these foundational components of state-of-the-art LLMs.
- We conduct extensive evaluations using real-world Internet datasets. The results show that our shadow-based attacks demonstrate strong performance against tokenizers.
- We further analyze tokenizers from commercial LLMs. The results show that the tokenizers, such as OpenAI-o200k [75] and DeepSeek-R1 [35], also contain distinctive tokens for implementing membership inference.

Main Findings. We have the following key findings:

- According to prior work [46, 66], scaling up the intelligence of LLMs involves expanding the tokenizer's vocabulary [97] and thus improving its compression efficiency [59]. However, our experimental results show that it also increases the tokenizer's vulnerability to effective MIAs.
- The membership status of the target dataset with more data samples is typically more accurately inferred by MIAs.
- While removing infrequent tokens from the tokenizer's vocabulary can partially reduce the effectiveness of MIAs, this approach also lowers the tokenizer's compression efficiency. Moreover, even with this mitigation, MIAs can remain effective for inferring large datasets.

Organization. The remainder of this paper is organized as follows. Section 2 presents preliminaries on tokenizer training and membership inference. Section 3 introduces the threat models for membership inference attacks. Section 4 presents three of our attack methods. Section 5 introduces two additional methods and reports the experimental results. Section 6 reviews related work. Section 7 discusses the limitations of

LLM dataset inference. Section 8 concludes the paper.

2 Preliminaries

2.1 Tokenizer Training

A tokenizer [109] is a fundamental component in LLMs, converting raw text into a format that the model can process. Formally, a tokenizer is defined as a function $f_{\mathcal{V}} : S \to \mathcal{V}^*$ that maps an input string $s \in S$ (e.g., a sentence or document) into a sequence of tokens from a vocabulary $\mathcal{V}$. In practice, this function is learned from a collection of text datasets $\mathcal{D}$. Specifically, its training objective is to segment and encode the data in a way that maximizes compression efficiency [109]. This process begins by initializing the vocabulary $\mathcal{V}$ with basic symbols, such as individual characters. During training, the tokenizer $f_{\mathcal{V}}$ iteratively merges the most frequent pairs of symbols in the data via the byte-pair encoding (BPE) algorithm [93]. This iterative process results in a token merge order: each token $t_i \in \mathcal{V}$ is assigned an index i corresponding to the iteration, where it was merged into the vocabulary $\mathcal{V}$.

In commercial LLM applications, the tokenizer also serves as a basis for token billing. As the tokenizer directly determines how users are charged based on the number of tokens in a message, its operation is critical for ensuring transparent billing [47]. To promote such transparency in token counting, the organizations behind major LLMs [4, 33, 35, 75] have open-sourced their tokenizers, making their vocabularies and token merge orders publicly available.

2.2 Membership Inference

The concept of membership inference attacks was first introduced by Shokri et al. [94], who demonstrated that an adversary can determine whether a specific data record was included in a model's training set. Specifically, they propose to train multiple shadow models that imitate the behavior of the target model. By comparing the output distributions of shadow models trained with and without a specific data record, the adversary can infer whether the data record was part of the target model's training data. [29, 72, 89].

Building upon this foundational research, subsequent studies [12, 13, 23, 56, 58] have investigated the effectiveness of MIAs on a variety of machine learning models, including ResNet-18 [40] and BERT [21]. Nonetheless, as models increase in size and are trained on larger datasets over fewer epochs, the overfitting signal for individual samples decreases, resulting in reduced MIA performance on LLMs [25]. To address this limitation, recent MIAs [42, 84] instead focus on dataset membership, which aggregates signals from individual samples to enhance the detection of membership. However, evaluating these attacks presents significant challenges, such as *mislabeled samples* [68], *distribution shifts* [26], and discrepancies in model size between experimental and real-world settings. Additionally, the effectiveness of these attacks typically depends on further assumptions. For example, some [63] assumes that an adversary has access to the LLM's output loss, while another [102] requires the ability to fine-tune the target LLM. However, these assumptions are not guaranteed to hold in closed-source LLMs. Moreover, they can be defended by adding noise during the model training process [108].

3 Threat Models

We consider an adversary who aims to determine whether a specific dataset was used to train the target tokenizer.

Adversary's Objective. Given a collection of datasets $\mathcal{D}_{\text{mem}}$ sampled from an underlying distribution $\mathbb{D}$ (denoted as $\mathcal{D}_{\text{mem}} \leftarrow \mathbb{D}$), we write $\mathcal{V}_{\text{target}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{mem}})$ to represent a tokenizer's vocabulary $\mathcal{V}_{\text{target}}$ is trained by running the BPE algorithm $\mathcal{T}$ [32] on $\mathcal{D}_{\text{mem}}$. This training process results in the target tokenizer $f_{\mathcal{V}_{\text{target}}}$. Given a target dataset $D \in \mathbb{D}$, the adversary's objective is to determine whether D was part of the training data used to construct the vocabulary $\mathcal{V}_{\text{target}}$. To achieve this, the adversary employs a membership inference attack $\mathcal{A}$, which can be formally defined as:

$$\mathcal{A}\colon D, f_{\mathcal{V}_{\text{target}}} \to \{0,1\}, \tag{1}$$

where 1 indicates $D \in \mathcal{D}_{\text{mem}}$, and 0 indicates $D \notin \mathcal{D}_{\text{mem}}$.

Adversary's Capabilities. We align the adversary's capabilities with the real-world conditions, where commercial LLMs such as OpenAI-o3 [75], Gemini-1.5 [33], and Claude-2 [4] have open-sourced their trained tokenizers to support transparent billing. Accordingly, we assume that the adversary has access to the tokenizer $f_{\mathcal{V}_{\text{target}}}$ with its associated vocabulary $\mathcal{V}_{\text{target}} = \{t_1, t_2, \ldots, t_{|\mathcal{V}_{\text{target}}|}\}$, where each token $t_i \in \mathcal{V}_{\text{target}}$ was merged at iteration i during the training process. Furthermore, we assume that the adversary is able to sample auxiliary datasets $\mathcal{D}_{\text{aux}}$ from the same distribution as the training data used by the target tokenizer, i.e., $\mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}$. Leveraging datasets $\mathcal{D}_{\text{aux}}$, the adversary can use the BPE algorithm $\mathcal{T}$ to train shadow tokenizers. This assumption is consistent with previous work [11, 58, 88]. It is also realistic in practice, as the training data for tokenizers is representative of that in LLMs [8, 9, 37], both of which are primarily sourced from publicly available web content [5, 35, 74].

4 Attack Methodology

In this section, we present our MIAs against pre-trained LLMs. For each method, we start by introducing our design intuition. Then we describe the attack methodology.

4.1 Baseline: MIA via Merge Similarity

Shadow-based MIAs [39, 50, 58, 88, 104] involve training auxiliary models to calibrate predictions. Inspired by these at-

tacks, we formalize MIA via Merge Similarity on tokenizers.

Design Intuition. Prior work [11] has revealed that the overfitting behavior of machine learning models can vary depending on whether a particular data point was present in the training data. Based on this insight, we hypothesize that tokenizers may also differ depending on whether a dataset was included in the training data. Specifically, we assume that the token merge order can serve as an indicator of the overfitting phenomenon. Thus, merge orders in vocabularies $\mathbb{V}_{\text{in}} = \{\mathcal{V}_{\text{in}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{aux}} \bigcup \{D\}) \mid \mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}\}$ and $\mathbb{V}_{\text{out}} = \{\mathcal{V}_{\text{out}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{aux}} \setminus \{D\}) \mid \mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}\}$ can differ depending on whether the target dataset D was included in the training data. Building on this hypothesis, an adversary can exploit this difference by comparing the similarity ρ of token merge orders for pairs $(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})$ and $(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})$. If the average value for $\rho(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})$ is higher than that of $\rho(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})$, it is more likely that $D \in \mathcal{D}_{\text{mem}}$.

Attack Method. This attack consists of four steps.

(i) The adversary randomly samples a collection of datasets $\mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}$ for N times, and trains N shadow tokenizers considering inclusion or exclusion of the target dataset D. Thus, the adversary obtains sets $\mathbb{V}_{\text{in}}$ and $\mathbb{V}_{\text{out}}$.

(ii) The adversary computes the similarity of token merge orders for each $\rho(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})$ and $\rho(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})$ via Spearman's rank correlation coefficient [91].

(iii) The membership signal for target dataset D is defined as:

$$\frac{1}{2} + \frac{\sum_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}} \rho(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})}{4|\mathbb{V}_{\text{in}}|} - \frac{\sum_{\mathcal{V}_{\text{out}} \in \mathbb{V}_{\text{out}}} \rho(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})}{4|\mathbb{V}_{\text{out}}|}, \quad (2)$$

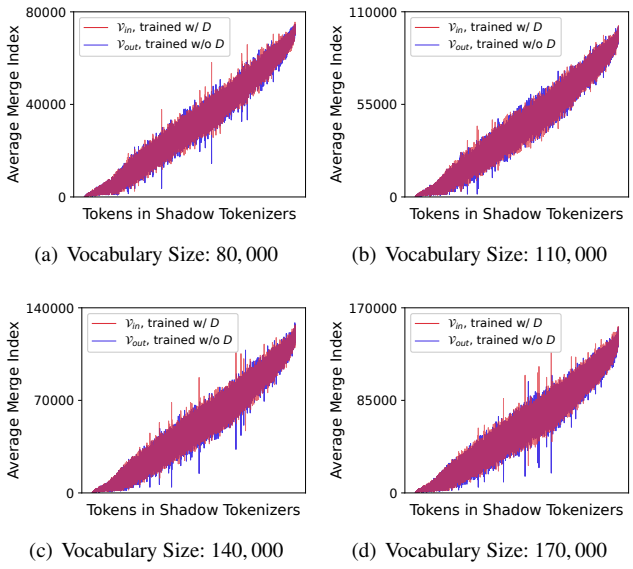
where it ranges from 0 to 1.

(iv) If the membership signal is larger than a decision-making threshold τ, output 1 (*member*). Otherwise, output 0.

We conduct validation experiments for this attack using real-world Internet data [81] (see Section 5.2 for detailed experiments). However, the results demonstrate unsatisfactory performance of MIA via Merge Similarity in distinguishing between *members* and *non-members*. These results are probably due to the overall distributions of token merge orders in $\mathcal{V}_{\text{in}}$ and $\mathcal{V}_{\text{out}}$ resembling each other, as illustrated in Figure 3. The minor discrepancies observed between these distributions suggest weak overfitting signals from the perspective of global tokens. Consequently, the correlation values of Spearman's rank $\rho(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})$ and $\rho(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})$ remain highly similar, making it hard for to infer the membership of the dataset D.

4.2 Improved MIA via Vocabulary Overlap

Building on the observation that global token distribution can obscure overfitting signals from the target dataset D, we shift our focus to a more fine-grained analysis. Specifically, we



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000

(c) Vocabulary Size: 140,000 (d) Vocabulary Size: 170,000

Figure 3: Average merge index for tokens in $\mathcal{V}_{\text{in}}$ and $\mathcal{V}_{\text{out}}$. It is shown that overall merge orders in $\mathcal{V}_{\text{in}}$ and $\mathcal{V}_{\text{out}}$ resemble.

solely examine those distinctive tokens whose merge index differs between the vocabularies $\mathbb{V}_{\text{in}}$ and $\mathbb{V}_{\text{out}}$. Our analysis suggests that only when the tokenizer is trained on dataset D, some distinctive tokens in D are more likely to be overfit in its vocabulary. Typically, these distinctive tokens more frequently appear in $\mathbb{V}_{\text{in}}$, but are seldom found in $\mathbb{V}_{\text{out}}$. As a result, there exist minor discrepancies between the vocabularies $\mathbb{V}_{\text{in}}$ and $\mathbb{V}_{\text{out}}$. Leveraging this insight, we propose an improved approach, MIA via Vocabulary Overlap.

Design Intuition. When a target tokenizer $f_{\mathcal{V}_{\text{target}}}$ is trained on a target dataset D, its vocabulary $\mathcal{V}_{\text{target}}$ is likely to overfit the distinctive tokens present in dataset D. In fact, existing analysis has shown that OpenAI's tokenizer contains tokens unique to the Reddit forum [36, 65]. Building on this, we hypothesize that: the more distinctive tokens from D that are found in $\mathcal{V}_{\text{target}}$, the more likely it is that $\mathcal{V}_{\text{target}}$ was trained on D. To quantify the overlap of distinctive tokens, one effective approach is to use the Jaccard index [6], which measures the similarity between two sets by focusing on the presence of shared elements. Specifically, an adversary can exploit this by computing the Vocabulary Overlap using Jaccard index J for pairs $(\mathcal{V}_{\text{in}}, \mathcal{V}_{\text{target}})$ and $(\mathcal{V}_{\text{out}}, \mathcal{V}_{\text{target}})$ in terms of these distinctive tokens. We write $\mathcal{V}_{\text{non}} = (\bigcup_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}} \mathcal{V}_{\text{in}}) \bigcap (\bigcup_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{out}}} \mathcal{V}_{\text{out}})$ to denote the set of non-distinctive tokens. If the average value for $J(\mathcal{V}_{\text{in}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})$ is higher than that for $J(\mathcal{V}_{\text{out}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})$, it is more likely $D \in \mathcal{D}_{\text{mem}}$.

