R.R.: Unveiling LLM Training Privacy through Recollection and Ranking

Wenlong Meng[1], Zhenyuan Guo[1], Lenan Wu[1], Chen Gong[2], Wenyan Liu[3], Weixian Li[3], Chengkun Wei[1†], Wenzhi Chen[1]
[1]Zhejiang University, [2]University of Virginia, [3]Ant Group
{mengwl, zhenyuanguo, weichengkun, chenwz}@zju.edu.cn

Abstract

Large Language Models (LLMs) pose significant privacy risks, potentially leaking training data due to implicit memorization. Existing privacy attacks primarily focus on membership inference attacks (MIAs) or data extraction attacks, but reconstructing specific personally identifiable information (PII) in LLMs' training data remains challenging. In this paper, we propose R.R. (Recollect and Rank), a novel two-step privacy stealing attack that enables attackers to reconstruct PII entities from scrubbed training data where the PII entities have been masked. In the first stage, we introduce a prompt paradigm named recollection, which instructs the LLM to repeat a masked text but fill in masks. Then we can use PII identifiers to extract recollected PII candidates. In the second stage, we design a new criterion to score each PII candidate and rank them. Motivated by membership inference, we leverage the reference model as a calibration to our criterion. Experiments across three popular PII datasets demonstrate that the R.R. achieves better PII identification performance than baselines. These results highlight the vulnerability of LLMs to PII leakage even when training data has been scrubbed. We release our code and datasets at GitHub.[1]

1 Introduction

Large Language Models (LLMs) have been widely adopted in various applications, like automated content generation (Girotra et al., 2023; Roziere et al., 2023) to personalized virtual assistants (Nam et al., 2024; Shao et al., 2024). This success is driven by massive training datasets, which raise significant concerns about privacy leakage. Previous research has shown that LLMs can memorize portions of their training data, even before overfitting (Carlini et al., 2019). These training datasets of LLM may contain sensitive personally identifiable information (PII), such as names, phone numbers, and email addresses. Additionally, LLMs are usually fine-tuned with custom datasets to meet application needs, which may introduce further privacy risks.

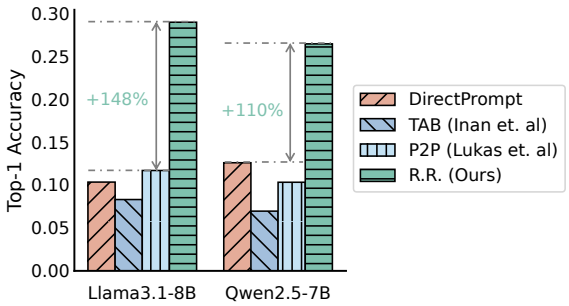


Figure 1: Comparison of PII prediction accuracies on NeurIPS LLM-PC dataset. R.R. achieves an improvement of over 100% compared to previous SOTAs.

LLM memorization is an inherent and inevitable part of training. Thus, the security of LLM privacy is a crucial and widespread topic of public concern. This paper focuses on attacks that steal the privacy of training datasets. One widely studied privacy attack is *membership inference attack* (MIA) (Mireshghallah et al., 2022a; Fu et al., 2024), which aims to determine whether a specific data record was included in the training set. However, MIAs targeting LLMs usually require the attacker to obtain complete text samples, which is often impractical in real-world scenarios (Mattern et al., 2023; Duan et al., 2024). Another line of attack is *data extraction* (Carlini et al., 2021; Mireshghallah et al., 2022b), where an adversary attempts to reconstruct training data as much as possible. However, the impact of data extraction is limited, as the extracted data may not contain the target information an attacker seeks.

Recently, Lukas et al., 2023 proposed *PII reconstruction attack*, which refers to the task where an attacker attempts to recover the masked PII entities

†Corresponding author
[1]https://github.com/meng-wenlong/RR

Findings of the Association for Computational Linguistics: ACL 2025, pages 17383–17397

in a piece of text. For example, in the masked sentence "The shipment for [MASK] was delivered to 1234 Elm Street," if the attacker successfully reconstructs [MASK], they can identify the individual residing at that address. This practical attack has attracted great attention in the community. Scrubbed datasets publicly released by LLM developers may serve as potential targets for such attacks. Nevertheless, existing PII reconstruction methods based on prefix continuation (Inan et al., 2021) or perplexity scoring (Lukas et al., 2023), lack contextual dependencies and insights from the original LLM, yielding a reconstruction accuracy below 10%.

Our Contributions. In this paper, we propose R.R. (Recollect and Rank), a novel and effective PII reconstruction method that assumes black-box access to the victim LLM. R.R. consists of two stages: (1) recollection, generating PII candidates, and (2) ranking, identifying the candidate most likely to be PII from candidates.

In the recollection stage, we provide the entire masked text as input to the victim LLM and prompt it to reproduce the text without masks. Unlike traditional prefix continuation methods, which rely solely on the preceding text, our approach leverages both preceding and succeeding contexts. Once the model finished generating the recalled text, R.R. uses a PII identifier to extract internal PII entities. After repeating recollection several times, we get a PII candidates pool. In the ranking stage, extracted PII candidates are inserted into the mask placeholder. We use the cross-entropy loss as the criterion for ranking, and the candidate with the lowest loss is selected as the reconstructed result. Inspired by reference-based MIAs (Ye et al., 2022; Zeng et al., 2024), we use the loss from the pre-trained model, on which the victim model is fine-tuned, as a reference to calibrate the ranking criterion. Interestingly, we observe that reference calibration does not always improve ranking accuracy. To address it, we propose a biased ranking criterion that blends uncalibrated and calibrated criteria. We theoretically prove that our proposed criterion preserves the strengths of both.

Experimental results of four LLMs show that R.R. achieves an average top-1 accuracy of 25.73% on the NeurIPS LLM-PC dataset (Li et al., 2024), marking a more than 100% improvement over previous SOTAs, as presented in Figure 1. In summary, our contributions are as follows:

- We propose a novel PII reconstruction attack named R.R. which leverages recollection to generate PII candidates and employs a biased cross-entropy loss for ranking. This paper shows that *scrubbed training data is not safe*, calling for increased attention to this threat.
- We take the first step in using reference calibration for PII ranking and propose a new criterion that combines calibrated and uncalibrated criteria.
- Extensive experiments across three PII datasets and four popular LLMs show that R.R. achieves an average 122% improvement in top-1 accuracy compared to baselines.

2 Preliminaries

2.1 LLM Training and PII

LLM Training. We train LLMs using next-token prediction, where the model learns to maximize the probability of correctly predicting the next token given a sequence of preceding tokens. The objective is formulated to minimize cross-entropy loss. Given a sequence of tokens $X = (x_1, x_2, \ldots, x_T)$, the cross-entropy loss is defined as:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T} \log P_\theta\left(x_t \mid x_{1:t-1}\right), \qquad (1)$$

where $P_\theta\left(x_t \mid x_{1:t-1}\right)$ denotes the probability assigned by LLM with parameter θ.

Personally Identifiable Information (PII). PII refers to any data that can be used to uniquely identify an individual, such as names, addresses, phone numbers, or social security numbers. Since LLMs are often trained on vast real-world texts, they inevitably ingest and memorize a significant amount of PII. To detect PII in natural language text, Named Entity Recognition (NER) models are commonly employed. NER is a token classification task, where each token in a sequence is assigned a label corresponding to a specific entity type. Several industry-grade PII detection services utilize NER-based models to identify and redact sensitive information (Mendels et al., 2018; Amazon Web Services, Inc., 2025).

