Retaliatory Attacks Against Federated Unlearning via Data Leakage

Xinyi Sheng, Wei Bao, Hequn Wang, Yuqin Liu, and Sen Fu

School of Computer Science, The University of Sydney, Australia
xinyi.sheng@sydney.edu.au, wei.bao@sydney.edu.au, hwan0565@uni.sydney.edu.au, yliu0720@uni.sydney.edu.au, sen.fu@sydney.edu.au

Abstract

Federated unlearning (FU) allows a participating client in a federated learning (FL) system to remove its contribution from the trained global model, thereby enforcing the client's "right to be forgotten" (RTBF). However, from the perspective of a client that does not request unlearning, the activation of the FU process may disrupt ongoing FL training and introduce additional computational and time overhead. In such cases, a client opposed to unlearning may be incentivized to retaliate against the unlearning client(s). In this work, we take the first step toward demonstrating the feasibility of such retaliatory behavior by exploiting the information leakage introduced during the FU process. Specifically, we propose a novel unlearning-induced membership inference attack (MIA) model, followed by a coarse-to-fine data generation method that enables an adversarial client to locally reconstruct the unlearned data. Building on this reconstruction, we introduce two targeted retaliatory attacks: (1) Anti-Unlearning Attack (AUA), which hinders the global model from successfully forgetting the data intended for removal, and (2) Discrimination-Unlearning Attack (DUA), which specifically degrades the global model's performance on the unlearned data. Extensive experiments across a variety of FU methods and settings validate the effectiveness of the proposed retaliatory attack framework.

Introduction

Recent data protection regulations, including the European Union's General Data Protection Regulation GDPR (Voigt and von dem Bussche 2017) and California Consumer Privacy Act CCPA (Harding et al. 2019), have emphasized the need to support the "right to be forgotten" (RTBF) for personal data used in training machine learning (ML) models. To address this need, machine unlearning (MU) techniques have been developed to enable the removal of specific data samples from trained ML models in centralized settings (Cao and Yang 2015; Bourtoule et al. 2021; Wang et al. 2024). Federated unlearning (FU) (Liu et al. 2023) extends this concept to federated learning (FL), a decentralized learning framework where multiple clients collaboratively train a shared global model while keeping their local data private (McMahan et al. 2017). In FU, participating clients can submit unlearning requests to remove their contributions from the shared model, thereby exercising their RTBF.

While most existing efforts on FU have focused on improving the efficiency of the unlearning process (Liu et al. 2022; Halimi et al. 2022; Liu et al. 2022; Zhang et al. 2023a; Che et al. 2023), enhancing the utility of the global model after unlearning (Liu et al. 2021; Wu, Zhu, and Mitra 2023; Xiong et al. 2023), and ensuring certified removal of the unlearned data (Wang et al. 2022; Wu, Zhu, and Mitra 2023; Gao et al. 2024), very little attention has been paid to the potential security vulnerabilities introduced by the FU mechanism itself (Wang, Li, and Li 2023). Although a few recent studies have begun to examine security issues within the FU framework (Sheng, Bao, and Ge 2024; Wang et al. 2025), they primarily focus on adversarial threats directly arising from malicious unlearning requests.[1] In contrast to these works, we take the first step to investigate a more fundamental security concern in FU: the data privacy leakage that can occur during the unlearning process.

To demonstrate that such data privacy leakage constitutes a realistic and practical threat, we introduce a novel class of attacks termed *retaliatory attacks*, which are launched by an FL participant who opposes the unlearning process. This scenario is both intuitive and plausible in practice. From the perspective of a client who does not request unlearning, the activation of the FU process may disrupt the ongoing federated training, incur additional computational and time overhead, and potentially degrade the performance of the global model after unlearning. These consequences can serve as incentives for a dissatisfied client to retaliate against the one who initiated the unlearning request.

Specifically, we propose two targeted retaliatory attacks: (1) *Anti-Unlearning Attack (AUA)*, which aims to prevent the global model from successfully forgetting the private data that was requested to be unlearned; and (2) *Discrimination-Unlearning Attack (DUA)*, which seeks to intentionally degrade the global model's performance on the data and the underlying distribution associated with the unlearned client(s). The core idea behind these attacks is that an adversarial client can exploit the data leakage introduced during

[1] Due to space limitation, a more comprehensive discussion of related work, including studies that are less directly aligned with our work, is provided in the Appendix A (Sheng et al. 2025).

the FU procedure to reconstruct the unlearned data originally contributed by the unlearning client(s). To reconstruct the unlearned data, one class of ideal attacks is the gradient inversion attack (Zhu, Liu, and Han 2019), which aims to recover private data from uploaded gradients. However, such attacks typically assume a compromised or curious server with direct access to client gradients (Zhang et al. 2023b; Lamri et al. 2025; Zhang et al. 2025), which is not feasible in our setting where the adversary is merely a normal client. Thus, we propose to leverage an alternative privacy-based attack, membership inference attack (MIA), to reconstruct the unlearned data. MIA involves training a set of attack models to determine whether a given sample was part of a target model's training set (Shokri et al. 2017; Yeom et al. 2018). However, in the privacy-preserving setting of FL, an adversarial client does not have access to the local model of the unlearned client and therefore cannot apply conventional MIAs on the target model. To address this challenge, inspired by (Chen et al. 2021), we propose an innovative unlearning-induced MIA that exploits the discrepancy between the global model before and after unlearning to indirectly infer the membership status of a given data sample.

Given that the unlearning-induced MIA model (attack model) functions only as a discriminative classifier, we design a coarse-to-fine data generation pipeline to reconstruct the unlearned samples. Specifically, we begin by generating coarse candidate samples using the predictions of the attack model. To reduce false positives (i.e., samples incorrectly identified as unlearned), we introduce a novel cross-model filtering mechanism that permutes the input posteriors of the attack models to mitigate the dominance of any single posterior in the final prediction. To further improve reconstruction quality, we apply a sample-level refinement to each coarse candidate, aiming to increase the attack model's confidence in classifying it as unlearned data and to enhance the overall diversity of the generated samples. Finally, AUA is executed by forcing the global model to relearn the reconstructed unlearned data, while DUA is performed by injecting targeted poisoning based on the reconstructed samples.

In conclusion, our work makes following contributions:

- We introduce a novel class of retaliatory attacks, initiated by an FL participant who opposes the unlearning process and seeks to retaliate against the client(s) requesting unlearning by exploiting the data leakage during FU.
- We develop an innovative unlearning-induced MIA model alongside a coarse-to-fine data generation pipeline to reconstruct the unlearned data of the unlearned client(s), which serves as the foundation for executing two targeted retaliatory attacks: AUA and DUA.
- We thoroughly evaluate the proposed retaliatory attacks across a range of existing FU methods and settings, revealing a consistent and realistic privacy vulnerability introduced by the FU process.

Problem Formulation

Federated Learning and Unlearning

We consider a typical FL scenario in which a set of n clients, denoted as $\mathcal{K} = \{k_1, k_2, \ldots, k_n\}$, collaboratively train a global model $\mathcal{M}$ via a central server. Each client k_i ($i \in \{1, 2, \ldots, n\}$) maintains a local dataset $\mathcal{D}_i$, and the entire training dataset is denoted as $\mathcal{D} = \bigcup_{i=1}^{n} \mathcal{D}_i$. The FL training process can then be formalized as a function $FL(\mathcal{D}) \rightarrow \mathcal{M}$, consisting of two key steps: (1) *Local training*, where each client k_i trains a local model $\mathcal{M}_i$ on its dataset $\mathcal{D}_i$ and uploads it to the server; and (2) *Global aggregation*, where the server aggregates all local models (e.g., using aggregation rules such as FedAvg (McMahan et al. 2017)) and distributes the updated global model back to the clients.

Then, during the FL training, a subset of clients $\mathcal{K}^u \subset \mathcal{K}$ may submit an unlearning request to remove the contribution of their local datasets from the trained global model. Upon receiving the request, the server interrupts the standard FL process and initiates the FU process. Let $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$ denote the global model before the FU process begins and after it finishes, respectively. The FU process can then be represented as a function $FU(\mathcal{M}_{\text{before}}, \mathcal{D}^u, \mathcal{D}^r, \mathcal{I}) \rightarrow \mathcal{M}_{\text{after}}$, where $\mathcal{D}^u = \bigcup_{k_i \in \mathcal{K}^u} \mathcal{D}_i$ denotes the set of local datasets to be removed (i.e., from the unlearned clients), $\mathcal{D}^r = \mathcal{D} \setminus \mathcal{D}^u$ denotes the remaining datasets belonging to the remaining clients $\mathcal{K}^r = \mathcal{K} \setminus \mathcal{K}^u$, and $\mathcal{I}$ denotes any additional information required to perform unlearning (e.g., historical checkpoints or intermediate states). Finally, when the FU process completes, the FL training resumes for all clients in $\mathcal{K}^r$, continuing from the unlearned model $\mathcal{M}_{\text{after}}$.

Threat and Adversary Model

In this study, we consider a threat model where a participating client in the FL system, although not issuing any unlearning request itself, is opposed to the unlearning mechanism and thus acts as an adversary by retaliating against those clients who request to be unlearned. Given that the adversary's ultimate goal is to retaliate against unlearning clients, it may adopt various specific attack strategies to achieve this objective. Specifically, we propose two such retaliatory attacks, namely: (1) *Anti-Unlearning Attack* (AUA), which aims to prevent the global model from successfully forgetting the private data intended to be unlearned, thereby undermining the privacy guarantees (i.e., RTBF) promised to the unlearned clients; and (2) *Discrimination-Unlearning Attack* (DUA), which aims to intentionally degrade the global model's performance on both the data and underlying distribution associated with the unlearned clients, thereby causing the model to systematically discriminate against their data.

We then consider a strictly privacy-preserving FL and FU environment, where the adversary does not have access to any additional information (e.g., external datasets, private data, or uploaded gradients from other clients), except for what is naturally available to a participating client in the FL or FU process (e.g., its own local data, the FL model architecture, and the global models distributed by the server). In addition, we assume that the adversary cannot interfere with the global aggregation or the FU process, and has no knowledge of the specific FU algorithm adopted by the server. The only assumption we make is that the adversary can observe the global model before ($\mathcal{M}_{\text{before}}$) and after ($\mathcal{M}_{\text{after}}$) the FU process. This assumption is reasonable, as the execution of

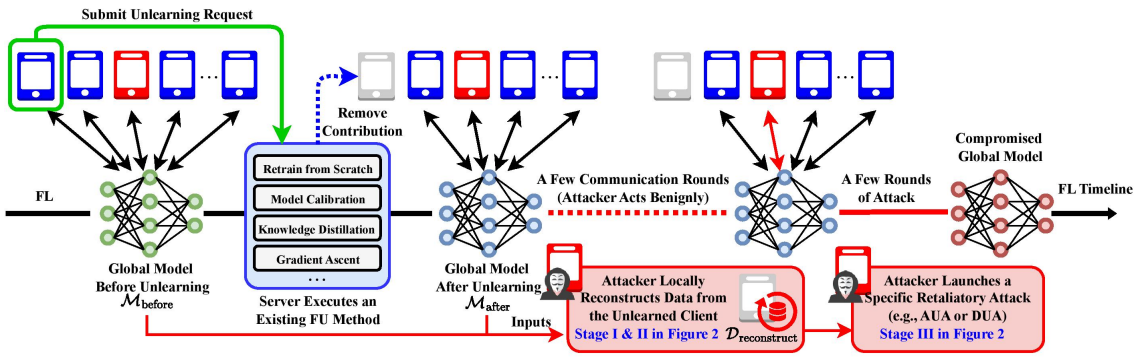


Figure 1: Overview and timeline of the proposed retaliatory attack on federated unlearning.

FU often introduces observable interruptions to the standard FL process, and in many FU implementations, the server must explicitly notify all participating clients that unlearning is being performed. Even in the absence of such explicit signals, we show in our Appendix C (Sheng et al. 2025) that an adversarial client can still reliably detect the occurrence of FU and estimate the post-unlearning global model by monitoring consistency patterns in global model updates. (See Appendix B (Sheng et al. 2025) for a summary of notation.)

Method

Attack Overview

The core of the proposed retaliatory attacks lies in the additional information inadvertently leaked during the unlearning process. In a conventional privacy-preserving FL system, it is infeasible for any single (adversarial) client to accurately identify or reconstruct private data belonging to other participating clients. However, we demonstrate that this becomes feasible when an adversarial client exploits the model discrepancies introduced by the unlearning process.

Figure 1 illustrates the overall workflow and timeline of the proposed retaliatory attack. The adversarial client behaves benignly throughout both the FL and FU phases, as the attack specifically targets the unlearning process and is therefore triggered only after the adversary receives the global model following unlearning. The attack proceeds in three stages. In Stage I, the adversarial client trains a set of unlearning-induced MIA models $\mathcal{A}$, each of which exploits the discrepancies between the global model before ($\mathcal{M}_{\text{before}}$) and after ($\mathcal{M}_{\text{after}}$) the unlearning process. These models enable the adversary to determine whether a given data sample belongs to the unlearned client(s), other participating clients, or to none of them. In Stage II, a coarse-to-fine data generation process is employed to reconstruct the unlearned data $\mathcal{D}_{\text{reconstruct}}$, leveraging the predictions of $\mathcal{A}$. Stages I and II are both carried out locally on the adversarial client's device and may span several rounds of global aggregation. During this period (illustrated by the red dotted line in Figure 1), the adversary continues to behave benignly and does not interfere with the training of the global model. Stage III begins once the reconstruction is complete, during which the adversarial client launches a specific retaliatory attack (e.g., AUA or DUA) against the global model using $\mathcal{D}_{\text{reconstruct}}$, which can rapidly compromise the global model within a few rounds of aggregation (illustrated by the red solid line in Figure 1). The detailed procedures of each attack stage are elaborated in the following sections.

Unlearning-Induced Membership Inference

In Stage I of the proposed retaliatory attack, the adversarial client locally trains a set of MIA models to determine the membership status of a given data sample during the FL and FU processes. Specifically, these models aim to identify whether a sample was part of the training data ($\mathcal{D}^u$) of the unlearned client(s). Traditional MIA pipelines (Shokri et al. 2017) typically require access to the target model trained on the data, which in this context corresponds to the local model of the unlearned client(s). However, such access is infeasible for the adversarial client in a typical FL setting due to its decentralized nature. To address this limitation, inspired by (Chen et al. 2021), we propose leveraging the discrepancies between the global models before and after FU. These differences implicitly capture the influence of the unlearned data, enabling membership inference without requiring direct access to the unlearned clients' local models. The detailed procedure for establishing these attack models is illustrated in Stage I of Figure 2 and described as follows.