Attack Method. We structure this attack in five steps.

(i) The adversary randomly samples a collection of datasets $\mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}$ for N times, and trains N shadow tokenizers considering inclusion or exclusion of the target dataset

Algorithm 1 MIA via Vocabulary Overlap**.** We train N shadow tokenizers with and without target dataset D, filter out non-distinctive tokens, and compute the membership signal. If the signal is larger than a threshold τ, the dataset D is inferred as a member. Otherwise, it is inferred as a non-member.

Input: Target dataset D, vocabulary of target tokenizer $\mathcal{V}_{\text{target}}$, underlying distribution $\mathbb{D}$, number of shadow tokenizers N, BPE algorithm $\mathcal{T}$, threshold τ

$\mathbb{V}_{\text{in}} \leftarrow \{\}, \quad \mathbb{V}_{\text{out}} \leftarrow \{\}$
 # Step 1: Train N shadow tokenizers
for N times **do**
  $\mathcal{D}_{\text{aux}} \xleftarrow{\$} \mathbb{D}$ ▷ *randomly sample auxiliary datasets*
  $\mathcal{V}_{\text{in}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{aux}} \bigcup \{D\})$ ▷ *train IN tokenizer*
  $\mathbb{V}_{\text{in}} \leftarrow \mathbb{V}_{\text{in}} \cup \{\mathcal{V}_{\text{in}}\}$
  $\mathcal{V}_{\text{out}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{aux}} \setminus \{D\})$ ▷ *train OUT tokenizer*
  $\mathbb{V}_{\text{out}} \leftarrow \mathbb{V}_{\text{out}} \cup \{\mathcal{V}_{\text{out}}\}$
end for
 # Step 2: Compute non-distinctive tokens
$\mathcal{V}_{\text{non}} \leftarrow \left(\bigcup_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}} \mathcal{V}_{\text{in}}\right) \cap \left(\bigcup_{\mathcal{V}_{\text{out}} \in \mathbb{V}_{\text{out}}} \mathcal{V}_{\text{out}}\right)$
$J_{\text{in}} \leftarrow 0, \quad J_{\text{out}} \leftarrow 0, \quad \tilde{\mathcal{V}}_{\text{target}} \leftarrow \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}}$
 # Step 3: Calculate Jaccard index
for each $\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}$ **do**
  $\tilde{\mathcal{V}}_{\text{in}} \leftarrow \mathcal{V}_{\text{in}} \setminus \mathcal{V}_{\text{non}}$ ▷ *filter non-distinctive tokens in* $\mathcal{V}_{in}$
  $J_{\text{in}} \leftarrow J_{\text{in}} + \frac{|\tilde{\mathcal{V}}_{\text{in}} \cap \tilde{\mathcal{V}}_{\text{target}}|}{|\tilde{\mathcal{V}}_{\text{in}} \cup \tilde{\mathcal{V}}_{\text{target}}|}$ ▷ *sum Jaccard index in* $\mathbb{V}_{in}$
end for
for each $\mathcal{V}_{\text{out}} \in \mathbb{V}_{\text{out}}$ **do**
  $\tilde{\mathcal{V}}_{\text{out}} \leftarrow \mathcal{V}_{\text{out}} \setminus \mathcal{V}_{\text{non}}$ ▷ *filter non-distinctive tokens in* $\mathcal{V}_{out}$
  $J_{\text{out}} \leftarrow J_{\text{out}} + \frac{|\tilde{\mathcal{V}}_{\text{out}} \cap \tilde{\mathcal{V}}_{\text{target}}|}{|\tilde{\mathcal{V}}_{\text{out}} \cup \tilde{\mathcal{V}}_{\text{target}}|}$ ▷ *sum Jaccard index in* $\mathbb{V}_{out}$
end for
 # Step 4: Compute membership signal
$\textsc{Signal} \leftarrow \frac{1}{2} + \frac{J_{\text{in}}}{2|\mathbb{V}_{\text{in}}|} - \frac{J_{\text{out}}}{2|\mathbb{V}_{\text{out}}|}$
 # Step 5: Infer the membership
return $\mathbb{1}[\textsc{Signal} > \tau]$

D. This process results in vocabulary sets $\mathbb{V}_{\text{in}}$ and $\mathbb{V}_{\text{out}}$.

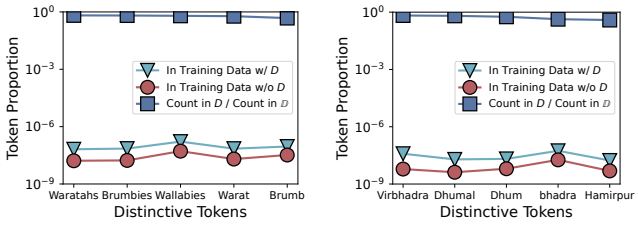
(ii) The adversary computes the non-distinctive tokens as:

$$\mathcal{V}_{\text{non}} = \Big(\bigcup_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}} \mathcal{V}_{\text{in}}\Big) \bigcap \Big(\bigcup_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{out}}} \mathcal{V}_{\text{out}}\Big). \tag{3}$$

(iii) The adversary calculates the overfitting signals using the Jaccard index [6] for each $J(\mathcal{V}_{\text{in}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})$ and $J(\mathcal{V}_{\text{out}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})$.

(iv) The membership signal for dataset D is defined as:

$$\frac{1}{2} + \frac{\sum_{\mathcal{V}_{\text{in}} \in \mathbb{V}_{\text{in}}} J(\mathcal{V}_{\text{in}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})}{2|\mathbb{V}_{\text{in}}|} - \frac{\sum_{\mathcal{V}_{\text{out}} \in \mathbb{V}_{\text{out}}} J(\mathcal{V}_{\text{out}} \setminus \mathcal{V}_{\text{non}}, \mathcal{V}_{\text{target}} \setminus \mathcal{V}_{\text{non}})}{2|\mathbb{V}_{\text{out}}|}, \tag{4}$$



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 140,000

Figure 4: Distinctive tokens in MIA via Vocabulary Overlap.

where it ranges from 0 to 1.

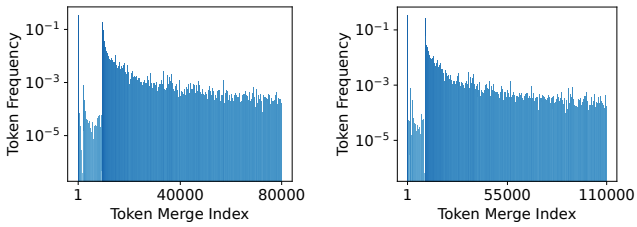
(v) If the membership signal is larger than a decision-making threshold τ, output 1 (*member*). Otherwise, output 0.

The detailed process of this attack is outlined in Algorithm 1. However, like other shadow-based MIAs [11,94,103], we find that MIA via Vocabulary Overlap requires multiple shadow tokenizers (e.g., 96) to effectively capture membership signals. As a result, training such a large number of shadow tokenizers incurs a substantial time cost.

4.3 Efficient MIA via Frequency Estimation

MIA via Vocabulary Overlap raises a natural question: Can we design an attack that relies on fewer shadow tokenizers and thus reduces the overall time cost? To address this, we investigate whether it is possible to identify distinctive tokens directly by analyzing their statistical characteristics. Motivated by this, we examine such distinctive tokens and derive two key insights: ❶ The distinctive tokens of dataset D appear infrequently in the training data of the tokenizer trained on D. ❷ As shown in Figure 4 and Figure 12, the majority of occurrences of these distinctive tokens in the underlying distribution $\mathbb{D}$ are found in the dataset D. Given these characteristics, if dataset D is excluded from the tokenizer's training data, the frequency of such distinctive tokens becomes lower. As a result, these tokens with low frequency are unlikely to be merged into the vocabulary during tokenizer training, since BPE primarily merges the most frequent tokens. This observation suggests that including dataset D in the training data is almost a necessary condition for some tokens to be merged into the tokenizer's vocabulary. Motivated by these insights, we introduce the MIA via Frequency Estimation.

Design Intuition. It is hypothesized that tokenizer training probably exhibits overfitting by incorporating distinctive tokens from the training datasets into its vocabulary [36, 65]. Building on this intuition, an adversary could exploit such overfitting by evaluating whether including dataset D in the training data is necessary for the merging of some tokens in vocabulary $\mathcal{V}_{\text{target}}$. If the presence of such distinctive tokens in $\mathcal{V}_{\text{target}}$ strongly depends on dataset D, it is likely $D \in \mathcal{D}_{\text{mem}}$.



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000

(c) Vocabulary Size: 140,000 (d) Vocabulary Size: 170,000

Figure 5: Relationship between token merge index and frequency in training data, indicating they follow a power law.

Necessity Evaluation. However, no existing metric evaluates this necessity. To fill this gap, we introduce a new metric: Relative Token Frequency with Self-information (RTF-SI).

Definition 4.1 (Relative Token Frequency with Self-information). *Let $\mathbb{D}$ denote a data distribution. Given a dataset $D \subseteq \mathbb{D}$ and a target tokenizer's vocabulary $\mathcal{V}_{target}$, the Relative Token Frequency with Self-Information (RTF-SI) of a token $t_i \in \mathcal{V}_{target}$ in D is defined as:*

$$\text{RTF-SI}(D, t_i, \mathcal{V}_{target}) := \text{RTF}(t_i, D) \cdot \text{SI}(t_i, \mathcal{V}_{target}), \quad (5)$$

where the relative token frequency (RTF) is calculated as:

$$\text{RTF}(t_i, D) = \frac{n_D(t_i)}{\sum_{D' \in \mathbb{D}} n_{D'}(t_i)}, \quad (6)$$

with $n_D(t_i)$ denoting the count of token t_i in the dataset D. The self-information (SI) is given by:

$$\text{SI}(t_i, \mathcal{V}_{target}) = -\log \Pr(t_i \mid \mathcal{V}_{target}), \quad (7)$$

where $\Pr(t_i \mid \mathcal{V}_{target})$ is the frequency of token t_i appearing in the training data $\mathcal{D}_{\text{mem}}$ associated with the vocabulary $\mathcal{V}_{target}$. Ideally, this probability is computed as:

$$\Pr(t_i \mid \mathcal{V}_{target}) = \frac{\sum_{D' \in \mathcal{D}_{\text{mem}}} n_{D'}(t_i)}{\sum_{t' \in \mathcal{V}_{target}} \sum_{D' \in \mathcal{D}_{\text{mem}}} n_{D'}(t')}. \quad (8)$$

RTF-SI evaluates whether it is necessary to include dataset D in the training data for constructing the target vocabulary, $\mathcal{V}_{\text{target}}$. Building on the classic TF-IDF definition [2, 82, 96], RTF-SI modifies the normalization used in the TF component [80]. A high RTF-SI for the target dataset D suggests that:

Table 1: Power-law fit on token frequency in training data.

$\lvert\mathcal{V}_{\text{target}}\rvert$	80,000	110,000	140,000	170,000	200,000
$x_{\min}$	9,782	9,782	9,782	9,782	9,782
α	1.717	1.604	1.537	1.493	1.460
Std. Error	0.003	0.002	0.001	0.001	0.001

❶ Token t_i is with high self-information, i.e., it appears infrequently in the training data of the target tokenizer. ❷ Dataset D contributes the majority of token t_i's relative frequency in the underlying distribution $\mathbb{D}$. As a result, the absence of dataset D may significantly affect some merged tokens in vocabulary $\mathcal{V}_{\text{target}}$, indicating that $D \in \mathcal{D}_{\text{mem}}$.

