Simple Yet Effective: Extracting Private Data Across Clients in Federated Fine-Tuning of Large Language Models

Anonymous ACL submission

Abstract

Federated fine-tuning of large language models (FedLLMs) presents a promising approach for achieving strong model performance while preserving data privacy in sensitive domains. However, the inherent memorization ability of LLMs makes them vulnerable to training data extraction attacks. To investigate this risk, we introduce simple yet effective extraction attack algorithms specifically designed for FedLLMs. In contrast to the "verbatim" extraction attacks, which assume access to fragments from all training data, our approach operates under a more realistic threat model, where the attacker only has access to a single client's data and aims to extract previously unseen personally identifiable information (PII) from other clients. This requires leveraging contextual prefixes held by the attacker to generalize across clients. To evaluate the effectiveness of our approaches, we propose two rigorous metrics—coverage rate and efficiency—and extend a real-world legal dataset with PII annotations aligned with CPIS, GDPR, and CCPA standards, achieving 89.9% human-verified precision. Experimental results show that our method can extract up to 56.57% of victim-exclusive PII, with "Address," "Birthday," and "Name" being the most vulnerable categories. Our findings underscore the pressing need for robust defense strategies and contribute a new benchmark and evaluation framework for future research in privacy-preserving federated learning. The data and code will be made publicly available to facilitate reproducibility.[1]

1 Introduction

Fine-tuning large language models (LLMs) in a federated learning (FL) setting (Ye et al., 2024a,b; Zhang et al., 2023; Chen et al., 2024; Yao et al., 2024) enables collaborative training across distributed clients without requiring centralized access to private data. Recent advancements (Bai et al., 2024; Wu et al., 2025; Li et al., 2020; Karimireddy et al., 2020) have primarily focused on improving the performance and efficiency of FL algorithms. However, the privacy risks—particularly the threat of training data extraction—from federated LLMs (FedLLMs) remain insufficiently explored.

LLMs are well known for memorizing and regurgitating parts of their training data (Carlini et al., 2021, 2023), including sensitive content such as personally identifiable information (PII) (Shao et al., 2024; Kim et al., 2023; Nakka et al., 2024). Although FL helps preserve data locality by exchanging model updates instead of raw data, our preliminary experiments (Appendix B.3) reveal that FedLLMs are still susceptible to *verbatim data extraction* (VDE) attacks—where an attacker recovers large fragments of training data from the aggregated global model. However, VDE relies on unrealistic assumptions such as the adversary having access to large portions of the training corpus (Yu et al., 2023; Huang et al., 2022).

In this work, we consider a more realistic threat model: a malicious client exploits its own local context to extract previously unseen PII from other clients via the shared global model. For example, a prefix like "`The trial of this case has now concluded. The plaintiff, [Name]...`" from the attacker's data can be used to elicit sensitive information memorized from other clients. Based on this scenario, we introduce three simple yet effective extraction strategies: (1) using all contextual prefixes from the attacker's dataset to query the global model, (2) querying the model with high-frequency prefixes from the attacker's dataset(FP Sampling), (3) locally fine-tuning on prefix–PII pairs to amplify memorization-based extraction(LAFt).

To evaluate these attacks, we construct a benchmark dataset by annotating a real-world legal corpus with PII labels in accordance with major pri-

[1]To preserve anonymity, code and data release details will be disclosed after acceptance.

vacy regulations such as CPIS, GDPR, and CCPA (see Acronyms List A). We assess attack success through two metrics: (i) *coverage*—the proportion of target PII extracted from other clients, and (ii) *efficiency*—the amount of PII extracted within a limited query budget.

Experiments show that our attack methods can achieve up to 56.57% coverage, with "Address," "Birthday," and "Name" being the most vulnerable PII types. We also observe diminishing returns in coverage as the prefix budget increases. Notably, FP Sampling and LAFt enhance the diversity of extracted PII under constrained budgets. These results highlight a concrete privacy risk in FedLLMs and call for stronger defense mechanisms.

This paper makes the following key contributions:

1. We propose three novel and effective data extraction attack strategies targeting FedLLMs, evaluated using the metrics of coverage and efficiency. These attacks are orthogonal to existing approaches such as gradient reconstruction and active membership inference.
2. We conduct comprehensive experiments showing that our attacks can extract up to 56.57% of victim-exclusive PII, and revealing a trade-off between extraction coverage and efficiency.
3. We build a real-world benchmark dataset by augmenting a legal corpus with fine-grained PII annotations aligned with regulatory standards (CPIS, GDPR, CCPA), addressing the scarcity of resources for studying privacy in FL.

2 Related Work

This study is related to the fields of data extraction attacks and federated learning. For the reader's convenience, a brief introduction to these concepts is provided in Appendix B. In this section, we review only the work directly related to our method.

2.1 PII Extraction Attacks in LLM

Large language models, due to their massive parameter scale, are capable of memorizing exact training data samples, making them vulnerable to data extraction attacks. These attacks can target different granularities of information: sample-level and entity-level.

At the sample level, an attacker with access to the full prefix of a training sample can query the LLM to regenerate the exact suffix (Yu et al., 2023; Shi et al., 2024; Zhang et al., 2024). This technique, known as *verbatim training data extraction* (VDE) (Carlini et al., 2021, 2023; Schwarzschild et al., 2024), is widely used to detect data contamination and copyright violations (Dong et al., 2024).

At the entity level, attackers may know a subset of PII entities—such as names or affiliations—about a particular subject. By combining these known details with prompt templates (either manually crafted or automatically generated (Kassem et al., 2025)), they can elicit the model to produce additional PII records about the same subject. This is known as an associative data extraction attack (Shao et al., 2024; Kim et al., 2023; Zhou et al., 2024).

Broadly, PII extraction attacks refer to any attack that aims at eliciting outputs from the model that contain PII (Lukas et al., 2023; Nakka et al., 2024; Huang et al., 2022). Both verbatim and associative techniques can be used to conduct such attacks.

While most prior work assumes centralized training with full data access, we investigate PII extraction under federated fine-tuning, where the attacker has limited observability and control. We elaborate on this in Section 4.1.

2.2 Privacy Threats in Federated Learning

Threats in Federated Learning can be categorized into two main areas: security and privacy (Wang et al., 2024; Xie et al., 2024; Li et al., 2024b). Security threats typically aim to disrupt the entire FL system by invalidating model training (Shejwalkar and Houmansadr, 2021) and introducing backdoors (Bagdasaryan et al., 2020; Chang et al., 2024). In contrast, privacy threats have attracted more attention from researchers and focus on stealing confidential information from the FL system, such as inferring sensitive properties (Melis et al., 2019), reconstructing clients' private datasets (Zhu et al., 2019; Geiping et al., 2020), and determining the membership and source of training data (Rashid et al., 2025; Vu et al., 2024; Hu et al., 2024). To achieve these attacks, researchers often make different assumptions regarding the attacker's knowledge. Common assumptions typically fall into two dimensions: whether the attacker is a client or a server (Chu et al., 2023), and whether the attacker is semi-honest (Applebaum, 2017; Hu et al., 2024) or malicious. These assumptions determine whether the attacker has access to gradients, local datasets, model parameters, and the ability to

manipulate them.

3 Dataset

3.1 Data Sources and Preprocessing

The majority of our dataset is sourced from the Challenge of AI in Law (CAIL) (Li et al., 2024a), supplemented by smaller portions from CJRC (Duan et al., 2019) and JEC-QA (Zhong et al., 2020). CAIL is a renowned annual competition featuring a variety of legal NLP tasks. In our study, we focus on two natural language generation tasks (i.e., Summary and Reading Comprehension) and three natural language understanding tasks (i.e., Match, Exam, and Classification). Detailed task descriptions are provided in Appendix E, with Table 6 illustrating representative examples for each task. Following prior work (Zhang et al., 2023; Yue et al., 2024) on LLM and FedLLM applications in the legal domain, we further preprocess and curate the dataset to fit our setting. The complete preprocessing procedure is outlined in Appendix F, where Table 7 presents comprehensive dataset statistics.[2]

3.2 PII Labeling

We reviewed the definitions and examples of PII in various legal provisions, including CPIS, GDPR, CCPA, and Singapore PDPC (see Acronyms List in Appendix A), and used them as references to establish a systematic PII labeling standard. We selected PII types relevant to the text modality and removed types that are unlikely to appear in legal texts (e.g., browser history, SMS content, IP & MAC addresses), as well as those that are difficult to describe or evaluate (e.g., medical examination reports, psychological trends). Ultimately, we defined labeling guidelines encompassing 7 major categories and 36 subcategories. These standards are summarized in Table 9, and the distribution of labeled PII types is shown in Figure 1.

We employed a combination of machine-assisted annotation and manual verification to label the data. For each major PII category, we designed a dedicated prompt (Figure 9) and used GPT-4o (OpenAI et al., 2024) to generate initial annotations. We then used Label Studio (Tkachenko et al., 2020-2025) and recruited students to review a subset of the annotated data. Details of the human annotation process—including annotator backgrounds and labeling instructions—are provided in Appendix J. For human review, we randomly selected 100 samples comprising over 6,000 PII annotation instances, yielding an F1 score of 89.9%. The evaluation results are summarized in Table 10.