Shadowing FU Processes. To conduct the unlearning-induced MIA, the adversarial client first shadows the entire FU process locally, simulating the global unlearning procedure. This requires access to a shadow dataset $\mathcal{D}_{\text{shadow}}$ that resembles the original training data $\mathcal{D}$ used by the global model. Since our work primarily targets tabular datasets, $\mathcal{D}_{\text{shadow}}$ can be constructed by generating high-confidence samples from $\mathcal{M}_{\text{before}}$ using some search-based approaches (e.g., the hill-climbing algorithm proposed in (Shokri et al. 2017)). Then, in each shadow process $s \in \{1, 2, ..., S\}$ (where S denotes the total number of shadowing processes), the shadow dataset $\mathcal{D}_{\text{shadow}}$ is randomly partitioned into three disjoint subsets: $\mathcal{D}^s_{\text{external}}$, representing samples unused in both the FL and FU processes; $\mathcal{D}^s_{\text{unlearned}}$, representing samples used to train $\mathcal{M}_{\text{before}}$ but excluded from training $\mathcal{M}_{\text{after}}$; and $\mathcal{D}^s_{\text{retained}}$, used in training both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$. To better mimic the FU process, $\mathcal{D}^s_{\text{unlearned}}$ and $\mathcal{D}^s_{\text{retained}}$ are further split and assigned to a group of simulated unlearned and

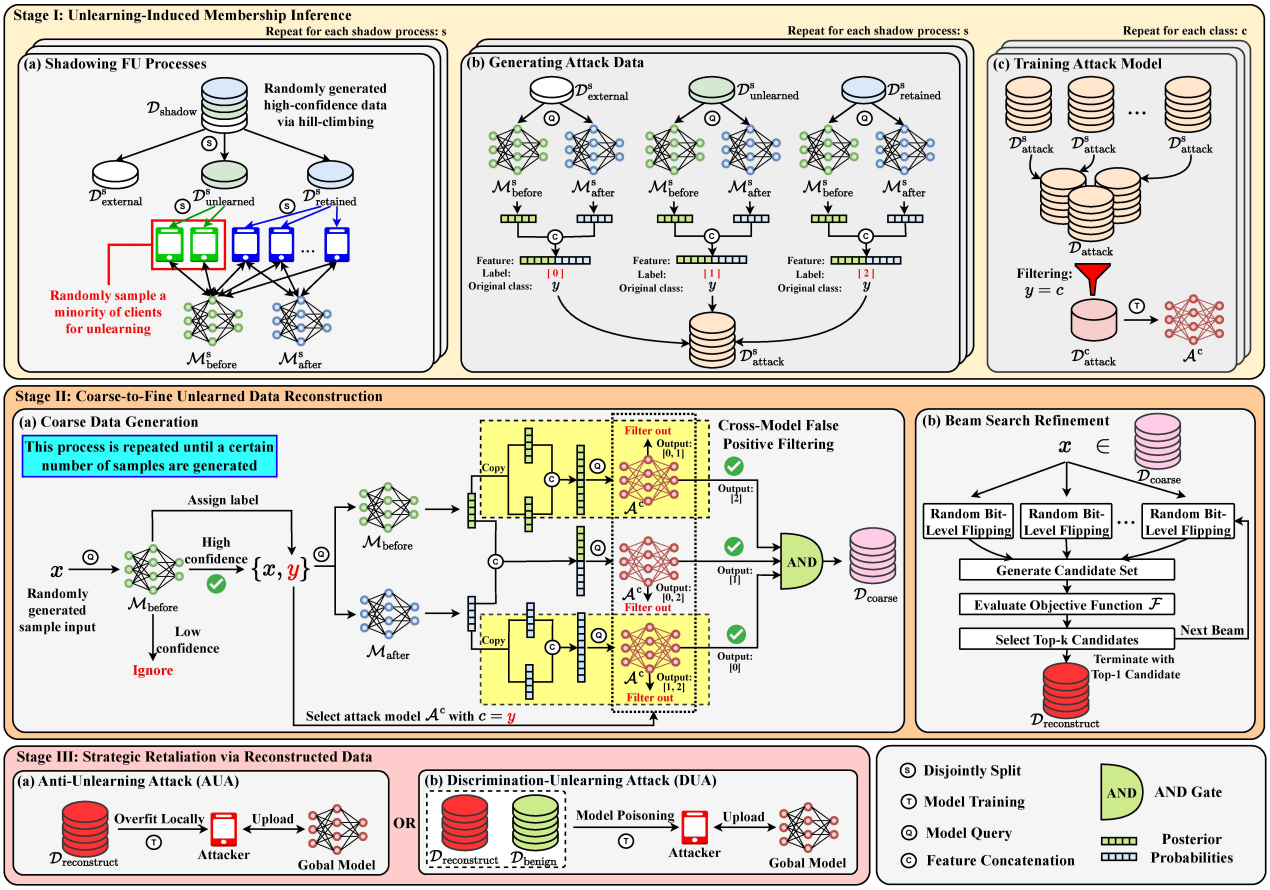


Figure 2: Detailed illustration of the proposed retaliatory attack on federated unlearning.

retained clients, respectively, where the unlearned clients are randomly sampled and constitute a minority among all simulated clients. Based on this client setup, the adversary can emulate both the FL and FU processes locally, thus obtaining the shadowed global models before and after unlearning, denoted as $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$, respectively.

Generating Attack Data. To prepare the attack data for training the attack models, the adversary queries the trained models $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$ using samples from $\mathcal{D}^s_{\text{external}}$, $\mathcal{D}^s_{\text{unlearned}}$, and $\mathcal{D}^s_{\text{retained}}$, respectively. Let $\mathbb{P}^s_{\text{before}}$ and $\mathbb{P}^s_{\text{after}}$ denote the posterior probabilities obtained by querying $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$, respectively. The input features of the attack data are constructed by concatenating the two posteriors, i.e., $\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}$. A label of 0, 1, or 2 is then assigned to the attack data generated from $\mathcal{D}^s_{\text{external}}$, $\mathcal{D}^s_{\text{unlearned}}$, and $\mathcal{D}^s_{\text{retained}}$, respectively, indicating their exact membership status during the shadowed FU process. These form the set of attack data $\mathcal{D}^s_{\text{attack}}$ generated from the shadowing process s. (Note that the original class label y of each queried data sample is also recorded for subsequent partitioning.)

Training Attack Models. After completing all S shadowing processes, the overall attack dataset is assembled as $\mathcal{D}_{\text{attack}} = \bigcup_{s=1}^{S} \mathcal{D}^s_{\text{attack}}$. For each class $c \in \{1, 2, \ldots, C\}$ (where C denotes the total number of classes), a separate attack model $\mathcal{A}^c$ is trained using a class-specific subset $\mathcal{D}^c_{\text{attack}} \subset \mathcal{D}_{\text{attack}}$, which contains all samples with class label $y = c$. These models form the final set of attack models $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$ (the unlearning-induced MIA models).

Coarse-to-Fine Unlearned Data Reconstruction

While the set of attack models $\mathcal{A}$ obtained in Stage I can effectively determine the membership status of a given data sample, it serves only as a discriminative classifier. A generative method is still necessary to enable the reconstruction of the unlearned data. Therefore, in Stage II of the proposed attack framework, we introduce a novel coarse-to-fine data generation method to recover the unlearned data from the FU process, as illustrated in Stage II of Figure 2.

Coarse Data Generation. The coarse data generation begins by randomly initializing the input feature vector x, with each attribute uniformly sampled within its domain. The resulting x is used to query the global model before unlearning ($\mathcal{M}_{\text{before}}$) to obtain its posterior probabilities $\mathbb{P}_{\text{before}}$. The confidence score is then calculated as the maximum value in $\mathbb{P}_{\text{before}}$, i.e., $\max(\mathbb{P}_{\text{before}})$, and the predicted label is assigned as $y = \arg\max(\mathbb{P}_{\text{before}})$. A generated sample is retained if its confidence score exceeds a predefined threshold; otherwise, it is discarded. This filtering step ensures that each retained sample has a reliable class label, which is essential for selecting the corresponding attack model $\mathcal{A}^c$ with $c = y$.

While these randomly generated high-confidence samples roughly capture the overall training data distribution of $\mathcal{M}_{\text{before}}$, they may not align with the data distribution of the unlearned client(s). To further narrow the candidate set toward the unlearned data, each sample x is queried on both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$ to obtain the corresponding posterior probability vectors $\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}$, which are then concatenated and fed into the corresponding attack model $\mathcal{A}^c$. The sample is retained only if $\mathcal{A}^c$ classifies it as class 1, indicating that it is inferred to belong to the unlearned data; otherwise, it is discarded.

Cross-Model False Positive Filtering. Although the set of attack models $\mathcal{A}$ provides valuable guidance for recovering the unlearned data, its predictions are still significantly affected by false positives (i.e., samples incorrectly classified as unlearned). Inspired by (Carlini et al. 2021), which proposes filtering out false positives by comparing predictions with those from a second model trained on a disjoint dataset when extracting training data from a large language model, we propose an alternative yet novel cross-model false positive filtering strategy that fully leverages the same attack model via permutation of input posteriors, as highlighted by the yellow-shaded regions in Stage II(a) of Figure 2. Specifically, in addition to the original input—formed by concatenating the posteriors $\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}$ and fed into the attack model $\mathcal{A}^c$ with class 1 as the desired output—we generate two auxiliary inputs: one by duplicating $\mathbb{P}_{\text{before}}$ (i.e., $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{before}}$) and the other by duplicating $\mathbb{P}_{\text{after}}$ (i.e., $\mathbb{P}_{\text{after}}||\mathbb{P}_{\text{after}}$). These are also fed into $\mathcal{A}^c$, with the expected outputs being class 2 and class 0, respectively.

The key insight here is that, given our attack model $\mathcal{A}^c$ actually considers two models ($\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$) as target models, each of the two posteriors fed into $\mathcal{A}^c$ encodes distinct training-related information for an unlearned data sample: the first indicates that the sample was seen (i.e., included in the training data) by the first target model ($\mathcal{M}_{\text{before}}$), while the second reflects that the sample was absent from the training data of the second target model ($\mathcal{M}_{\text{after}}$). In this case, when the input is $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{before}}$, it represents a sample that is seen by both target models, and thus should be assigned a membership status of 2. Conversely, when the input is $\mathbb{P}_{\text{after}}||\mathbb{P}_{\text{after}}$, it indicates that the sample is absent from the training sets of both models and should therefore correspond to membership status 0. This strategy effectively filters out false positives in the initial screening stage, where the prediction of $\mathcal{A}^c$ may be dominated by either $\mathbb{P}_{\text{before}}$ or $\mathbb{P}_{\text{after}}$, leading to the incorrect classification of non-unlearned samples as class 1. Finally, the sample is retained and added to $\mathcal{D}_{\text{coarse}}$ only when all three conditions are satisfied.

Beam Search Refinement. While $\mathcal{D}_{\text{coarse}}$ provides an initial set of candidate samples that approximate the distribution of the unlearned data, we further introduce a sample-level refinement procedure to enhance the fidelity of these samples with respect to the original unlearned data. The key idea is to increase the confidence of the attack model $\mathcal{A}^c$ in classifying a sample as class 1, i.e., to maximize $\mathbb{P}_{\text{attack}}[1]$, where $\mathbb{P}_{\text{attack}}$ denotes the posterior obtained from querying $\mathcal{A}^c$. In addition, we incorporate a diversity penalty to discourage the refined samples from collapsing into similar patterns, thereby promoting sample-level variability. Furthermore, the confidence score $\mathbb{P}_{\text{before}}[c]$ of $\mathcal{M}_{\text{before}}$ is encouraged to remain high, as it directly determines the selection of the corresponding attack model $\mathcal{A}^c$. Formally, our objective function $\mathcal{F}$ is defined as:

$$\mathcal{F}(x) = \alpha \cdot (\mathbb{P}_{\text{before}}[c]) + \beta \cdot (\mathbb{P}_{\text{attack}}[1]) - \gamma \cdot \big(1 - \min_{x' \in \tilde{\mathcal{D}}_{\text{reconstruct}}} \text{dist}(x, x')\big), \quad (1)$$

where α, β, and γ are weighting coefficients; $\text{dist}(\cdot,\cdot)$ denotes the distance function between two data samples (we adopt the Jaccard distance in our implementation); and $\tilde{\mathcal{D}}_{\text{reconstruct}}$ represents the set of samples that have already been refined. Given the black-box and non-differentiable nature of $\mathcal{F}(x)$, we employ a heuristic beam search algorithm that iteratively refines candidate samples via random bit-level flipping in the input space, as illustrated in Stage II(b) of Figure 2. Finally, after refining all samples in $\mathcal{D}_{\text{coarse}}$, we obtain $\mathcal{D}_{\text{reconstruct}}$ as the final reconstructed set of unlearned data, thus concluding Stage II of the attack.

Strategic Retaliation via Reconstructed Data

With $\mathcal{D}_{\text{reconstruct}}$ obtained, the adversarial client can now proceed to launch a targeted retaliatory attack, either AUA or DUA, against the global model.

Anti-Unlearning Attack. Recall that the objective of AUA is to prevent the global model from successfully forgetting the training data of the unlearned client(s). As illustrated in Stage III(a) of Figure 2, this can be achieved by having the adversarial client repeatedly upload poisoned local model updates that are deliberately overfitted to $\mathcal{D}_{\text{reconstruct}}$ (e.g., by using an increased number of local epochs and a reduced learning rate). This strategy effectively forces the global model to relearn the reconstructed unlearned data within just a few communication rounds.

Discrimination-Unlearning Attack. DUA aims to introduce discrimination against the unlearning process by intentionally degrading the performance of the global model on the distribution corresponding to the unlearned client(s). As illustrated in Stage III(b) of Figure 2, the adversarial client achieves this by conducting a model poisoning attack that leverages both $\mathcal{D}_{\text{reconstruct}}$ and its own local benign dataset $\mathcal{D}_{\text{benign}}$. Specifically, during local training, the adversarial client optimizes the following objective:

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{benign}}) - \lambda \cdot \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{reconstruct}}), \quad (2)$$

where $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss, and λ controls the strength of the negative gradient signal derived from the unlearned data. By inverting the loss gradient on $\mathcal{D}_{\text{reconstruct}}$, the adversary forces the model to perform poorly on the reconstructed unlearned samples, while maintaining nominal performance on its own benign data. (See Appendix D (Sheng et al. 2025) for the full Stage I–III procedure of the proposed retaliatory attack.)

Experiments

Experimental Setup

Datasets. To evaluate the proposed retaliatory attacks, we conduct experiments on two tabular datasets that are widely

Dataset	Metric	Federated Unlearning Method									
		Retrain		FedEraser		KD-based FU		SGA-based FU		RobustFU	
		Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack
Location	MIA	0.502	0.629 (↑0.127)	0.505	0.648 (↑0.143)	0.509	0.725 (↑0.216)	0.514	0.713 (↑0.199)	0.505	0.601 (↑0.096)
	UA	0.606	0.950 (↑0.344)	0.585	0.981 (↑0.396)	0.568	0.991 (↑0.423)	0.610	0.984 (↑0.374)	0.608	0.977 (↑0.369)
Purchase100	MIA	0.499	0.597 (↑0.098)	0.502	0.609 (↑0.107)	0.504	0.627 (↑0.123)	0.511	0.590 (↑0.079)	0.501	0.573 (↑0.072)
	UA	0.578	0.975 (↑0.397)	0.621	0.997 (↑0.376)	0.566	0.998 (↑0.432)	0.628	0.966 (↑0.338)	0.616	0.980 (↑0.364)

Table 1: Attack performance of the proposed AUA across different federated unlearning methods.