Frequency Estimation. In practical implementation, an adversary can estimate the RTF component using auxiliary datasets $\mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}$. However, the SI component is not directly observable, as the frequency $\Pr(t_i \mid \mathcal{V}_{\text{target}})$ in the training data is not available. To estimate this, we draw on the power law [17], which has been widely applied to approximate word frequency [31, 78]. According to the power law, when a list of measured values exceeding a threshold $x_{\min} \in \mathbb{Z}_{>0}$ is sorted in decreasing order, the n-th value is approximately proportional to $1/n^{\alpha}$, where $\alpha \in \mathbb{R}_{>0}$ is a constant. As shown in Figure 5, there is a power-law relationship between token merge order and frequency. For rigorous verification, we fit the frequency $\Pr(t_i \mid \mathcal{V}_{\text{target}})$ in a power-law distribution [17]:

$$\Pr(t_i \mid \mathcal{V}_{\text{target}}) \propto \frac{1}{i^{\alpha}}, \text{ where } t_i \in \mathcal{V}_{\text{target}} \text{ and } i > x_{\min}. \quad (9)$$

The estimation results in Table 1 show a small standard error between the estimated and actual values, which supports using the power law to approximate the value $\Pr(t_i \mid \mathcal{V})$ in the SI component. Given this, RTF-SI can also be computed as:

Theorem 4.2 (RTF-SI under the Power Law). *Under the power-law distribution [17], the frequency $\Pr(t_i \mid \mathcal{V}_{target})$ of a token $t_i \in \mathcal{V}_{target}$ is proportional to $1/i^{\alpha}$:*

$$\Pr(t_i \mid \mathcal{V}_{target}) \propto \frac{1}{i^{\alpha}}, \quad (10)$$

where $i > x_{\min}$, and $\alpha \in \mathbb{R}_{>0}$, $x_{\min} \in \mathbb{Z}_{>0}$ are constants defined by the power law. Then, RTF-SI can be approximated by its lower bound:

$$\text{RTF-SI}(D, t_i, \mathcal{V}_{target}) \geq \frac{n_D(t_i)}{\sum_{D' \in \mathbb{D}} n_{D'}(t_i)} \cdot \log\Big(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{target}|} i^{\alpha}/j^{\alpha}\Big). \quad (11)$$

The detailed proof of Theorem 4.2 is provided in Appendix A. The power law allows an adversary to estimate RTF-SI without directly accessing the frequency $\Pr(t_i \mid \mathcal{V}_{\text{target}})$. Since the power law estimates the frequency of tokens $t_i \in \mathcal{V}_{\text{target}}$ with merge index $i > x_{\min}$, MIA via Frequency Estimation also concentrates on them.

Attack Method. We outline this attack in four steps below.

Algorithm 2 MIA via Frequency Estimation**.** We train a shadow tokenizer to fit the power-law distribution of token frequency, approximate RTF-SI for each token $t_i \in \mathcal{V}_{\text{target}}$ where $i > x_{\min}$, and compute the membership signal based on the maximum RTF-SI. If the membership signal is larger than a decision-making threshold τ, the dataset D is inferred as a member. Otherwise, it is inferred as a non-member.

Input: Target dataset D, vocabulary of target tokenizer $\mathcal{V}_{\text{target}}$, underlying distribution $\mathbb{D}$, sampling times N, BPE algorithm $\mathcal{T}$, power-law fit function `pl.fit`, threshold τ

$\tilde{\mathbb{D}} \leftarrow \{\}$
 # Step 1: Prepare for frequency estimation
for N times **do**
  $\mathcal{D}_{\text{aux}} \xleftarrow{\$} \mathbb{D}$ ▷ *randomly sample auxiliary datasets*
  $\tilde{\mathbb{D}} \leftarrow \tilde{\mathbb{D}} \cup \mathcal{D}_{\text{aux}}$
end for
$\mathcal{V}_{\text{shadow}} \leftarrow \mathcal{T}(\mathcal{D}_{\text{aux}})$ ▷ *train shadow tokenizer*
 # Step 2: Estimate components in RTF-SI
$\alpha, x_{\min} \leftarrow$ `pl.fit`$(\mathcal{V}_{\text{shadow}}, \mathcal{D}_{\text{aux}})$ ▷ *fit token frequency*
for $i = x_{\min}+1$ to $|\mathcal{V}_{\text{target}}|$ **do**
  $\text{RTF}(D, t_i) \leftarrow \dfrac{n_D(t_i)}{\sum_{D' \in \tilde{\mathbb{D}} \cup \{D\}} n_{D'}(t_i)}$
  $\text{SI}(t_i, \mathcal{V}_{\text{target}}) \leftarrow \log\left(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} \dfrac{i^\alpha}{j^\alpha}\right)$ ▷ *apply Theorem 4.2*
end for
 # Step 3: Compute membership signal
for $i = x_{\min}+1$ to $|\mathcal{V}_{\text{target}}|$ **do**
  $\text{RTF-SI}(D, t_i, \mathcal{V}_{\text{target}}) \leftarrow \text{RTF}(t_i, D) \cdot \text{SI}(t_i, \mathcal{V}_{\text{target}})$
end for
$\text{RTF-SI}_{\max} \leftarrow \max_{t_i \in \mathcal{V}_{\text{target}}, i > x_{\min}} \text{RTF-SI}(D, t_i, \mathcal{V}_{\text{target}})$
$\textsc{Signal} \leftarrow \dfrac{1}{1 + e^{-\text{RTF-SI}_{\max}}}$ ▷ *normalize by sigmoid*
 # Step 4: Infer the membership
return $\mathbb{1}[\textsc{Signal} > \tau]$

(i) The adversary randomly samples a collection of datasets $\mathcal{D}_{\text{aux}} \leftarrow \mathbb{D}$ N times, comprising a set $\tilde{\mathbb{D}}$. Then, the adversary trains a shadow tokenizer $f_{\mathcal{V}_{\text{shadow}}}$ using a $\mathcal{D}_{\text{aux}} \subseteq \tilde{\mathbb{D}}$.

(ii) The adversary fits the power-law distribution in Equation 9 using the vocabulary $\mathcal{V}_{\text{shadow}}$ and its training data. For each token $t_i \in \mathcal{V}_{\text{target}}$ where $i > x_{\min}$, the adversary approximate its $\text{RTF}(D, t_i)$ on set $\tilde{\mathbb{D}} \bigcup \{D\}$, and estimate its $\text{SI}(t_i, \mathcal{V}_{\text{target}})$ via the fitted power-law distribution.

(iii) Based on the Theorem 4.2, the membership signal for target dataset D is defined as follows:

$$\sigma\Big(\max_{t_i \in \mathcal{V}_{\text{target}}, i > x_{\min}} \underbrace{\frac{n_D(t_i)}{\sum_{D' \in \tilde{\mathbb{D}} \cup \{D\}} n_{D'}(t_i)}}_{\text{RTF}(D, t_i)} \cdot \underbrace{\log\Big(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} i^\alpha / j^\alpha\Big)}_{\text{SI}(t_i, \mathcal{V}_{\text{target}})}\Big), \quad (12)$$

where σ denotes the sigmoid function [85]. Thereby, the value of Equation 12 ranges from 0 to 1.

(iv) If the membership signal is larger than a decision-making threshold τ, output 1 (*member*). Otherwise, output 0.

The detailed process of this attack is shown in Algorithm 2. The membership signal for MIA via Frequency Estimation is defined as the maximum RTF-SI value for the target dataset D. Specifically, if including dataset D in the training data is necessary for at least one token t_i to be merged into the vocabulary $\mathcal{V}_{\text{target}}$, it suggests that the absence of dataset D has a significant influence on the already constructed vocabulary. Consequently, it is likely that dataset D is a *member*.

5 Attack Evaluation

In this section, we first introduce the experimental settings. Next, we develop two shadow-free membership inference methods to serve as additional exploration and baselines. Finally, we present the evaluation results.

5.1 Experimental Setup

Datasets. According to disclosures from existing LLMs [5, 8, 35, 100], the training data for tokenizers is primarily sourced from publicly available web content. To ensure a realistic evaluation of our attacks, we therefore utilize real-world web data from the *C4* corpus [81] in our evaluation. Specifically, the *C4* corpus is created by cleaning and filtering web pages from Common Crawl, widely used for training and evaluating natural language processing models. In our evaluation, we utilize 1,681,296 web pages across 4,133 websites (i.e., $\mathbb{D}$) from the *C4* corpus, with each website treated as a dataset D.

Tokenizer Training. Following prior work [11, 94], we randomly select half of the datasets in $\mathbb{D}$ to serve as training data. Consistent with DeepSeek [7], we train the target tokenizers using the HuggingFace library [48]. The tokenizer vocabulary size ranges from 80,000 to 200,000 tokens, with the upper bound matching that of OpenAI's latest tokenizer, o200k [75]. For MIA via Merge Similarity and MIA via Vocabulary Overlap, the adversary trains 96 shadow tokenizers. For MIA via Frequency Estimation, the adversary samples the auxiliary datasets 10 times to compose a set $\tilde{\mathbb{D}}$.

Verification of No Distribution Shifts. As discussed in previous works [20, 25, 68], distribution shifts between *members* and *non-members* can invalidate the evaluation of MIAs. In such scenarios, an evaluator [20, 68] can exploit bag-of-words features extracted from test samples and train a random forest classifier to detect whether a sample was part of the training data, even without access to the target model. To ensure that there is no distribution shift in our evaluation, we follow the methodology of prior work [20] by training a random forest classifier and leveraging the bag-of-words features to distinguish between *members* and *non-members*. The experimental

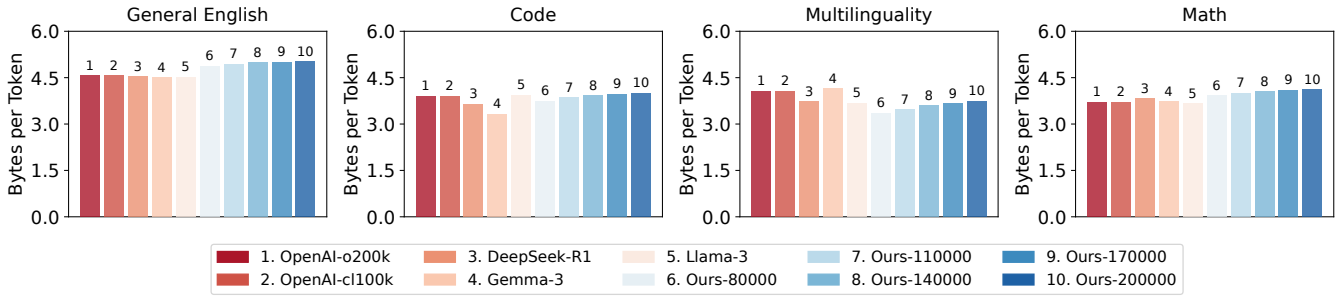


Figure 6: Comparison of tokenizer utility based on the metric of bytes per token. Specifically, "Ours-80000" refers to our trained target tokenizer with a vocabulary size of 80,000 tokens. The above experimental results indicate that the utility performance of the target tokenizers utilized in our evaluations is comparable to that of tokenizers used in state-of-the-art LLMs [33, 35, 75].

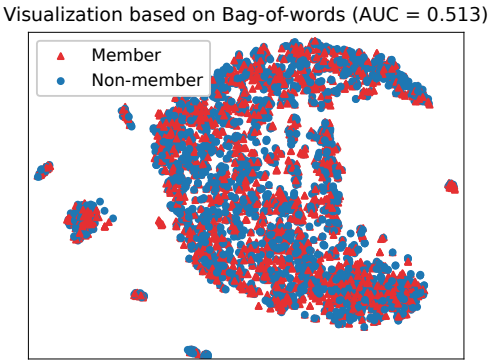


Figure 7: Visualization of the test set using t-SNE [62]. The results confirm no distribution shifts in our evaluation.

results are illustrated in Figure 7. The AUC score of 0.513 via this approach to distinguish *members* and *non-members* confirms the absence of distribution shifts in our evaluations.

Comparison to Commercial Tokenizers. To assess how well our trained target tokenizers in experimental settings mimic the commercial tokenizers in the real world, we compare their utility performance using a standard compression metric: the bytes per token [59]. Specifically, following prior work [19, 95], we compute the ratio of UTF-8 bytes in a given text to the number of tokens generated by the tokenizer. A higher score is desired. We conduct this evaluation across widely used benchmarks: general English (*WikiText-103* [69]), code (*GitHub Code* [49]), multilingual content (*MGSM* [90]), and mathematics (*GPQA* [83]). As shown in Figure 6, the utility performance of our trained tokenizers is comparable to that of the commercial tokenizers [33, 35, 75, 100].