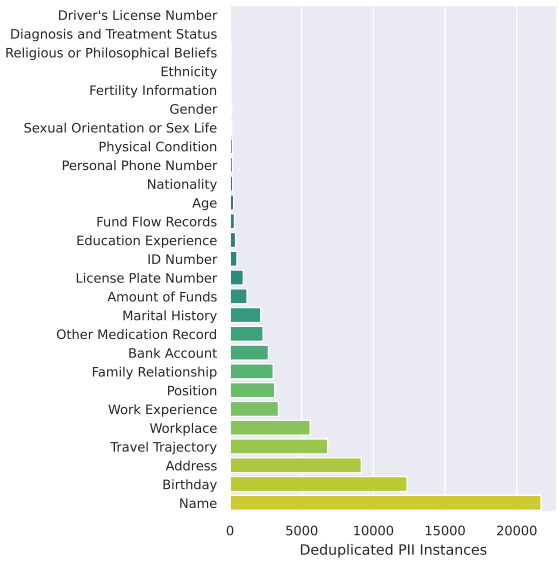

Figure 1: Distribution of de-duplicated PII instances by label category.

4 Method

4.1 Problem Definition

We study a novel extraction attack tailored to federated LLMs (FedLLMs), which differs from traditional verbatim data extraction (VDE) in three key aspects:

Assumptions. Unlike VDE that assumes the attacker has access to most or all of the training data, our setting limits the attacker to a small, isolated subset of the overall training corpus.

Setup. In our formulation, the prefix and its corresponding target suffix are not drawn from a contiguous span of training data. Instead, extraction prefixes are sampled from the attacker's local dataset D_a, while the target suffixes reside exclusively in other clients' private data and are absent from D_a. Thus, each prefix must generalize beyond local context to trigger the generation of unseen suffixes.

Goals. The attacker does not aim to recover all training completions, but instead focuses on extracting specific, high-value information—most notably, personally identifiable information (PII)—from the

[2]The datasets contain PII from publicly available government-published legal documents. They were de-identified and used in prior work (e.g., Yue et al. (2024)). We use curated versions from these papers. Since our study concerns privacy risks in FedLLMs, real-world PII is necessary to evaluate model vulnerabilities.

global model.

Formally, we consider an FL system comprising c clients $\mathcal{C} = \{C_1, C_2, \ldots, C_c\}$, where each client C_i holds a local dataset D_i. Among them, one client—denoted as the attacker $C_a \in \mathcal{C}$—is assumed to be semi-honest (Applebaum, 2017; Hu et al., 2024). That is, C_a faithfully follows the FL protocol (e.g., does not poison data or manipulate model weights), but acts adversarially in a passive manner, attempting to infer PII contained in other clients' datasets by analyzing the global model θ.

In this setting, the attacker issues queries to the model θ to extract data without knowing which client any particular output originates from. However, for evaluation purposes, we designate one client as the reference victim to measuring the attack's effectiveness. Let S_a and S_v denote the sets of PII instances held by the attacker and the victim client, respectively. The attacker constructs a prompt set P and queries the federated language model θ, obtaining a corresponding output set Y. We formalize key definitions and evaluation metrics in the following subsections.

Definition 1 (Extracted)**.** *A PII instance $s \in S_v$ is considered successfully* extracted *if there exists a prompt $p \in P$ and a corresponding model output $y \in Y$ such that:*

$$\exists u \in \Sigma^* \text{ such that } y = s \oplus u, \tag{1}$$

where Σ^* is the set of all finite-length strings over the vocabulary, and $\oplus$ denotes string concatenation. In other words, the model output y begins with s.

Definition 2 (Coverage Rate)**.** *The coverage rate measures how thoroughly the attacker recovers the PII unique to the victim client. It is defined as:*

$$S_E = \{s_i \mid \exists y \in Y \text{ such that } s_i \text{ is extracted by } y\},$$
$$CR = \frac{|(S_v \setminus S_a) \cap S_E|}{|S_v \setminus S_a|}. \tag{2}$$

A higher CR indicates that a larger fraction of the victim's unique PII has been successfully extracted.

Definition 3 (Efficiency)**.** *Efficiency quantifies the precision of extraction with respect to the number of queries. Let Q denote the number of queries, the efficiency is defined as:*

$$EF = \frac{|(S_v \setminus S_a) \cap S_E|}{Q}. \tag{3}$$

A higher EF indicates that more PII is extracted with fewer queries.

Building upon these definitions, the central challenge is to design algorithms that enable the attacker to extract PII both comprehensively and efficiently—that is, achieving high coverage and high efficiency.

4.2 Attacking Algorithms

4.2.1 PII-contextual Prefix Sampling

We begin with a straightforward method for constructing query prompts using PII-contextual prefixes—text segments that immediately precede PII instances in the attacker's own dataset D_i. This approach is motivated by the observation that manually crafted prompts such as "my phone number is" may not align with the model's training distribution and are often ineffective at extracting attacker-defined PII instances.

Let the attacker's training example be denoted as $U_a = \{t_0, t_1 \cdots t_{|U_a|}\}$, formed by concatenating all samples in D_i, where each t_i represents a word token. Let $\mathcal{S}$ be the multiset of PII instances labeled in U_a. Define the function $\text{Loc}(s)$ as the index of the first word of a PII instance s in U_a. We further define a contextual prefix extraction function $\mathcal{T}_\lambda(U, s)$ that returns the λ-length prefix ending just before s:

$$\mathcal{T}_\lambda(U, s) = t_{\text{Loc}(s)-\lambda} \cdots t_{\text{Loc}(s)-1}.$$

The resulting set of PII-contextual prompts is given by:

$$P_c = \{\mathcal{T}_\lambda(U_a, s) \mid s \in \mathcal{S}\}. \tag{4}$$

Once P_c is constructed, the attacker $\mathcal{C}_a$ uses the global model θ to generate candidate PII completions. For each prefix $p \in P_c$, a suffix y of at most m tokens is sampled according to the model's conditional distribution:

$$y = \{x_1, x_2 \ldots x_m\} \sim \mathbf{P}(y \mid p; \theta).$$

To increase the diversity of generations, $\mathcal{C}_a$ may sample n independent suffixes for each prefix p, forming:

$$Y_p = \{y_1, y_2, \ldots, y_n\}.$$

The final set of candidate generations is the union across all prefixes:

$$Y = \bigcup_{p \in P_c} Y_p.$$

This results in a total query cost of $Q = n \cdot |P_c|$.

We can further generalize the contextual prompt set P_c by collecting a multiset of all prefix substrings that end immediately before each PII instance:

$$\text{SUP}(P_c) = \{t_i \cdots t_{\text{Loc}(s)-1} \mid (\text{Loc}(s) - i) \in [1, \lambda]\}.$$

This produces a much larger set of contextual prefixes, which may yield a higher extraction coverage rate but an extremely low efficiency due to the massive number of query prompts.

4.2.2 Frequency-Prioritized Prefix Sampling

Motivated by prior work (Shao et al., 2024), which suggests that extraction effectiveness may be closely related to co-occurrence frequency, we hypothesize that prefixes occurring more frequently immediately before PII entities are more strongly associated with diverse PII instances. Based on this intuition, our objective is to prioritize such high-frequency prefixes in order to construct a more compact and informative prefix set.

To formalize this, we partition the multiset $\text{SUP}(P_c)$ based on prefix frequency. For each integer $\sigma \geq 1$, let:

$$P_\sigma = \{p \in \text{SUP}(P_c) \mid \text{Count}_{\text{SUP}(P_c)}(p) = \sigma\}.$$

which defines subsets of prefixes that occur exactly σ times. This induces a frequency-based decomposition of the unique prefixes in $\text{SUP}(P_c)$:

$$\text{Set}(\text{SUP}(P_c)) = \bigcup_{\sigma \geq 1} P_\sigma.$$

Given a frequency threshold σ_a, we define the set of frequent prefixes as:

$$P_{f \geq \sigma_a} = \bigcup_{\sigma \geq \sigma_a} P_\sigma, \tag{5}$$

which is then sorted in descending order of frequency. Notably, setting $\sigma_a = 1$ yields the full set of generalized contextual prefixes, $P_{f \geq 1} = \text{Set}(\text{SUP}(P_c))$, sorted in descending order of frequency.

Given a prefix budget B, we construct the final prefix set by selecting the top-B prefixes from $P_{f \geq \sigma_a}$, thereby prioritizing high-frequency prefixes during sampling.

4.2.3 Latent Association Fine-tuning

We hypothesize that a model's susceptibility to extraction attacks stems from its latent ability to associate two underlying conceptual distributions: (1) $\mathcal{A}$ — the distribution over prefixes that are semantically or syntactically likely to precede PII; and (2) $\mathcal{B}$ — the distribution over actual PII instances. Let $\text{Dist}(\mathcal{A}, \mathcal{B}; \theta)$ denote the degree of association between these two distributions under model parameters θ.