Dataset	Metric	Federated Unlearning Method									
		Retrain		FedEraser		KD-based FU		SGA-based FU		RobustFU	
		Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack
Location	UA	0.606	0.544 (↓0.062)	0.585	0.530 (↓0.055)	0.568	0.523 (↓0.045)	0.610	0.547 (↓0.063)	0.608	0.551 (↓0.057)
	TA	0.611	0.598 (↓0.013)	0.601	0.584 (↓0.017)	0.597	0.582 (↓0.015)	0.615	0.597 (↓0.018)	0.621	0.603 (↓0.018)
Purchase100	UA	0.578	0.497 (↓0.081)	0.621	0.494 (↓0.127)	0.566	0.513 (↓0.053)	0.628	0.526 (↓0.102)	0.616	0.542 (↓0.074)
	TA	0.637	0.619 (↓0.018)	0.635	0.610 (↓0.025)	0.629	0.606 (↓0.023)	0.630	0.617 (↓0.013)	0.639	0.605 (↓0.034)

Table 2: Attack performance of the proposed DUA across different federated unlearning methods.

used in the data privacy literature: Location (Yang et al. 2015) and Purchase (Shokri et al. 2017). For both datasets, we follow the same preprocessing procedure as in (Shokri et al. 2017). The resulting Location dataset consists of 30 geosocial classes, with each sample represented by 446 binary features indicating whether a user has visited specific regions or location types. We use the Purchase100 version of the Purchase dataset, which includes 100 purchase styles, with each sample represented by 600 binary features indicating whether a user purchased a specific product.

While the proposed retaliatory attacks are primarily tailored for tabular classification tasks, we further demonstrate that the privacy leakage exploited in Stage I of our framework is also applicable to image classification tasks. To validate this, we evaluate such unlearning-induced data leakage on two additional image datasets: CIFAR-10 (Krizhevsky, Hinton et al. 2009) and SVHN (Netzer et al. 2011).

Evaluation Metrics. To evaluate the effectiveness of AUA, we adopt the standard **MIA accuracy (MIA)** (Shokri et al. 2017), which is widely used as a certified test for assessing whether the unlearned data has been successfully forgotten (Wang et al. 2022; Halimi et al. 2022; Sheng, Bao, and Ge 2024). In addition, we measure the **Unlearned Accuracy (UA)**, the prediction accuracy of the global model on the unlearned dataset, where a higher UA indicates that the unlearned data has not been effectively forgotten. For DUA, we similarly report UA to quantify the performance degradation on the unlearned dataset. To ensure a fair evaluation, we also include the **Test Accuracy (TA)** on a held-out test set, which reflects the global model's performance on normal, non-unlearned data. For each of these metrics, we report both the values before (i.e., immediately after the FU process) and after the attack, thereby highlighting the extent to which AUA and DUA compromise the global model. Furthermore, since Stage I of our proposed retaliatory framework also involves a novel **unlearning-induced MIA (U-MIA)**, which infers membership status based on discrepancies between the global model before and after unlearning, we report its performance as a direct measure of privacy leakage caused by the FU process. For both the standard MIA and U-MIA, higher attack accuracy indicates greater privacy leakage and weaker unlearning guarantees.

FU Methods. We consider several state-of-the-art FU methods that the server may adopt, including:

- **FedEraser** (Liu et al. 2021), which leverages stored historical parameter updates from participating clients and incorporates a calibration mechanism during retraining to efficiently reconstruct the unlearned model.
- **SGA-based FU** (Wu et al. 2022), which integrates elastic weight consolidation (EWC) with reverse stochastic gradient ascent (SGA) to enable effective FU.
- **KD-based FU** (Wu, Zhu, and Mitra 2023), which achieves FU by subtracting accumulated historical updates from the trained global model and restoring its performance through knowledge distillation (KD).
- **RobustFU** (Sheng, Bao, and Ge 2024), which performs robust FU by reintroducing high-information-gain samples into the remaining clients during retraining.

We also evaluate the retraining-from-scratch golden baseline (**Retrain**), where the unlearned model is retrained on the remaining clients using FedAvg (McMahan et al. 2017).

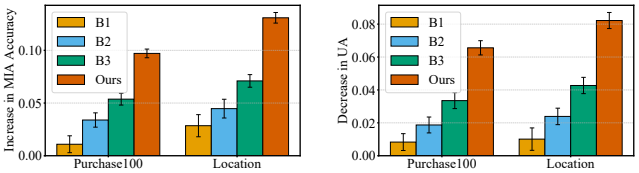
Implementation Details. Please refer to our Appendix E (Sheng et al. 2025) for implementation and training details.

Experimental Results

Overall Attack Performance. Table 1 and Table 2 present the attack performance of the proposed AUA and DUA, respectively. For AUA, the results show that the MIA accuracy is close to 0.5 across all evaluated FU methods prior to the attack, suggesting that the unlearned global model (i.e., before being attacked) has effectively forgotten the data requested for unlearning. However, after launching the AUA, the MIA accuracy on the unlearned data increases substantially, indicating that the compromised global model has re-

Dataset	Federated Unlearning Method (U-MIA)			
	Retrain	FedEraser	KD-based FU	SGA-based FU
Location	0.709	0.688	0.657	0.672
Purchase100	0.804	0.786	0.763	0.781
CIFAR-10	0.654	0.630	0.625	0.634
SVHN	0.693	0.679	0.640	0.661

Table 3: Evaluation of privacy leakage in FU methods.



(a) AUA (b) DUA

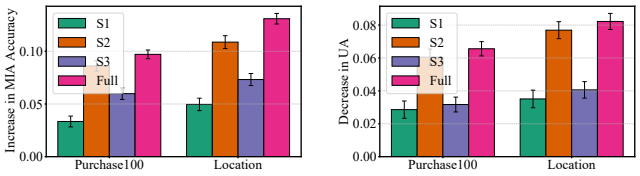
Figure 3: Comparison between our method and baselines.

learned information that was intended to be removed. Similarly, a significant increase in the prediction accuracy on the unlearned data is observed after the attack, further confirming that the global model fails to preserve the forgetting effect and continues to memorize the unlearned samples.

For DUA, a substantial drop in prediction accuracy on the unlearned data is observed after the attack, indicating that the compromised global model has successfully introduced additional discrimination against the unlearned data. While a slight decrease is also noted in the prediction accuracy on the test data, this drop is relatively minor compared to that on the unlearned data, suggesting that the attack remains primarily targeted at the distribution of the unlearned data.

Privacy Leakage Evaluation. Since the effectiveness of the proposed retaliatory attacks hinges on the privacy leakage exposed during the FU process, the set of unlearning-induced MIA models ($\mathcal{A}$) proposed in Stage I of our attack provides a direct and quantifiable measure of this leakage, where a higher U-MIA accuracy suggests a greater degree of privacy leakage. We evaluate this leakage across two tabular and two image datasets under various FU methods (For image classification tasks, we assume the adversarial client maintains a local shadow set to emulate the FU process.). As shown in Table 3, the proposed unlearning-induced MIA models consistently achieve high attack accuracy (U-MIA) across all settings, thereby highlighting the vulnerability of existing FU mechanisms to privacy leakage.

Baseline Comparison. To further demonstrate the effectiveness of the proposed attack, we compare it with several baseline methods. We first consider a naive membership inference baseline (denoted as **B1**), which randomly generates data samples and retains those on which the global model before unlearning yields high confidence. We then propose an improved baseline (**B2**), which additionally incorporates the global model after unlearning. This baseline is based on the intuition that the confidence of a model should decrease on samples that have been removed through unlearning. Specifically, B2 generates candidate samples and retains those that receive high confidence from the pre-unlearning



(a) AUA (b) DUA

Figure 4: Ablation studies of key design components.

model but significantly lower confidence from the post-unlearning model. Moreover, we adopt a more advanced MIA-based baseline method (**B3**), which trains two separate MIA models (Shokri et al. 2017) for the global models before and after unlearning, respectively. A randomly generated candidate sample is considered as an unlearned training sample if it is classified as a member by the MIA model corresponding to the pre-unlearning global model, but classified as a non-member by the MIA model corresponding to the post-unlearning global model.

Figure 3 demonstrates the performance gain (i.e., the increase in MIA accuracy for AUA and the decrease in UA for DUA) for each of the baseline methods and our proposed method on both the Location and Purchase100 datasets (using Retrain as the FU method). It can be seen that our method achieves better performance than all baselines, demonstrating its superior ability to reconstruct the unlearned data and conduct more effective retaliatory attacks.

Ablation Studies. We further conduct ablation studies to evaluate the contribution of each key component in our coarse-to-fine data generation pipeline. Specifically, we examine several settings: (**S1**) using only a single prediction from the attack model $\mathcal{A}^c$ on class 1, without false positive filtering or beam search refinement; (**S2**) using a single prediction without false positive filtering but incorporating the beam search refinement; (**S3**) applying only the coarse data generation stage (i.e., $\mathcal{D}_{\text{coarse}}$), without beam search refinement; and the full version of our method with all components enabled. The results, shown in Figure 4, demonstrate how each design component influences the overall attack performance through different configurations.

Extended Analysis. For additional experiments, please refer to our Appendix F (Sheng et al. 2025).

Conclusion

In this paper, we introduce a new concept of retaliatory attacks against FU, which refer to potential attacks launched by malicious users who oppose the unlearning mechanism in FL. We propose two such attacks, AUA and DUA, which aim to either undermine the forgetting effect or induce targeted discrimination against the unlearned user. These attacks leverage privacy leakage during the FU process by first conducting an unlearning-induced MIA, followed by a coarse-to-fine data generation method to reconstruct the unlearned data. We evaluate the effectiveness and robustness of the proposed attacks under various FU methods and settings.

Acknowledgments

This work is supported by Australian Research Council/Linkage Project LP230100294.

References

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, 141–159. IEEE.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 463–480. IEEE.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. 2021. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, 2633–2650.

Che, T.; Zhou, Y.; Zhang, Z.; Lyu, L.; Liu, J.; Yan, D.; Dou, D.; and Huan, J. 2023. Fast federated machine unlearning with nonlinear functional theory. In *International conference on machine learning*, 4241–4268. PMLR.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, 896–911.

Gao, X.; Ma, X.; Wang, J.; Sun, Y.; Li, B.; Ji, S.; Cheng, P.; and Chen, J. 2024. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*.

Halimi, A.; Kadhe, S. R.; Rawat, A.; and Angel, N. B. 2022. Federated Unlearning: How to Efficiently Erase a Client in FL? In *International Conference on Machine Learning*.

Harding, E.; Vanto, J. J.; Clark, R.; Ji, L. H.; and Ainsworth, S. C. 2019. Understanding the scope and impact of the California Consumer Privacy Act of 2018. *Journal of Data Protection & Privacy*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Lamri, H.; Alam, M.; Jiang, H.; and Maniatakos, M. 2025. DRAUN: An Algorithm-Agnostic Data Reconstruction Attack on Federated Unlearning Systems. arXiv:2506.01777.

Liu, G.; Ma, X.; Yang, Y.; Wang, C.; and Liu, J. 2021. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 1–10. IEEE.

Liu, Y.; Xu, L.; Yuan, X.; Wang, C.; and Li, B. 2022. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, 1749–1758. IEEE.

Liu, Z.; Jiang, Y.; Shen, J.; Peng, M.; Lam, K.-Y.; Yuan, X.; and Liu, X. 2023. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and Arcas, B. A. y. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 7. Granada.

Sheng, X.; Bao, W.; and Ge, L. 2024. Robust federated unlearning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2034–2044.

Sheng, X.; Bao, W.; Wang, H.; Liu, Y.; and Fu, S. 2025. GitHub: Retaliatory Attacks Against Federated Unlearning via Data Leakage. https://github.com/stcebra/Retaliatory-Attacks-Against-Federated-Unlearning. Appendix & Code.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Voigt, P.; and von dem Bussche, A. 2017. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Cham, Switzerland: Springer.

Wang, F.; Li, B.; and Li, B. 2023. Federated unlearning and its privacy threats. *IEEE Network*, 38(2): 294–300.

Wang, J.; Guo, S.; Xie, X.; and Qi, H. 2022. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, 622–632.

Wang, W.; Ma, Q.; Zhang, Z.; Liu, Y.; Liu, Z.; and Fang, M. 2025. Poisoning Attacks and Defenses to Federated Unlearning. In *Companion Proceedings of the ACM on Web Conference 2025*, 1365–1369.

Wang, W.; Tian, Z.; Zhang, C.; and Yu, S. 2024. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*.

Wu, C.; Zhu, S.; and Mitra, P. 2023. Unlearning Backdoor Attacks in Federated Learning. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

Wu, L.; Guo, S.; Wang, J.; Hong, Z.; Zhang, J.; and Ding, Y. 2022. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5): 129–135.

Xiong, Z.; Li, W.; Li, Y.; and Cai, Z. 2023. Exact-fun: an exact and efficient federated unlearning approach. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1439–1444. IEEE.

Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.

Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, 268–282. IEEE.

Zhang, F.; Li, W.; Hao, Y.; Yan, X.; Cao, Y.; and Lim, W. Y. B. 2025. Verifiably Forgotten? Gradient Differences Still Enable Data Reconstruction in Federated Unlearning. arXiv:2505.11097.

Zhang, L.; Zhu, T.; Zhang, H.; Xiong, P.; and Zhou, W. 2023a. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18: 4732–4746.

Zhang, R.; Guo, S.; Wang, J.; Xie, X.; and Tao, D. 2023b. A Survey on Gradient Inversion: Attacks, Defenses and Future Directions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5678–685.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. *Advances in neural information processing systems*, 32.

A Related Work

In this section, we provide a detailed discussion of existing works related to the proposed retaliatory attacks, which were briefly summarized in the *Introduction* section of the main paper.

A.1 Machine and Federated Unlearning

Machine unlearning (MU) has been proposed as a privacy-preserving technique that enables a ML model to forget specific portions of its training data (Cao and Yang 2015; Wang et al. 2024a). A variety of MU methods have been developed in centralized settings, aiming to achieve either *exact* (Bourtoule et al. 2021; Golatkar et al. 2021; Graves, Nagisetty, and Ganesh 2021; Brophy and Lowd 2021; Yan et al. 2022) or *approximate* (Ginart et al. 2019; Nguyen, Low, and Jaillet 2020; Guo et al. 2020; Sekhari et al. 2021; Neel, Roth, and Sharifi-Malvajerdi 2021; Mehta et al. 2022; Cha et al. 2024) unlearning. Federated unlearning (FU) extends MU to the FL setting (Liu et al. 2021, 2024c). Existing FU methods can be categorized into three types: (1) *Client-level unlearning*, which removes the entire contribution of participating clients (Su and Li 2023; Zhang et al. 2023a; Yuan et al. 2023; Tao et al. 2024); (2) *Sample-level unlearning*, which removes specific data samples from within a client's dataset (Che et al. 2023; Xiong et al. 2023; Zhu, Li, and Hu 2023; Li et al. 2023); and (3) *Class-level unlearning*, which removes data associated with particular classes (Wang et al. 2022; Zhao et al. 2023b; Wang et al. 2024b). Our work mainly focuses on client-level unlearning, while also providing an adaptation for sample-level unlearning (see Appendix F.9). The primary goals of FU methods include improving the *effectiveness* of the unlearned model (e.g., by leveraging historical information (Liu et al. 2021) or applying knowledge distillation (Wu, Zhu, and Mitra 2023)), enhancing the *efficiency* of the unlearning process (e.g., through rapid retraining (Liu et al. 2022a) or projected gradient descent (Halimi et al. 2022)), and ensuring *certified removal* of the unlearned data (e.g., via MIA (Wang et al. 2022), backdoor verification (Wu, Zhu, and Mitra 2023), or removal metrics (Gao et al. 2024)).