2.2 PII Leakage Attacks

Attacks Taxonomy. According to Lukas et al., 2023 proposed, PII leakage attacks can be categorized into three types based on the adversary's ca-

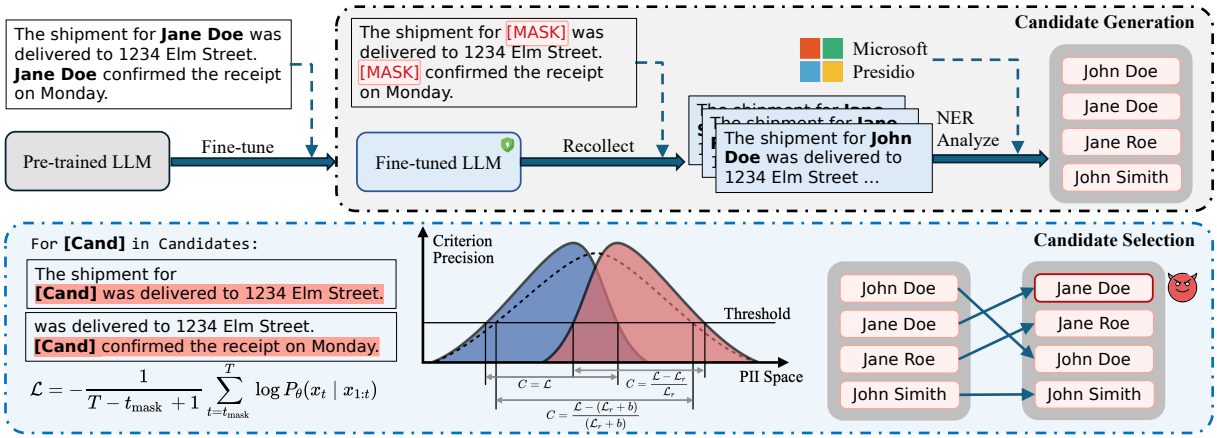


Figure 2: Overview of R.R.. R.R. has two steps: candidate generation and selection. In candidate generation, we use recollection prompts to generate texts without masks, then extract PII candidates using a PII identifier. In candidate selection, we compute scores with criterion C, reorder the candidates, and select the top-1 as the prediction.

pabilities: PII extraction, reconstruction, and inference attacks. Extraction attackers have no auxiliary information, while reconstruction attackers have knowledge of masked training texts (e.g., scrubbed datasets) and aim to recover the missing PII. Inference attacks go a step further, where the attacker possesses both masked training texts and PII candidates. The goal of inference is to associate a piece of PII with a given context.

Threat Model. This paper focuses on PII reconstruction attacks because obtaining masked training data is practical in real-world scenarios. We assume an adversary with black-box API access to the victim LLM, allowing them to request a continuation or query next-token probabilities to compute cross-entropy loss and PPL. Since API calls entail monetary costs, the adversary should minimize request or query times. Experimental results (Section 4.4) prove that our R.R. requires fewer queries to achieve the same PII recall in candidate generation. In candidate selection, we neglect the content preceding PII when computing loss, reducing the query count. Additionally, we assume the adversary knows the reference model, i.e., the base model on which the victim LLM is fine-tuned. It is a reasonable assumption since modern LLM development is usually based on open-source LLMs. And according to popular open-source LLM licenses, such as OpenRAIL-M (BigCode, 2022) and Apache-2.0 (Apache, 2004), the developer needs to manifest the source when employing a modification of the LLM for commercial use. Even if the attacker uses an incorrect reference model, we show that it causes a small degradation in reconstruction accuracy (Appendix C).

3 R.R. Methodology

The overall pipeline of R.R. is illustrated in Figure 2. Our R.R. reconstructs masked PII in two steps: candidate generation and candidate selection. During candidate generation, R.R. iteratively induces the victim LLM to generate possible PII candidates, which are then ranked in candidate selection. The top-1 candidate is selected as the reconstructed result.

3.1 Problem Statement

Our attack involves two participants: an LLM developer and an attacker. The LLM developer fine-tunes a publicly available pre-trained LLM, denoted as π_θ, on a private dataset $\mathcal{D}_p = \{X_i \mid i = 1, 2, \ldots\}$, obtaining a fine-tuned LLM $\pi_{\theta'}$. The developer then deploys the custom LLM on the cloud and provides a query API. The attacker possesses several masked texts, denoted as $\mathcal{D}_a = \left\{\widetilde{X}_i \mid i = 1, 2, \ldots\right\}$, where each masked text takes the form:

$$\widetilde{X}_i = \left(x_{i,1}, x_{i,2}, \ldots, \texttt{[MASK]}, \ldots, x_T\right). \quad (2)$$

For each $\widetilde{X}_i \in \mathcal{D}_a$, there exists a sequence $M = (m_i)_{i=1}^k$ such that

$$X_i = \widetilde{X}_i \circ M \in \mathcal{D}_p, \quad (3)$$

where $\circ$ represents the operation of replacing [MASK] in $\widetilde{X}_i$ with M. The attacker's goal is to recover M. For brevity, we use $\mathcal{L}(M)$ to represent $\mathcal{L}(\widetilde{X} \circ M)$ in the rest of this paper.

In practice, $\widetilde{X}_i$ may contain multiple [MASK]s, which indicate entity types related to PII, such as [DATE_TIME]. The entity types considered in this paper are listed in Table 6 of Section B.1.

3.2 Candidate Generation

This step uses designed prompts to induce the victim LLM to output memorized PII. We construct prompts using a recollection approach, where the masked text $\widetilde{X}_i$ is directly fed into the LLM, prompting it to repeat the given text while omitting the masked tokens. We then use Microsoft Presidio to extract PII of specific types from the recollected text. Our motivation is to leverage all available information as much as possible. Through the recollection paradigm, we could leverage all contexts the attacker has.

The most widely used previous candidate generation method is TAB (Inan et al., 2021), which generates completions using the true prefix. TAB behaves like the Tab autocompletion method which automatically fills in table entries, mimicking the behavior of the Tab key. However, its major limitation is that it cannot leverage information from the text following the `[MASK]`. For instance, in the sentence: "The shipment for `[MASK]` was delivered to 1234 Elm Street." TAB fails to utilize the address "1234 Elm Street" to infer the missing PII. Additionally, if other masks appear in the preceding text, TAB requires truncation to prevent the LLM from generating mask tokens. For example, if "shipment" is also masked in the above sentence, TAB can only use "for" as the continuation prefix. Lukas et al., 2023 addressed this issue by filling in previous masks using a *Masked Language Model* (MLM). However, this approach assumes that the token length of the masked PII is known in advance, which is often impractical in real-world scenarios. Conversely, recollection maximizes the use of both preceding and following context while aligning with the LLMs' training paradigm.

3.3 Candidate Selection

Assume that, during the candidate generation step, R.R. produces n PII candidates $\mathcal{M} = \{M_i\}_{i=1}^{n}$ for a given mask. In this step, we aim to assign a score, which indicates fitness to the mask, to each $M_i \in \mathcal{M}$. Then, we rank them and select the top-1 candidate as the final prediction. In short, our ranking criterion is:

$$C\left(M_i\right) = \frac{\mathcal{L}^p\left(M_i\right) - \left(\mathcal{L}_r^p\left(M_i\right) + b\right)}{\mathcal{L}_r^p\left(M_i\right) + b}, \quad (4)$$

where $\mathcal{L}^p$ and $\mathcal{L}_r^p$ represent the *partial cross-entropy loss* in the victim model and the reference model, respectively. b is a constant positive number specific to each LLM.