Because this association is implicitly encoded in the model's internal representations, we propose a method to reduce this distance through parameter updates—a technique we term Latent Association Fine-tuning (LAFt). The core idea is to fine-tune the model to minimize $\text{Dist}(\mathcal{A}, \mathcal{B}; \theta)$, thereby reinforcing its internal linkage between indicative prefixes and corresponding PII, ultimately improving its capacity for PII extraction.

To implement this, we construct a fine-tuning dataset D_{ft} by pairing frequent prefixes with known PII instances from the attacker's dataset:

$$D_{\text{ft}} = \{(p, s) \mid p \in P_f,\ s \in S_a\}, \tag{6}$$

where P_f is the set of frequent prefixes derived from D_a, and S_a is the attacker's known PII set. We then fine-tune the model using the standard causal language modeling objective:

$$\theta' = \arg\min_\theta \sum_{(p,s) \in D_{\text{ft}}} \sum_{t=1}^{|s|} -\log \mathbf{P}(s_t \mid p, s_{<t}; \theta).$$

The updated model θ' is then used for extraction, using prefixes from either P_f or P_c.

5 Experiment

5.1 Experimental Setup

Federated Setup. We simulate a federated system with 5 clients using a label-skewed non-IID data partitioning method based on clustering of language embeddings (Li et al., 2023). Each client receives a comparable number of samples. Federated fine-tuning is conducted on legal tasks using the OpenFedLLM framework (Ye et al., 2024b), with FedAVG as the aggregation method over 10 communication rounds. All clients adopt parameter-efficient fine-tuning (LoRA) and a shared prompt template. Hyperparameter settings and implementation details are detailed in Appendix K.

After federated fine-tuning, we evaluate the utility of the final global model on a held-out global test set. Following common practice, we compare it against a centrally trained (non-FL) baseline evaluated on the same test set. The results are reported in Table 8 in the Appendix.

Models and Metrics. We use Qwen1-8B (Bai et al., 2023) and Baichuan2-7B (Yang et al., 2023), both pre-trained primarily on Chinese corpora.[3] The model performance is evaluated with Coverage Rate(**CR**), Efficiency(**EF**), and Victim-exclusive Extracted PII(**VxPII**, defined as $|(S_v \setminus S_a) \cap S_E|$), defined to measure extraction accuracy and completeness.

Attack Strategies. We designate client 0 as the attacker and client 1 as the victim, and evaluate three strategies: (1) PII-contextual prefix sampling. The attacker builds a prefix set P_c from its local dataset D_0 with prefix length $\lambda = 50$. Each prefix queries the global model 15 times, generating up to $m = 10$ tokens per query—sufficient to recover most PIIs with manageable cost. (2) Frequency-prioritized sampling. Prefixes in $\text{Set}(\text{SUP}(P_c))$ are ranked by frequency to form $P_{f\geq 1}$ and used in descending order. Sweeping the prefix budget B varies the frequency threshold σ_a, enabling analysis of coverage–efficiency trade-offs. (3) Latent association fine-tuning. The attacker fine-tunes the global model (1 epoch, LR = 5e-5, LoRA: $r = 16$, $\alpha = 32$) using 10k frequent prefixes and 10k randomly sampled PIIs from its own data to reinforce prefix–PII associations. Further implementation details are provided in Appendix K.3.2.

Evaluation Protocol. To ensure fair evaluation, we define the set of victim-exclusive PIIs as $(S_v \setminus S_a)$ and apply two filters: (1) only PIIs $s_i \in S_v$ that do not appear in the attacker's training corpus ($s_i \notin U_a$) are retained; and (2) we enforce a longest common prefix constraint to eliminate interference between PIIs—e.g., those sharing prefixes—which may confuse the determination of which PII has been extracted (see Equation (1)):

$$\text{LCP}(s_i, s_j) = 0, \quad \forall s_i \neq s_j \in S_v$$

Metrics are computed on this filtered and prefix-disjoint set, as defined in Equations (2)–(3).

5.2 Main Results

RQ1: How effective is the PII extraction attack using contextual prefixes? We first evaluate the coverage rate (CR) and efficiency (EF) of our extraction attacks by querying federated fine-tuned LLMs using the PII-contextual prefix set P_c. Table 1 presents the results. With P_c, our attack achieves a considerable CR of 22.93% on Qwen1-8B and 28.95% on Baichuan2-7B.

[3]Both models are publicly available. See licenses on Huggingface pages: Qwen1 and Baichuan2.

Table 1: Summary of attack results using the PII-contextual prefix sampling method (with and without LAFt), where client 0 (attacker) targets client 1 (victim). Additional statistics used to compute CR and EF are available in Appendix Table 5.

Model	Prefix Set	CR	EF
	wo LAFt		
Qwen1-8B	P_c	22.93%	0.1910%
	$\text{Set}(\text{SUP}(P_c))$	56.20%	0.0110%
Baichuan2-7B	P_c	28.95%	0.2411%
	$\text{Set}(\text{SUP}(P_c))$	53.56%	0.0105%
	w LAFt		
Qwen1-8B	P_c	28.30%	0.2357%
	$\text{Set}(\text{SUP}(P_c))$	56.57%	0.0111%
Baichuan2-7B	P_c	28.46%	0.2370%
	$\text{Set}(\text{SUP}(P_c))$	52.16%	0.0102%

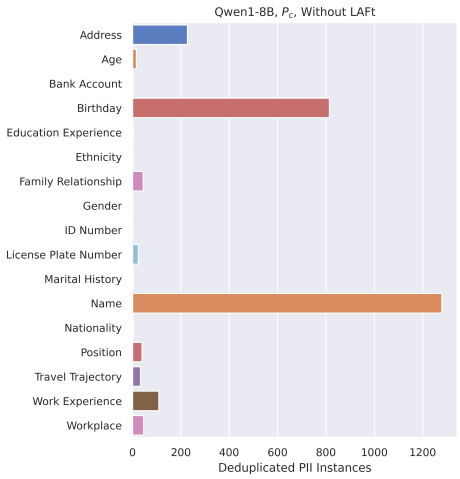

Figure 2: Label distribution of deduplicated victim-exclusive PII extracted by Qwen1-8B (without LAFt, using prefix set P_c). Results for Baichuan2-7B are shown in Appendix Figure 13.

To understand what types of PII are most vulnerable, we analyze the extracted instances. Figure 2 shows the label distribution of deduplicated victim-exclusive PII extracted by Qwen1-8B (without LAFt). The results for Baichuan2-7B are provided in Appendix Figure 13.

The most frequently extracted PII categories include "Address", "Birthday", and "Name", while others such as "Work Experience" and "Work Place" occur less often but remain notable. More complex types like "Medication Record" are not extracted at all. This is primarily due to the evaluation protocol, which only credits model outputs that match ground truth exactly. Complex PII often appears as long free-text spans, making verbatim

Figure 3: Coverage rate (CR) and efficiency (EF) under varying prefix budgets B for prefix sets P_c and $P_{f\geq 1}$. Prefix set $P_{f\geq 1}$ is frequency-sorted in descending order (see Section 4.2.2). Budget values are scaled exponentially (base 10); model used is Qwen1-8B.

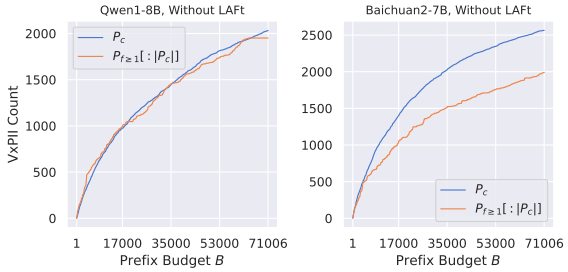

Figure 4: VxPII counts under varying prefix budgets (B) for prefix sets P_c and $P_{f\geq 1}$. Prefix set $P_{f\geq 1}$ is frequency-sorted in descending order (see Section 4.2.2) and truncated to match the size of P_c here.

reproduction difficult.

To estimate an upper bound of extraction capability, we evaluate with the generalized prefix set $\mathrm{Set}(\mathrm{SUP}(P_c))$, which includes all potential contextual prefixes. As shown in Table 1, expanding P_c to $\mathrm{Set}(\mathrm{SUP}(P_c))$ increases CR to 56.57% (Qwen1-8B) and 53.56% (Baichuan2-7B). However, this gain comes at a steep cost in efficiency—dropping EF to only 0.01%—indicating most queries yield redundant or irrelevant content.

We further investigate this CR–EF tradeoff in Figure 3, which illustrates how CR and EF vary with prefix budget B for prefix sets P_c and $P_{f\geq 1}$. As B increases, CR improves, but EF declines sharply. This suggests diminishing returns in efficiency when scaling up the number of queries to discover new PII instances.