Attack Baselines. As our work is the first to investigate MIAs targeting tokenizers, there are no existing baselines from prior studies. Therefore, we establish our own baselines by comparing MIA via Vocabulary Overlap and MIA via Frequency Estimation with three other methods: MIA via Merge Similarity (see Section 4.1), as well as two attack methods we developed. These two attacks are described below:

- **MIA via** Naive Bayes**.** Since every token originates from at least one of the tokenizer's training datasets, the adversary can approximate the empirical probability $\Pr(t_i \to D)$ that the token $t_i \in \mathcal{V}_{\text{target}}$ comes from the dataset D. Specifically, this probability $\Pr(t_i \to D)$ can be represented by:

$$\Pr(t_i \to D) = \frac{n_D(t_i)}{\sum_{D' \in \tilde{\mathbb{D}} \cup \{D\}} n_{D'}(t_i)}, \tag{13}$$

where $n_D(t_i)$ is the frequency of t_i in D, and $\tilde{\mathbb{D}}$ is constructed by sampling auxiliary datasets $\mathcal{D}_{\text{aux}}$ N times as an estimation for distribution $\mathbb{D}$. We default to setting $N = 10$ in evaluation. The probability that $D \in \mathcal{D}_{\text{mem}}$ is given by:

$$\Pr(D \in \mathcal{D}_{\text{mem}}) = 1 - \Pr\Big(\bigcap_{t_i \in \mathcal{V}_{\text{target}}} t_i \not\to D \Big), \tag{14}$$

where $t_i \not\to D$ means t_i was not sourced from D. Assuming rare tokens in training data typically come from disjoint datasets, the probability $\Pr(D \in \mathcal{D}_{\text{mem}})$ thus can be approximated via the Naive Bayes [101]:

$$\Pr(D \in \mathcal{D}_{\text{mem}}) \approx 1 - \prod_{t_i \in \mathcal{V}_{\text{rare}}} \Pr(t_i \not\to D) \tag{15}$$

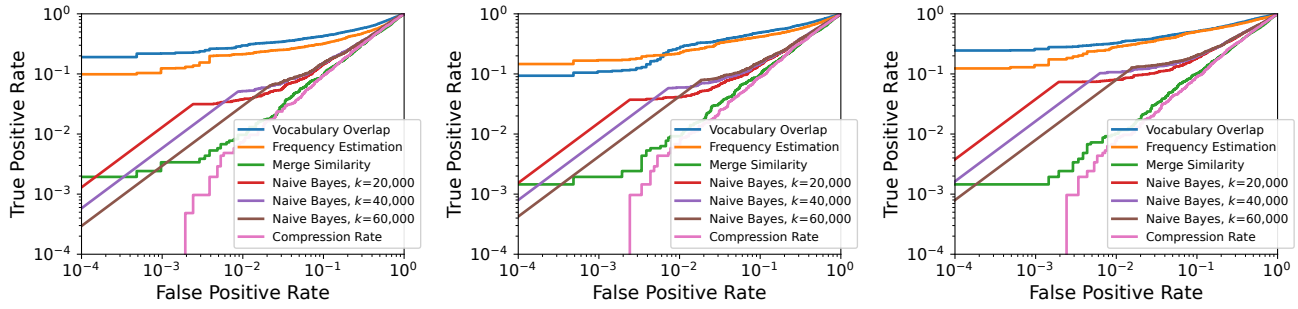
$$\approx 1 - \prod_{t_i \in \mathcal{V}_{\text{rare}}} \big(1 - \Pr(t_i \to D)\big), \tag{16}$$

where $\mathcal{V}_{\text{rare}} \subseteq \mathcal{V}_{\text{target}}$ is the set of tokens with the top k merge orders, selected as likely rare tokens in training data (see Equation 9). The membership signal for the target dataset D is defined by the probability $\Pr(D \in \mathcal{D}_{\text{mem}})$. If it is larger than a threshold τ, output 1 (*member*); or, output 0.

- **MIA via** Compression Rate**.** The objective of tokenizer training is to maximize the compression rate of a given text corpus [109]. Based on this optimization objective, we hypothesize that a tokenizer achieves higher compression rates on datasets it was trained on [38]. Leveraging this insight, an adversary can calculate the compression rate of a given dataset D using the metric of bytes per token [59].

Table 2: Comparison of MIAs against target tokenizers. Here, BA denotes the metric of balanced accuracy, and TPR refers to the metric of TPR @ 1.0% FPR. The bold values indicate the best performance, while the underlined values denote the second-best. It is observed that MIA via Vocabulary Overlap and MIA via Frequency Estimation outperform other baseline methods.

Attack Approach	Shadow Tokenizers	Auxiliary Datasets	$\lvert\mathcal{V}_{\text{target}}\rvert = 80{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 110{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 140{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 170{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 200{,}000$		
			AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Compression Rate	○	○	0.507	0.513	0.72%	0.509	0.517	0.71%	0.508	0.514	0.68%	0.508	0.513	0.82%	0.509	0.516	0.77%
Naive Bayes, $k=20{,}000$	○	●	0.534	0.526	3.78%	0.526	0.524	3.44%	0.535	0.533	4.11%	0.546	0.538	5.86%	0.564	0.551	8.03%
Naive Bayes, $k=40{,}000$	○	●	0.543	0.530	5.18%	0.546	0.533	6.10%	0.542	0.537	5.95%	0.550	0.542	7.60%	0.572	0.557	10.70%
Naive Bayes, $k=60{,}000$	○	●	0.543	0.530	2.22%	0.551	0.538	5.57%	0.543	0.536	2.80%	0.553	0.545	4.94%	0.572	0.557	8.86%
Merge Similarity	●	●	0.493	0.509	1.06%	0.494	0.507	1.02%	0.494	0.508	0.97%	0.495	0.506	0.92%	0.495	0.508	0.87%
Vocabulary Overlap	●	●	**0.693**	**0.666**	**26.77%**	**0.718**	**0.672**	**28.75%**	**0.736**	**0.696**	**29.72%**	**0.761**	**0.709**	**32.53%**	**0.771**	**0.711**	**34.61%**
Frequency Estimation	●	●	0.610	0.614	21.30%	0.645	0.641	22.00%	0.676	0.660	22.41%	0.707	0.681	25.61%	0.740	0.705	27.88%



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 140,000 (c) Vocabulary Size: 200,000

Figure 8: Success rate of our attacks on tokenizers with different vocabulary sizes. The experimental results demonstrate that, when evaluated at low false positive rates, both MIA via Vocabulary Overlap and MIA via Frequency Estimation consistently outperform other methods. Notably, even at a 0.01% false positive rate, both attacks achieve a true positive rate of nearly 10%.

The membership signal for the target dataset D is defined by this compression rate. If it is larger than a decision-making threshold τ, output 1 (*member*); otherwise, output 0.

Evaluation Metrics. Following prior studies [11, 25, 94], we report the MIA performance using three convincing metrics:

- **AUC.** This metric quantifies the overall distinguishability of an MIA by computing the area under the receiver operating characteristic (ROC) curve [13].
- **Balanced Accuracy.** This metric (denoted as BA) measures the overall correct predictions on membership by averaging the true positive rate and the true negative rate.
- **TPR at Low FPR.** Proposed by [11], this metric evaluates the true positive rate (TPR) when the false positive rate (FPR) is low. Following prior work [41, 67], we report TPR @ 1.0% FPR in our evaluations (denoted as TPR).

5.2 Main Results

Finding 1. *According to prior work [46, 66], scaling up the intelligence of LLMs involves expanding the tokenizer's vocabulary [97] and thus improving its compression efficiency [59]. Yet, the results show it also increases a tokenizer's vulnerability to MIAs.*

Overall Performance. As shown in Table 2, the MIA via Vocabulary Overlap and the MIA via Frequency Estimation consistently demonstrate strong performance and outperform other baseline methods across different vocabulary sizes. For instance, the MIA via Vocabulary Overlap achieves an AUC score of 0.771 when evaluated on a target tokenizer with 200,000 tokens, whereas the MIA via Frequency Estimation achieves a comparable AUC score of 0.740. Beyond these results, we observe that the performance of both attacks improves as vocabulary size increases. This trend suggests that MIAs targeting tokenizers in LLMs may become even more effective as state-of-the-art models continue to scale and adopt larger vocabularies in their tokenizers [46, 66, 97]. One possible explanation is that larger vocabularies contain more tokens, which may increase the likelihood of merging the distinctive tokens from the training data. As a result, expanding the tokenizer's vocabulary may unintentionally increase its vulnerability to effective MIAs.

ROC Analysis. The prior study [11] has highlighted the importance of MIAs being able to reliably infer even a small number of a model's training data. To demonstrate this capability of our MIAs, Figure 8 presents the full log-scale ROC curves for various attack methods across different vocabulary sizes. It is observed that both MIA via Vocabulary Overlap

Table 3: Impact of the target dataset size on MIAs. BA denotes balanced accuracy. TPR refers to TPR @ 1.0% FPR. It is observed that MIA via Vocabulary Overlap and MIA via Frequency Estimation perform better for target datasets with larger sizes.

Attack Approach	#Dataset Size	$\lvert\mathcal{V}_{\text{target}}\rvert = 80{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 110{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 140{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 170{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert = 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	$\lvert D\rvert \in [0,400)$	0.672	0.652	21.58%	0.696	0.655	24.76%	0.714	0.676	27.56%	0.737	0.687	27.82%	0.747	0.692	29.22%
	$\lvert D\rvert \in [400,800)$	0.739	0.695	33.73%	0.758	0.718	41.27%	0.791	0.755	42.77%	0.808	0.761	43.37%	0.808	0.766	43.67%
	$\lvert D\rvert \in [800,1200)$	0.773	0.720	33.75%	0.785	0.757	43.75%	0.797	0.767	45.00%	0.826	0.813	47.50%	0.882	0.838	62.50%
Frequency Estimation	$\lvert D\rvert \in [0,400)$	0.599	0.608	18.91%	0.631	0.632	19.73%	0.662	0.648	20.31%	0.695	0.668	21.83%	0.729	0.691	25.84%
	$\lvert D\rvert \in [400,800)$	0.629	0.624	23.49%	0.683	0.662	29.27%	0.728	0.697	30.42%	0.747	0.713	31.63%	0.774	0.736	32.83%
	$\lvert D\rvert \in [800,1200)$	0.758	0.734	33.75%	0.772	0.756	40.00%	0.774	0.761	41.25%	0.814	0.789	50.00%	0.843	0.810	53.75%

Table 4: Time cost (hours) for training N shadow tokenizers, where each tokenizer has a vocabulary size of 200,000.

Tokenizer Count	N=1	N=32	N=64	N=96	N=128
Training Time	0.024	0.731	1.498	2.251	3.054

Table 5: Time cost (hours) for MIAs via Vocabulary Overlap and Frequency Estimation inferring 4,133 target datasets.

Vocabulary Size	80,000	110,000	140,000	170,000	200,000
Vocabulary Overlap	1.230	1.608	2.067	2.613	3.375
Frequency Estimation	0.170	0.235	0.303	0.373	0.432

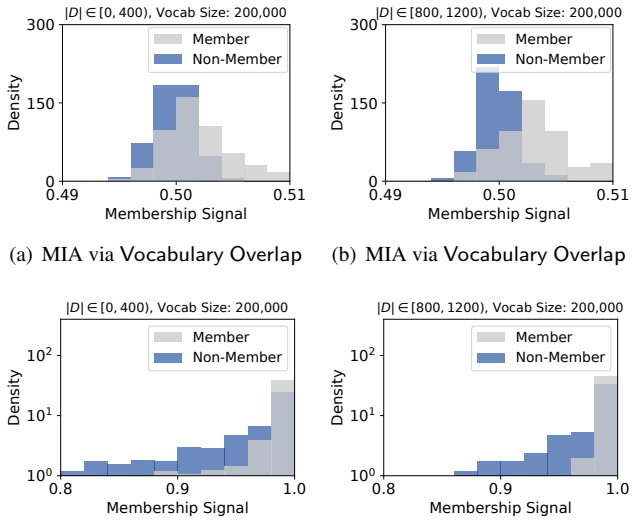
and MIA via Frequency Estimation can reliably infer the membership of datasets, particularly in regions with low false positive rates. For example, when applied to a tokenizer with 140,000 tokens, these two attacks achieve true positive rates ranging from approximately 10% to 30% at a false positive rate below 1%. Notably, even at a false positive rate of 0.01%, both attacks can still achieve a true positive rate of nearly 10%. Additional ROC curve results can be found in Figure 13.

Efficiency Analysis. We further analyze the computational cost for both MIA via Vocabulary Overlap and MIA via Frequency Estimation across two phases: shadow tokenizer training and the remaining inference. In the phase of shadow tokenizer training, MIA via Vocabulary Overlap trains multiple tokenizers (e.g., 96), resulting in a high computational cost. In contrast, MIA via Frequency Estimation requires training only a single tokenizer, significantly reducing the overall cost. As shown in Table 4, this leads to substantial savings in training time. In the inference phase, MIA via Vocabulary Overlap involves frequent comparisons across different tokenizers, whereas MIA via Frequency Estimation primarily estimates a power-law distribution, a much simpler computation. Table 5 confirms the shorter inference time of the latter method. For example, MIA via Vocabulary Overlap takes over two hours to infer the membership of 4,133 datasets from a tokenizer with 140,000 tokens. However, MIA via Frequency Estimation accomplishes the same task in under 20 minutes, making it efficient for large-scale attacks.