More recently, *security* and *privacy* concerns surrounding unlearning algorithms have received increasing attention (Liu et al. 2024a; Wang, Li, and Li 2023; Liu et al. 2024b). Both MU and FU have been shown to be vulnerable to various types of adversarial attacks. Among these, poisoning attacks have emerged as the most effective threat against both MU (e.g., mislead the MU model (Di et al. 2022; Qian et al. 2023; Zhao et al. 2023a)) and FU (e.g., degrade the fairness (Sheng, Bao, and Ge 2024) or utility (Wang et al. 2025) of the FU model). In addition, (Marchant, Rubinstein, and Alfeld 2022) proposes a slow-down attack that increases the computational cost of data removal in MU, while (Liu et al. 2024d) demonstrates that MU can be leveraged to conduct backdoor attacks by strategically selecting the data samples to be unlearned. In this work, diverging from all existing attack paradigms, we propose a completely new attack surface, which we term retaliatory attacks against FU—representing attacks launched by clients opposed to unlearning as a form of retaliation against unlearning clients.

A.2 Privacy Attacks

Two of the most effective privacy attacks against ML models are model inversion attacks (Fredrikson, Jha, and Ristenpart 2015) and membership inference attacks (MIAs) (Shokri et al. 2017). A model inversion attack aims to reconstruct sensitive attributes of a model's training data by exploiting its prediction outputs (He, Zhang, and Lee 2019; Zhang et al. 2020; Wang et al. 2021; Nguyen et al. 2023). Recent studies have extended model inversion attacks to FL settings by exploiting client gradients (also known as gradient inversion attacks), enabling the reconstruction of private data from participating clients (Zhu, Liu, and Han 2019; Geiping et al. 2020; Huang et al. 2021; Balunovic et al. 2022). However, these attacks typically assume a malicious server or a neighbor that have access to the gradients of the victim (Zhang et al. 2023b). Such assumptions are overly strong in the context of both FL and FU, and are incompatible with our setting, where the attacker is merely a benign-looking client without access to any internal information from the server or the unlearned client.

MIA, on the other hand, aims to determine whether a specific data sample was included in the training set of a ML model by analyzing the model's responses to that sample (Choquette-Choo et al. 2021; Hu et al. 2022; Carlini et al. 2022). The most widely used MIA approaches rely on either shadow model training (Shokri et al. 2017; Long et al. 2020) or model prediction statistics, such as loss (Yeom et al. 2018; Liu et al. 2022b) or entropy (Salem et al. 2018). A fundamental prerequisite for these approaches is having access to the target (victim) model. However, this assumption does not hold in our FU setting, as the privacy-preserving nature of FL prevents an adversarial client from accessing the unlearned client's local model. (Chen et al. 2021) pioneers an MIA approach for MU by leveraging the differences between models before and after unlearning. While this approach opens a promising direction for enabling MIA in unlearning contexts, it is primarily effective on classical ML models such as logistic regression and decision trees, but shows limited effectiveness on deep models like multi-layer perceptrons (MLPs). This significantly restricts its applicability in FL and FU settings. In this work, we draw inspiration from (Chen et al. 2021) and propose a novel unlearning-induced MIA for FU by introducing several key adaptations. This unlearning-induced MIA serves as a critical foundation for enabling the retaliatory attacks proposed in our study.

B Summary of Notations

In this section, we summarize the notations used in our main paper (*Problem Formulation* and *Method* sections) in Table 1. All symbols are also clearly defined within the main text to ensure the paper is self-contained.

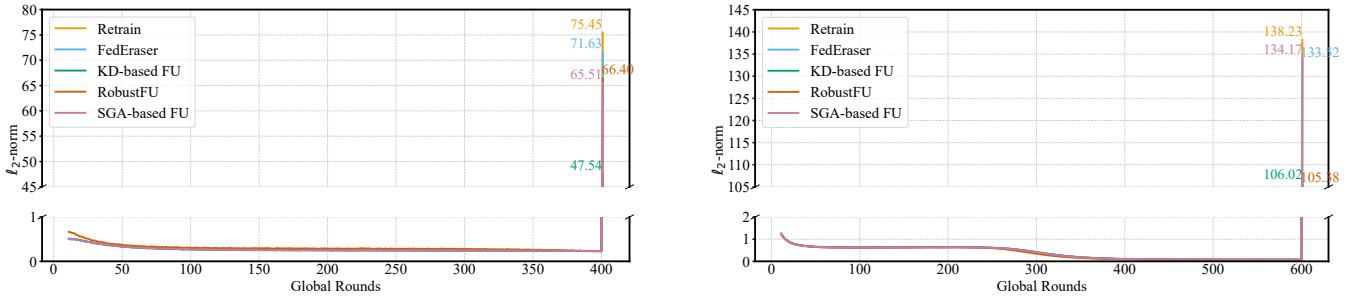
Symbols	Explanation
n	total number of participating clients in FL
$\mathcal{K}$	set of all participating clients in FL
k_i	i-th client
$\mathcal{D}_i$	local dataset of client k_i
$\mathcal{D}$	entire training dataset
$\mathcal{M}$	global model in the FL system
$\mathcal{M}_{\text{before}}, \mathcal{M}_{\text{after}}$	global models before and after the FU process
$\mathcal{K}^u$	set of unlearned clients (i.e., clients who submit unlearning requests)
$\mathcal{K}^r$	set of retained clients (i.e., clients who do not request unlearning)
$\mathcal{D}^u$	unlearned data associated with clients in $\mathcal{K}^u$
$\mathcal{D}^r$	retained data associated with clients in $\mathcal{K}^r$
$\mathcal{I}$	auxiliary information required for unlearning
S	total number of shadowing processes
$\mathcal{D}_{\text{shadow}}$	shadow dataset constructed by the adversarial client
$\mathcal{M}^s_{\text{before}}, \mathcal{M}^s_{\text{after}}$	shadowed global models before and after unlearning in the s-th shadowing process
$\mathcal{D}^s_{\text{external}}$	subset of shadow data not used in training either $\mathcal{M}^s_{\text{before}}$ or $\mathcal{M}^s_{\text{after}}$ in the s-th shadowing process
$\mathcal{D}^s_{\text{unlearned}}$	subset of shadow data used to train $\mathcal{M}^s_{\text{before}}$ but excluded from $\mathcal{M}^s_{\text{after}}$ in the s-th shadowing process
$\mathcal{D}^s_{\text{retained}}$	subset of shadow data used to train both $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$ in the s-th shadowing process
$\mathbb{P}^s_{\text{before}}, \mathbb{P}^s_{\text{after}}$	posterior probabilities from querying $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$
$\mathcal{D}^s_{\text{attack}}$	attack data generated in the s-th shadowing process
$\mathcal{D}_{\text{attack}}$	aggregated attack data from all shadowing processes
C	total number of classes
x	input feature vector of a data sample
y	class label of a data sample
$\mathcal{D}^c_{\text{attack}}$	attack data samples with original class label $y = c$
$\mathcal{A}^c$	attack model trained on class-specific attack data $\mathcal{D}^c_{\text{attack}}$
$\mathcal{A}$	set of all class-specific attack models
$\mathbb{P}_{\text{before}}, \mathbb{P}_{\text{after}}$	posterior probabilities from querying $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$
$\mathbb{P}_{\text{attack}}$	posterior probability from querying the attack model
$\mathcal{D}_{\text{coarse}}$	coarsely generated samples approximating the distribution of unlearned data
$\tilde{\mathcal{D}}_{\text{reconstruct}}$	intermediate reconstructed data refined via beam search
$\mathcal{D}_{\text{reconstruct}}$	final reconstructed unlearned data after beam search refinement
$\mathcal{D}_{\text{benign}}$	benign local dataset held by the adversarial client
$\mathcal{F}$	objective function used in the beam search process
α, β, γ	weighting coefficients in the beam search objective $\mathcal{F}$
$\mathcal{L}$	objective function used in DUA
$\mathcal{L}_{CE}$	standard cross-entropy loss
λ	weighting coefficient in the DUA objective $\mathcal{L}$

Table 1: Summary of notations used in the main paper (ordered by appearance)

C Detecting Federated Unlearning via Consistency Patterns in Global Model Updates

In the *Threat and Adversary Model* section of the main paper, we assume that the adversary (a normal participating client who does not issue unlearning requests) can observe the global model before ($\mathcal{M}_{\text{before}}$) and after ($\mathcal{M}_{\text{after}}$) the FU process. This is a reasonable assumption, as the execution of FU often introduces observable disruptions to the standard FL process. Moreover, in many FU implementations, the server explicitly notifies all participating clients when unlearning is being performed. In this section, we show that even in the absence of such explicit signals, an adversarial client can still reliably detect the occurrence of FU and estimate the post-unlearning global model.

The key insight for detecting the occurrence of a FU process lies in the observation that unlearning often introduces model-level inconsistencies compared to the standard FL process. For instance, retraining-based FU methods may involve broadcasting an untrained (Liu et al. 2021; Sheng, Bao, and Ge 2024) or partially reinitialized (Wu et al. 2022) global model, which can deviate significantly from the previous global model. Even for non-retraining-based approaches (Wu, Zhu, and Mitra 2023), the removal of the influence of unlearned data can cause substantial shifts in the global model's parameters. To exploit this phenomenon, we propose a simple yet effective detection method that monitors global model update consistency throughout the FL process. Specifically, we compute the ℓ_2-norm of the difference between two consecutive global models to quantify update variance. An update is flagged as anomalous (potentially indicating the initiation of a FU process) if its magnitude exceeds a



(a) Location (b) Purchase100

Figure 1: Visualization of model update inconsistency induced by the FU process.

threshold defined as the mean μ plus two standard deviations σ (i.e., $\mu + 2\sigma$), where μ and σ are calculated over the historical sequence of global update magnitudes observed during training. These statistics are updated incrementally over rounds. We choose this threshold based on the empirical observation that global model update magnitudes approximately follow a normal distribution, where roughly 95% of the values lie within two standard deviations of the mean. Hence, any update exceeding this range is considered a statistically significant deviation and a reliable indicator of a FU-triggered inconsistency.

Figure 1 visualizes the model-level inconsistency induced by the FU process during FL. We consider five FU methods: Retrain (i.e., retraining from scratch), FedEraser (Liu et al. 2021), KD-based FU (Wu, Zhu, and Mitra 2023), SGA-based FU (Wu et al. 2022), and RobustFU (Sheng, Bao, and Ge 2024). On both the Location and Purchase100 datasets, when an unlearning process is triggered at the 401st and 601st global rounds, we observe a significant deviation between the received global model and the one from the previous round. Based on this observation, the adversarial client can accurately identify the global models in the 400th and 600th rounds as corresponding to the global model before unlearning ($\mathcal{M}_{\text{before}}$).

To identify the global model after unlearning ($\mathcal{M}_{\text{after}}$), we propose a performance-based estimation approach. Specifically, the adversarial client first evaluates the performance (e.g., accuracy) of the global model before unlearning on a local validation set (e.g., a small subset held out from its training data). Once the FU process is detected, any subsequently received global model is estimated to be the post-unlearning model if its performance on the validation set recovers to a certain proportion (e.g., 90%) of the pre-unlearning performance. This method is elegantly applicable to both retraining-based FU methods, which may require several rounds to recover model performance, and non-retraining-based FU methods, where the post-unlearning model may already retain high accuracy.

D Detailed Procedure of the Retaliatory Attack

In this section, we provide a detailed description of the proposed retaliatory attack, as introduced in the *Method* section of the main paper. We follow the same stage-wise structure and elaborate on each component of the attack framework: Stage I, unlearning-induced membership inference; Stage II, coarse-to-fine unlearned data reconstruction; and Stage III, strategic retaliation via reconstructed data.

Stage I: Unlearning-Induced Membership Inference

Algorithm 1 illustrates the detailed procedure of Stage I of the proposed retaliatory attack, which trains a set of unlearning-induced MIA models by leveraging the discrepancy between the pre-unlearning global model $\mathcal{M}_{\text{before}}$ and the post-unlearning global model $\mathcal{M}_{\text{after}}$.

In Stage I(a), the adversary begins by locally shadowing the entire FU process (Algorithm 1, Lines 1–10). Specifically, the adversarial client first generates a shadow dataset $\mathcal{D}_{\text{shadow}}$ based on $\mathcal{M}_{\text{before}}$ using a search-based algorithm, such as the hill-climbing method proposed in (Shokri et al. 2017) (Algorithm 1, Line 2). Then, for each shadowing process $s \in \{1, 2, \ldots, S\}$ (where S denotes the total number of shadowing processes), the shadow dataset $\mathcal{D}_{\text{shadow}}$ is randomly partitioned into three disjoint subsets according to a predefined partition rate $\boldsymbol{r}$: $\mathcal{D}^s_{\text{external}}$, representing samples unused in both the FL and FU processes; $\mathcal{D}^s_{\text{unlearned}}$, representing samples used to train $\mathcal{M}_{\text{before}}$ but excluded from training $\mathcal{M}_{\text{after}}$; and $\mathcal{D}^s_{\text{retained}}$, used in training both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$ (Algorithm 1, Line 4). To better mimic the FU process, the adversary simulates a set of $\tilde{n}$ local clients, denoted as $\tilde{\mathcal{K}}$ (Algorithm 1, Line 5). Then, m clients are randomly sampled from $\tilde{\mathcal{K}}$ to simulate the set of unlearned clients $\tilde{\mathcal{K}}^u$, where m is uniformly sampled from $\{1, 2, \ldots, \lfloor \frac{\tilde{n}}{3} \rfloor\}$ to ensure that the unlearned clients constitute a minority of the total (Algorithm 1, Line 6). After that, $\mathcal{D}^s_{\text{unlearned}}$ and $\mathcal{D}^s_{\text{retained}}$ are evenly partitioned and assigned to each client in $\tilde{\mathcal{K}}^u$ and $\tilde{\mathcal{K}}^r$, respectively, to simulate their local datasets (Algorithm 1, Lines 7-8). Finally, the shadowed pre-unlearning global model $\mathcal{M}^s_{\text{before}}$ is trained using all simulated clients in $\tilde{\mathcal{K}}$, and the shadowed post-unlearning global model $\mathcal{M}^s_{\text{after}}$ is trained using only the retained clients in $\tilde{\mathcal{K}}^r$ (Algorithm 1, Lines 9-10). It is worth noting that both training processes strictly follow the standard FL procedure and

Algorithm 1 Unlearning-Induced Membership Inference (Stage I)

Input: Pre-unlearning global model $\mathcal{M}_{\text{before}}$, post-unlearning global model $\mathcal{M}_{\text{after}}$, total number of shadowing processes S, partition rate $\boldsymbol{r}$, total number of simulated clients $\tilde{n}$, and total number of classes C.

Output: The set of unlearning-induced MIA models $\mathcal{A}$.