Partial Cross-Entropy Loss. We use cross-entropy loss as the base criterion because it allows us to easily sum multiple criteria in cases where a PII mask appears multiple times. As shown in Figure 2, when a PII mask appears multiple times in X_i, our proposed R.R. employs a greedy strategy that divides X_i into segments at each PII mask location, computes the score for each segment, and then sums the scores. In this way, we reduce the computational complexity of loss calculation from $\mathcal{O}\left({n_c}^{n_t}\right)$ to $\mathcal{O}\left(n_c \cdot n_t\right)$, where n_c and n_t are the number of candidates for reach PII type and the number of PII types, respectively. Since most LLMs use unidirectional attention, the prediction probabilities for tokens preceding a PII mask will not change due to variations in M_i. Therefore, when calculating the cross-entropy loss, we only consider the text from M_i and onwards,

$$\mathcal{L}^p = -\frac{\sum_{t=t_{\text{mask}}}^{T} \log P_\theta\left(x_t \mid x_{1:t}\right)}{T - t_{mask} + 1}, \quad (5)$$

where t_{mask} is the position of the mask token.

The effect of the preceding text on cross-entropy loss can be disregarded when a PII mask appears only once. However, when the PII mask appears multiple times, the varying lengths of the clauses can cause the normalization factor $\frac{1}{T}$ in Equation 1 to diminish the contribution of longer clauses. Since $\mathcal{L}^p$ more accurately reflects the impact of inserting M_i in most cases, we use $\mathcal{L}$ to represent the partial cross-entropy loss in the rest of this paper.

Revising Biased Reference Calibration. The pre-trained model has already been trained on some texts. Even if the victim model's fine-tuning private dataset does not include those texts, their loss may still be lower than private texts. Actually, the victim model is trained on two datasets, (1) the pretraining dataset and (2) the fine-tuning private dataset. The overlap between the pre-training and fine-tuning datasets reduces the accuracy of the attack on the private fine-tuning dataset. This challenge motivates us to adjust Equation 5 to avoid confusion between these two datasets and exclude the pretraining dataset's terrible impact.

In MIA, researchers use reference models (i.e., the pre-trained models) to adjust the criteria used for determining whether a text is included in the private training data. Inspired by reference-based MIAs, we propose reference calibration into the PII inference attack. To our knowledge, we are the first to do so. The most commonly used calibration method in MIA is to measure the ratio of the

decrease in $\mathcal{L}$ relative to $\mathcal{L}_r$, that is,

$$C_r = \frac{\mathcal{L} - \mathcal{L}_r}{\mathcal{L}_r}. \tag{6}$$

This approach helps to refine the criterion by accounting for the reference model's influence. However, we empirically find that *reference-based criterion is not always superior to simple loss; rather, they perform comparably in most cases.*

From our investigation, The PII in the training dataset correctly identified using $\mathcal{L}$ for calibration is different from that identified using $\mathcal{C}_r$. For example, when reconstructing the LLM-PC dataset on a fine-tuned Llama3.1-8B model, $\mathcal{L}$ correctly predicts 1,627 PII entities, while C_r correctly predicts 1,689 PII entities. Among these, 132 PII entities are correctly predicted by $\mathcal{L}$ but incorrectly predicted by C_r, and 194 PII entities are correctly predicted by C_r but incorrectly predicted by $\mathcal{L}$. These mismatched PII account for 10% of the total correctly predicted PII. If we can propose a new criterion that combines $\mathcal{L}$ and C_r, we could have a theoretically 10% improvement. Our solution is to add a bias b to $\mathcal{L}_r$, resulting in Equation 4. The core idea is that by adding b to $\mathcal{L}_r$, we control how much influence $\mathcal{C}_r$ has in comparison to $\mathcal{L}$. When a PII entity is ranked as top-1 by both $\mathcal{L}$ and C_r, it will be also ranked as top-1 by Equation 4. Specifically, we introduce the following theorem.

Theorem 3.1. *Let* $\mathcal{M} = \{M_i\}_{i=1}^n$. *For* $j \neq k$, $1 \leq j, k \leq n$, *if* $\mathcal{L}(M_j) < \mathcal{L}(M_k)$ *and* $\frac{\mathcal{L}(M_j) - \mathcal{L}_r(M_j)}{\mathcal{L}_r(M_j)} < \frac{\mathcal{L}(M_k) - \mathcal{L}_r(M_k)}{\mathcal{L}_r(M_k)}$, *then*

$$\frac{\mathcal{L}(M_j) - (\mathcal{L}_r(M_j) + b)}{\mathcal{L}_r(M_j) + b} < \frac{\mathcal{L}(M_k) - (\mathcal{L}_r(M_k) + b)}{\mathcal{L}_r(M_k) + b}.$$

Theorem 3.1 indicates that our proposed criterion C in Equation 4 can maintain the overlapped correct PII entities predicted by $\mathcal{L}$ and C_r. We present the detailed proof of Theorem 3.1 in the appendix. When $b \to 0$, $C \to C_r$. Through Taylor's expansion,

$$C = -1 + \frac{\mathcal{L}}{b} + \mathcal{O}\left(\frac{1}{b^2}\right). \tag{7}$$

Based on the equation above, we know that as b increases, C becomes increasingly dominated by $\mathcal{L}$. Therefore, by adjusting b, we can control whether C is closer to C_r or $\mathcal{L}$. Through extensive experiments, we find that the optimal b depends mainly on the model type (see Section 4.3). An attacker can determine the optimal b by testing on a public PII dataset using the reference model.

4 Evaluation

4.1 Experimental Setup

We briefly introduce our datasets, models, metrics, and baselines in this section. Please refer to Appendix B for more attack settings.

Datasets and LLMs. We choose three PII text datasets to evaluate PII reconstruction attacks: ECHR (Poudyal et al., 2020), ENRON (Klimt and Yang, 2004), and LLM-PC (Li et al., 2024). We employ Microsoft Presidio to obtain the masked texts and the corresponding PII data. Our dataset statistics are shown in Table 4 of Section B.1. As for victim LLMs, we choose four open-source LLMs for the evaluation: Llama3.1-8B, Llama3.2-3B, Qwen2.5-7B, and Phi3.5-Mini.

Metrics. We report the top-1 accuracy for PII prediction. We consider a prediction successful when the true PII is included in the predicted sequence M, regardless of the letter case. For evaluating PII candidate generation, we use PII recall, which measures how many PII entities are successfully generated in the candidates set $\mathcal{M}$.

Baselines. This paper considers three PII reconstruction attack methods as baselines: (1) DirectPrompt, which feeds the masked text directly to the LLM and prompts it to generate the best candidate to fill a specific PII mask; (2) TAB (Inan et al., 2021), which generates completions using the true prefix; (3) P2P (Prefix, Presidio, and Perplexity) (Lukas et al., 2023), which performs multiple rounds of generation using true prefix and ranks the candidates using perplexity.

4.2 Overall Effectiveness

Table 1 presents the PII prediction accuracy of R.R. compared to our baseline methods. The results demonstrate that R.R. consistently outperforms all baselines across different models and datasets. Compared to previous SOTAs, R.R. achieves a 122% improvement in top-1 accuracy, effectively reconstructing twice as many PII entities as before. Among the baselines, DirectPrompt performs the worst, which we attribute to the alignment mechanisms of LLMs. Directly querying for PII entities may sometimes trigger these mechanisms, causing the model to refuse to respond. Although P2P introduces improvements over TAB, its prediction accuracy shows no significant advantage compared to TAB. Additionally, we observe a clear trend: larger models achieve higher PII prediction accu-

Table 1: Top-1 accuracy of R.R. and baselines across different models and datasets.