5.3 Ablation Study

Finding 2. *The membership status of the target dataset with more data samples is typically more accurately inferred by MIAs from the tokenizer.*

Impact of Dataset Size $|D|$. Recent studies [42, 63, 67, 79]



(a) MIA via Vocabulary Overlap (b) MIA via Vocabulary Overlap

(c) MIA via Frequency Estimation (d) MIA via Frequency Estimation

Figure 9: Distribution of *members* and *non-members*.

have shown that increasing the amount of data used for membership inference can improve the attack performance. This finding is particularly relevant in the context of high-value litigation nowadays, where the datasets at stake are often massive [3, 73]. Motivated by this, we investigate whether MIAs can more effectively infer the membership of larger datasets from the tokenizers. Table 3 presents the performance of MIA via Vocabulary Overlap and MIA via Frequency Estimation for varying-size target datasets. The results show that both attacks become more effective as the dataset size increases, achieving particularly strong performance on large datasets. For instance, the MIA via Vocabulary Overlap achieves an AUC score of 0.882 on datasets containing 800 to 1,200 data samples, whereas the MIA via Frequency Estimation

Table 6: MIAs against tokenizers with the min count defense. Here, n_{min} denotes a threshold. If a token appears fewer than n_{min} times in training data, it is likely a distinctive token and will be excluded from the vocabulary $\mathcal{V}_{\text{target}}$. $|\mathcal{V}_{\text{target}}| \leq 80{,}000$ indicates the vocabulary size prior to applying defense is 80,000. BA denotes balanced accuracy. TPR refers to TPR @ 1.0% FPR.

Attack Approach	#Min Count	$\lvert\mathcal{V}_{\text{target}}\rvert \leq 80{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 110{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 140{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 170{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	w/o defense	0.693	0.666	26.77%	0.718	0.672	28.75%	0.736	0.696	29.72%	0.761	0.709	32.53%	0.771	0.711	34.61%
	w/ $n_{\text{min}} = 32$	0.663	0.638	23.57%	0.699	0.657	23.76%	0.718	0.671	26.48%	0.736	0.686	28.41%	0.746	0.691	30.83%
	w/ $n_{\text{min}} = 48$	0.663	0.638	23.57%	0.697	0.655	23.76%	0.714	0.671	25.79%	0.717	0.665	26.09%	0.734	0.683	30.83%
	w/ $n_{\text{min}} = 64$	0.663	0.638	21.93%	0.685	0.640	23.57%	0.699	0.657	23.77%	0.707	0.663	26.48%	0.717	0.671	26.48%
Frequency Estimation	w/o defense	0.610	0.614	21.30%	0.645	0.641	22.00%	0.676	0.660	22.56%	0.707	0.681	25.61%	0.740	0.705	27.88%
	w/ $n_{\text{min}} = 32$	0.600	0.602	18.25%	0.633	0.630	21.30%	0.664	0.647	22.41%	0.695	0.669	24.15%	0.730	0.693	28.80%
	w/ $n_{\text{min}} = 48$	0.598	0.598	18.05%	0.633	0.626	21.06%	0.663	0.645	21.78%	0.690	0.663	23.23%	0.692	0.664	25.41%
	w/ $n_{\text{min}} = 64$	0.596	0.600	17.13%	0.630	0.624	20.72%	0.661	0.645	23.81%	0.666	0.648	22.36%	0.668	0.648	23.72%

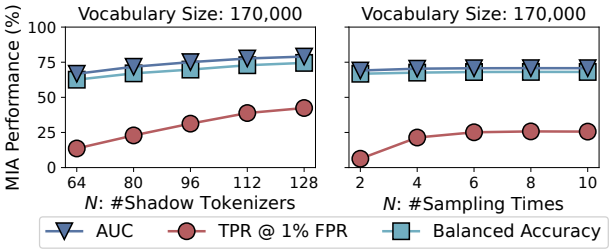


Figure 10: Impact of N. Left: MIA via Vocabulary Overlap, training N shadow tokenizers. Right: MIA via Frequency Estimation, sampling auxiliary datasets N times.

achieves a competitive AUC score of 0.843. Figure 9 further illustrates the relationship between dataset size and the membership signal in both attacks. As the dataset size increases, the overlap between the membership signal distributions for *members* and *non-members* decreases, thereby enhancing the discriminative power of the attacks. This reduced overlap is likely attributable to the presence of more distinctive tokens in larger datasets, which serve to strengthen the membership signal. Additional evaluations illustrating the membership signal distributions are presented in Figure 14 and Figure 15.

Impact of Hyperparameter N. We further analyze the impact of the hyperparameter N on the performance of MIAs. Specifically, in the case of MIA via Vocabulary Overlap, an adversary trains N shadow tokenizers to capture distinctive tokens. As shown in the left plot of Figure 10, the effectiveness of this attack improves steadily as N increases. However, after training more than 112 shadow tokenizers, the performance gain begins to plateau. In another attack, MIA via Frequency Estimation, the adversary samples auxiliary datasets N times to estimate probabilities in membership inference. As shown in the right plot of Figure 10, the effectiveness of this attack also increases with larger N. Nevertheless, the performance of MIA via Frequency Estimation tends to be stable once N exceeds 6. Although the results demonstrate that both attacks benefit from increasing N, a larger value of N also incurs greater resource consumption during inference.

Table 7: Tokenizer utility measured by bytes per token. Utility scores that decrease after applying the defense are in red.

Benchmark	$\lvert\mathcal{V}_{\text{target}}\rvert \leq$	Tokenizer Utility Measured by Bytes per Token			
		w/o defense	w/ $n_{\text{min}} = 32$	w/ $n_{\text{min}} = 48$	w/ $n_{\text{min}} = 64$
WikiText [69]	80,000	4.873	4.873 (-0.0)	4.873 (-0.0)	4.873 (-0.0)
	110,000	4.943	4.943 (-0.0)	4.943 (-0.0)	4.943 (-0.0)
	140,000	4.986	4.986 (-0.0)	4.986 (-0.0)	4.986 (-0.0)
	170,000	5.008	5.008 (-0.0)	5.006 (-0.002)	4.995 (-0.013)
	200,000	5.025	5.025 (-0.0)	5.012 (-0.013)	5.000 (-0.025)
Github [49]	80,000	3.740	3.739 (-0.001)	3.739 (-0.001)	3.738 (-0.002)
	110,000	3.853	3.851 (-0.002)	3.851 (-0.002)	3.849 (-0.004)
	140,000	3.924	3.921 (-0.003)	3.921 (-0.003)	3.918 (-0.006)
	170,000	3.973	3.970 (-0.003)	3.967 (-0.006)	3.947 (-0.026)
	200,000	4.009	4.006 (-0.003)	3.990 (-0.019)	3.965 (-0.044)
MGSM [90]	80,000	3.357	3.356 (-0.001)	3.356 (-0.001)	3.355 (-0.002)
	110,000	3.476	3.475 (-0.001)	3.474 (-0.002)	3.473 (-0.003)
	140,000	3.601	3.601 (-0.0)	3.601 (-0.0)	3.600 (-0.001)
	170,000	3.663	3.663 (-0.0)	3.662 (-0.001)	3.639 (-0.024)
	200,000	3.740	3.740 (-0.0)	3.739 (-0.001)	3.716 (-0.024)
GPQA [83]	80,000	3.925	3.924 (-0.001)	3.924 (-0.001)	3.923 (-0.002)
	110,000	4.008	4.007 (-0.001)	4.006 (-0.002)	4.003 (-0.005)
	140,000	4.056	4.055 (-0.001)	4.053 (-0.003)	4.050 (-0.006)
	170,000	4.094	4.094 (-0.0)	4.092 (-0.002)	4.086 (-0.008)
	200,000	4.117	4.116 (-0.001)	4.109 (-0.008)	4.093 (-0.024)

5.4 Adaptive Defense

Finding 3. *Removing infrequent tokens from the target tokenizer's vocabulary can partially reduce the effectiveness of MIAs. However, this mitigation comes at the cost of reduced tokenizer utility. Moreover, MIAs remain effective when inferring large datasets.*

While defense against MIAs is not the primary focus of this work, we have also explored the defense mechanism to mitigate membership leakage in tokenizers. Specifically, previous studies [42, 94, 103] have demonstrated that overfitting signals are a key requirement for the success of MIAs. This suggests that methods designed to reduce overfitting may function as effective defense mechanisms [1, 108]. Building on this insight, we assume an adaptive defender who mitigates our attacks by employing mechanisms that reduce the overfitting of distinctive tokens in the target tokenizer's vocabulary.

Defender's Objective. Given the target tokenizer $f_{\mathcal{V}_{\text{target}}}$, the defender's goal is to reduce the inference accuracy of MIAs

Table 8: Impact of the target dataset size on defense mechanism ($n_{\min} = 64$). $|\mathcal{V}_{\text{target}}| \leq 80{,}000$ indicates the vocabulary size prior to applying defense is 80,000. TPR refers to TPR @ 1.0% FPR. The results show our MIAs remain effective on large datasets.

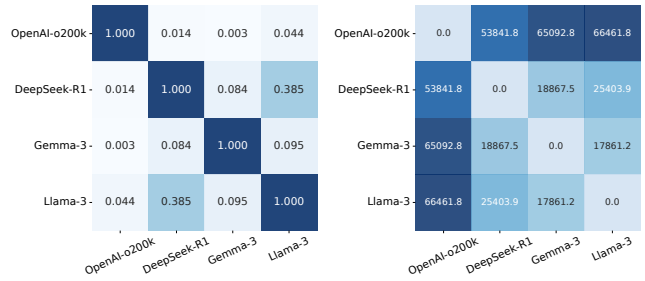
Attack Approach	#Dataset Size	$\lvert\mathcal{V}_{\text{target}}\rvert \leq 80{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 110{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 140{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 170{,}000$			$\lvert\mathcal{V}_{\text{target}}\rvert \leq 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	$\lvert D\rvert \in [0,400)$	0.646	0.632	19.16%	0.662	0.636	21.25%	0.677	0.642	21.25%	0.683	0.647	23.30%	0.694	0.655	24.06%
	$\lvert D\rvert \in [400,800)$	0.703	0.666	21.25%	0.731	0.675	21.58%	0.751	0.699	22.85%	0.761	0.711	32.50%	0.772	0.717	37.95%
	$\lvert D\rvert \in [800,1200)$	0.712	0.670	25.30%	0.743	0.702	27.11%	0.768	0.729	30.12%	0.795	0.746	33.73%	0.797	0.761	38.75%
Frequency Estimation	$\lvert D\rvert \in [0,400)$	0.591	0.598	17.50%	0.620	0.619	18.20%	0.651	0.636	20.11%	0.654	0.637	21.07%	0.656	0.639	21.39%
	$\lvert D\rvert \in [400,800)$	0.619	0.615	22.29%	0.672	0.653	28.31%	0.717	0.682	30.42%	0.718	0.683	31.93%	0.723	0.686	32.83%
	$\lvert D\rvert \in [800,1200)$	0.734	0.717	26.25%	0.739	0.731	33.75%	0.744	0.736	33.75%	0.745	0.742	35.00%	0.748	0.746	38.75%

on $f_{\mathcal{V}_{\text{target}}}$, while preserving its utility as much as possible.

Defender's Capabilities. We assume that the defender is aware of the MIA strategy targeting the tokenizer $f_{\mathcal{V}_{\text{target}}}$, including the conditions that are important for their success. Specifically, our attacks rely on the distinctive tokens, which appear infrequently in the training data and overfit into the vocabulary $\mathcal{V}_{\text{target}}$. As a defense, the defender may modify the vocabulary $\mathcal{V}_{\text{target}}$ by identifying and removing such infrequent tokens. Thereby, it can mitigate the membership inference without significantly degrading the tokenizer utility.

Min Count Mechanism. We introduce the min count mechanism as an adaptive defense against our attacks. In this mechanism, the defender post-processes the trained vocabulary $\mathcal{V}_{\text{target}}$ by filtering infrequent tokens. Let $n_{\min} \in \mathbb{Z}_{>0}$ denote the filtering threshold, and let $n_{D'}(t_i)$ represent the count of token t_i in dataset D'. For each token $t_i \in \mathcal{V}_{\text{target}}$, if the aggregated count $\sum_{D' \in \mathcal{D}_{\text{mem}}} n_{D'}(t_i)$ across the tokenizer's training data is less than $n_{\min}$, the defender removes t_i from trained vocabulary $\mathcal{V}_{\text{target}}$. Table 6 shows the results of MIAs against tokenizers with the min count mechanism. It is observed that, as the threshold $n_{\min}$ increases, the defense can partially reduce the effectiveness of MIAs against tokenizers. However, this comes at the cost of the tokenizer's utility. Table 7 shows that the compression efficiency of bytes per token diminishes under more strict filtering rules. While the min count mechanism can mitigate some inference risks, our MIAs remain effective, particularly for large datasets. For example, Table 8 reports an AUC of 0.797 when applying MIA via Vocabulary Overlap to infer the membership of datasets ranging in size from 800 to 1,200 samples. Additional experimental results under the min count mechanism are provided in Figure 9 and Figure 10.

Differentially Private Mechanism. Prior study [57] has demonstrated that differentially private mechanisms can typically reduce the vulnerability of membership leakage in machine learning models. Represented by DP-SGD [1], these defense methods [22, 105] add noise to gradients during model training, thereby obfuscating the distinction between *members* and *non-members*. However, to the best of our knowledge, existing research has not investigated the application of differentially private mechanisms specifically designed for LLM tokenizers. As a result, it remains unclear how differential privacy can be leveraged to mitigate membership leakage through tokenizers. Given this, we hope to evaluate our proposed attacks on differentially private tokenizers, should such mechanisms become available.