RQ2: How effective is frequency-prioritized prefix sampling? As shown in Figure 4, frequency-prioritized (FP) sampling does not extract more VxPII instances than the contextual prefix set P_c, contrary to our hypothesis in Section 4.2.2. This result suggests that the contextual cues embedded in P_c are already strong indicators of LLM memorization, and that memorization cannot be inferred solely from co-occurrence frequency. Instead, it likely arises from more complex interactions between corpus semantics, model architecture, and pre-training dynamics.

Despite this, FP sampling captures highly distinct subsets of memorized PII. As shown in Figure 5(a), the Venn diagram comparison reveals that 49.9% of the VxPII extracted by FP sampling on Qwen1-8B and 65.02% on Baichuan2-7B are not discovered by the P_c method. This highlights FP sampling's complementary strength in uncovering diverse memorized content.

RQ3: How effective is Latent Association Fine-tuning? As shown in Table 1, applying Latent Association Fine-tuning (LAFt) significantly improves the CR of Qwen1-8B by 5.37%, raising it to 28.30%, and increases EF to 0.24%, indicating enhanced extraction performance. For Baichuan2-7B, LAFt does not yield a direct improvement in CR, but, as depicted in Figure 5(b), it facilitates the identification of additional distinct PII instances.

These results demonstrate that LAFt is an effective method for increasing the diversity of extracted PII, complementing the FP sampling approach. The extent of the improvement achieved by LAFt is influenced by the construction of the fine-tuning dataset D_{ft} and the choice of hyperparameters. In this study, we adopt a consistent setting by constructing D_{ft} through pairing frequent prefixes with randomly sampled PII and fine-tuning the model for one epoch to ensure a fair comparison. However, further exploration of personalized strategies—tailored to models with different architectures and pre-training conditions—could potentially yield better performance.

5.3 Cross-Client Evaluation of Extraction Robustness

To assess the robustness of our PII extraction method across different clients, we perform a cross-client evaluation where each client is iteratively designated as the attacker, while the remaining clients act as victims. This setup ensures that the extraction performance is not biased toward any particular client.

As shown in Table 2, our method achieves consistently high coverage rates across all attacker–victim pairings, demonstrating its generalizability and effectiveness in diverse settings.

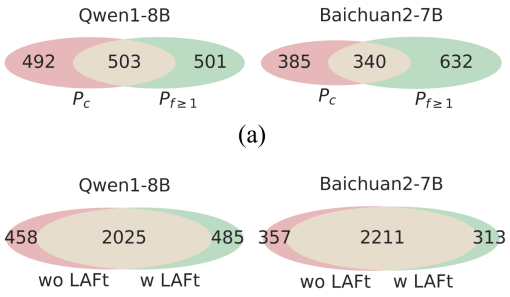

Figure 5: Venn diagrams showing overlap between VxPII sets extracted by different methods. (a) Comparison of VxPII sets using PII-contextual prefixes P_c vs. frequency-prioritized prefixes $P_{f\geq 1}$ at prefix budget $B = 10{,}000$ (without LAFt). (b) Comparison of VxPII sets extracted with and without LAFt on Qwen1-8B and Baichuan2-7B using the full P_c prefix set.

Table 2: Coverage rates (CR) of extraction attacks across different attacker–victim client pairs with a prefix budget $B = 10000$. Prefixes are randomly sampled from each attacker's corresponding set P_c. "–" indicates self-attack scenarios, which are not applicable.

Attacker ID	**Victim ID**				
	0	1	2	3	4
0	-	10.91%	12.89%	10.93%	11.88%
1	11.97%	-	12.41%	11.46%	12.35%
2	12.56%	11.39%	-	11.65%	12.74%
3	12.07%	10.82%	12.04%	-	11.99%
4	12.26%	11.36%	13.25%	11.21%	-

5.4 PII Sanitization Defense

We evaluate the effectiveness of a simple data sanitization strategy that masks PII using existing annotations. Each PII instance in the training data is replaced with a string of asterisks (*) of equal length. We then re-fine-tune FedLLM on this sanitized dataset and perform the PII-contextual Prefix Sampling attack once again. Table 3 compares the attack performance with and without the PII masking defense.

As shown in the results, the number of extracted VxPII is only slightly reduced. To understand this, we examine the document frequency of VxPII instances—that is, how often each appears in the training data. Figure 7 shows that PII masking significantly lowers the frequency of most VxPII, suggesting that our annotations effectively cover the majority of PII. Interestingly, some VxPII with zero document frequency—indicating they were absent from the masked dataset—were still extracted.

Based on these observations, we identify two main reasons for the limited effectiveness of the masking defense: (1) Pretraining data contamination: Our training data, sourced from publicly available legal documents, likely overlaps with the pretraining corpora of models like Qwen1-8B and Baichuan2-7B. (2) Incomplete PII labeling: Some PII instances may remain unlabeled—and therefore unmasked—in the training data. In the realistic scenarios, attackers can customize PII definitions, making exhaustive and comprehensive coverage inherently challenging.

Table 3: Attack performance with and without PII masking using the contextual prefix set P_c. The model is Qwen1-8B.

	VxPII	**CR**	**EF**
With PII Masking	2017	22.74%	0.1894%
Without Defense	2034	22.93%	0.1910%

Table 4: Comparison of VxPII sets between attacks on FedLLM and its un-fine-tuned base model.

Prefix Set	Model	$\lvert F \setminus B \rvert$	$\lvert B \setminus F \rvert$	$\lvert F \cap B \rvert$
P_c	Qwen1	682	518	1801
$P_{f\geq 1}$	Qwen1	554	308	4611
P_c	Baichuan2	407	405	2161

Pretraining data contamination is difficult to eliminate, as LLM providers rarely disclose their pretraining corpora. To mitigate this, we adopt a conservative strategy: we compare the VxPII extracted from FedLLM (F) with that from its un-fine-tuned base model (B), and subtract B to isolate PII memorized during federated fine-tuning. Table 4 shows that even after removing B, a substantial number of VxPII remain in $F \setminus B$, confirming memorization during fine-tuning. Figure 12 further shows that $F \setminus B$ exhibits a similar label distribution of VxPII as observed in Figure 2, supporting the validity of our method.

6 Conclusion

To investigate the privacy risks of data extraction attacks in realistic settings, we introduce a new class of attacks targeting FedLLMs. We extend a legal dataset with systematic PII annotations aligned with major privacy regulations, and evaluate attack performance using two key metrics: coverage rate and efficiency. Extensive experiments demonstrate that certain PII types are highly vulnerable, and our proposed methods can achieve substantial extraction performance. These findings highlight a critical privacy gap in FedLLMs and underscore the urgent need for stronger defense mechanisms in future federated learning systems.

7 Limitations

This work investigates the privacy risks of FedLLMs using a legal-domain dataset. Future research can extend our proposed methods to other sensitive domains such as healthcare and finance, where privacy concerns are equally critical. Additionally, there is a need for further exploration of defense mechanisms that can preserve the privacy of FedLLMs while maintaining their performance.

8 Ethic

This paper presents PII extraction attacks on federated fine-tuned LLMs to expose potential privacy risks. While designed for research and defense purposes, such methods could be misused to recover sensitive user data in real-world FL systems. We conduct all experiments on legal datasets with anonymized PII, and highlight the need for stronger safeguards in FedLLM deployments.

References

Benny Applebaum. 2017. *Garbled Circuits as Randomized Encodings of Functions: a Primer*, pages 1–44. Springer International Publishing, Cham.

Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 2938–2948. PMLR.

Jiamu Bai, Daoyuan Chen, Bingchen Qian, Liuyi Yao, and Yaliang Li. 2024. Federated fine-tuning of large language models under heterogeneous tasks and client resources. In *Advances in Neural Information Processing Systems*, volume 37, pages 14457–14483. Curran Associates, Inc.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. *Preprint*, arXiv:2012.07805.

Shan Chang, Ye Liu, Zhijian Lin, Hongzi Zhu, Bingzhu Zhu, and Cong Wang. 2024. Fedtrojan: Corrupting federated learning via zero-knowledge federated trojan attacks. In *2024 IEEE/ACM 32nd International Symposium on Quality of Service (IWQoS)*, pages 1–10.

Chaochao Chen, Xiaohua Feng, Yuyuan Li, Lingjuan Lyu, Jun Zhou, Xiaolin Zheng, and Jianwei Yin. 2024. Integration of Large Language Models and Federated Learning. *Patterns*, 5(12):101098.

Hong-Min Chu, Jonas Geiping, Liam H. Fowl, Micah Goldblum, and Tom Goldstein. 2023. Panning for gold in federated learning: Targeted text extraction under arbitrarily large-scale aggregation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. 2024. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12039–12050, Bangkok, Thailand. Association for Computational Linguistics.

Xingyi Duan, Baoxin Wang, Ziyue Wang, Wentao Ma, Yiming Cui, Dayong Wu, Shijin Wang, Ting Liu, Tianxiang Huo, Zhen Hu, Heng Wang, and Zhiyuan Liu. 2019. Cjrc: A reliable human-annotated benchmark dataset for chinese judicial reading comprehension. In *Chinese Computational Linguistics*, pages 439–451, Cham. Springer International Publishing.