Stealer	Llama3.1-8B			Llama3.2-3B			Qwen2.5-7B			Phi3.5-Mini		
	ECHR	Enron	LLM-PC	ECHR	Enron	LLM-PC	ECHR	Enron	LLM-PC	ECHR	Enron	LLM-PC
DirectPrompt	6.07	2.55	10.33	3.56	2.11	10.73	3.24	2.09	12.58	2.11	0.62	10.95
TAB	13.51	19.00	8.30	7.20	9.12	6.77	8.61	13.31	6.95	4.09	5.62	4.86
P2P	13.19	19.14	11.68	6.91	8.38	8.65	8.99	13.50	10.31	4.28	5.74	7.41
R.R.	**25.68**	**33.31**	**28.93**	**14.79**	**20.61**	**26.48**	**16.35**	**25.38**	**26.41**	**11.10**	**16.71**	**22.13**

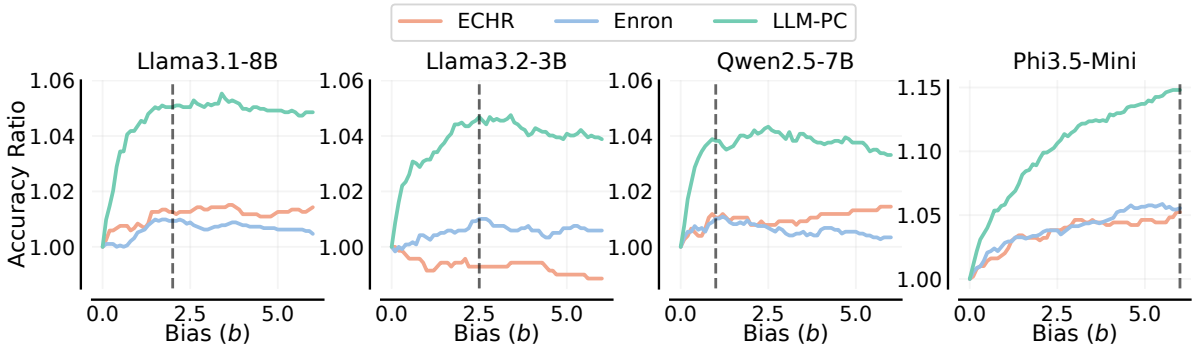


Figure 3: Accuracy ratio versus b. The gray dashed line indicates the optimal value of b chosen during candidate selection. We can see that the accuracy ratio has a similar trend in the same model.

racy. This suggests that larger models have more redundant parameters, enabling them to memorize subtle information during training. Finally, ECHR seems a more difficult dataset than Enron and LLM-PC. This may be due to its shorter text length and the limited occurrence of most PII entities, which appear only once in the text.

4.3 Effectiveness of Biased Criterion

To visually illustrate the impact of bias b on candidate selection, we adjust b and record the accuracy ratio, which is defined as the ratio of the current accuracy to the accuracy when $b = 0$. The observed results are plotted in Figure 3. From the figure, we can see that different datasets tend to exhibit similar accuracy ratio trends on the same model. The only exception in our experiments is ECHR fine-tuned Llama3.2. However, despite this anomaly, R.R. still achieves a decent prediction accuracy on ECHR fine-tuned LLama3.2 at $b = 2.5$. Except for Phi-3.5, the accuracy ratios of the other three models first increase and then decrease, indicating that three exits non-overlapping space between L and C_r. By combining these two criteria, we can achieve a more effective attack. We also observe that LLM-PC dataset is the most sensitive to b, suggesting that the non-overlapping space between L and C_r is larger in this dataset. This is likely due to the higher diversity of texts in LLM-PC. Some PII entities experience a greater decrease in L after training, while others show a more significant reduction in C_r.

4.4 Ablation Studies

In this section, we conduct ablation studies on two steps of R.R.. For candidate generation and candidate selection, we replace Recollect and C in Equation 4 with alternative methods, respectively. We also compare hybrid attacks that combine different candidate generation and candidate selection methods. Due to the space limitation, we defer hybrid attack results to Appendix D.

Ablation of Candidate Generation. We compare Recollect with DirectPrompt and TAB as alternative candidate generation methods. For each approach, we query the victim model 50 times and compare their PII recall. The experimental results are presented in Figure 4. We observe that both Recollect and DirectPrompt showcase an increasing trend in PII recall as the number of query iterations increases. However, TAB reaches a peak with the first few iterations and then stabilizes, indicating that the candidates generated by TAB remain relatively fixed over multiple queries. This also explains why P2P, while doing multiple true-prefix queries, fails to improve prediction accuracy. In contrast, Recollect generally increases PII recall at a faster rate than DirectPrompt and continues to show an upward trend even after 50 iterations, demonstrating the scalability of R.R.. We also con-

Table 2: Ablation study on candidate selection. We compare top-1 accuracy while ranking with different criteria.

Criterion	Llama3.1-8B			Llama3.2-3B			Qwen2.5-7B			Phi3.5-Mini		
	ECHR	Enron	LLM-PC	ECHR	Enron	LLM-PC	ECHR	Enron	LLM-PC	E CHR	Enron	LLM-PC
$\mathcal{L}$ (vanilla)	25.42	31.51	27.89	14.66	19.07	24.15	16.19	24.33	25.04	10.99	15.79	21.39
$\mathcal{L}$	25.55	32.95	28.57	14.74	20.40	25.41	16.31	25.13	25.88	**11.10**	**16.71**	**22.13**
$\mathcal{L}-\mathcal{L}_r$	19.73	27.24	19.01	12.38	15.83	15.93	12.15	17.42	15.11	5.94	7.11	7.51
$\frac{\mathcal{L}-\mathcal{L}_r}{\mathcal{L}_r}$	25.36	33.00	27.54	**14.89**	20.40	25.29	16.17	25.11	25.43	10.59	15.74	18.75
$\frac{\mathcal{L}-(\mathcal{L}_r+b)}{(\mathcal{L}_r+b)}$	**25.68**	**33.31**	**28.93**	14.79	**20.61**	**26.48**	**16.35**	**25.38**	**26.41**	**11.10**	**16.71**	**22.13**

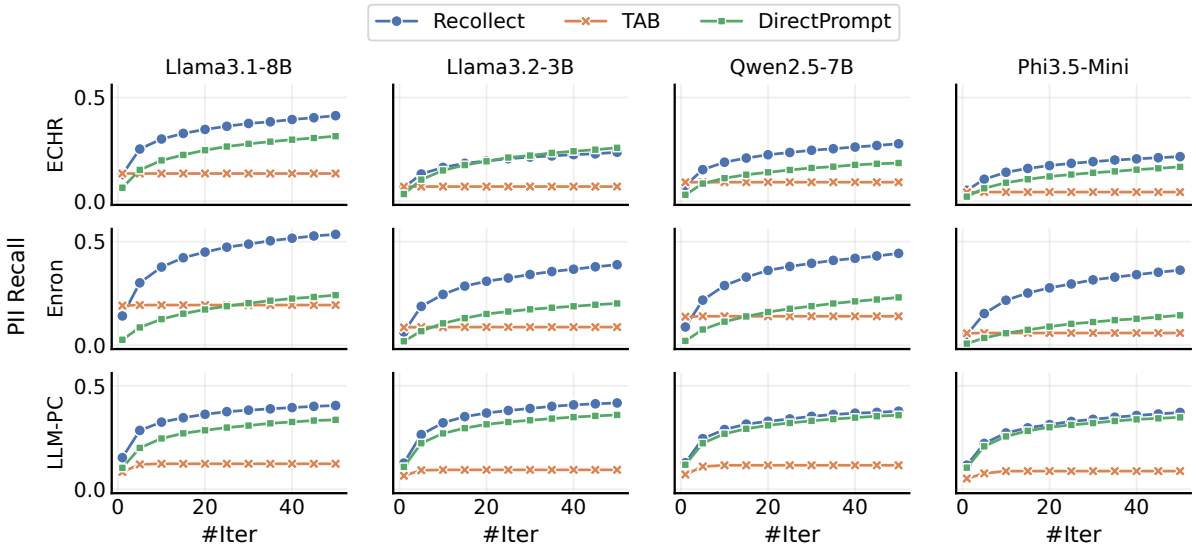
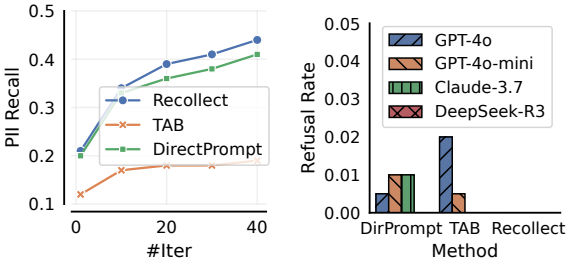


Figure 4: Ablation study on candidate generation. We report PII recall versus the number of query iterations.