1 *// Stage I(a): Shadowing FU Processes*
2 Generate a shadow dataset $\mathcal{D}_{\text{shadow}}$ based on $\mathcal{M}_{\text{before}}$ using a search-based algorithm (e.g., the hill-climbing algorithm);
3 **for** *each shadow process* $s = 1$ **to** S **do**
4 Randomly partition $\mathcal{D}_{\text{shadow}}$ into $\mathcal{D}^s_{\text{external}}$, $\mathcal{D}^s_{\text{unlearned}}$, and $\mathcal{D}^s_{\text{retained}}$ based on the partition ratio $\boldsymbol{r}$;
5 Simulate a set of $\tilde{n}$ local clients and denote it by $\tilde{\mathcal{K}}$;
6 Uniformly sample an integer $m \in \{1, 2, \ldots, \lfloor \frac{\tilde{n}}{3} \rfloor\}$, and randomly select m local clients as the simulated unlearned clients $\tilde{\mathcal{K}}^u$ (i.e., a minority subset of $\tilde{\mathcal{K}}$);
7 Uniformly partition $\mathcal{D}^s_{\text{unlearned}}$ among the unlearned clients in $\tilde{\mathcal{K}}^u$;
8 Uniformly partition $\mathcal{D}^s_{\text{retained}}$ among the retained clients in $\tilde{\mathcal{K}}^r$;
9 Locally train the shadowed pre-unlearning global model $\mathcal{M}^s_{\text{before}}$ with all simulated clients $\tilde{\mathcal{K}}$;
10 Locally train the shadowed post-unlearning global model $\mathcal{M}^s_{\text{after}}$ with the retained clients $\tilde{\mathcal{K}}^r = \tilde{\mathcal{K}} \setminus \tilde{\mathcal{K}}^u$;
11 *// Stage I(b): Generating Attack Data*
12 **init** $\mathcal{D}^s_{\text{attack}} \leftarrow \{\}$;
13 **for** *each data sample* $(x_e, y) \in \mathcal{D}^s_{\textit{external}}$ **do**
14 Use x_e to query $\mathcal{M}^s_{\text{before}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{before}}$;
15 Use x_e to query $\mathcal{M}^s_{\text{after}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{after}}$;
16 Add attack sample $(\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}, 0, y)$ to $\mathcal{D}^s_{\text{attack}}$;
17 **for** *each data sample* $(x_u, y) \in \mathcal{D}^s_{\textit{unlearned}}$ **do**
18 Use x_u to query $\mathcal{M}^s_{\text{before}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{before}}$;
19 Use x_u to query $\mathcal{M}^s_{\text{after}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{after}}$;
20 Add attack sample $(\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}, 1, y)$ to $\mathcal{D}^s_{\text{attack}}$;
21 **for** *each data sample* $(x_r, y) \in \mathcal{D}^s_{\textit{retained}}$ **do**
22 Use x_r to query $\mathcal{M}^s_{\text{before}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{before}}$;
23 Use x_r to query $\mathcal{M}^s_{\text{after}}$ and obtain the posterior probability $\mathbb{P}^s_{\text{after}}$;
24 Add attack sample $(\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}, 2, y)$ to $\mathcal{D}^s_{\text{attack}}$;
25 *// Stage I(c): Training Attack Models*
26 Construct the overall attack dataset as $\mathcal{D}_{\text{attack}} = \bigcup_{s=1}^{S} \mathcal{D}^s_{\text{attack}}$;
27 Partition $\mathcal{D}_{\text{attack}}$ into class-specific subsets $\{\mathcal{D}^c_{\text{attack}}\}_{c=1}^{C}$;
28 **for** *each class* $c \in \{1, 2, ..., C\}$ **do**
29 Train the class-specific attack model $\mathcal{A}^c$ with the corresponding dataset $\mathcal{D}^c_{\text{attack}}$;
30 **return** the final set of unlearning-induced MIA (attack) models $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$;

the retrain-from-scratch FU method.

In Stage I(b), the adversary generates training data for the attack model by querying each shadowed data sample to both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$, and assigning the corresponding labels (Algorithm 1, Lines 11–24). Specifically, for each data sample $(x_e, y) \in \mathcal{D}^s_{\text{external}}$, the adversary queries both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$ to obtain the corresponding posterior probabilities, denoted as $\mathbb{P}^s_{\text{before}}$ and $\mathbb{P}^s_{\text{after}}$, respectively (Algorithm 1, Lines 13-15). These posteriors capture how each model interprets the sample—both in terms of predicted class likelihoods and overall confidence (e.g., the maximum probability among all classes). The input feature for the attack model is then constructed by concatenating the two posteriors: $\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}$. Since (x_e, y) belongs to $\mathcal{D}^s_{\text{external}}$, it was not used to train either $\mathcal{M}^s_{\text{before}}$ or $\mathcal{M}^s_{\text{after}}$; therefore, the corresponding label is set to 0. The resulting attack sample, $(\mathbb{P}^s_{\text{before}}||\mathbb{P}^s_{\text{after}}, 0, y)$, with y being the original class label, is then added to the attack dataset $\mathcal{D}^s_{\text{attack}}$ (Algorithm 1, Line 16). A similar procedure is then applied to each data sample $(x_u, y) \in \mathcal{D}^s_{\text{unlearned}}$ (Algorithm 1, Lines 17-19) and $(x_r, y) \in \mathcal{D}^s_{\text{retained}}$ (Algorithm 1, Lines 21-23). The only difference lies in the label assignment: samples from $\mathcal{D}^s_{\text{unlearned}}$ are assigned a label of 1 (Algorithm 1, Line 20), indicating they were unlearned during the shadowed FU process, while samples from $\mathcal{D}^s_{\text{retained}}$ are assigned a label of 2 (Algorithm 1, Line 24), indicating they were used to train both $\mathcal{M}^s_{\text{before}}$ and $\mathcal{M}^s_{\text{after}}$.

In Stage I(c), the adversary trains class-specific attack models using the aggregated attack data generated from all shadowing processes (Algorithm 1, Lines 25–30). Specifically, let $\mathcal{D}_{\text{attack}} = \bigcup_{s=1}^{S} \mathcal{D}^s_{\text{attack}}$ denote the assembled attack dataset from all S shadowing processes (Algorithm 1, Line 26). This dataset is further partitioned into class-specific subsets $\{\mathcal{D}^c_{\text{attack}}\}_{c=1}^{C}$ based on the recorded class label y, where C is the total number of classes and each $\mathcal{D}^c_{\text{attack}}$ contains only attack samples with $y = c$

Algorithm 2 Coarse Data Generation (Stage II(a))

Input: Pre-unlearning global model $\mathcal{M}_{\text{before}}$, post-unlearning global model $\mathcal{M}_{\text{after}}$, confidence threshold τ, set of class-specific attack models $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$, and total number of generated data samples R.
Output: The set of coarsely generated data samples $\mathcal{D}_{\text{coarse}}$.

1 **init** $\mathcal{D}_{\text{coarse}} \leftarrow \{\}$;
2 **while** $|\mathcal{D}_{coarse}| < R$ **do**
3 Randomly generate sample input x;
4 Query $\mathcal{M}_{\text{before}}$ with x to obtain $\mathbb{P}_{\text{before}}$;
5 **if** $\max(\mathbb{P}_{before}) \geq \tau$ **then**
6 Assign the label $y = \arg\max(\mathbb{P}_{\text{before}})$ to $\boldsymbol{x}$;
7 Select the attack model $\mathcal{A}^c$ corresponding to class $c = y$;
8 Query $\mathcal{M}_{\text{after}}$ with x to obtain $\mathbb{P}_{\text{after}}$;
9 Query $\mathcal{A}^c$ with $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{after}}$ to obtain the predicted membership status z;
10 **if** $z = 1$ **then**
11 *// Cross-Model False Positive Filtering*
12 Query $\mathcal{A}^c$ with $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{before}}$ to obtain the predicted membership status $z^{\text{FP}}_{\text{before}}$;
13 Query $\mathcal{A}^c$ with $\mathbb{P}_{\text{after}}||\mathbb{P}_{\text{after}}$ to obtain the predicted membership status $z^{\text{FP}}_{\text{after}}$;
14 **if** $z^{FP}_{before} = 2$ ***and*** $z^{FP}_{after} = 0$ **then**
15 **accept** the sample (x, y) and add it to $\mathcal{D}_{\text{coarse}}$;
16 **else**
17 **continue** to the next iteration;
18 **else**
19 **continue** to the next iteration;
20 **else**
21 **continue** to the next iteration;
22 **return** $\mathcal{D}_{\text{coarse}}$;

(Algorithm 1, Line 27). A class-specific attack model $\mathcal{A}^c$ is then trained on $\mathcal{D}^c_{\text{attack}}$ for each class c (Algorithm 1, Lines 28-29), and the final set of unlearning-induced MIA (attack) models is assembled as $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$ (Algorithm 1, Line 30).

Stage II: Coarse-to-Fine Unlearned Data Reconstruction

In Stage II of the proposed retaliatory attack framework, we introduce a novel coarse-to-fine data generation pipeline that reconstructs the unlearned data based on the attack models obtained in Stage I. This section details the procedures for coarse data generation (Stage II(a), Algorithm 2) and beam search refinement (Stage II(b), Algorithm 3) within the pipeline.

In Stage II(a), the coarse data generation begins by randomly initializing the input feature vector x, with each attribute uniformly sampled within its domain (Algorithm 2, Line 3). The resulting x is used to query the pre-unlearning global model $\mathcal{M}_{\text{before}}$ to obtain its posterior probabilities $\mathbb{P}_{\text{before}}$ (Algorithm 2, Line 4). The confidence score is then calculated as the maximum value in $\mathbb{P}_{\text{before}}$, i.e., $\max(\mathbb{P}_{\text{before}})$. A generated sample is retained if its confidence score exceeds a predefined threshold τ (Algorithm 2, Line 5); otherwise, it is discarded (Algorithm 2, Lines 20-21). For a retained sample, its label is assigned as $y = \arg\max(\mathbb{P}_{\text{before}})$ (Algorithm 2, Line 6). This filtering step ensures that each retained sample has a reliable class label, which is essential for selecting the corresponding class-specific attack model $\mathcal{A}^c$ with $c = y$ (Algorithm 2, Line 7). While these randomly generated high-confidence samples roughly capture the overall training data distribution of $\mathcal{M}_{\text{before}}$, they may not align with the data distribution of the unlearned client(s). To further narrow the candidate set toward the unlearned data, each sample x is queried on both $\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$ to obtain the corresponding posterior probability vectors $\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}$ (Algorithm 2, Line 8), which are then concatenated and fed into the corresponding attack model $\mathcal{A}^c$ to obtain the predicted membership status z (Algorithm 2, Line 9). The sample is retained only if $z = 1$, indicating that it is inferred to belong to the unlearned data (Algorithm 2, Line 10); otherwise, it is discarded (Algorithm 2, Lines 18-19).

To minimize the impact of false positive predictions from the attack model set $\mathcal{A}$, we propose an innovative cross-model false positive filtering strategy that fully leverages the same attack model via permutation of input posteriors (Algorithm 2, Lines 11-17). Specifically, in addition to the original judgment (Algorithm 2, Lines 9–10), we generate two auxiliary inputs by duplicating $\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}$ (i.e., $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}||\mathbb{P}_{\text{after}}$). These are then fed into $\mathcal{A}^c$ to obtain the corresponding predicted membership statuses, $z^{\text{FP}}_{\text{before}}$ and $z^{\text{FP}}_{\text{after}}$ (Algorithm 2, Lines 12-13). The sample is accepted and added to the final set of coarsely generated samples $\mathcal{D}_{\text{coarse}}$ only if $z^{\text{FP}}_{\text{before}} = 2$ and $z^{\text{FP}}_{\text{after}} = 0$ (Algorithm 2, Lines 14-15); otherwise, it is discarded

Algorithm 3 Beam Search Refinement (Stage II(b))

Input: Pre-unlearning global model $\mathcal{M}_{\text{before}}$, post-unlearning global model $\mathcal{M}_{\text{after}}$, set of class-specific attack models $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$, beam size B, number of candidate samples generated from each beam sample U; number of features flipped per step V, maximum number of search steps M, and weighting coefficients α, β, and γ.

Output: Reconstructed set of unlearned data samples $\mathcal{D}_{\text{reconstruct}}$.

init $\mathcal{D}_{\text{reconstruct}} \leftarrow \{\}$;
for *each* $(x, y) \in \mathcal{D}_{coarse}$ **do**
  Compute initial score $\mathcal{F}(x)$;
  init BeamSet $\leftarrow \{(x, \mathcal{F}(x))\}$;
  for $m = 1$ **to** M **do**
    init CandidateSet $\leftarrow \{\}$;
    for *each* $(x_{cur}, \mathcal{F}(x_{cur})) \in$ *BeamSet* **do**
      for $u = 1$ **to** U **do**
        Randomly sample V feature indices;
        For each sampled index, flip its value within its domain to obtain x_{flip};
        Compute $\mathcal{F}(x_{\text{flip}})$;
        Add $(x_{\text{flip}}, \mathcal{F}(x_{\text{flip}}))$ to CandidateSet;
    Sort CandidateSet by score in descending order;
    Set BeamSet $\leftarrow$ top B samples from CandidateSet;
  Select sample x^* with highest score $\mathcal{F}(x^*)$ from BeamSet;
  Add (x^*, y) to $\mathcal{D}_{\text{reconstruct}}$;
return $\mathcal{D}_{\text{reconstruct}}$;

(Algorithm 2, Lines 16-17). The key insight here is that, given our attack model $\mathcal{A}^c$ actually considers two models ($\mathcal{M}_{\text{before}}$ and $\mathcal{M}_{\text{after}}$) as target models, each of the two posteriors fed into $\mathcal{A}^c$ encodes distinct training-related information for an unlearned data sample: the first indicates that the sample was seen (i.e., included in the training data) by the first target model ($\mathcal{M}_{\text{before}}$), while the second reflects that the sample was absent from the training data of the second target model ($\mathcal{M}_{\text{after}}$). In this case, when the input is $\mathbb{P}_{\text{before}}||\mathbb{P}_{\text{before}}$, it represents a sample that is seen by both target models, and thus should be assigned a membership status of 2. Conversely, when the input is $\mathbb{P}_{\text{after}}||\mathbb{P}_{\text{after}}$, it indicates that the sample is absent from the training sets of both models and should therefore correspond to membership status 0. This strategy effectively filters out false positives in the initial screening stage, where the prediction of $\mathcal{A}^c$ may be dominated by either $\mathbb{P}_{\text{before}}$ or $\mathbb{P}_{\text{after}}$, leading to the incorrect classification of non-unlearned samples as class 1. Finally, the entire procedure is repeated until a total of R samples have been generated (Algorithm 2, Line 2).

In Stage II(b), we further introduce a sample-level refinement procedure to enhance the fidelity of these samples with respect to the original unlearned data. The key idea is to increase the confidence of the attack model $\mathcal{A}^c$ in classifying a sample as class 1, i.e., to maximize $\mathbb{P}_{\text{attack}}[1]$, where $\mathbb{P}_{\text{attack}}$ denotes the posterior obtained from querying $\mathcal{A}^c$. In addition, we incorporate a diversity penalty to discourage the refined samples from collapsing into similar patterns, thereby promoting sample-level variability. Furthermore, the confidence score $\mathbb{P}_{\text{before}}[c]$ of $\mathcal{M}_{\text{before}}$ is encouraged to remain high, as it directly determines the selection of the corresponding attack model $\mathcal{A}^c$. Formally, our objective function $\mathcal{F}$ is defined as:

$$\mathcal{F}(x) = \alpha \cdot (\mathbb{P}_{\text{before}}[c]) + \beta \cdot (\mathbb{P}_{\text{attack}}[1]) - \gamma \cdot \big(1 - \min_{x' \in \tilde{\mathcal{D}}_{\text{reconstruct}}} \text{dist}(x, x')\big) \tag{1}$$

where α, β, and γ are weighting coefficients; $\text{dist}(\cdot, \cdot)$ denotes the distance function between two data samples (we adopt the Jaccard distance in our implementation, see Appendix E.5); and $\tilde{\mathcal{D}}_{\text{reconstruct}}$ represents the set of samples that have already been refined. Given the black-box and non-differentiable nature of $\mathcal{F}(x)$, we employ a heuristic beam search algorithm that iteratively refines candidate samples via random bit-level flipping in the input space, as illustrated in Algorithm 3.