European Union. 2016. General Data Protection Regulation (GDPR).

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients - how easy is it to break privacy in federated learning? In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Kuangpu Guo, Yuhe Ding, Jian Liang, Ran He, Zilei Wang, and Tieniu Tan. 2024. Exploring vacant classes in label-skewed federated learning. *Preprint*, arXiv:2401.02329.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Hongsheng Hu, Xuyun Zhang, Zoran Salcic, Lichao Sun, Kim-Kwang Raymond Choo, and Gillian Dobbie. 2024. Source inference attacks: Beyond membership inference attacks in federated learning. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3012–3029.

Jie Huang, Hanyin Shao, and Kevin Chen-Chuan Chang. 2022. Are large pre-trained language models leaking your personal information? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2038–2047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank J. Reddi, Sebastian U. Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic Controlled Averaging for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning,ICML*, Vienna, Austria (virtual conference).

Aly M. Kassem, Omar Mahmoud, Niloofar Mireshghallah, Hyunwoo Kim, Yulia Tsvetkov, Yejin Choi, Sherif Saad, and Santu Rana. 2025. Alpaca against vicuna: Using llms to uncover memorization of llms. *Preprint*, arXiv:2403.04801.

Siwon Kim, Sangdoo Yun, Hwaran Lee, Martin Gubri, Sungroh Yoon, and Seong Joon Oh. 2023. Propile: Probing privacy leakage in large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 20750–20762. Curran Associates, Inc.

Vladimir I. Levenshtein. 1965. Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Soviet physics. Doklady*, 10:707–710.

Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024a. Lexeval: A comprehensive chinese legal benchmark for evaluating large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 25061–25094. Curran Associates, Inc.

Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. 2023. From quantity to quality: Boosting LLM performance with self-guided data selection for instruction tuning. *CoRR*, abs/2308.12032.

Shenghui Li, Fanghua Ye, Meng Fang, Jiaxu Zhao, Yun-Hin Chan, Edith C. H. Ngai, and Thiemo Voigt. 2024b. Synergizing foundation models and federated learning: A survey. *Preprint*, arXiv:2406.12844.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Preprint*, arXiv:1812.06127.

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

H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2023. Communication-efficient learning of deep networks from decentralized data. *Preprint*, arXiv:1602.05629.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706.

Krishna Kanth Nakka, Ahmed Frikha, Ricardo Mendes, Xue Jiang, and Xuebing Zhou. 2024. Pii-scope: A benchmark for training data pii leakage assessment in llms. *Preprint*, arXiv:2410.06704.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Personal Data Protection Commission, Singapore. 2012. Personal Data Protection Act (PDPA).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Md Rafi Ur Rashid, Vishnu Asutosh Dasu, Kang Gu, Najrin Sultana, and Shagufta Mehnaz. 2025. Fltrojan: Privacy leakage attacks against federated language models through selective weight tampering. *Preprint*, arXiv:2310.16152.

Google Research. 2022. Training data extraction challenge.

Avi Schwarzschild, Zhili Feng, Pratyush Maini, Zachary C. Lipton, and J. Zico Kolter. 2024. Rethinking llm memorization through the lens of adversarial compression. In *Advances in Neural Information Processing Systems*, volume 37, pages 56244–56267. Curran Associates, Inc.

Hanyin Shao, Jie Huang, Shen Zheng, and Kevin Chang. 2024. Quantifying association capabilities of large language models and its implications on privacy leakage. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 814–825, St. Julian's, Malta. Association for Computational Linguistics.

Virat Shejwalkar and Amir Houmansadr. 2021. Manipulating the byzantine: Optimizing model poisoning attacks and defenses for federated learning. In *NDSS*.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. 2024. Detecting pretraining data from large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Standardization Administration of China (SAC). 2020. GB/T 35273-2020 | Information Security Technology: Personal Information Security Specification. Available at: https://openstd.samr.gov.cn/. Replaces GB/T 35273-2017, National Standard, Number: GB/T 35273—2020.

State of California, US. 2018. California Consumer Privacy Act (CCPA).

Maxim Tkachenko, Mikhail Malyuk, Andrey Holmanyuk, and Nikolai Liubimov. 2020-2025. Label Studio: Data labeling software. Open source software available from https://github.com/HumanSignal/label-studio.

Minh N. Vu, Truc Nguyen, Tre' R. Jeter, and My T. Thai. 2024. Analysis of privacy leakage in federated large language models. *Preprint*, arXiv:2403.04784.

Linlin Wang, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. 2024. Linkage on security, privacy and fairness in federated learning: New balances and new perspectives. *Preprint*, arXiv:2406.10884.

Yebo Wu, Chunlin Tian, Jingguang Li, He Sun, Kahou Tam, Li Li, and Chengzhong Xu. 2025. A survey on federated fine-tuning of large language models. *Preprint*, arXiv:2503.12016.

Xianghua Xie, Chen Hu, Hanchi Ren, and Jingjing Deng. 2024. A survey on vulnerability of federated learning: A learning algorithm perspective. *Neurocomputing*, 573:127225.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, and 36 others. 2023. Baichuan 2: Open large-scale language models. *Preprint*, arXiv:2309.10305.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2).

Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan Rossi, Ang Li, Lina Yao, Julian McAuley, Yiran Chen, and Carlee Joe-Wong. 2024. Federated large language models: Current progress and future directions. *Preprint*, arXiv:2409.15723.

Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Yaxin Du, Yang Liu, Yanfeng Wang, and Siheng Chen. 2024a. Fedllm-bench: Realistic benchmarks for federated learning of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 111106–111130. Curran Associates, Inc.

Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. 2024b. Openfedllm: Training Large Language Models on Decentralized Private Data Via Federated Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, page 6137–6147, New York, NY, USA. Association for Computing Machinery.

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. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 40306–40320. PMLR.

Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, and Zhongyu Wei. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *Preprint*, arXiv:2309.11325.

Shengbin Yue, Shujun Liu, Yuxuan Zhou, Chenchen Shen, Siyuan Wang, Yao Xiao, Bingxuan Li, Yun Song, Xiaoyu Shen, Wei Chen, and 1 others. 2024. Lawllm: Intelligent legal system with legal reasoning and verifiable retrieval. In *International Conference on Database Systems for Advanced Applications*, pages 304–321. Springer.

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

Zhuo Zhang, Xiangjing Hu, Jingyuan Zhang, Yating Zhang, Hui Wang, Lizhen Qu, and Zenglin Xu. 2023. FEDLEGAL: The first real-world federated learning benchmark for legal NLP. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3492–3507, Toronto, Canada. Association for Computational Linguistics.

Haoxi Zhong, Chaojun Xiao, Cunchao Tu, Tianyang Zhang, Zhiyuan Liu, and Maosong Sun. 2020. Jec-qa: A legal-domain question answering dataset. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):9701–9708.

Zhenhong Zhou, Jiuyang Xiang, Chaomeng Chen, and Sen Su. 2024. Quantifying and analyzing entity-level memorization in large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(17):19741–19749.

Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA. Curran Associates Inc.

A Acronyms List

- **GDPR** - General Data Protection Regulation (European Union, 2016)

- **CCPA** - California Consumer Privacy Act (State of California, US, 2018)
- **CPIS** - Chinese Information Security Technology: Personal Information Security Specification (GB/T 35273-2020) (Standardization Administration of China (SAC), 2020)
- **Singapore PDPC** - Personal Data Protection Commission (Singapore) (Personal Data Protection Commission, Singapore, 2012)
- **Non-IID** - Non-independent and identically distributed

B Preliminary Knowledge

B.1 Data Extraction Attack

Early research on training data extraction attacks has broadly categorized them into untargeted and targeted attacks (Research, 2022; Yu et al., 2023). Untargeted extraction aims to recover any memorized training samples without specifying a target (Lukas et al., 2023), whereas targeted extraction attempts to reconstruct specific training samples, often by providing a known prefix and recovering the remaining content (Carlini et al., 2021). The latter type, often referred to as Verbatim Data Extraction, has become a standard approach for evaluating memorization in LLMs (Carlini et al., 2023; Dong et al., 2024) and for detecting potential data contamination (Dong et al., 2024). We briefly outline the core methodology of verbatim data extraction below.

Given an LLM θ and a training dataset X, each training sample $x_i \in X$ is partitioned into two segments: a prefix a_i and a suffix b_i, such that $x_i = a_i b_i$. The model is then prompted with a_i to generate a completion g_i of the same length as b_i. If g_i exactly matches b_i, the sample is considered successfully extracted.

In practice, model outputs may not exactly replicate the original suffix but can still be lexically close. To accommodate this, a similarity-based metric such as Edit Distance (Levenshtein, 1965) is often employed. A sample is deemed extracted if the similarity score between g_i and b_i exceeds a predefined threshold t. By computing this similarity-based extraction score across all samples in a dataset D, one can quantify the model's memorization behavior or assess its vulnerability to training data extraction attacks.