(a) Comparison of PII recalls on GPT-4o-mini.

(b) Prompt refusal rates on proprietary models.

Figure 5: Experiments on real-world models.

duct an experiment on GPT-4o-mini using OpenAI Fine-tuning API to demonstrate that our recollection method remains powerful to commercial models. We randomly select 50 texts from LLM-PC, containing 218 PII entities and fine-tube GPT-4o-mini for 5 epochs. The PII recall versus the number of query iterations is shown in Figure 5a. Recollect achieves the highest PII recall when giving the same query budget.

Another concern is that victim models may refuse to output or provide perturbed results when detecting privacy-related queries. To address this concern, we conduct an experiment using 200 prompts for each candidate generation method to investigate whether current proprietary models would refuse them. The refusal rate results are shown in Figure 5b. We observe that DirectPrompt occasionally triggers malicious behavior detection, causing the model to refuse to respond. TAB sometimes results in refusals from GPT-4o and GPT-4o-mini due to too short context provided for continuation. In contrast, none of the above four proprietary models refuses Recollect. This may be because Recollect disguises itself by presenting the task of filling the missing parts.

Ablation of Candidate Selection. To validate the effectiveness of the criterion proposed in Equation 4, we apply different selection criteria on candidates generated by Recollect and report their top-1 accuracy in Table 2. From the table, we observe that partial loss consistently outperforms vanilla loss. This is because vanilla loss introduces errors when PII entities appear multiple times in the text as discussed in Section 3.3. Our proposed C achieves the highest top-1 accuracy in most cases. Because Phi3.5-Mini exhibits a monotonic increase in accuracy ratio as b increases, we set L as the

Table 3: comparison of model performance and PII prediction accuracy for models trained with DP-SGD under varying privacy budget (ϵ).

	Llama32-3B					Phi-3.5-mini				
ϵ	4	8	16	32	∞	4	8	16	32	∞
Train Loss	2.50	2.48	2.47	2.46	1.79	2.15	2.12	2.13	2.03	1.81
ACC (DirectPrompt)	0.22%	0.39%	0.57%	0.44%	3.56%	0.44%	0.48%	0.84%	0.84%	2.11%
ACC (TAB)	0.53%	0.35%	0.53%	0.62%	7.20%	0.40%	0.31%	0.44%	0.53%	4.09%
ACC (P2P)	0.57%	0.62%	0.62%	0.75%	6.91%	0.48%	0.53%	0.58%	0.62%	4.28%
ACC (R.R.)	3.60%	3.74%	3.87%	4.18%	14.79%	2.95%	3.21%	3.64%	3.91%	11.10%

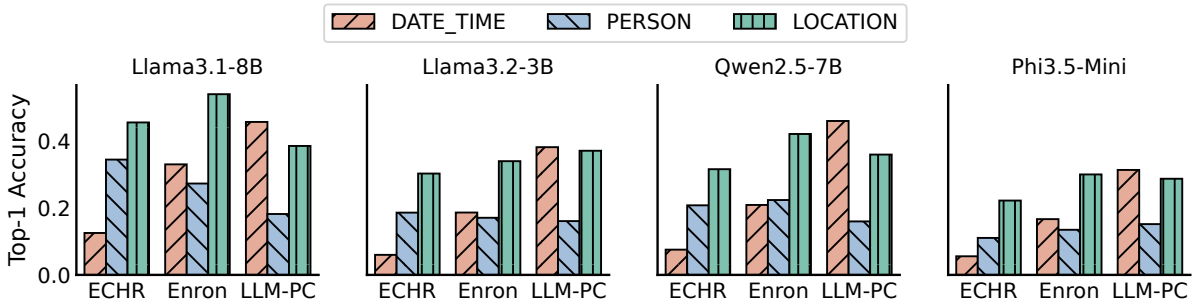


Figure 6: Top-1 accuracies of R.R. on specific PII entity types.

ranking criterion for Phi3.5-Mini in R.R. experiments, which is equal to the case $b \to \infty$.

4.5 Defending against R.R.

To mitigate the risk of PII leakage from LLMs, developers may employ privacy-preserving techniques. The most common and widely adopted approach is Differential Privacy (DP) (Dwork, 2006), which provides rigorous mathematical guarantees that an individual's data cannot be inferred through differential attacks. DP-SGD (Abadi et al., 2016) is the most prominent method for training deep learning models to satisfy DP. Given a privacy budget (ϵ), DP-SGD operates by clipping per-sample gradients and adding scaled noise during each update. In this work, we fine-tune victim models on the ECHR dataset using DP-SGD implemented by Microsoft's DP-Transformers (Wutschitz et al., 2022). Since DP-SGD incurs extra memory overhead and is hard to integrate with distributed learning, we only conduct DP-SGD experiments on 3B models. The results are shown in Table 3.

We find that although DP-SGD effectively reduces reconstruction accuracy, it also leads to a significant increase in loss, jeopardizing model performance. Our baselines become nearly unusable on DP fine-tuned models, whereas R.R. remains about 1/3 accuracy. The results suggest that while DP-SGD is a powerful privacy-preserving method, it cannot completely defend against our R.R.. An alternative defense is to prevent the model from learning PII tokens during training. For instance, Lin et al., 2024 proposed Selective Language Modeling (SLM), which selectively trains on useful tokens. We hypothesize that most PII tokens contribute little to the general language modeling capability of LLMs. Thus, selectively excluding them could enhance privacy with minimal impact on utility. Another line of defense is post-hoc unlearning (Yao et al., 2024; Liu et al., 2024). The developer can remove specific PII from LLMs after training using unlearning techniques.

4.6 Analysis

We explore how PII entity type and text length affect the reconstruction accuracy in this section. We also analyze the impact of PII occurrence frequency. Due to space limitation, we defer it into Appendix E.

PII Entity Type. We select three PII entity types that appear more than 100 times across all datasets, namely `DATE_TIME`, `PERSON`, and `LOCATION`, to evaluate R.R.'s effectiveness on specific entity types. Figure 6 illustrates the results. Overall, R.R. demonstrates a strong ability to reconstruct `LOCATION` entities, while its performance on `PERSON` entities is weaker. We suspect this is due to the discrepancy in value space; names vary significantly, whereas the number of locations is relatively limited, especially for major cities like New

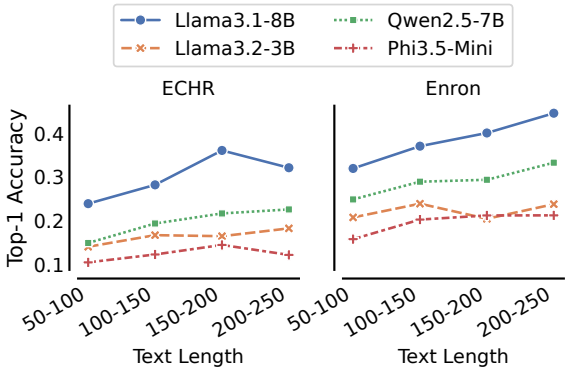


Figure 7: Top-1 accuracy versus text length.