Specifically, for each data sample $(x, y) \in \mathcal{D}_{\text{coarse}}$, we first compute its initial score $\mathcal{F}(x)$ using our objective function $\mathcal{F}$ (Algorithm 3, Line 3) and initialize a BeamSet with $\{(x, \mathcal{F}(x))\}$ (Algorithm 3, Line 4). Then, for each beam search step $m \in \{1, 2, \ldots, M\}$ (where M is the maximum number of search steps), an empty CandidateSet is initialized to store the candidate samples generated during the beam search (Algorithm 3, Lines 5-6). Next, for each beam sample $(x_{\text{cur}}, \mathcal{F}(x_{\text{cur}}))$ in the BeamSet, a total of U candidate samples are generated from x_{cur} by first randomly sampling V feature indices and then flipping the value of each sampled index within its domain (Algorithm 3, Lines 7-10). Let x_{flip} denote the modified sample; it is added to the CandidateSet along with its updated score $\mathcal{F}(x_{\text{flip}})$ (Algorithm 3, Lines 11-12). Once all candidate samples have been added to the CandidateSet, they are sorted in descending order based on their score values (Algorithm 3, Line 13), and the top B samples form the updated BeamSet for the current step (Algorithm 3, Line 14). Finally, after all M beam steps

Algorithm 4 AUA: Anti-Unlearning Attack (Stage III(a))

Input: Reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$, learning rate η, global round at which the attack starts T, number of local epochs E, and number of global poisoning rounds G.

1 **for** *each global round* $t = T$ **to** $T + G - 1$ **do**
2 Receive the global model $\mathcal{M}^{t-1}$ from the server;
3 **init** $\mathcal{M}_0^t \leftarrow \mathcal{M}^{t-1}$;
4 **for** *each local epoch* $e = 1$ **to** E **do**
5 $\mathcal{M}_e^t \leftarrow$ train $\mathcal{M}_{e-1}^t$ on $\mathcal{D}_{\text{reconstruct}}$ with learning rate η;
6 Set $\tilde{\mathcal{M}}^t \leftarrow \mathcal{M}_E^t$;
7 Upload the poisoned local model $\tilde{\mathcal{M}}^t$ to the central server;

Algorithm 5 DUA: Discrimination-Unlearning Attack (Stage III(b))

Input: Reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$, learning rate η, global round at which the attack starts T, number of local epochs E, number of global poisoning rounds G, local benign dataset of the adversarial client $\mathcal{D}_{\text{benign}}$, and weighting coefficient λ.

1 **for** *each global round* $t = T$ **to** $T + G - 1$ **do**
2 Receive the global model $\mathcal{M}^{t-1}$ from the server;
3 **init** $\mathcal{M}_0^t \leftarrow \mathcal{M}^{t-1}$;
4 **for** *each local epoch* $e = 1$ **to** E **do**
5 $\mathcal{M}_e^t \leftarrow$ train $\mathcal{M}_{e-1}^t$ by minimizing the loss $\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{benign}}) - \lambda \cdot \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{reconstruct}})$ with learning rate η;
6 Set $\tilde{\mathcal{M}}^t \leftarrow \mathcal{M}_E^t$;
7 Upload the poisoned local model $\tilde{\mathcal{M}}^t$ to the central server;

are completed, the sample (x^*, y) with the highest score $\mathcal{F}(x^*)$ is selected as the refined version of the coarse sample (x, y) (Algorithm 3, Line 15) and is added to the final reconstructed set $\mathcal{D}_{\text{reconstruct}}$ (Algorithm 3, Line 16).

Stage III: Strategic Retaliation via Reconstructed Data

Once the adversarial client reconstructs the unlearned dataset $\mathcal{D}_{\text{reconstruct}}$ locally, it can now proceed to launch a targeted retaliatory attack, either AUA or DUA, against the global model.

Algorithm 4 illustrates the detailed procedure for launching the AUA (Stage III(a)). Recall that the objective of AUA is to prevent the global model from successfully forgetting the training data of the unlearned client(s). This is achieved by having the adversarial client repeatedly upload poisoned local model updates that are deliberately overfitted to $\mathcal{D}_{\text{reconstruct}}$, so that the global model is forced to relearn the unlearned data samples and compromise the intended forgetting effect. Specifically, let T denote the global round at which the attack starts. For each global round $t \in \{T, T+1, \ldots, T+G\}$, where G is the number of poisoning rounds, the adversarial client first receives the global model $\mathcal{M}^{t-1}$ from the previous communication round (Algorithm 4, Lines 1-2). It then trains this model on the reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$ using a small learning rate η for E local epochs (Algorithm 4, Lines 3-5). Finally, the resulting poisoned local model $\tilde{\mathcal{M}}^t$ is uploaded to the central server for global aggregation (Algorithm 4, Lines 6-7). Following this, the AUA effectively compromises the global model after unlearning within just a few communication rounds (i.e., small G).

Algorithm 5 illustrates the detailed procedure for launching the DUA (Stage III(b)). Recall that the goal of DUA is to introduce discrimination into the unlearning process by deliberately degrading the global model's performance on the data distribution associated with the unlearned client(s). This is accomplished through a targeted model poisoning attack, where the adversarial client jointly leverages the reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$ and its own local benign dataset $\mathcal{D}_{\text{benign}}$. Specifically, similar to AUA, the adversarial client initiates the attack at the T-th global communication round. For each subsequent round $t \in \{T, T+1, \ldots, T+G\}$, the adversary trains the received global model $\mathcal{M}^{t-1}$ for E local epochs using a learning rate η, by minimizing the following loss function (Algorithm 5, Lines 1–5):

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{benign}}) - \lambda \cdot \mathcal{L}_{\text{CE}}(\mathcal{D}_{\text{reconstruct}}), \tag{2}$$

where $\mathcal{L}_{\text{CE}}$ denotes the standard cross-entropy loss, and λ controls the strength of the negative gradient signal derived from the unlearned data. By inverting the loss gradient on $\mathcal{D}_{\text{reconstruct}}$, the adversary forces the model to perform poorly on the reconstructed unlearned samples, while maintaining nominal performance on its own benign data. Finally, the resulting poisoned local model $\tilde{\mathcal{M}}^t$ is uploaded to the central server for global aggregation (Algorithm 5, Lines 6-7). In doing so, DUA also effectively compromises the global model after unlearning within just a few communication rounds (i.e., a small G).

E Detailed Experimental Setup

E.1 Detailed Description of Datasets

Location (Yang et al. 2015). Our Location dataset is derived from Foursquare check-in data collected in the Bangkok area between April 2012 and September 2013. Following the methodology in prior work on MIA (Shokri et al. 2017), we preprocess the dataset by filtering out users with fewer than 25 check-ins and locations with fewer than 100 visits, resulting in 5,502 user profiles. Each user is represented by a 446-dimensional binary feature vector indicating visits to specific geographic regions and semantic location types. To define classification labels, we apply the k-means algorithm to cluster the user records into 30 groups based on their feature patterns, where each group corresponds to a distinct geosocial type. To ensure a balanced class distribution, we remove a small number of samples. After this balancing step, we retain 4,057 user records, which are used for training and evaluation in the subsequent tasks.

Purchase (Shokri et al. 2017). Our Purchase dataset is derived from Kaggle's "Acquire Valued Shoppers Challenge" dataset, which involves offering promotional coupons to customers and predicting which recipients will become repeat buyers. The dataset provides at least one year of shopping history per customer, including full transaction records with product name, store chain, quantity, and date. Following the methodology proposed in (Shokri et al. 2017), we use a preprocessed version released by the authors, which simplifies the original data into $197,324$ user records. Each user is represented by a 600-dimensional binary vector indicating whether each of 600 products has been purchased. Based on these features, the authors clustered users into 2, 10, 20, 50, or 100 classes to represent different purchasing styles. In our experiments, we directly adopt the 100-class setting (Purchase100) without further modification and extract the first 35,000 user records in the subsequent tasks.

CIFAR-10 (Krizhevsky, Hinton et al. 2009). CIFAR-10 is a widely used image classification benchmark consisting of $60,000$ color images in 10 categories, each of size 32×32 pixels. The dataset is split into $50,000$ training images and $10,000$ test images, with $6,000$ images per class. We use the standard training set provided by the dataset and do not perform any additional preprocessing or label transformation.

Street View House Numbers (Netzer et al. 2011). The Street View House Numbers (SVHN) dataset contains real-world digit images collected from Google Street View. Each image is a 32×32 color image representing a digit (0–9), with over $600,000$ labeled examples in total. In our experiments, we use the standard format of the dataset, consisting of $73,257$ training images and $26,032$ test images. Similar to CIFAR-10, no extra processing or augmentation is applied.

E.2 Detailed Evaluation Metrics

In this subsection, we provide detailed descriptions of the evaluation metrics used in this paper.

Member Inference Attack Accuracy (MIA). MIA is widely used as a certified test for assessing whether the unlearned data has been successfully forgotten from the global model during FU (Wang et al. 2022; Halimi et al. 2022; Sheng, Bao, and Ge 2024). Specifically, a MIA model takes a given sample along with the posterior probabilities obtained by querying the target model on that sample as input, and outputs a binary prediction of either 0 or 1. A prediction of 0 indicates that the sample was not part of the target model's training set, while a prediction of 1 indicates that it was. In this setting, a higher prediction accuracy suggests greater membership information leakage from the target model, whereas an accuracy close to 0.5 implies the MIA model is performing no better than random guessing, indicating no effective leakage (Shokri et al. 2017).

In the FU scenario, the MIA model is trained using the global model before unlearning ($\mathcal{M}_{\text{before}}$) as the target model. A successful MIA model is expected to achieve high prediction accuracy on the unlearned data prior to the FU process, as this data was part of the training set of $\mathcal{M}_{\text{before}}$. After the FU process, however, the target model becomes the global model after unlearning ($\mathcal{M}_{\text{after}}$), in which the unlearned data should have been removed from the training set. A successful FU method (i.e., one that provides certified removal) is therefore expected to significantly reduce the MIA model's prediction accuracy on the unlearned data, ideally approaching random guess (i.e., around 0.5).

In our evaluation, we leverage this MIA setup to assess the effectiveness of the proposed AUA by measuring the increase in MIA prediction accuracy when using the global model compromised by AUA (after attack) as the target model. A larger increase in MIA accuracy (compared to using $\mathcal{M}_{\text{after}}$ as the target model) indicates stronger attack effectiveness and more severe privacy leakage. We adopt the standard MIA approach proposed by (Shokri et al. 2017) for our implementation. To ensure fair evaluation, we use a balanced evaluation set consisting of an equal number of unlearned samples (labeled as class 0) and training samples (labeled as class 1) when computing the MIA accuracy.

Unlearned Accuracy (UA). UA refers to the prediction accuracy of the global model on the unlearned dataset. We employ UA as an evaluation metric for both AUA and DUA. In the context of AUA, UA serves as an indicator of whether the unlearned data has truly been forgotten by the global model: a higher UA suggests the model still memorizes the data, while a lower UA (closer to the test accuracy on unseen data) indicates successful unlearning. Similar to MIA, we report UA both before and after applying AUA. An increase in UA after the attack implies that the proposed AUA has effectively compromised the global model by forcing it to relearn the unlearned data samples.

In the context of DUA, the goal is to introduce additional discrimination against the unlearned data and its underlying distribution. To evaluate this, we also report the UA before and after applying DUA, where a larger decrease in UA indicates stronger attack effectiveness. It is worth noting that the FU process inherently reduces the UA of the global model, as the

unlearned data is removed from its training set. To accurately assess the effect of DUA, we use the UA measured after the FU process (i.e., before the attack) as a reference point, allowing us to isolate the impact introduced solely by the attack.

Test Accuracy (TA). TA is simply the classification accuracy of the global model on a held-out test set. We include TA as an auxiliary metric when evaluating the performance of DUA. Specifically, during the DUA's targeted model poisoning attack, it is important to demonstrate that the degradation in performance is focused on the unlearned data and its underlying distribution, rather than on the overall model. In this context, a small decrease in TA is desirable, as it indicates that the attack does not harm the global model's general utility, but instead discriminates specifically against the unlearned data.

Unlearning-Induced Membership Inference Attack Accuracy (U-MIA). Given that the unlearning-induced MIA model ($\mathcal{A}$), proposed in Stage I of our retaliatory attack framework, serves as a direct and quantifiable measure of the privacy leakage caused by the FU process, we report its accuracy as an indicator of the extent to which different FU methods leak private information. Unlike the standard MIA model, this attack model jointly considers the global model before ($\mathcal{M}_{\text{before}}$) and after unlearning ($\mathcal{M}_{\text{after}}$) as target models, and takes as input a given sample along with the concatenation of posterior probabilities ($\mathbb{P}_{\text{before}}$ and $\mathbb{P}_{\text{after}}$) obtained by querying both models. Its output is a three-class prediction, where class 0 indicates the sample was never seen during training (i.e., not used in either $\mathcal{M}_{\text{before}}$ or $\mathcal{M}_{\text{after}}$), class 1 indicates the sample was used in training $\mathcal{M}_{\text{before}}$ but removed from $\mathcal{M}_{\text{after}}$ (i.e., an unlearned sample), and class 2 indicates the sample was used in training both models (i.e., a retained sample). In this setting, the expected accuracy of random guessing is approximately 0.33, and a higher U-MIA accuracy reflects a greater degree of privacy leakage during the FU process. To ensure fair evaluation, we also use a balanced test set with equal numbers of samples from each class when computing the U-MIA accuracy.

E.3 Implementation Details of FU Methods

In this subsection, we provide detailed implementation settings for the FU methods used as benchmarks in the main paper.

FedEraser (Liu et al. 2021). For FedEraser, we adopt a calibration ratio of $r = 0.5$ and a retraining interval of $\Delta t = 2$, as specified in the original paper. During retraining, the global epochs are set to 20, and the local epochs to 10, following the official implementation provided by the authors.

SGA-based FU (Wu et al. 2022). For SGA-based FU, we perform adaptive rounds of SGA, terminating when the loss reaches a threshold of training loss ≤ 5 (i.e., we return the model from the round immediately preceding the loss exceeding 5).

KD-based FU (Wu, Zhu, and Mitra 2023). For KD-based FU, we adhere to the settings described in the original paper, where the server has access to an equal number of unlabeled data samples from a single client. These unlabeled samples are used to perform 10 rounds of knowledge distillation following the removal of historical parameter updates from the unlearned client.

RobustFU (Sheng, Bao, and Ge 2024). For RobustFU, we set the number of generated data samples to 150 for the Location dataset and 400 for the Purchase100 dataset, following the recommended range in the original paper (i.e., between $\frac{1}{5}$ and $\frac{1}{3}$ of a single client's data samples).

E.4 Training Details

In our experiments, we adopt a multi-layer perceptron (MLP) with two fully connected layers for both the Location and Purchase100 datasets. The models are trained for 400 and 600 global rounds, respectively, each with one local epoch per round and batch sizes of 64 and 128. For CIFAR-10, we use ResNet-18 as the FL model, trained for 100 global rounds with one local epoch per round and a batch size of 128. For SVHN, we employ a simple convolutional neural network (CNN) consisting of two convolutional layers followed by two fully connected layers, trained for 10 global rounds with 10 local epochs per round and a batch size of 64. The learning rate η is set to 0.001 for all datasets during standard training.