B.2 Federated Learning

Federated Learning (FL) is a solution to address data isolation issues (Yang et al., 2019), where a central server and multiple clients collaborate to complete the training process. A key feature of FL is that the training datasets are stored locally on each client and remain invisible to other clients. FL is commonly used in industrial scenarios where each client represents an independent organization, such as hospitals collaborating to train a medical model without combining their datasets due to legal restrictions or business competition. Federated Learning enables the training of stronger models compared to training on data from a single client alone.

Given c clients and their private datasets $D_1, D_2, \ldots, D_c$, the federated learning process aims to learn a global model θ by solving the following optimization problem:

$$\theta^* = \arg\min_{\theta} \frac{1}{c} \sum_{i=1}^{c} \mathcal{L}(D_i, \theta)$$

To solve this problem, many federated optimization algorithms have been proposed, such as FedAVG (McMahan et al., 2023) and FedProx (Li et al., 2020). Typically, these algorithms consist of two alternating phases: local updating and central aggregation. In the local updating phase, each client independently optimizes the global model using its own dataset. In the central aggregation phase, the server aggregates the models from the clients using an aggregation algorithm, obtaining a global model, which is then sent back to each client for the next round of local updating. A typical procedure of federated learning is illustrated in Algorithm 1.

B.3 Preliminary Assessment of Verbatim Data Extraction Risks in FedLLMs

To examine the memorization behavior of federated fine-tuned large language models (FedLLMs) and evaluate their potential risks of leaking sensitive information, we conduct a preliminary experiment simulating a *verbatim data extraction* (VDE) attack. The results are referenced in the main paper (Section 1) to empirically motivate our study.

We adapt the experimental setup from (Dong et al., 2024) to the federated setting, where an attacker is assumed to possess prefix fragments of the training data from all participating clients and attempts to recover the subsequent suffix tokens. For each training sample, we extract a prefix from

the original sequence and query the trained model to generate a continuation. The generated suffix is compared against the ground truth using **Edit Distance** (ED) (Levenshtein, 1965), where a lower ED indicates stronger memorization. Specifically:

- ED = 0 indicates the model has perfectly memorized and reproduced the suffix;
- ED values are capped at 50, as we restrict suffixes to a maximum of 50 tokens.

We perform the attack on the global models aggregated after 10 rounds of federated training. To ensure a comprehensive assessment, we consider three popular FL algorithms—**FedAvg**, **FedProx**, and **Scaffold**—each under both **IID** and **Non-IID** data distributions. Two baseline settings are also included:

- **Centralized**: All client data is pooled and the model is fine-tuned in a conventional non-federated manner;
- **Untrained**: The base model is evaluated without any fine-tuning.

Figure 6 summarizes the results across five downstream tasks. Our key observations are:

- FedLLMs consistently exhibit higher ED scores (i.e., lower memorization) than centralized models, suggesting that the FL aggregation process reduces susceptibility to verbatim extraction.
- However, compared to untrained models, FedLLMs still show non-negligible memorization, with noticeably lower ED scores, indicating partial leakage of training data.

These findings highlight a trade-off between collaborative model training and privacy preservation, and they serve as the motivation for our in-depth investigation of privacy risks in FedLLMs.

C Federated Learning Framework

Algorithm 1 outlines a general framework for Federated Learning (FL), where a central server coordinates multiple clients to collaboratively train a global model without sharing local data. At each round, the server distributes the current model to all clients, each of which performs local updates based on its private data and sends the updated parameters back. The server then aggregates the received updates to produce a new global model.

Algorithm 1 A Federated Learning Framework

Input: Clients set $\mathcal{C} = \{c_1, c_2, \ldots, c_c\}$ with local datasets $D_1, D_2, \ldots, D_c$; total FL rounds R; initial global model θ_0; server aggregation function f_{agg}; client loss function $\mathcal{L}$
Output: Learned global model θ_R

ServerExecute:
for round $r = 1$ to R **do**
  for each client $c_i \in \mathcal{C}$ (in parallel) **do**
    $\theta_r^i \leftarrow \text{CLIENTUPDATE}(c_i, \theta_{r-1})$
  end for
  $\theta_r \leftarrow f_{\text{agg}}(\{\theta_r^i | c_i \in \mathcal{C}\})$
end for

ClientExecute:
function $\text{CLIENTUPDATE}(c_i, \theta_{r-1})$
  $\theta_r^i \leftarrow \arg\min_{\theta} \mathcal{L}(\theta_{r-1}, D_i)$
  return θ_r^i
end function

D Full Table for PII-contextual Prefix Attack Results

Table 5 provides the complete results corresponding to Table 1, showing detailed data across all settings.

E Task Descriptions and Examples

1. **Judicial Summarization (Sum)**: The task of judicial summarization aims to extract key information from court judgments and generate concise summaries. The input to this task is a legal document, and the output is a summary of its content. The performance of this task is evaluated using the Rouge-L metric, which effectively measures the similarity between the generated and reference texts based on the longest common subsequence (LCS). Rouge-L is a widely used metric in text generation tasks. In this study, we adopt Rouge-L because it captures both semantic and structural similarities between texts, making it suitable for summarizing judicial documents.

2. **Judicial Reading Comprehension (RC)**: This task focuses on answering legal questions based on court documents to evaluate the model's reading comprehension ability. The input consists of a piece of legal material and a question, and the task requires answering the question based on the content of the material. The performance metric for this task is Rouge-L.

3. **Similar Case Matching (Match)**: In this task,

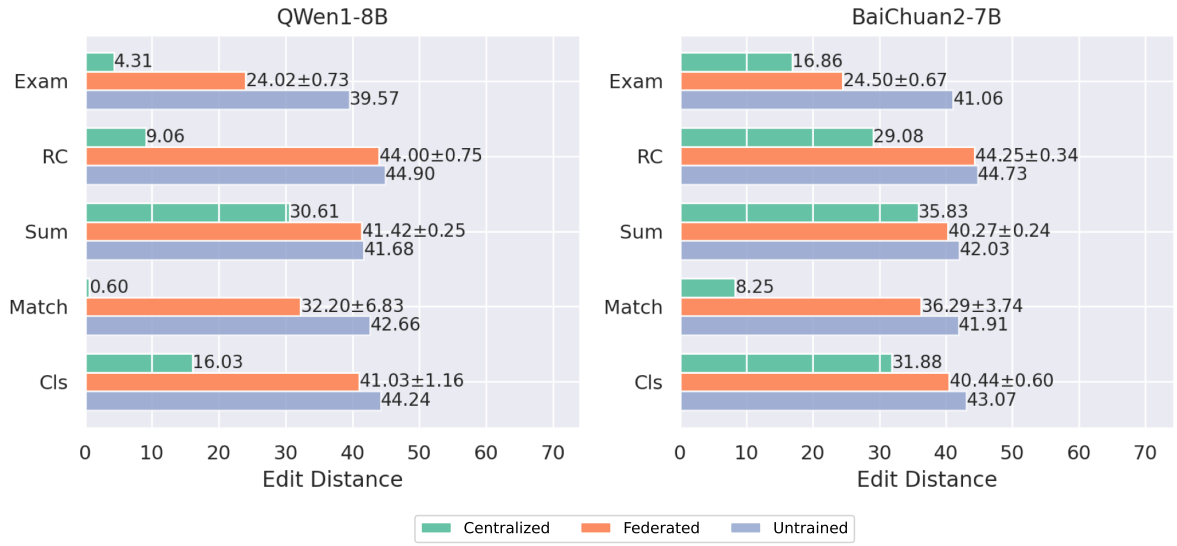

Figure 6: Edit Distance results of verbatim data extraction attacks after 10 training rounds. We evaluate six federated configurations (FedAvg, FedProx, Scaffold × IID/Non-IID) and report the mean and standard deviation. Lower values indicate stronger memorization. Centralized and untrained models serve as baselines.

Table 5: Detailed attack performance of client 0 (attacker) targeting client 1 (victim) under various settings. The victim-exclusive set $(S_v \setminus S_a)$ includes 8,870 unique PII items.

Model	Prefix Set	LAFt	CR	EF	VxPII Count	Prefix Set Size
Qwen1-8B	P_c	without	22.93%	0.1910%	2034	71006
	P_c	with	28.30%	0.2357%	2510	71006
	$\text{Set}(\text{SUP}(P_c))$	without	56.20%	0.0110%	4985	3013161
	$\text{Set}(\text{SUP}(P_c))$	with	56.57%	0.0111%	5018	3013161
Baichuan2-7B	P_c	without	28.95%	0.2411%	2568	71006
	P_c	with	28.46%	0.2370%	2524	71006
	$\text{Set}(\text{SUP}(P_c))$	without	53.56%	0.0105%	4751	3013161
	$\text{Set}(\text{SUP}(P_c))$	with	52.16%	0.0102%	4627	3013161

the input includes three case documents, and the model is required to determine which of the latter two documents is more similar to the first one. The model selects the most similar document by computing the similarity between the first case and each of the other two. The evaluation metric for this task is accuracy.