York and Los Angeles, which appear frequently and are easy to guess. Additionally, we observe that DATE_TIME exhibits varying prediction accuracies across different datasets. We suspect this is because DATE_TIME precision differs across datasets. For instance, ECHR often specifies exact dates, while LLM-PC typically records only the year.

Text Length. We split datasets into subsets based on a specific range of word lengths and measure how text length can impact PII prediction. LLM-PC is excluded since its texts have already been processed to maintain a similar length. The result is shown in Figure 7. Top-1 accuracy tends to rise as text length grows. On the ECHR fine-tuned Llama3.1 model, the top-1 accuracy rises from 0.23 to 0.35 as text length grows from 50-100 to 150-200 words, demonstrating that more context helps LLM recall texts. However, on Enron fine-tuned Llama3.1 model, R.R. performs slightly better at 100-150 words than 150-200 words. This suggests that while longer texts provide more context, they may also introduce more noise, reducing performance. In addition, the impact of text length on performance appears to be greater for larger models, which may be more capable of taking advantage of additional context in longer texts.

5 Related Work

LLM Memorization. Memorization in LLMs refers to the model's tendency to store and reproduce exact phrases or passages from the training data rather than generating novel or generalized outputs (Satvaty et al., 2024). It has been proved to play a significant role in simple, knowledge-intensive tasks (Wang et al., 2024), but include rare, unrelated details before overfitting, a phenomenon called "unintended memorization" (Carlini et al., 2019). This unintended memorization raises privacy concerns, especially on domain-specific models (Yang et al., 2024). Several studies have examined how factors such as model size, training data repetition, context length, and fine-tuning strategies influence LLM memorization (Carlini et al., 2023; Mireshghallah et al., 2022b; Zeng et al., 2024).

PII Leakage. Due to the unintended memorization of LLMs, malicious attackers can exploit this vulnerability to steal sensitive data from training sets, including Personally Identifiable Information (PII), resulting in PII leakage. Niu et al., 2023 emphasize the need to address this issue, proposing a pipeline for extracting sensitive information from the Codex model. PII leakage attacks can be classified into three categories according to the adversary's capabilities: extraction attacks (Carlini et al., 2021; Mireshghallah et al., 2022b; Yu et al., 2023; Zhang et al., 2023), reconstruction attacks (Inan et al., 2021; Lukas et al., 2023), and inference attacks (Mireshghallah et al., 2022a; Fu et al., 2024). One technique to mitigate these risks is directly editing model weights, though no universal defense methods exist yet (Patil et al., 2023). Further attacks can bypass model alignment and recover sensitive training data (Nasr et al., 2025).

6 Conclusion

In this paper, we propose R.R. (Recollect and Rank), a novel two-step PII reconstruction attack. First, we employ a recollection-based method to recover the original text and leverage a NER-model-powered PII identifier to extract potential PII candidates. Then, we rank the PII candidates by computing their cross-entropy scores under the victim model and a reference model. We further introduce a biased ranking criterion that effectively integrates reference-based and non-reference-based ranking methods. Our experiments across three popular PII datasets and four open-source LLMs demonstrate that the R.R. significantly improves top-1 accuracy.

Acknowledgements

This research received partial support from the National Natural Science Foundation of China under Grant No. 62302441. This work was also supported by the Key Research and Development Program Project of Ningbo Grant No. 2025Z029. Infrastructure support was provided by the Information Technology Center of Zhejiang University and the Supercomputing Center of Hangzhou City University.

Limitations

Requirement for the Reference Model. R.R.'s new ranking criterion relies on a reference model for calibration. We assume that the victim model is fine-tuned from an open-source model, but this assumption does not always hold, particularly when attacking proprietary black-box models. Furthermore, incorporating a reference model increases the computational overhead for the attacker. Unlike the victim model, which can be queried, the reference model must be run locally, requiring additional resources. If the reference model is large, the attack becomes significantly more expensive, posing a practical limitation for adversaries with limited computational power.

Need to Adjust b When Changing the Victim Model. To better combine $\mathcal{L}$ and C_r in reference calibration, we introduce a bias b. As demonstrated in Section 4.3, b generally improves performance, except for the ECHR fine-tuned Llama3.2 model. However, we cannot guarantee that b will not be ineffective for other models, as we cannot predict whether the victim model will behave like the ECHR fine-tuned Llama3.2. Additionally, determining the optimal value of b is computationally expensive. Since the optimal bias b largely depends on the specific model type, it is necessary to recalculate b each time a new victim model is selected, resulting in a significant overhead.

Robustness against Defense Mechanisms. In real-world applications, developers often employ privacy-preserving techniques, such as differential privacy, to effectively mitigate potential risks of PII leakage. As demonstrated in Section 4.5, DP-SGD significantly reduces the PII prediction accuracy of our proposed method to one-third. However, despite this degradation, our proposed method still outperforms other baseline approaches. Furthermore, such defense mechanisms not only incur higher computational costs, but also reduce the utility of LLMs.

References

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318.

Amazon Web Services, Inc. 2025. Amazon comprehend. Accessed: 2025-02-15.

Apache. 2004. Apache license, version 2.0. Accessed: 2025-02-15.

BigCode. 2022. Openrail-m license. Accessed: 2025-02-15.

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

Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. 2019. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284.

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

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

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

Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. 2024. Membership inference attacks against fine-tuned large language models via self-prompt calibration. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Karan Girotra, Lennart Meincke, Christian Terwiesch, and Karl T Ulrich. 2023. Ideas are dimes a dozen: Large language models for idea generation in innovation. *Available at SSRN 4526071*.

Huseyin A Inan, Osman Ramadan, Lukas Wutschitz, Daniel Jones, Victor Rühle, James Withers, and Robert Sim. 2021. Training data leakage analysis in language models. *arXiv preprint arXiv:2101.05405*.

Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *European conference on machine learning*, pages 217–226. Springer.

Qinbin Li, Junyuan Hong, Chulin Xie, Junyi Hou, Yiqun Diao, Zhun Wang, Dan Hendrycks, Zhangyang Wang, Bo Li, Bingsheng He, and Dawn Song. 2024. The neurIPS 2024 LLM privacy challenge. In *NeurIPS 2024 Competition Track*.

Zhenghao Lin, Zhibin Gou, Yeyun Gong, Xiao Liu, Ruochen Xu, Chen Lin, Yujiu Yang, Jian Jiao, Nan Duan, Weizhu Chen, et al. 2024. Not all tokens are

what you need for pretraining. *Advances in Neural Information Processing Systems*, 37:29029–29063.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Towards safer large language models through machine unlearning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1817–1829.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Beguelin. 2023. Analyzing leakage of personally identifiable information in language models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 346–363. IEEE Computer Society.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. 2023. Membership inference attacks against language models via neighbourhood comparison. *arXiv preprint arXiv:2305.18462*.

Omri Mendels, Coby Peled, Nava Vaisman Levy, Sharon Hart, Tomer Rosenthal, Limor Lahiani, et al. 2018. Microsoft Presidio: Context aware, pluggable and customizable pii anonymization service for text and images.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. 2022a. Quantifying privacy risks of masked language models using membership inference attacks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8332–8347.