In our default FL setting, we consider a system with $n = 5$ clients, where all clients participate in every global aggregation round (i.e., cross-silo FL). During FU, we assume that one participating client requests to remove its contribution from the global model. While this setting represents a relatively small-scale FL and FU scenario, we further evaluate the scalability of the proposed retaliatory attacks in larger and more complex setups in Appendix F.1 and F.2, including cases with more participating clients and multiple unlearning requests. In all settings, one participating client that does not submit an unlearning request is designated as the adversarial client. Unless otherwise specified, all experiments are conducted on a server equipped with an NVIDIA RTX 3090 GPU and an Intel(R) Core(TM) i9-10980XE CPU. All reported results are averaged over 10 runs with different random seeds.

E.5 Detailed Attack Setup

In this section, we provide detailed configurations and hyperparameters used in the proposed retaliatory attacks.

Unlearning-Induced MIA (Stage I). To conduct the proposed unlearning-induced MIA, we generate shadow datasets $\mathcal{D}_{\text{shadow}}$ consisting of $1,900$ and $15,000$ samples for the Location and Purchase100 datasets, respectively. We run $S = 60$ shadowing processes for Location and $S = 20$ for Purchase100. In each process s, $\mathcal{D}_{\text{shadow}}$ is partitioned into external ($\mathcal{D}^s_{\text{external}}$), unlearned ($\mathcal{D}^s_{\text{unlearned}}$), and retained ($\mathcal{D}^s_{\text{retained}}$) subsets using a fixed ratio of $\boldsymbol{r} = [0.4, 0.18, 0.42]$. The unlearned and retained subsets are then distributed across $\tilde{n} = 6$ simulated clients, with $m \in \{1, 2\}$ clients randomly selected as the unlearned client set $\tilde{\mathcal{K}}^u$.

For CIFAR-10 and SVHN, we allow the adversarial client to possess a separate dataset $\mathcal{D}_{\text{shadow}}$ that is disjoint from both the training and test sets used in FL. The shadow dataset contains $24,000$ and $12,000$ samples for CIFAR-10 and SVHN,

respectively. We run $S = 20$ shadowing processes for CIFAR-10 and $S = 10$ for SVHN. For both datasets, the participation ratio is fixed as $\boldsymbol{r} = [0.167, 0.167, 0.666]$, and as in the tabular setting, the unlearned ($\mathcal{D}^s_{\text{unlearned}}$) and retained ($\mathcal{D}^s_{\text{retained}}$) subsets are further distributed across $\tilde{n} = 6$ simulated clients. For training the shadow models across all datasets, we adopt the same model architectures and training configurations as described in Appendix E.4.

Then, to train the final set of class-specific attack models $\mathcal{A} = \{\mathcal{A}^c\}_{c=1}^{C}$, we adopt a 4-layer MLP architecture and generate $114,000$, $300,000$, and $24,000$ attack samples to train $C = 30$, 100, and 10 class-specific models for the Location, Purchase100, and CIFAR-10 datasets, respectively. Each model is trained for 100 epochs with a learning rate of 0.001. For SVHN, we use a 2-layer MLP and train $C = 10$ class-specific models on $12,000$ attack samples for 50 epochs with a learning rate of 0.001.

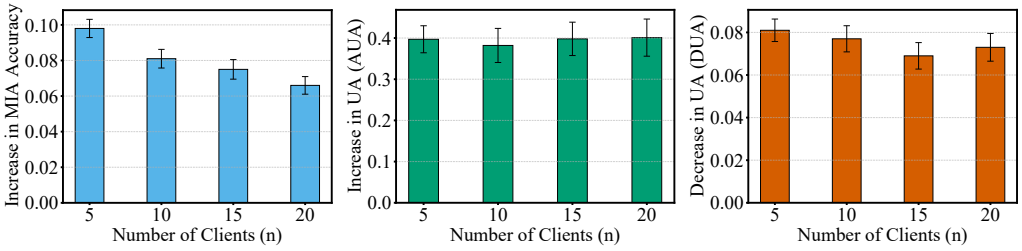
Coarse-to-Fine Unlearned Data Reconstruction (Stage II). For coarse data generation (Algorithm 2), we generate $R = 320$ and $2,000$ samples for the Location and Purchase100 datasets, respectively, using a confidence threshold of $\tau = 0.95$. The resulting coarse samples are then refined via beam search (Algorithm 3) with a beam size of $B = 5$, number of generated candidate samples $U = 5$, number of features flipped per step $V = 5$, and a maximum number of search steps $M = 10$. For the objective function $\mathcal{F}$, we set the weighting coefficients to $\alpha = 2$, $\beta = 1$, and $\gamma = 0.5$, selected by grid search to yield stable performance.

For the distance function $\text{dist}(\cdot, \cdot)$, we adopt the Jaccard distance for the Location and Purchase100 datasets, as both consist exclusively of binary attributes. (For tabular datasets with non-binary attributes, other distance metrics, such as Euclidean distance, can be applied instead.) In our implementation, a positive γ leads to the use of the *complement* of the minimum Jaccard distance between the sample x under evaluation and the set of previously generated samples $\tilde{\mathcal{D}}_{\text{reconstruct}}$. This encourages diversity by assigning a smaller penalty to samples that are more dissimilar from the generated set.

Strategic Retaliation via Reconstructed Data (Stage III). For AUA, the adversarial client replaces its original dataset with the reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$. The attack is performed over $G = 5$ global rounds, where the adversarial client trains locally for $E = 200$ epochs per round using a small learning rate of $\eta = 0.0005$. For DUA, the adversarial client conducts a targeted poisoning attack using both its original benign dataset $\mathcal{D}_{\text{benign}}$ and the reconstructed unlearned dataset $\mathcal{D}_{\text{reconstruct}}$. The attack proceeds for $G = 4$ global rounds, with $E = 50$ local training epochs per round and the same learning rate of $\eta = 0.0005$. The loss weighting coefficient λ is set to 1.5 and 0.5 for the Location and Purchase100 datasets, respectively.

F Extended Experimental Results

F.1 Scalability with Respect to the Number of Clients



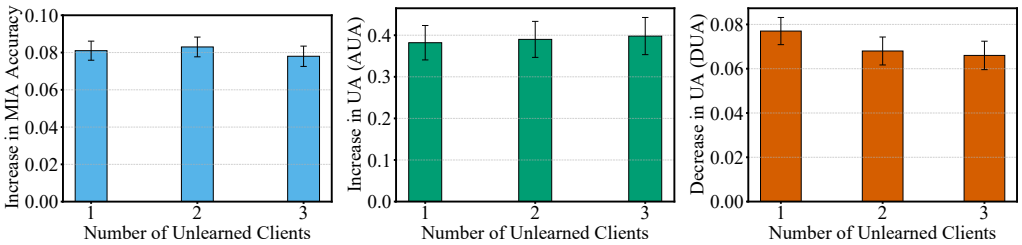
(a) AUA (MIA) (b) AUA (UA) (c) DUA (UA)

Figure 2: Scalability of the proposed retaliatory attacks under varying number of clients (n).

To evaluate the scalability of the proposed retaliatory attacks (AUA and DUA), we conduct additional experiments with varying numbers of total clients n. Specifically, we assess the performance of AUA and DUA under $n \in \{5, 10, 15, 20\}$ using the Purchase100 dataset (using Retrain as the FU method), which offers sufficient training samples to support larger-scale FL setups. Figure 2(a) and (b) present the performance of AUA, evaluated by the increase in MIA accuracy and UA, respectively, while Figure 2(c) illustrates the performance of DUA, evaluated by the decrease in UA under varying client counts. The results indicate that both attacks remain consistently effective as the scale of the FL system increases.

F.2 Impact of the Number of Unlearned Clients

We further evaluate the robustness of the proposed retaliatory attacks (AUA and DUA) under varying numbers of unlearning requests. Specifically, we use the Purchase100 dataset with a FL setup of $n = 10$ clients, where 1, 2, or 3 clients submit unlearning requests. We adopt Retrain as the FU method and assume that all unlearning requests are processed jointly in a single unlearning process. Figure 3 shows the performance of both AUA (measured by the increase in MIA accuracy and UA) and DUA (measured by the decrease in UA). It can be observed that both attacks maintain consistent effectiveness across different numbers of unlearning clients, highlighting their robustness and reliability.



(a) AUA (MIA) (b) AUA (UA) (c) DUA (UA)

Figure 3: Effectiveness of the proposed retaliatory attacks under varying numbers of unlearned clients.

Dataset	Metric	Concentration Level (α)									
		0.1		0.5		1		10		1000	
		Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack
Location	MIA	0.495	0.579 (↑0.084)	0.499	0.595 (↑0.096)	0.501	0.613 (↑0.112)	0.500	0.620 (↑0.120)	0.504	0.631 (↑0.127)
	UA	0.243	0.752 (↑0.509)	0.430	0.891 (↑0.461)	0.508	0.924 (↑0.416)	0.597	0.959 (↑0.362)	0.617	0.964 (↑0.347)
Purchase100	MIA	0.497	0.551 (↑0.054)	0.501	0.565 (↑0.064)	0.502	0.573 (↑0.071)	0.501	0.586 (↑0.085)	0.502	0.594 (↑0.092)
	UA	0.218	0.659 (↑0.441)	0.345	0.856 (↑0.511)	0.427	0.930 (↑0.503)	0.548	0.968 (↑0.420)	0.572	0.981 (↑0.409)

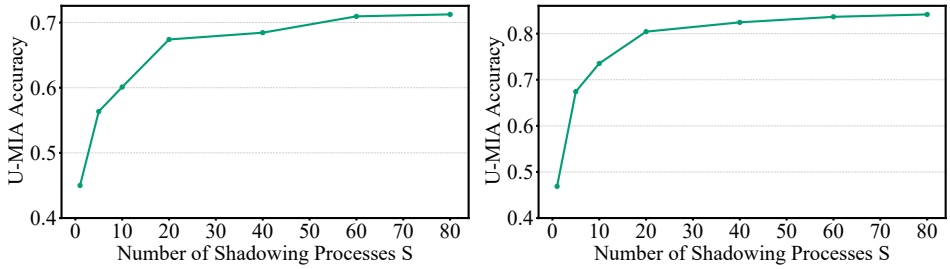
Table 2: Attack performance of the proposed AUA under varying non-IID levels.

F.3 Attack Effectiveness under Non-IID Data Distributions

To evaluate the robustness of the proposed retaliatory attacks, we further assess the attack performance of AUA and DUA under non-identically distributed (non-IID) data settings. To generate non-IID distributions across clients, we adopt the Dirichlet-based partitioning approach from (Hsu, Qi, and Brown 2019), varying the concentration parameter among $\{0.1, 0.5, 1, 10, 1000\}$. A smaller concentration value (e.g., $0.1, 0.5, 1$) indicates a higher degree of data heterogeneity across clients, while a larger value (e.g., $10, 1000$) yields distributions that approximate the IID setting.

Table 2 and Table 3 summarize the attack performance of AUA and DUA under varying levels of non-IID data distributions (using Retrain as the FU method). We observe that AUA consistently achieves strong attack performance, evidenced by increased MIA and UA values after the attack. For DUA, the attack becomes even more effective under higher non-IID levels, as reflected by a greater reduction in UA. This improvement can be attributed to the increased divergence between the data distribution of the unlearned client and that of the retained clients in non-IID settings, which allows the targeted model poisoning to more effectively degrade performance on the unlearned data distribution.

F.4 Impact of the Number of Shadowing Processes S



(a) Location (b) Purchase100

Figure 4: Impact of the number of shadowing processes S on the proposed unlearning-induced MIA (measured by U-MIA).

Given that Stage I of the proposed retaliatory attack is an unlearning-induced MIA that leverages shadow model training, we investigate the impact of the number of shadowing processes S on the performance of the trained attack models $\mathcal{A}$. Specifically, we evaluate the U-MIA accuracy of the attack models $\mathcal{A}$ under varying numbers of shadowing processes $S \in \{1, 5, 10, 20, 40, 60, 80\}$.

Figure 4 presents the results on both the Location (a) and Purchase100 (b) datasets. As shown, increasing the number of shadowing processes S generally improves the performance of the resulting MIA models. This observation is consistent with

Dataset	Metric	Concentration Level (α)									
		0.1		0.5		1		10		1000	
		Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack
Location	UA	0.243	0.145 (↓0.098)	0.430	0.323 (↓0.107)	0.508	0.425 (↓0.083)	0.597	0.519 (↓0.078)	0.617	0.550 (↓0.067)
	TA	0.434	0.442 (↑0.008)	0.484	0.461 (↓0.023)	0.536	0.517 (↓0.019)	0.610	0.589 (↓0.021)	0.624	0.606 (↓0.018)
Purchase100	UA	0.218	0.103 (↓0.115)	0.345	0.234 (↓0.111)	0.427	0.320 (↓0.107)	0.548	0.463 (↓0.085)	0.572	0.501 (↓0.071)
	TA	0.425	0.423 (↓0.002)	0.478	0.470 (↓0.008)	0.555	0.542 (↓0.013)	0.621	0.599 (↓0.022)	0.639	0.612 (↓0.027)

Table 3: Attack performance of the proposed DUA under varying non-IID levels.

prior findings (Shokri et al. 2017), which suggest that a larger number of shadowing processes enhances MIA effectiveness by providing more diverse training scenarios. However, in our context, excessively increasing S also introduces significant computational overhead, thereby reducing the overall attack efficiency. As a trade-off, we select a U-MIA accuracy around 0.8 as a practical threshold for both datasets, which corresponds to $S = 60$ for Location and $S = 20$ for Purchase100, to balance effectiveness and efficiency.

F.5 Impact of the Overfitting Degree

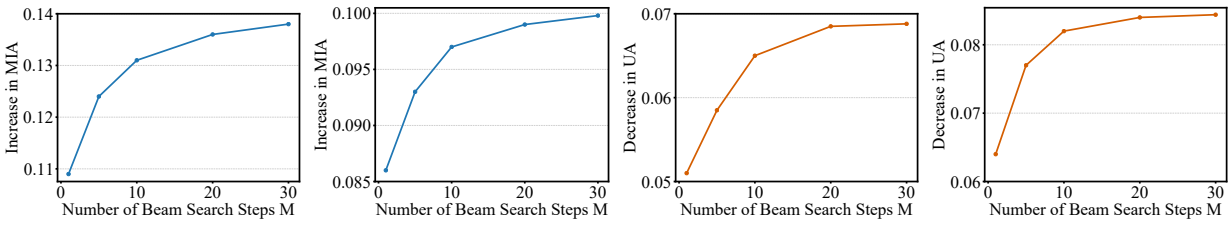
Dataset	Train Accuracy	Test Accuracy	Overfitting	U-MIA Accuracy
Location	0.961	0.734	0.227	0.341
	0.987	0.649	0.338	0.402
	0.996	0.631	0.365	0.668
	1.000	0.605	0.395	0.714
	1.000	0.587	0.413	0.745
Purchase100	0.885	0.628	0.257	0.439
	0.923	0.687	0.236	0.543
	0.991	0.671	0.320	0.798
	1.000	0.635	0.365	0.810
	1.000	0.601	0.399	0.826

Table 4: Impact of model overfitting on the effectiveness of the proposed unlearning-induced MIA (measured by U-MIA).