4. **Judicial Exam (Exam)**: This task simulates multiple-choice questions from legal examinations to assess the model's knowledge of legal concepts. Given a judicial exam question with multiple options, the model is expected to choose the correct answer. The performance is evaluated using accuracy.

5. **Legal Case Classification (Cls)**: This task requires the model to classify the cause of action in a case, assisting legal retrieval systems in automatically categorizing case types. The input is a description of the case facts, and the model is required to output the corresponding case category. The performance metric is accuracy.

F Data Preprocessing

Previous works (Zhang et al., 2023; Yue et al., 2023) have used these datasets for LLM and FedLLM research. In this work, we use the processed datasets from these prior studies and further curate the data for our experiments. We applied the following preprocessing steps to prepare the datasets:

Deduplication and Cleansing. To ensure the quality of our data, we remove duplicate samples with logically equivalent meanings. For example, in the RC tasks, some samples only differ in the order of two legal cases. We also clean out samples

Table 6: Input and Output Examples for Each Task

Task	Input	Output
Judicial Summarization (SUM)	First-instance civil judgment on inheritance dispute between Han and Su Shenyang Dadong District People's Court Plaintiff: Han, female, born June 6, 1927, Han ethnicity... ... Clerk: Li Dan	Summary: This case involves an inheritance dispute between the plaintiff and the defendant. The plaintiff requests...
Judicial Reading Comprehension (RC)	Case: Upon trial, it was found that on February 11, 2014, the plaintiff... Question: When did the plaintiff and defendant agree on the travel plan?	The plaintiff and defendant agreed on the travel plan on February 11, 2014.
Similar Case Matching (Match)	Determine whether Case A is more similar to Case B or Case C. A: Plaintiff: Zhou Henghai, male, born October 17, 1951... B: Plaintiff: Huang Weiguo, male, Han ethnicity, resident of Zhoushan City... C: Plaintiff: Zhang Huaibin, male, resident of Suzhou City, Anhui Province, Han ethnicity...	B
Judicial Exam (Exam)	Wu was lawfully pursued by A and B... Which of the following analyses is correct? A. If Wu missed both A and B, and the bullet... B. If Wu hit A, resulting in A's death... C. If Wu hit both A and B, causing A's death and B's serious injury... D. If Wu hit both A and B, causing both to die...	A
Legal Case Classification (Cls)	Legal document: Plaintiff Yan Qiang submitted the following claims to this court:...	Private Loan Dispute

containing garbled characters or large segments with a mixture of multiple languages.

Unifying Prompt Template and Instruction Reshaping. Some tasks, such as Exam, contain instructions that appear in different parts of the sample (either at the beginning or the end). To standardize the format, we reshape the data so that the instruction always appears at the beginning, followed by the legal document. Additionally, we employ hierarchical hyper markers such as "<Case A>", "<Case B>", and "<Answer>" to clearly segment the prompt, making the structure more transparent for the LLM.

G Supplementary Dataset Statistics and Analysis

Table 7 summarizes the basic statistics of the five datasets used in our experiments. Each dataset corresponds to a different downstream task for fine-tuning the model.

Figure 7 presents the document frequency distribution of the 2017 VxPII instances extracted from the model trained on the masked dataset (see Section 5.4). Most VxPII exhibit low frequency, indicating that PII masking significantly reduces memorization.

Table 7: Dataset Statistics

	Exam	RC	SUM	Match	Cls
#Samples	2399	3500	2651	3848	4196

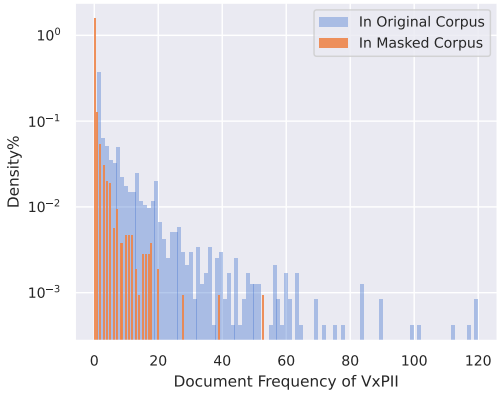

Figure 7: Document frequency distribution of the 2017 VxPII instances extracted from the model trained on the masked dataset.

H Prompt Template and Utility Fine-tuning Results for FedLLMs

Figure 8 shows the unified prompt template used for all federated utility fine-tuning tasks. Table 8 reports the evaluation results across multiple tasks, comparing different federated learning algorithms and base models.

```
Below is a task related to judicial and legal
    matters. Output an appropriately completed
    response to the request.

<### Input>
{{Task Input}}

<### Output>
{{Task Output}}
```

Figure 8: Unified Utility Fine-tuning Template for All Tasks.

Table 8: Utility Performance over Different Tasks.

FL Algorithms	Models	SUM(rouge-l)	RC(rouge-l)	Match(Acc)	Exame(Acc)	Cls(Acc)
FedAvg	Qwen1-8B	50.0	14.2	50.0	37.5	90.0
FedAvg	Baichuan2-7B	57.6	42.4	50.0	33.3	89.5
Non-FL	Qwen1-8B	50.0	18.9	50.0	40.8	87.0

I Machine Annotation Standards for PII Labeling

This section provides details on the machine annotation protocol we use to identify Personally Identifiable Information (PII) in our dataset. Table 9 defines our categorization schema, which includes seven major categories and their corresponding fine-grained subtypes. To ensure annotation consistency and scalability, we utilize a templated prompting approach for automated PII labeling. Figure 9 shows the machine annotation prompt used to instruct the LLM annotator. The prompt dynamically incorporates category definitions and format constraints to standardize the output.

```
I would like you to assist in reviewing the provided
     document and labeling all sections containing
    {{Major Categories of PII}} according to the
    following requirements.

1. **Types of personal information to identify
    include:**
   {{PII Subcategories}}
2. **Output format:**
   {{Output Format Description}}
3. **Input instructions:**
   {{Input Format Description}}

Please provide the output directly in accordance
    with the format requirements above, without any
     additional explanation or comments. Thank you
    for your assistance!
```

Figure 9: PII Machine Annotation Prompt Template

J Details of Human Evaluation for PII Annotation

To validate the quality of the machine-generated PII annotations, we recruited four Chinese-speaking students with foundational knowledge of Chinese law to manually annotate PII on a selected subset of the dataset. Prior to annotation, all annotators underwent thorough training on the annotation guidelines and usage of the Label Studio tool. The instructions provided to annotators are detailed in Figure 10, while Figure 11 illustrates the annotation interface used. All annotators were fairly compensated upon completion of their tasks.

The human evaluation results, reported in terms of precision, recall, and F1 score, are summarized in Table 10, indicating high agreement both in exact span matching and in combined span-and-label matching, confirming the reliability of the machine annotations.

K Experiment Implementation Details

K.1 Federated Dataset Partitioning.

We use the preprocessed and labeled datasets (see Section 3) for our experiments, splitting the data into training and testing sets. In the federated learning setup, we simulate a system with five clients. The testing set remains global, while the training set is heterogeneously partitioned across the clients using a balanced Non-IID distribution (see Acronyms List A). To achieve this, We employ a clustering-based method (Li et al., 2023) for partitioning, where a language encoder first generates embeddings, which are then clustered using K-means. Finally, a Dirac distribution with $\alpha = 0.5$ is applied to create a label-skewed partitioning (Guo et al., 2024), ensuring each client receives a comparable number of samples.

K.2 Hardware and Computation Budget

All experiments are conducted on a single NVIDIA A6000 GPU with 48 GB of memory, using bfloat16 precision. Most sampling-based attack experiments are completed within 200 GPU hours.

K.3 Experiment Procedure

K.3.1 Federated Utility Fine-Tuning

We begin by performing federated fine-tuning of the LLM(Zhang et al., 2023; Wu et al., 2025) on the partitioned dataset, adapting it to the legal tasks. The fine-tuning is conducted using the OpenFedLLM framework (Ye et al., 2024b). We set the total number of FL rounds to 10 and use FedAVG as the aggregation algorithm.