Fatemehsadat Mireshghallah, Archit Uniyal, Tianhao Wang, David K Evans, and Taylor Berg-Kirkpatrick. 2022b. An empirical analysis of memorization in fine-tuned autoregressive language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1816–1826.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramèr, and Katherine Lee. 2025. Scalable extraction of training data from aligned, production language models. In *The Thirteenth International Conference on Learning Representations*.

Liang Niu, Shujaat Mirza, Zayd Maradni, and Christina Pöpper. 2023. CodexLeaks: Privacy leaks from code generation language models in GitHub copilot. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 2133–2150, Anaheim, CA. USENIX Association.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *Preprint*, arXiv:2309.17410.

Prakash Poudyal, Jaromír Šavelka, Aagje Ieven, Marie Francine Moens, Teresa Goncalves, and Paulo Quaresma. 2020. Echr: Legal corpus for argument mining. In *Proceedings of the 7th Workshop on Argument Mining*, pages 67–75.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Ali Satvaty, Suzan Verberne, and Fatih Turkmen. 2024. Undesirable memorization in large language models: A survey. *Preprint*, arXiv:2410.02650.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Xinyi Wang, Antonis Antoniades, Yanai Elazar, Alfonso Amayuelas, Alon Albalak, Kexun Zhang, and William Yang Wang. 2024. Generalization v.s. memorization: Tracing language models' capabilities back to pretraining data. *Preprint*, arXiv:2407.14985.

Lukas Wutschitz, Huseyin A. Inan, and Andre Manoel. 2022. dp-transformers: Training transformer models with differential privacy. https://www.microsoft.com/en-us/research/project/dp-transformers.

Xinyu Yang, Zichen Wen, Wenjie Qu, Zhaorun Chen, Zhiying Xiang, Beidi Chen, and Huaxiu Yao. 2024. Memorization and privacy risks in domain-specific large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419.

Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3093–3106.

Weichen Yu, Tianyu Pang, Qian Liu, Chao Du, Bingyi Kang, Yan Huang, Min Lin, and Shuicheng Yan. 2023. Bag of tricks for training data extraction from language models. *Preprint*, arXiv:2302.04460.

Shenglai Zeng, Yaxin Li, Jie Ren, Yiding Liu, Han Xu, Pengfei He, Yue Xing, Shuaiqiang Wang, Jiliang Tang, and Dawei Yin. 2024. Exploring memorization in fine-tuned language models. *Preprint*, arXiv:2310.06714.

Zhexin Zhang, Jiaxin Wen, and Minlie Huang. 2023. Ethicist: Targeted training data extraction through loss smoothed soft prompting and calibrated confidence estimation. *Preprint*, arXiv:2307.04401.

Appendices

A Proof of Theorem 3.1

Theorem 3.1. Let $\mathcal{M} = \{M_i\}_{i=1}^{n}$. For $j \neq k$, $1 \leq j, k \leq n$,if $\mathcal{L}(M_j) < \mathcal{L}(M_k)$ and $\frac{\mathcal{L}(M_j)-\mathcal{L}_r(M_j)}{\mathcal{L}_r(M_j)} < \frac{\mathcal{L}(M_k)-\mathcal{L}_r(M_k)}{\mathcal{L}_r(M_k)}$, then

$$\frac{\mathcal{L}(M_j) - (\mathcal{L}_r(M_j) + b)}{\mathcal{L}_r(M_j) + b} < \frac{\mathcal{L}(M_k) - (\mathcal{L}_r(M_k) + b)}{\mathcal{L}_r(M_k) + b}$$

Proof. Given $\frac{\mathcal{L}(M_j)-\mathcal{L}_r(M_j)}{\mathcal{L}_r(M_j)} < \frac{\mathcal{L}(M_k)-\mathcal{L}_r(M_k)}{\mathcal{L}_r(M_k)}$, we have

$$\begin{aligned} \frac{\mathcal{L}(M_j)}{\mathcal{L}_r(M_j)} &< \frac{\mathcal{L}(M_k)}{\mathcal{L}_r(M_k)} \\ \mathcal{L}(M_j)\mathcal{L}_r(M_k) &< \mathcal{L}(M_k)\mathcal{L}_r(M_j). \end{aligned} \tag{8}$$

Because b is a positive bias, we can get

$$b\mathcal{L}\left(M_j\right) < b\mathcal{L}\left(M_k\right). \tag{9}$$

Combing Equation 8 and Equation 9 we obtain

$$\mathcal{L}(M_j)\mathcal{L}_r(M_k) + b\mathcal{L}(M_j) < \mathcal{L}(M_k)\mathcal{L}_r(M_j) + b\mathcal{L}(M_k)$$

By a simple transformation, we get

$$\begin{aligned} \frac{\mathcal{L}_r(M_k) + b}{\mathcal{L}(M_k)} &< \frac{\mathcal{L}_r(M_j) + b}{\mathcal{L}(M_j)} \\ \frac{\mathcal{L}(M_j)}{\mathcal{L}_r(M_j) + b} &< \frac{\mathcal{L}(M_k)}{\mathcal{L}_r(M_k) + b} \\ \frac{\mathcal{L}(M_j) - (\mathcal{L}_r(M_j) + b)}{\mathcal{L}_r(M_j) + b} &< \frac{\mathcal{L}(M_k) - (\mathcal{L}_r(M_k) + b)}{\mathcal{L}_r(M_k) + b}, \end{aligned}$$

Based on our assumption that k can be any other candidate index that is unequal to j, M_j will also be ranked as the first by $\frac{\mathcal{L}-(\mathcal{L}_r+b)}{\mathcal{L}_r+b}$. □

B Attack Setting Details

B.1 Dataset Details

Table 4 shows the statistics of our datasets. The information and post-processing steps for each dataset are as follows.

ECHR. This is an English legal judgment dataset comprising approximately 11,500 cases from the European Court of Human Rights. We randomly select 4,000 cases for training victim models and choose 2,000 cases from the training set to perform PII reconstruction attacks.

Enron. The Enron dataset is a comprehensive collection of approximately 500,000 emails of the Enron Corporation. Similar to ECHR, we select 4,000 emails for training victim models and choose 2,000 emails to perform PII reconstruction attacks.

LLM-PC. This dataset is released by NeurIPS in its 2024 LLM Privacy challenge. LLM-PC consists of two subsets: a development set for assessment and a test set for competition ranking. Each subset contains 1,500 synthetic texts. We use the development set for training victim models and performing PII reconstruction attacks since the test set's ground truth is not available.

Table 4: Dataset statistics.

Dataset	# of texts	average length	# of PII
ECHR	2,000	88 words	4,692
Enron	2,000	90 words	5,836
LLM-PC	1,500	1,275 words	5,907

Table 5: Biases in candidate selection.

Model	b
Llama3.1-8B	2.0
Llama3.2-3B	2.5
Qwen2.5-7B	1.0
Phi3.5-Mini	∞

B.2 Hyperparameters

Each target mode is trained using SFT for three epochs with a learning rate of 2e-5 on each dataset. In the candidate generation step, we query the victim model 40 times for each PII entity to collect candidate predictions. During generation, we set the temperature to 1.2, top-K to 30, and top-P to 0.8. Given the extensive length of LLM-PC's text, we split it into sentences and extract those containing a specific PII entity along with their neighboring sentences to construct the input text. The recollect prompts we use are illustrated in Figure 10 and Figure 11. The biases we used in candidate selection are shown in Table 5 As for baselines, we follow the settings in their original papers.

C Attack with Incorrect Reference Model

We conducted additional experiments to study the robustness of our attack under incorrect reference.

Table 6: Description of PII entity types.