(a) Jaccard Similarity (b) Differences in Merge Index

Figure 11: Comparison of tokenizers in real-world LLMs.

5.5 Additional Investigations

Distinctive Tokens in Real-world Tokenizers. Our MIAs exploit distinctive tokens, which can vary across different tokenizers, considering both their occurrence and their merge indices. A key question is whether real-world tokenizers differ in such ways that might enable effective MIAs. To investigate this, we compare the vocabularies of tokenizers used in real-world LLMs [33, 35, 75, 100]. Specifically, we limit our analysis to the first 120,000 tokens of each tokenizer to enable a fair comparison across varying sizes. Figure 11(a) presents the similarity between tokenizers' vocabularies based on the Jaccard index [6]. If there were no distinctive tokens, we would expect high similarity scores between any pair of tokenizers. However, the evaluation results show that real-world tokenizers contain a significant number of distinctive tokens, as evidenced by the maximum Jaccard index of only 0.385 observed between DeepSeek-R1 [35] and Llama-3 [100]. In addition, Figure 11(b) illustrates the average absolute differences in the merge index for tokens between any two vocabularies. If there were no significant differences between the two vocabularies, we would expect only slight variations in the token merge indices. Nevertheless, the results suggest that token merge indices also vary largely in real-world tokenizers.

6 Related Work

Tokenizers. Tokenizers play a crucial role in enabling the generation and comprehension capabilities of LLMs [35, 70]. By converting raw text into discrete tokens, tokenizers provide the input representations that models require for inference [109]. Recent research has highlighted the connection between tokenizers and scaling laws, suggesting that larger models benefit from a larger tokenizer vocabulary, leading to improved performance under the same training cost [46, 97]. Nevertheless, our work reveals that scaling up the tokenizer's vocabulary increases its vulnerability to MIAs. We underscore the necessity of paying attention to these overlooked risks.

MIAs on Classifiers. Membership inference attacks [30, 45, 60, 76] are designed to determine whether a specific entity was included in the training data of a machine learning (ML) model. These attacks have become fundamental tools for quantifying privacy leakage in various scenarios [14, 51, 77]. The first MIA is proposed by Shokri et al. [94], which focuses on demining record-level membership from ML-based classifiers. Building on the first work, subsequent research has examined MIAs under different assumptions regarding the adversary's access to the classifier. Sablayrolles et al. [86] investigate the black-box setting [50], where the adversary can only access the classifier's output. They employ Bayesian learning [28] to approximate optimal strategies for membership inference in this context. In contrast, Leino et al. [53] explore MIAs in the white-box setting, where the adversary can access the model's internal components. They proposed using confidence scores to improve attack performance for membership inference. Choquette-Choo et al. [16] develop the label-only MIAs. They exploit the sensitivity of the output label to the input perturbation for membership inference.

MIAs on LLMs. Recent studies have underscored the importance of detecting whether specific data was used during the pre-training of LLMs, as such data may contain copyrighted or sensitive information [98, 99]. To address these, several membership inference methods have been proposed. Shi et al. [92] introduced MIN-K% PROB, a technique that detects membership by identifying outlier words with unusually low probabilities in previously unseen examples. Duarte et al. [26] proposed DE-COP, which probes LLMs using multiple-choice questions that include both verbatim and paraphrased versions of candidate sentences. Zhang et al. [107] presented MIN-K%++, which leverages the insight that training samples are likely to be local maxima under maximum likelihood training. However, these MIA methods face significant evaluation challenges. Duan et al. [25] argue that existing benchmarks suffer from a temporal distribution shift between *members* and *non-members*, potentially invalidating evaluation results. Meeus et al. [68] further observe that many methods may incorporate mislabeled samples and use impractical evaluation models in their experiments. To mitigate these privacy threats, differential privacy (DP) serves as a principal defense mechanism [27]. Defenders can leverage these techniques [1, 22] to add noise to the gradients during model training, thereby obfuscating the distinction between *members* and *non-members*. Yet, it is still unknown how to apply DP [24] to tokenizers.

7 Discussion

LLM Dataset Inference. Recent research [64] has emphasized the importance of dataset inference in LLMs. In particular, existing methods [15, 63, 102] attempt to predict whether a specific dataset was used to train a target LLM by analyzing the model's output. However, as noted in Section 1, these methods face significant challenges during evaluation, such as mislabeled samples, distribution shifts, and mismatches in target model sizes compared to real-world models. Furthermore, these attacks typically introduce additional assumptions about the adversary, such as access to model output loss or fine-tuning the target model, which are not guaranteed to hold in closed-source LLMs. Moreover, they can be defended by adding noise during the model training process [108].

Limitations of Tokenizer Inference. Our work has two main limitations in its broader evaluation. First, due to the absence of ground-truth training data for commercial tokenizers, we are unable to evaluate our attacks on them. Instead, we conduct evaluations on our trained tokenizers with vocabulary sizes and utility comparable to those of commercial tokenizers (see Figure 6). Notably, this limitation is not unique to our work but is a common challenge in the field of membership inference, as also observed in other related studies [42, 50, 55, 60, 63]. Second, our attack evaluations are restricted to tokenizers used in LLMs. As a result, the feasibility of MIAs on tokenizers for classification models [21, 61] and diffusion-based language models [34, 87] remains unexplored, representing a promising direction for future work.

8 Conclusion

In this paper, we review the limitations of existing MIAs against pre-trained LLMs and introduce the tokenizer as a new attack vector to address these challenges. To demonstrate its feasibility for membership inference, we present the first study of MIAs on tokenizers of LLMs. By analyzing overfitting signals during tokenizer training, we proposed five attack methods for inferring dataset membership. Extensive evaluations on millions of Internet data demonstrate that our shadow-based attacks achieve strong performance. To mitigate these attacks, we further propose an adaptive defense mechanism. Although our proposed defense can reduce the membership leakage, it does so at the cost of tokenizer utility. Our findings highlight the vulnerabilities associated with LLMs' tokenizers. Through this endeavor, we hope our research contributes to the design of privacy-preserving tokenizers, towards building secure machine learning systems.

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *CCS*, pages 308–318, 2016.

[2] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[3] Bobby Allyn. Judge allows new york times copyright lawsuit to go forward. www.npr.org/2025/03/26/nx-s1-5288157/new-york-times-openai-copyright-case-goes-forward, 2025.

[4] Anthropic. Tokenizer for anthropic large language models. Tokenizer for Use with Anthropic's Models, 2024.

[5] Anthropic. Claude opus 4 & claude sonnet 4. System Card, 2025. Anthropic System Card.

[6] Sujoy Bag, Sri Krishna Kumar, and Manoj Kumar Tiwari. An efficient recommendation generation using relevant jaccard similarity. *Information Sciences*, 483:53–64, 2019.

[7] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, et al. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

[8] 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. Pythia: A suite for analyzing large language models across training and scaling. In *ICML*, pages 2397–2430. PMLR, 2023.

[9] Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. Gpt-neox-20b: An open-source autoregressive language model. *Challenges & Perspectives in Creating Large Language Models*, page 95, 2022.

[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020.

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

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

[13] André M Carrington, Douglas G Manuel, Paul W Fieguth, Tim Ramsay, Venet Osmani, Bernhard Wernly, Carol Bennett, Steven Hawken, Olivia Magwood, Yusuf Sheikh, et al. Deep roc analysis and auc as balanced average accuracy, for improved classifier selection, audit and explanation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):329–341, 2022.

[14] Yuetian Chen, Zhiqi Wang, Nathalie Baracaldo, Swanand Ravindra Kadhe, and Lei Yu. Evaluating the dynamics of membership privacy in deep learning. *arXiv preprint arXiv:2507.23291*, 2025.

[15] Hyeong Kyu Choi, Maxim Khanov, Hongxin Wei, and Yixuan Li. How contaminated is your benchmark? measuring dataset leakage in large language models with kernel divergence. In *Forty-second International Conference on Machine Learning*, 2025.

[16] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *ICML*, pages 1964–1974. PMLR, 2021.

[17] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[18] United States Code. U.s. copyright act, title 17, section 107. https://www.copyright.gov/title17/92chap1.html, 1976.

[19] Gautier Dagan, Gabriel Synnaeve, and Baptiste Rozière. Getting the most out of your tokenizer for pre-training and domain adaptation. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9784–9805, 2024.

[20] Debeshee Das, Jie Zhang, and Florian Trantèr. Blind baselines beat membership inference attacks for foundation models. In *2025 IEEE Security and Privacy Workshops (SPW)*, pages 118–125. IEEE, 2025.

[21] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, 2019.

[22] Minxin Du, Xiang Yue, Sherman SM Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. Dp-forward: Fine-tuning and inference on language models with differential privacy in forward pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pages 2665–2679, 2023.

[23] Yuntao Du, Jiacheng Li, Yuetian Chen, Kaiyuan Zhang, Zhizhen Yuan, Hanshen Xiao, Bruno Ribeiro, and Ninghui Li. Cascading and Proxy Membership Inference Attacks. In *33th Annual Network and Distributed System Security Symposium (NDSS)*, 2026.

[24] Yuntao Du and Ninghui Li. Systematic assessment of tabular data synthesis algorithms. *arXiv preprint arXiv:2402.06806*, 2024.

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

[26] André V Duarte, Xuandong Zhao, Arlindo L Oliveira, and Lei Li. De-cop: detecting copyrighted content in language models training data. In *Proceedings of the 41st International Conference on Machine Learning*, pages 11940–11956, 2024.

[27] Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.

[28] Anita Faul and Michael Tipping. Analysis of sparse bayesian learning. *Advances in neural information processing systems*, 14, 2001.

[29] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An {End-to-End} case study of personalized warfarin dosing. In *23rd USENIX security symposium (USENIX Security 14)*, pages 17–32, 2014.

[30] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX security symposium (USENIX Security 22)*, pages 1397–1414, 2022.

[31] Xavier Gabaix. Zipf's law and the growth of cities. *American Economic Review*, 89(2):129–132, 1999.

[32] Matthias Gallé. Investigating the effectiveness of bpe: The power of shorter sequences. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 1375–1381, 2019.

[33] Google. Counting gemini text tokens locally with the vertex ai sdk, July 2024. Tokenizer for Use with Google's Models.

[34] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36:16693–16715, 2023.

[35] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[36] Hacker News. Weird gpt-4 behavior for "davidjl". news.ycombinator.com/item?id=36242914, 2023.

[37] Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. Data mixture inference attack: Bpe tokenizers reveal training data compositions. *Advances in Neural Information Processing Systems*, 37:8956–8983, 2024.

[38] Jonathan Hayase, Alisa Liu, Yejin Choi, Sewoong Oh, and Noah A Smith. Data mixture inference attack: Bpe tokenizers reveal training data compositions. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[39] Jamie Hayes, Ilia Shumailov, Christopher A Choquette-Choo, Matthew Jagielski, George Kaissis, Katherine Lee, Milad Nasr, Sahra Ghalebikesabi, Niloofar Mireshghallah, Meenatchi Sundaram Mutu Selva Annamalai, et al. Strong membership inference attacks on massive datasets and (moderately) large language models. *arXiv preprint arXiv:2505.18773*, 2025.

[40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 770–778, 2016.

[41] Yu He, Boheng Li, Liu Liu, Zhongjie Ba, Wei Dong, Yiming Li, Zhan Qin, Kui Ren, and Chun Chen. Towards label-only membership inference attack against pre-trained large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, 2025.

[42] Yuke He, Zheng Li, Yang Zhang, Zhan Qin, Kui Ren, and Chun Chen. Membership inference attacks against vision-language models. In *34th USENIX Security Symposium (USENIX Security 25)*, 2025.

[43] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B Brown, Prafulla Dhariwal, Scott Gray, et al. Scaling laws for autoregressive generative modeling. *arXiv preprint arXiv:2010.14701*, 2020.

[44] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

[45] Hongwei Huang, Weiqi Luo, Guoqiang Zeng, Jian Weng, Yue Zhang, and Anjia Yang. Damia: Leveraging domain adaptation as a defense against membership inference attacks. *IEEE Transactions on Dependable and Secure Computing*, 19(5):3183–3199, 2021.

[46] Hongzhi Huang, Defa Zhu, Banggu Wu, Yutao Zeng, Ya Wang, Qiyang Min, et al. Over-tokenized transformer: Vocabulary is generally worth scaling. In *Forty-second International Conference on Machine Learning*, 2025.

[47] Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. *arXiv preprint arXiv:2508.05337*, 2025.

[48] Hugging Face. Tokenizer. https://huggingface.co/docs/transformers/main_classes/tokenizer, 2025.

[49] Hugging Face Datasets. Codeparrot github code dataset. https://huggingface.co/datasets/codeparrot/github-code, 2025.