Each client performs multi-task fine-tuning by mixing all local tasks and applying a unified prompt template, as illustrated in Figure 8, following the approach in Raffel et al. (2023). In each round of federated learning, the client fine-tunes the received global model for one epoch using parameter-

Table 9: Categorization of Personally Identifiable Information (PII) Types in Our Labeling Standards

Major Category	**Minor Category**
Personal Basic Information	Name, Birthday, Address, Gender, Ethnicity, Family Relationship, Age, Nationality, Personal Phone Number
Personal Identity Information	ID Number, Social Security Number, Driver's License Number, Employee Number, License Plate Number
Health Related Information	Physical Condition, Fertility Information, Current Medical History, Diagnosis and Treatment Status, Other Medication Record
Work and Education Information	Workplace, Position, Work Experience, Education Experience, Grades
Personal Property Information	Bank Account, Amount of Funds, Fund Flow Records, Virtual Assets, Other Financial Records
Personal Location Information	Precise Location, Accommodation Information, Travel Trajectory
Others	Marital History, Religious or Philosophical Beliefs, Sexual Orientation or Sex Life, Unpublished Criminal Records

```
# **PII Annotation Guidelines for Labelers**
## **1. Task Objective**
**Core Task**: Proofread legal texts to accurately identify and annotate **Personally Identifiable
    Information (PII)**. Each annotation task includes:
1. **Localization**: Mark the exact character offsets of each PII instance in the text;
2. **Categorization**: Assign each PII instance to the appropriate **major category (7 total)** and **minor
    category (36 total)**, ensuring precise classification.
## **2. PII Category System**
| Major Category | Minor Categories |
| - | - |
| Personal Basic Information | Name, Birthday, Address, Gender, Ethnicity, Family Relationship, Age,
    Nationality, Personal Phone Number |
...(omitted)...
## **3. Annotation Workflow and Standards**
### **Step-by-Step Process**
1. **Read the Full Text**: Understand the context to detect all potential PII entities;
2. **Sentence-by-Sentence Annotation**: For each PII instance, annotate its **start position**, **text span
    **, and corresponding **major + minor category**;
3. **Special Cases**: For ambiguous expressions (e.g., "a certain district of a certain city"), determine
    PII status based on contextual clues.
### **Annotation Guidelines**
* **Accuracy**: Ensure all annotated content is verifiably present in the text. Avoid false positives or
    over-labeling;
* **Support Channel**: If any uncertain cases arise during annotation, promptly reach out to the *Annotation
     Support Team* for clarification.
```

Figure 10: Markdown-style guideline for PII annotation, covering task objectives, taxonomy, and labeling procedures.

Table 10: Human Evaluation of PII Labeling Quality

Evaluation Criteria	**Precision (P)**	**Recall (R)**	**F1 Score (F1)**
Identical Span Only	0.89	0.93	0.91
Identical Span and Label	0.89	0.90	0.89

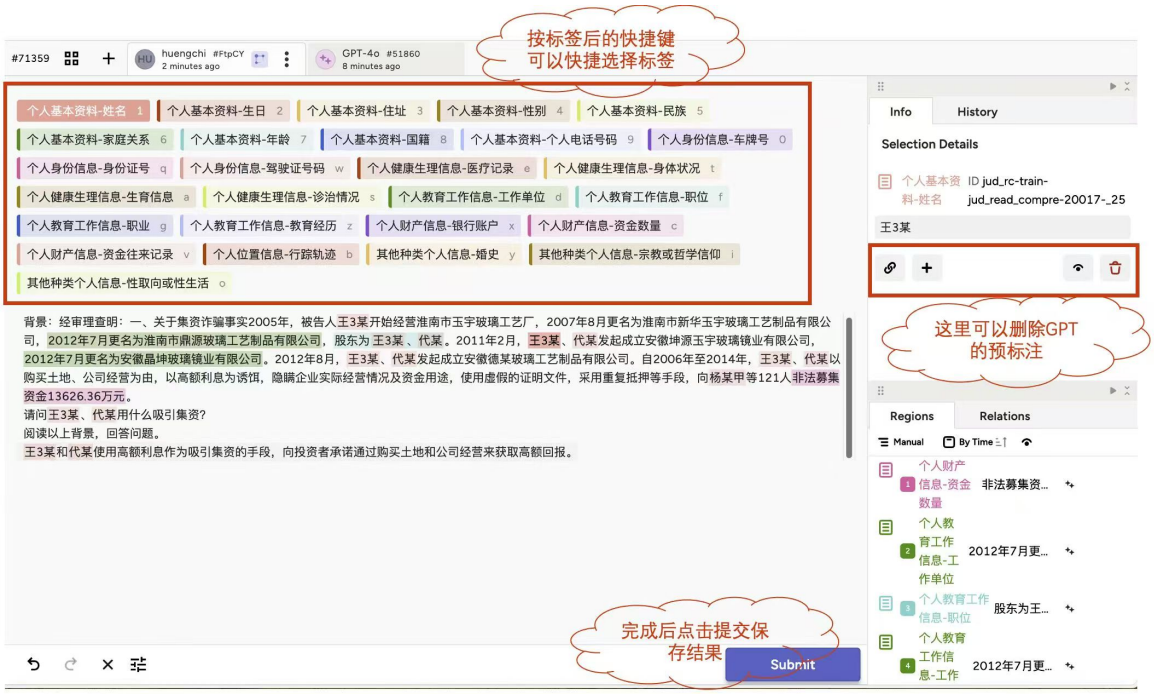

Figure 11: Human annotation interface in the Label Studio tool for PII labeling. Annotators are familiar with the Label Studio environment and are instructed to label PII spans based on predefined PII categories. Machine-generated labels are provided as references to assist the human annotators.

efficient fine-tuning (PEFT) techniques of LoRA (Hu et al., 2021). The learning rate is set to 3e-4 with a linear decay schedule. The maximum input sequence length is 3072 tokens. We use a batch size of 1 and apply gradient accumulation with a factor of 8. The LoRA configuration is set to $r = 16$ and $\alpha = 32$.

After federated fine-tuning is complete, we evaluate the utility performance of the final global model on a held-out global test set. In line with standard practices in federated learning research, we also compare this performance with that of a centrally (non-FL) trained model on the same test set. The results are summarized in Table 8.

K.3.2 PII Extraction

In the main experiments, we designate client 0 as the attacker and client 1 as the victim. We construct the prefix set P_c for PII-contextual prefix sampling from the local dataset D_0. During this construction, we set the length parameter λ to 50. Each prefix is used to independently query the utility fine-tuned global model $n = 15$ times. For each query, the model is allowed to generate up to $m = 10$ new tokens. This generation length is sufficient to cover most labeled PII instances while keeping the computational cost acceptable.

For Frequency-Prioritized Prefix Sampling, we construct $\mathrm{Set}(\mathrm{SUP}(P_c))$ from the aforementioned P_c, and sort it in descending order of prefix frequency (as described in Section 4.2.1). The model θ is then queried using prefixes in this frequency-descending order. Although we do not explicitly define a frequency threshold σ_a, we sweep the prefix budget B exponentially in base 10. Because $\mathrm{Set}(\mathrm{SUP}(P_c))$ is sorted by decreasing frequency, this sweep over B implicitly corresponds to sweeping σ_a from $+\infty$ to 1.

K.3.3 Latent Association Fine-tuning

To construct the fine-tuning dataset D_{ft}, we select the top 10000 most frequent prefixes from $\mathrm{Set}(\mathrm{SUP}(P_c))$ and randomly sample 10000 PII instances from the attacker's (client 0's) PII set S_a. Although alternative strategies could be explored for prefix and PII selection, this approach is relatively straightforward and effective. We then fine-tune the model θ to obtain θ' using one epoch and a small learning rate of 5e-5. LoRA is applied with $r = 16$ and $\alpha = 32$, consistent with the initial federated fine-tuning setup.

L Additional PII Label Distribution Results

Figure 12 illustrates the label distribution of FedLLM-exclusive victim PII extracted by Qwen1-

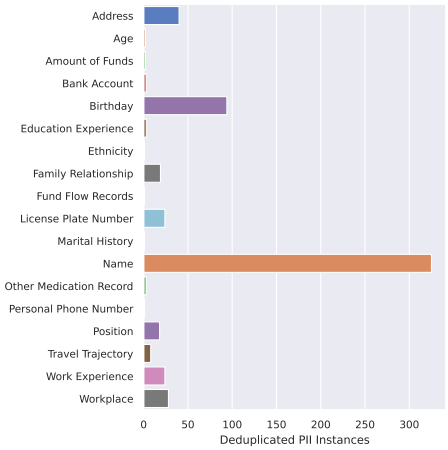

Figure 12: Label distribution of FedLLM-exclusive Vx-PII extracted using prefix set P_c and the Qwen1-8B model.

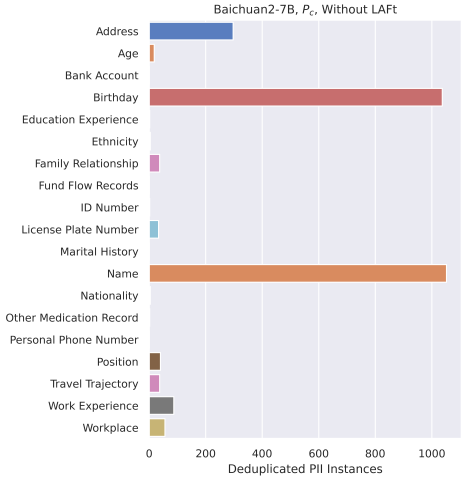

Figure 13: Label distribution of deduplicated victim-exclusive PII instances extracted by the Baichuan2-7B model (without LAFt, using prefix set P_c). This figure complements Figure 2 in the main text, which presents the corresponding results for Qwen1-8B.

8B. This result corresponds to the experiment described in Section 5.4.

Figure 13 presents the label distribution of deduplicated victim-exclusive PII instances extracted by the Baichuan2-7B model.