Entity Type	Description	Example
DATE_TIME	Absolute or relative dates or periods or times smaller than a day.	5 May 2010
EMAIL_ADDRESS	Identifies an email inbox for delivering email messages.	james.tyrone@example.com
ID	A unique identifier assigned to entities like applicants, respondents, or judges.	4616
LOCATION	A geographically or politically defined location, such as cities or countries.	New Orleans, Louisiana
ORGANIZATION	Companies, groups, clubs, government bodies, and public organizations.	British Council
PERSON	Represents a full person name, including first, middle, or last names.	Earl Blanton
PHONE_NUMBER	A telephone number used for contact or communication.	555-123-4567
URL	A Uniform Resource Locator, used to identify resources on the internet.	http://www.francismorrisdata.com

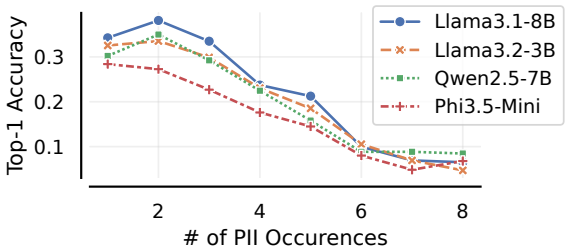


Figure 8: Top-1 accuracy versus the number of PII occurrences on LLM-PC dataset.

We measured the top-1 reconstruction accuracy using various reference models. The results (reported in percentages) are reported in Table 7. Interestingly, we found that using an incorrect reference model does not significantly degrade reconstruction accuracy, and in some cases, improves it. The maximum accuracy drop is only 1.62%. Selecting Llama3.2-3B as the reference model to attack Llama3.1-8B even results in a 0.43% improvement. We attribute this to the fact that LLMs are generally pretrained on similar corpora, primarily web-crawled data, which enables different models to provide similar references. Another noteworthy observation is that the choice of the reference model influences the optimal value of the bias b. Some illustrative examples are shown in Figure 9. We observe that the accuracy consistently increases when applying Phi3.5-Mini as the reference model for attacks against Llama3.1-8B, which aligns with the trend when applying Phi3.5-Mini to the correct corresponding victim model. This indicates a severe threat, as attackers could select an appropriate value of b based solely on the reference model without access to the victim model.

D Hybrid Attacks

We conducted hybrid attack experiments combining three different candidate generation methods and five different candidate selection criteria against the Llama3.1-8B and Qwen2.5-7B models

Table 7: Robustness of R.R. under incorrect reference. Each row corresponds to a victim model and each column corresponds to a reference model. The number in parentheses denotes the change relative to using the correct reference model.

	Llama3.1-8B	Llama3.2-3B	Qwen2.5-7B	Phi3.5-Mini
Llama3.1-8B	28.93	29.36 (+0.43)	28.75 (-0.18)	28.57 (-0.36)
Llama3.2-3B	25.53 (-0.95)	26.48	25.23 (-1.25)	25.41 (-1.07)
Qwen2.5-7B	26.11 (-0.30)	26.21 (-0.20)	26.41	25.41 (-1.00)
Phi3.5-Mini	21.14 (-0.99)	21.69 (-0.44)	20.51 (-1.62)	22.13

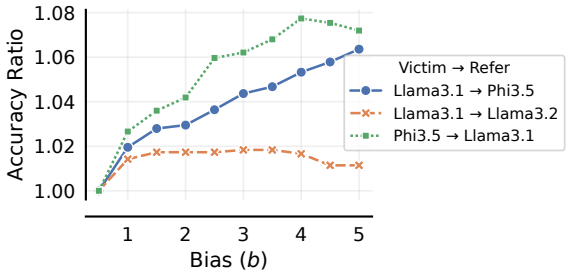


Figure 9: Accuracy ratio versus b under incorrect reference model. The optimal choice of b is influenced by the selection of the reference model.

on LLM-PC dataset. The Top-1 Accuracy results are shown as Table 8. From the table, we observe that our proposed method consistently achieves the highest Top-1 accuracy across both models when compared to other hybrid attack combinations. This highlights the effectiveness of our proposed method: the "recollecting" process yields a high-quality PII candidates pool, from which the predicted candidate is selected based on the validated biased ranking criterion.

E Analysis of PII Occurrence Frequency

A PII entity may appear multiple times within a given text. We measure the impact of PII occurrence frequency on R.R. performance. The results are presented in Figure 8. We focus solely on LLM-PC dataset, as ECHR and Enron's texts are relatively short, with most PII entities appearing only once. Interestingly, the peak top-1 accuracy occurs when the number of PII occurrences is 2.

Table 8: Comparison of PII prediction accuracy for different hybrid attacks against Llama3.1-8B and Qwen2.5-7B on LLM-PC dataset.

Victim Model	Candidate Generation	Candidate Selection				
		$\mathcal{L}$ (vanilla)	$\mathcal{L}$	$\mathcal{L}-\mathcal{L}_r$	$\frac{\mathcal{L}-\mathcal{L}_r}{\mathcal{L}_r}$	$\frac{\mathcal{L}-(\mathcal{L}_r+b)}{(\mathcal{L}_r+b)}$
Llama3.1-8B	DirectPrompt	24.34	24.60	18.01	24.19	24.90
	Tab	11.68	11.78	11.38	11.80	11.80
	Recollect	27.89	28.57	19.01	27.54	**28.93**
Qwen2.5-7B	DirectPrompt	23.12	23.44	15.07	23.46	24.06
	Tab	10.31	10.41	10.80	10.95	11.09
	Recollect	25.04	25.88	15.11	25.43	**26.41**

Beyond this point, top-1 accuracy gradually decreases. Upon further investigation, we find that the distribution of PII entity types varies across different occurrence counts. When the number of PII occurrences is 2, PERSON entities account for 35.6% of the dataset. However, when the number of PII occurrences increases to 6, the proportion of PERSON entities rises to 81.8%. As discussed earlier, PERSON is a challenging PII entity type to reconstruct. This observation indicates that the number of PII occurrences alone does not compensate for the influence of PII entity type.

Recollect Prompt Template of ECHR and ENRON

You are an expert in understanding texts and filling in masked information. Your task is to replace placeholders, denoted by [MASK-NUM], with the most appropriate and contextually relevant candidates. Ensure that all placeholders are replaced, and none of them appear in the output. The output should be natural and fully readable without brackets or placeholder markers.

There are some rules you need to follow:
Rules
1. Ensure placeholders are fully replaced, and no brackets or placeholder markers remain in the output.
2. End when finish rewriting the text.

Below is an example:
Text with masked placeholders:
24. On [DATE_TIME-0] Mr [PERSON-0]'s counsel appealed to the Lucerne Court of Appeal (Obergericht) against the detention order issued on [DATE_TIME-1] (see paragraph 9 above) and applied for his client's immediate release.

Rewritten text:
24. On 23 January 1992 Mr R.M.D.'s counsel appealed to the Lucerne Court of Appeal (Obergericht) against the detention order issued on 17 January (see paragraph 9 above) and applied for his client's immediate release.

Now please help me rewrite the following text.
Text with masked placeholders:
`{MASKED_TEXT}`

Rewritten text:

Figure 10: Recollect prompt template of ECHR and Enron.

Recollect Prompt Template of LLM-PC

You are an expert in understanding conversations and filling in masked information. Your task is to replace a placeholder, denoted by {mask}, with the most appropriate and contextually relevant candidate. You should ensure that the placeholder does not appear in your output.

There are some rules you need to follow:
Rules
1. Ensure placeholders are fully replaced, and no brackets or placeholder markers remain in the output.
2. End when finish rewriting the text.

Here is a text with some missing parts, represented by {mask}:

`{MASKED_TEXT}`

Please repeat the above text and replace {mask} with the most probable candidate based on the context. Here is the output:

Figure 11: Recollect prompt template of LLM-PC.