[50] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *ISOC Network and Distributed System Security Symposium (NDSS)*, 2021.

[51] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *CCS*, pages 259–274, 2019.

[52] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

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

[54] Hao Li, Zheng Li, Siyuan Wu, Chengrui Hu, Yutong Ye, Min Zhang, Dengguo Feng, and Yang Zhang. Seqmia: sequential-metric based membership inference attack. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3496–3510, 2024.

[55] Hao Li, Zheng Li, Siyuan Wu, Yutong Ye, Min Zhang, Dengguo Feng, and Yang Zhang. Enhanced label-only membership inference attacks with fewer queries. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security '25)*. USENIX Association, 2025.

[56] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership inference attacks and defenses in classification models. In *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, pages 5–16, 2021.

[57] Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. *arXiv preprint arXiv:2110.05679*, 2021.

[58] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *CCS*, pages 880–895, 2021.

[59] Alisa Liu, Jonathan Hayase, Valentin Hofmann, Sewoong Oh, Noah A Smith, and Yejin Choi. SuperBPE: Space travel for language models. In *Second Conference on Language Modeling*, 2025.

[60] Han Liu, Yuhao Wu, Zhiyuan Yu, and Ning Zhang. Please tell me more: Privacy impact of explainability through the lens of membership inference attack. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 4791–4809. IEEE, 2024.

[61] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[62] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[63] Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. Llm dataset inference: Did you train on my dataset? *Advances in Neural Information Processing Systems*, 37:124069–124092, 2024.

[64] Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. *arXiv preprint arXiv:2104.10706*, 2021.

[65] Matthew Watkins. Tokens used by gpt-4 probably come from the reddit users. https://x.com/SoC_trilogy/status/1666714127438434304, 2023.

[66] Prasanna Mayilvahanan, Thaddus Wiedemer, Sayak Mallick, Matthias Bethge, and Wieland Brendel. LLMs on the line: Data determines loss-to-loss scaling laws. In *Forty-second International Conference on Machine Learning*, 2025.

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

[68] Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. Sok: Membership inference attacks on llms are rushing nowhere (and how to fix it). In *IEEE Conference on Secure and Trustworthy Machine Learning ((SaTML, 2025)*. IEEE, 2025.

[69] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

[70] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[71] Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. *Advances in Neural Information Processing Systems*, 36:50358–50376, 2023.

[72] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 739–753. IEEE, 2019.

[73] New York Times. Reddit sues anthropic over its data scraping to train large language models. https://www.nytimes.com/2025/06/04/technology/reddit-anthropic-lawsuit-data.html, 2025.

[74] OpenAI. System card of chatgpt-o1. https://cdn.openai.com/o1-system-card-20241205.pdf, 2024.

[75] OpenAI. tiktoken: Tokenizer for openai models. https://github.com/openai/tiktoken, 2025.

[76] Yan Pang and Tianhao Wang. Black-box membership inference attacks against fine-tuned diffusion models. *arXiv preprint arXiv:2312.08207*, 2023.

[77] Yan Pang, Tianhao Wang, Xuhui Kang, Mengdi Huai, and Yang Zhang. White-box membership inference attacks against diffusion models. *arXiv preprint arXiv:2308.06405*, 2023.

[78] Steven T Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic bulletin & review*, 21(5):1112–1130, 2014.

[79] Haritz Puerto, Martin Gubri, Sangdoo Yun, and Seong Joon Oh. Scaling up membership inference: When and how attacks succeed on large language models. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 4165–4182, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics.

[80] Shahzad Qaiser and Ramsha Ali. Text mining: use of tf-idf to examine the relevance of words to documents. *International journal of computer applications*, 181(1):25–29, 2018.

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

[82] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA, 2003.

[83] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024.

[84] Jie Ren, Kangrui Chen, Chen Chen, Vikash Sehwag, Yue Xing, Jiliang Tang, and Lingjuan Lyu. Self-comparison for dataset-level membership inference in large (vision-) language model. In *Proceedings of the ACM on Web Conference 2025*, pages 910–920, 2025.

[85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[86] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*, pages 5558–5567. PMLR, 2019.

[87] Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024.

[88] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

[89] Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

[90] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations*, 2023.

[91] Philip Sedgwick. Spearman's rank correlation coefficient. *Bmj*, 349, 2014.

[92] Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *ICLR*, 2024.

[93] Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. Byte pair encoding: A text compression scheme that accelerates pattern matching. , 1999.

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

[95] Kevin Slagle. Spacebyte: Towards deleting tokenization from large language modeling. *Advances in Neural Information Processing Systems*, 37:124925–124950, 2024.

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

[97] Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. *Advances in Neural Information Processing Systems*, 37:114147–114179, 2024.

[98] Meng Tong, Kejiang Chen, Xiaojian Yuan, Jiayang Liu, Weiming Zhang, Nenghai Yu, and Jie Zhang. On the vulnerability of text sanitization. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5150–5164, 2025.

[99] Meng Tong, Kejiang Chen, Jie Zhang, Yuang Qi, Weiming Zhang, Nenghai Yu, Tianwei Zhang, and Zhikun Zhang. Inferdpt: Privacy-preserving inference for black-box large language models. *IEEE Transactions on Dependable and Secure Computing*, 2025.

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

[101] Geoffrey I Webb. Naïve bayes. In *Encyclopedia of machine learning and data mining*, pages 895–896. Springer, 2017.

[102] Chen Xiong, Zihao Wang, Rui Zhu, Tsung-Yi Ho, Pin-Yu Chen, Jingwei Xiong, Haixu Tang, and Lucila Ohno-Machado. Hey, that's my data! label-only dataset inference in large language models. *arXiv preprint arXiv:2506.06057*, 2025.

[103] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. Enhanced membership inference attacks against machine learning models. In *CCS*, pages 3093–3106, 2022.

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

[105] Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. *arXiv preprint arXiv:2110.06500*, 2021.

[106] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *Proceedings of the 41st International Conference on Machine Learning*, pages 58244–58282, 2024.

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

[108] Kaiyuan Zhang, Siyuan Cheng, Hanxi Guo, Yuetian Chen, Zian Su, Shengwei An, Yuntao Du, Charles Fleming, Ashish Kundu, Xiangyu Zhang, and Ninghui Li. SOFT: Selective Data Obfuscation for Protecting LLM Fine-tuning against Membership Inference Attacks. In *Proceedings of the 34th USENIX Security Symposium (USENIX Security 2025)*, 2025.

[109] Vilém Zouhar, Clara Meister, Juan Gastaldi, Li Du, Tim Vieira, Mrinmaya Sachan, and Ryan Cotterell. A formal perspective on byte-pair encoding. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 598–614, 2023.

A Proof of Theorem

Theorem 4.2 (RTF-SI under the Power Law). *Under the power-law distribution* [17]*, the frequency* $\Pr(t_i \mid \mathcal{V}_{target})$ *of a token* $t_i \in \mathcal{V}_{target}$ *is proportional to* $1/i^{\alpha}$*:*

$$\Pr(t_i \mid \mathcal{V}_{target}) \propto \frac{1}{i^{\alpha}}, \tag{17}$$

where $i > x_{\min}$*, and* $\alpha \in \mathbb{R}_{>0}$*,* $x_{\min} \in \mathbb{Z}_{>0}$ *are constants defined by the power law. Then, RTF-SI can be approximated by its lower bound:*

$$\text{RTF-SI}(D, t_i, \mathcal{V}_{target}) \geq \frac{n_D(t_i)}{\sum_{D' \in \mathbb{D}} n_{D'}(t_i)} \cdot \log\Big(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{target}|} i^{\alpha}/j^{\alpha}\Big). \tag{18}$$

Proof. Given that $\Pr(t_i \mid \mathcal{V}_{\text{target}}) \propto \frac{1}{i^{\alpha}}$, let us assume

$$\Pr(t_i \mid \mathcal{V}_{\text{target}}) = C/i^{\alpha}, \tag{19}$$

where $C \in \mathbb{R}_{>0}$ is a constant. Since the sum of the frequencies $\Pr(t_i \mid \mathcal{V}_{\text{target}})$ for all $t_i \in \mathcal{V}_{\text{target}}$ is at most 1, it follows that

$$1 \geq \sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} \Pr(t_j \mid \mathcal{V}_{\text{target}}) = \sum_{j=1+x_{\min}}^{|\mathcal{V}_{\text{target}}|} C/j^{\alpha}. \tag{20}$$

Therefore, the upper bound for the constant C is

$$C \leq \frac{1}{\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} \frac{1}{j^{\alpha}}}. \tag{21}$$

Using Equations 19 and 21, we derive the upper bound for the frequency $\Pr(t_i \mid \mathcal{V}_{\text{target}})$:

$$\Pr(t_i \mid \mathcal{V}_{\text{target}}) \leq \frac{1}{\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} \frac{1}{j^{\alpha}}} \cdot \frac{1}{i^{\alpha}}. \tag{22}$$

Taking the negative logarithm, the self-information of t_i is bounded from below:

$$\text{SI}(t_i, \mathcal{V}_{\text{target}}) = -\log \Pr(t_i \mid \mathcal{V}_{\text{target}}) \geq \log\Big(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} i^{\alpha}/j^{\alpha}\Big). \tag{23}$$

Therefore, the RTF-SI satisfies:

$$\text{RTF-SI}(D, t_i, \mathcal{V}_{\text{target}}) \geq \frac{n_D(t_i)}{\sum_{D' \in \mathbb{D}} n_{D'}(t_i)} \cdot \log\Big(\sum_{j=x_{\min}+1}^{|\mathcal{V}_{\text{target}}|} i^{\alpha}/j^{\alpha}\Big). \tag{24}$$

This theorem allows an adversary to approximate the RTF-SI using its lower bound under a power-law distribution. □

Table 9: Impact of the target dataset size on defense mechanism ($n_{\min} = 32$). $|\mathcal{V}_{\text{target}}| \leq 80{,}000$ indicates the vocabulary size prior to applying defense is 80,000. TPR refers to TPR @ 1.0% FPR. The results show our MIAs remain effective on large datasets.

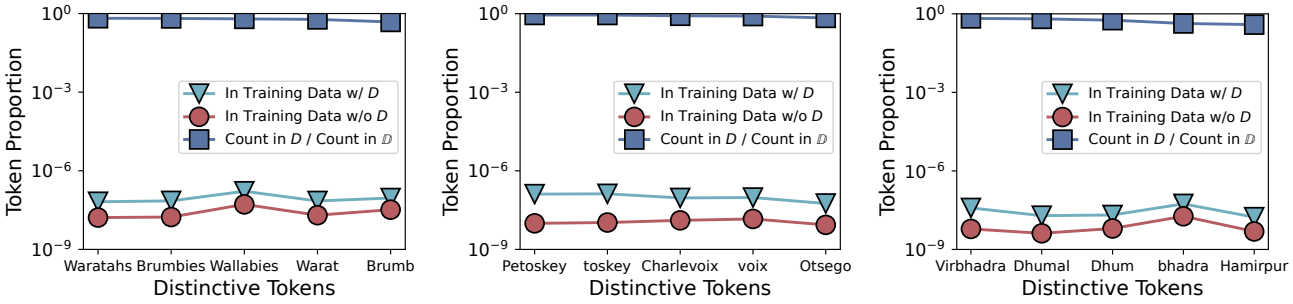
Attack Approach	#Dataset Size	$\lvert\mathcal{V}_{\textbf{target}}\rvert = 80{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 110{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 140{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 170{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	$\lvert D\rvert \in [0,400)$	0.646	0.632	21.25%	0.677	0.642	21.25%	0.694	0.655	23.30%	0.706	0.663	24.44%	0.717	0.669	25.14%
	$\lvert D\rvert \in [400,800)$	0.703	0.667	21.25%	0.761	0.711	22.85%	0.772	0.717	33.73%	0.793	0.730	34.94%	0.810	0.742	40.06%
	$\lvert D\rvert \in [800,1200)$	0.712	0.670	25.30%	0.768	0.729	30.12%	0.798	0.761	38.75%	0.823	0.775	43.75%	0.840	0.800	46.25%
Frequency Estimation	$\lvert D\rvert \in [0,400)$	0.586	0.591	16.36%	0.617	0.616	20.11%	0.649	0.635	20.62%	0.682	0.657	21.01%	0.720	0.683	26.99%
	$\lvert D\rvert \in [400,800)$	0.611	0.615	17.78%	0.675	0.650	25.00%	0.720	0.683	29.52%	0.739	0.702	30.42%	0.763	0.721	33.43%
	$\lvert D\rvert \in [800,1200)$	0.723	0.714	25.00%	0.742	0.716	35.00%	0.758	0.729	35.00%	0.785	0.759	47.50%	0.820	0.805	52.50%

Table 10: Impact of the target dataset size on defense mechanism ($n_{\min} = 48$). $|\mathcal{V}_{\text{target}}| \leq 80{,}000$ indicates the vocabulary size prior to applying defense is 80,000. TPR refers to TPR @ 1.0% FPR. The results show MIAs remain effective on large datasets.

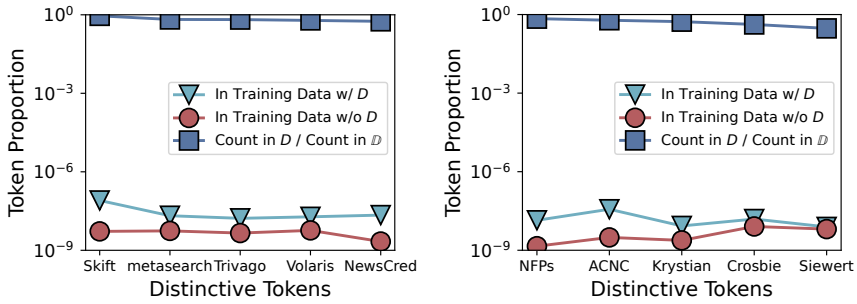
Attack Approach	#Dataset Size	$\lvert\mathcal{V}_{\textbf{target}}\rvert = 80{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 110{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 140{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 170{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert = 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	$\lvert D\rvert \in [0,400)$	0.646	0.632	21.25%	0.677	0.642	21.25%	0.687	0.645	22.85%	0.694	0.655	23.30%	0.710	0.667	27.12%
	$\lvert D\rvert \in [400,800)$	0.704	0.667	21.58%	0.761	0.711	22.22%	0.772	0.717	30.00%	0.775	0.721	34.64%	0.790	0.730	34.94%
	$\lvert D\rvert \in [800,1200)$	0.712	0.670	25.30%	0.768	0.729	30.12%	0.774	0.738	30.72%	0.798	0.761	38.75%	0.823	0.767	45.00%
Frequency Estimation	$\lvert D\rvert \in [0,400)$	0.586	0.591	16.36%	0.617	0.616	18.20%	0.649	0.634	19.92%	0.680	0.652	21.01%	0.680	0.654	22.79%
	$\lvert D\rvert \in [400,800)$	0.588	0.597	16.93%	0.619	0.623	20.11%	0.650	0.635	20.62%	0.682	0.657	21.64%	0.720	0.683	26.99%
	$\lvert D\rvert \in [800,1200)$	0.616	0.611	20.48%	0.681	0.659	28.01%	0.719	0.692	28.92%	0.731	0.693	31.93%	0.736	0.699	32.53%

Table 11: Impact of the target dataset size on defense mechanism ($n_{\min} = 64$). $|\mathcal{V}_{\text{target}}| \leq 80{,}000$ indicates the vocabulary size prior to applying defense is 80,000. TPR refers to TPR @ 1.0% FPR. The results show MIAs remain effective on large datasets.

Attack Approach	#Dataset Size	$\lvert\mathcal{V}_{\textbf{target}}\rvert \leq 80{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert \leq 110{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert \leq 140{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert \leq 170{,}000$			$\lvert\mathcal{V}_{\textbf{target}}\rvert \leq 200{,}000$		
		AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR	AUC	BA	TPR
Vocabulary Overlap	$\lvert D\rvert \in [0,400)$	0.646	0.632	19.16%	0.662	0.636	21.25%	0.677	0.642	21.25%	0.683	0.647	23.30%	0.694	0.655	24.06%
	$\lvert D\rvert \in [400,800)$	0.703	0.666	21.25%	0.731	0.675	21.58%	0.751	0.699	22.85%	0.761	0.711	32.50%	0.772	0.717	37.95%
	$\lvert D\rvert \in [800,1200)$	0.712	0.670	25.30%	0.743	0.702	27.11%	0.768	0.729	30.12%	0.795	0.746	33.73%	0.797	0.761	38.75%
Frequency Estimation	$\lvert D\rvert \in [0,400)$	0.591	0.598	17.50%	0.620	0.619	18.20%	0.651	0.636	20.11%	0.654	0.637	21.07%	0.656	0.639	21.39%
	$\lvert D\rvert \in [400,800)$	0.619	0.615	22.29%	0.672	0.653	28.31%	0.717	0.682	30.42%	0.718	0.683	31.93%	0.723	0.686	32.83%
	$\lvert D\rvert \in [800,1200)$	0.734	0.717	26.25%	0.739	0.731	33.75%	0.744	0.736	33.75%	0.745	0.742	35.00%	0.748	0.746	38.75%

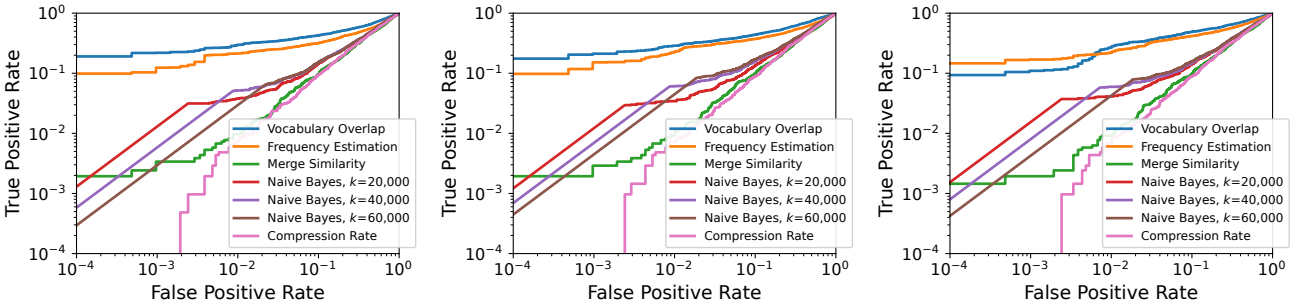


(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000 (c) Vocabulary Size: 140,000

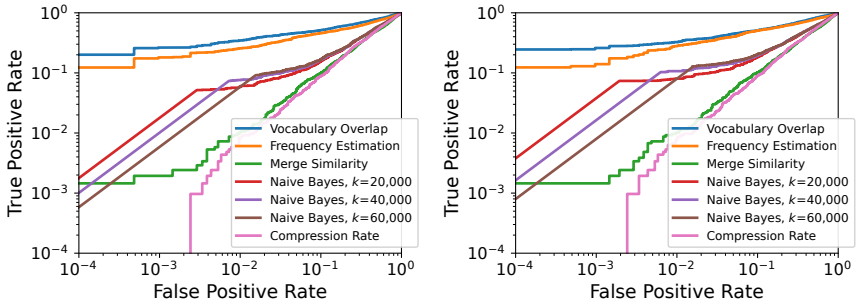


(d) Vocabulary Size: 170,000 (e) Vocabulary Size: 200,000

Figure 12: Distinctive tokens in MIA via Vocabulary Overlap.

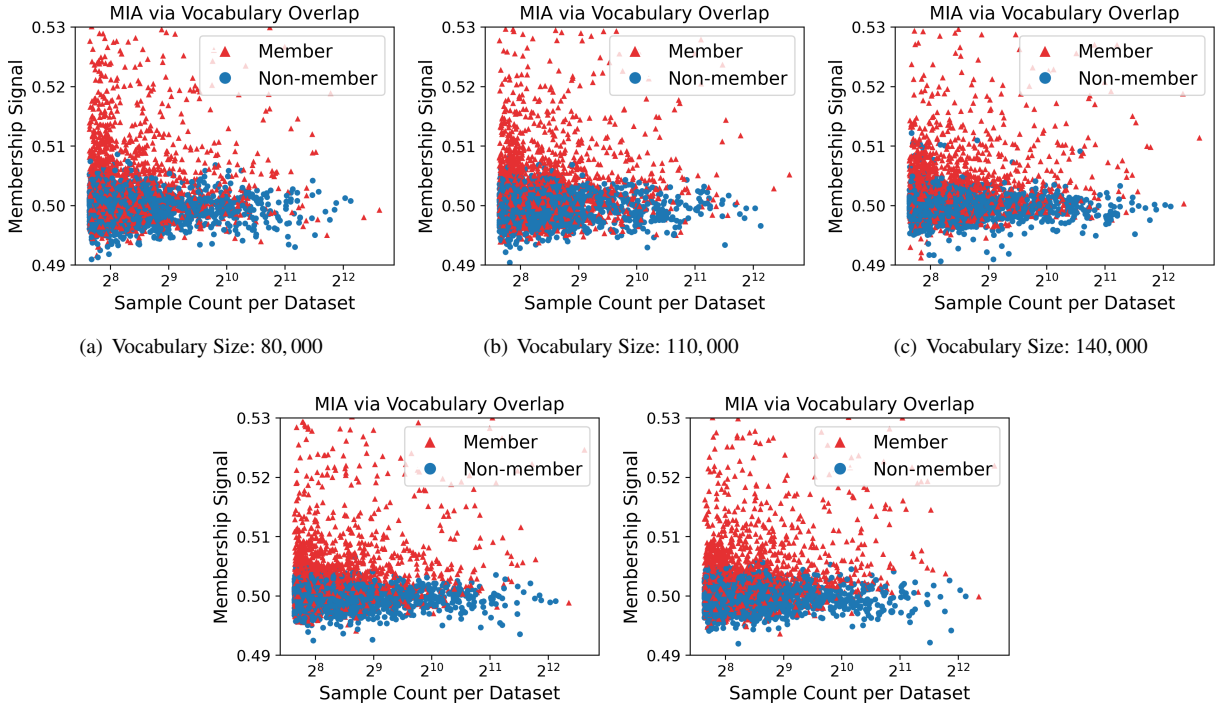


(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000 (c) Vocabulary Size: 140,000



(d) Vocabulary Size: 170,000 (e) Vocabulary Size: 200,000

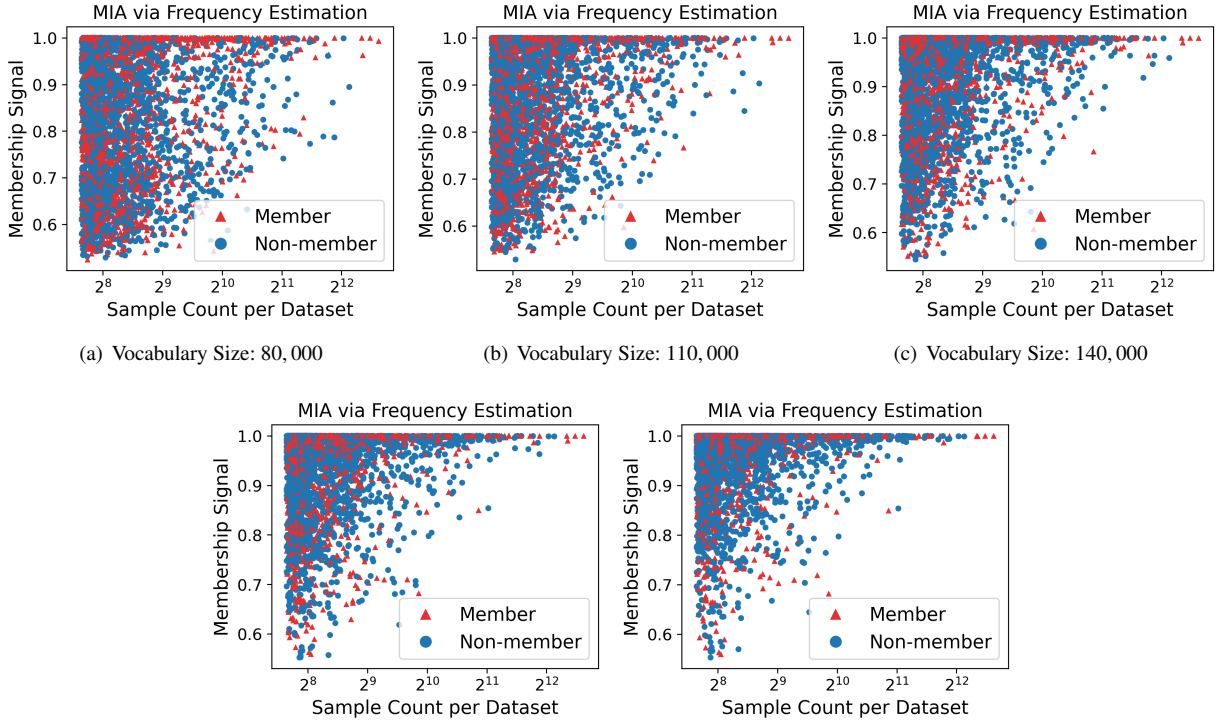
Figure 13: ROC curves for MIAs using different methods.



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000 (c) Vocabulary Size: 140,000

(d) Vocabulary Size: 170,000 (e) Vocabulary Size: 200,000

Figure 14: Dataset distribution based on MIA via Vocabulary Overlap.



(a) Vocabulary Size: 80,000 (b) Vocabulary Size: 110,000 (c) Vocabulary Size: 140,000

(d) Vocabulary Size: 170,000 (e) Vocabulary Size: 200,000

Figure 15: Dataset distribution based on MIA via Frequency Estimation.