Given that the effectiveness of MIA fundamentally relies on the overfitting of the target model, we further investigate how different levels of overfitting impact the proposed unlearning-induced MIA. Specifically, we vary the overfitting level of the global model before unlearning ($\mathcal{M}_{\text{before}}$), quantified as the difference between training and testing accuracy, and evaluate the attack performance using the U-MIA accuracy.

Table 4 presents the U-MIA accuracy achieved by the proposed unlearning-induced MIA on both the Location and Purchase100 datasets under different overfitting levels. We observe that a higher degree of overfitting in $\mathcal{M}_{\text{before}}$ consistently leads to improved attack performance, as indicated by increased U-MIA accuracy. This suggests that model overfitting is a primary contributing factor to information leakage during the FU process, and also serves as a key enabler of the proposed retaliatory attack's success.

F.6 Impact of the Maximum Number of Beam Search Steps M



(a) AUA (Location) (b) AUA (Purchase100) (c) DUA (Location) (d) DUA (Purchase100)

Figure 5: Impact of the maximum number of beam search steps M on attack performance.

In this section, we investigate the impact of the maximum number of beam search steps M on the effectiveness of the proposed retaliatory attacks. Specifically, we vary the number of steps as $M \in \{1, 5, 10, 20, 30\}$ and evaluate the resulting

performance gains for both AUA and DUA on the Location and Purchase100 datasets. The results are shown in Figure 5. As observed, increasing the number of beam search steps consistently improves attack performance across both datasets and both attack types.

However, this performance improvement comes at the cost of increased computational overhead, especially since the beam search operates at the sample level. To balance effectiveness and efficiency, we adopt $M = 10$ as a practical trade-off in our main experiments. We have also conducted additional experiments varying other hyperparameters used in the beam search refinement (e.g., beam size B, candidate count U, and number of features flipped per step V). These parameters show relatively smaller impact on the final attack performance compared to M.

F.7 Complexity Analysis

Dataset	Avg. Time per Global Round (s)	Avg. Time for Reconstruction (s)	Required Global Rounds
Location	87.73	4648.26	53
Purchase100	35.97	1271.55	36

Table 5: Complexity analysis of the proposed retaliatory attacks.

Given the sophisticated design of the proposed retaliatory attacks, we provide a detailed complexity analysis in this section. As illustrated in Figure 1 of the main paper, the primary computational overhead arises in Stage I and Stage II of the attack, where the adversarial client reconstructs the unlearned data from the unlearned client(s). This process involves training a set of unlearning-induced MIA models followed by a coarse-to-fine data generation procedure. As claimed in the main paper, this phase takes place over several global communication rounds, during which the adversarial client conducts local data reconstruction while behaving benignly within the FL process.

To assess the feasibility and efficiency of this process in practical FL deployments, we conduct a system-level complexity evaluation. Specifically, we consider a typical resource-constrained FL setting (Luo et al. 2021), where each participating client is simulated using a Raspberry Pi 4B to represent an edge device. In contrast, the adversarial client is assumed to have access to more powerful computing resources (modeled using an NVIDIA RTX 3090), reflecting the realistic assumption that such attacks are likely launched by clients with both technical expertise and access to high-performance hardware.

We measure two key metrics: (i) the average time required for a benign client to complete one global communication round (including local training and communication latency with the server), and (ii) the average time required for the adversarial client to complete the unlearned data reconstruction process locally. Based on these, we calculate the number of global rounds that would be required to complete the reconstruction in parallel with normal FL training.

Table 5 presents the results. For the Location and Purchase100 datasets, the reconstruction process takes an equivalent of 53 and 36 global rounds, respectively. After this, the adversarial client can proceed with executing either AUA or DUA. These findings confirm that the proposed retaliatory attacks are not only effective but also practically feasible within realistic FL environments.

F.8 Potential Defenses

Defense	Location						Purchase100					
	Stage I	AUA		DUA		Runtime (s)	Stage I	AUA		DUA		Runtime (s)
	U-MIA	MIA	UA	UA	TA		U-MIA	MIA	UA	UA	TA	
Before Attack	-	0.502	0.606	0.606	0.611	-	-	0.499	0.578	0.578	0.637	-
No Defense (Retrain)	0.709	0.629	0.950	0.544	0.598	9.01	0.804	0.597	0.975	0.497	0.619	49.25
Gradient Pruning	0.340	0.509	0.620	0.582	0.585	24.55	0.418	0.510	0.599	0.562	0.601	102.81
Differential Privacy	0.365	0.507	0.591	0.566	0.580	13.33	0.450	0.507	0.603	0.569	0.608	50.48

Table 6: Performance of the proposed defenses against retaliatory attacks.

While this paper primarily focuses on the attack perspective from an adversarial client, in this section, we also explore two possible defenses against the proposed retaliatory attacks. Since the foundation of these attacks lies in the exploitation of additional information inadvertently leaked during the unlearning process, our defense efforts are directed toward mitigating such information leakage during the FU phase itself, rather than defending against the subsequent AUA or DUA poisoning steps. By eliminating this root cause of leakage, one can proactively prevent not only the specific retaliatory attacks presented in this paper, but also other potential variants that exploit similar vulnerabilities.

Gradient Pruning. Our first defense incorporates gradient pruning during the retraining phase of the global model in the FU process. Gradient pruning is a gradient sparsification technique that removes local parameter updates with small magnitudes,

retaining only the most significant gradient signals. This method has been widely applied in privacy-preserving training to reduce the risk of sensitive information leakage through gradients (Zhu, Liu, and Han 2019; Xue et al. 2024). Here, we explore its effectiveness against the proposed retaliatory attack. Inspired by (Gao et al. 2021), we adopt a simplified pruning strategy that preserves only the top-k updates with the largest absolute magnitudes. Specifically, during each local training phase in the FU process, we first compute the parameter-wise updates (i.e., the difference between the model parameters before and after local training), then rank them by absolute value in descending order. A pruning threshold is determined based on the k-th largest update, and all updates with smaller magnitudes are set to zero. In our implementation, k is chosen to retain the top 10% of updates. By sparsifying the gradients in this manner, the defense reduces the amount of fine-grained information encoded in client updates, thereby weakening the adversary's ability to infer training data of the unlearned client(s) based on the discrepancy between the pre-unlearning and post-unlearning global models.

Differential Privacy (DP). Our second defense incorporates DP into the retraining phase of the FU process. DP mitigates information leakage by ensuring that the inclusion or exclusion of any individual training sample has a limited impact on the model output (Dwork et al. 2006). Previous studies have shown that DP is effective in mitigating MIAs (Biswas et al. 2020; Nasr et al. 2021; Chen et al. 2021). Given that Stage I of the proposed retaliatory attack introduces a novel unlearning-induced MIA, we further explore the effectiveness of DP in countering such threats within the FU context. Specifically, we adopt the differentially-private stochastic gradient descent (DP-SGD) mechanism (Abadi et al. 2016) during client-side training when retraining the model from scratch. At each update step, calibrated Gaussian noise is added to the original gradient vector g, resulting in $g \leftarrow g + \mathcal{N}(0, \Delta_f^2 \sigma^2 \mathbf{I})$, where $\mathbf{I}$ denotes the identity matrix, $\Delta_f = \frac{1}{|\mathcal{B}|}$ represents the gradient sensitivity (with $|\mathcal{B}|$ being the local batch size), and σ is the standard deviation of the noise. The value of σ is determined by the target privacy parameters ε (privacy budget) and δ (failure probability), following the standard DP-SGD formula: $\sigma = \frac{\sqrt{2\ln(1.25/\delta)}}{\varepsilon}$. In our implementation, we set $\varepsilon = 3$ and $\delta = 10^{-5}$. By injecting noise directly into local updates during retraining, this defense mitigates potential information leakage throughout the FU process, thereby countering the proposed retaliatory attacks at their source.

Results and Discussion. Table 6 summarizes the effectiveness of the two proposed defense strategies against the proposed retaliatory attacks (AUA and DUA) on both the Location and Purchase100 datasets. For comparison, we also include baseline results obtained before the attack and without any defense (Retrain). As shown, both defenses substantially mitigate the privacy leakage in Stage I of the attack (i.e., the unlearning-induced MIA), as evidenced by a significant reduction in U-MIA accuracy to a level close to random guessing. Additionally, the results across all evaluation metrics from both AUA and DUA indicate a clear suppression of attack effectiveness. However, since both defenses are implemented as additional components during the retraining phase of unlearning, they introduce notable computational overhead, thereby compromising a core objective of FU methods—efficiency (as reflected in the reported runtime, which captures the total retraining cost). This trade-off highlights the urgent need for FU approaches that are both secure-aware and computationally efficient, which we identify as an important direction and challenge for future research.

F.9 Extension to Sample-level Federated Unlearning

Dataset	AUA				DUA			
	MIA		UA		UA		TA	
	Before	After Attack	Before	After Attack	Before	After Attack	Before	After Attack
Location	0.505	0.598 (↑0.093)	0.611	0.959 (↑0.348)	0.611	0.558 (↓0.053)	0.625	0.599 (↓0.026)
Purchase100	0.503	0.581 (↑0.078)	0.610	0.964 (↑0.354)	0.610	0.539 (↓0.071)	0.642	0.617 (↓0.025)

Table 7: Attack effectiveness of AUA and DUA in sample-level FU.

While the proposed retaliatory attack is primarily designed for client-level FU, where participating clients can request the removal of their entire contribution from the global model, we demonstrate in this section that our attack framework can be readily adapted to sample-level FU (where clients are allowed to remove only a portion of their training data) with minimal modifications. Specifically, the only adaptation required is in Stage I(a) of our attack pipeline during shadow model training. For simulated unlearned clients, instead of removing their entire training data, only a subset (e.g., $\frac{1}{2}$) is designated as unlearned data ($\mathcal{D}^s_{\text{unlearned}}$), while the remaining data is treated as retained data ($\mathcal{D}^s_{\text{retained}}$) and used to train the shadow global model after unlearning ($\mathcal{M}^s_{\text{after}}$). This slight modification enables a more effective unlearning-induced MIA by allowing the shadowing process to better mimic the behavior of sample-level FU.

Table 7 presents the primary results of the proposed AUA and DUA under sample-level FU on the Location and Purchase100 datasets, using retraining-from-scratch as the FU method. We observe that both attacks achieve similar performance to the client-level FU setting, with AUA showing a substantial increase in both MIA and UA, and DUA causing a clear degradation specifically in UA. This extension highlights the generalizability of the proposed retaliatory attacks and underscores that privacy leakage remains a common vulnerability across different unlearning scenarios.

G Ethical Statement and Societal Impacts

Since this study introduces a novel class of retaliatory attacks against FU that could be exploited by malicious users to compromise the global model during FL, we acknowledge the clear ethical implications of studying such attack vectors. However, the primary aim of this research is to investigate a previously underexplored security vulnerability, namely, the potential privacy leakage introduced by the incorporation of FU mechanisms. This is particularly important given that MU and FU have recently emerged as rapidly developing areas of research. Moreover, we propose several potential defenses that can mitigate the risks posed by these attacks (see Appendix F.8), demonstrating that such vulnerabilities are addressable, albeit at the cost of unlearning efficiency. We hope this work raises awareness of broader security considerations within the unlearning research community and contributes to the development of more robust and security-aware FU methods.

References

Abadi, M.; Chu, A.; Goodfellow, I.; McMahan, H. B.; Mironov, I.; Talwar, K.; and Zhang, L. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 308–318.

Balunovic, M.; Dimitrov, D. I.; Staab, R.; and Vechev, M. 2022. Bayesian Framework for Gradient Leakage. In *International Conference on Learning Representations*.

Biswas, S.; Dong, Y.; Kamath, G.; and Ullman, J. 2020. Coinpress: Practical private mean and covariance estimation. *Advances in Neural Information Processing Systems*, 33: 14475–14485.

Bourtoule, L.; Chandrasekaran, V.; Choquette-Choo, C. A.; Jia, H.; Travers, A.; Zhang, B.; Lie, D.; and Papernot, N. 2021. Machine unlearning. In *2021 IEEE symposium on security and privacy (SP)*, 141–159. IEEE.

Brophy, J.; and Lowd, D. 2021. Machine unlearning for random forests. In *International Conference on Machine Learning*, 1092–1104. PMLR.

Cao, Y.; and Yang, J. 2015. Towards making systems forget with machine unlearning. In *2015 IEEE symposium on security and privacy*, 463–480. IEEE.

Carlini, N.; Chien, S.; Nasr, M.; Song, S.; Terzis, A.; and Tramer, F. 2022. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, 1897–1914. IEEE.

Cha, S.; Cho, S.; Hwang, D.; Lee, H.; Moon, T.; and Lee, M. 2024. Learning to unlearn: Instance-wise unlearning for pre-trained classifiers. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, 11186–11194.

Che, T.; Zhou, Y.; Zhang, Z.; Lyu, L.; Liu, J.; Yan, D.; Dou, D.; and Huan, J. 2023. Fast federated machine unlearning with nonlinear functional theory. In *International conference on machine learning*, 4241–4268. PMLR.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2021. When machine unlearning jeopardizes privacy. In *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, 896–911.

Choquette-Choo, C. A.; Tramer, F.; Carlini, N.; and Papernot, N. 2021. Label-only membership inference attacks. In *International conference on machine learning*, 1964–1974. PMLR.

Di, J. Z.; Douglas, J.; Acharya, J.; Kamath, G.; and Sekhari, A. 2022. Hidden poison: Machine unlearning enables camouflaged poisoning attacks. In *NeurIPS ML Safety Workshop*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, 265–284. Springer.

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, 1322–1333.

Gao, W.; Guo, S.; Zhang, T.; Qiu, H.; Wen, Y.; and Liu, Y. 2021. Privacy-preserving collaborative learning with automatic transformation search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 114–123.

Gao, X.; Ma, X.; Wang, J.; Sun, Y.; Li, B.; Ji, S.; Cheng, P.; and Chen, J. 2024. Verifi: Towards verifiable federated unlearning. *IEEE Transactions on Dependable and Secure Computing*.

Geiping, J.; Bauermeister, H.; Dröge, H.; and Moeller, M. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in neural information processing systems*, 33: 16937–16947.

Ginart, A.; Guan, M.; Valiant, G.; and Zou, J. Y. 2019. Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems*, 32.

Golatkar, A.; Achille, A.; Ravichandran, A.; Polito, M.; and Soatto, S. 2021. Mixed-privacy forgetting in deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 792–801.

Graves, L.; Nagisetty, V.; and Ganesh, V. 2021. Amnesiac machine learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 11516–11524.

Guo, C.; Goldstein, T.; Hannun, A.; and Van Der Maaten, L. 2020. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning*, 3832–3842.

Halimi, A.; Kadhe, S. R.; Rawat, A.; and Angel, N. B. 2022. Federated Unlearning: How to Efficiently Erase a Client in FL? In *International Conference on Machine Learning*.

He, Z.; Zhang, T.; and Lee, R. B. 2019. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, 148–162.

Hsu, T.-M. H.; Qi, H.; and Brown, M. 2019. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*.

Hu, H.; Salcic, Z.; Sun, L.; Dobbie, G.; Yu, P. S.; and Zhang, X. 2022. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s): 1–37.

Huang, Y.; Gupta, S.; Song, Z.; Li, K.; and Arora, S. 2021. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in neural information processing systems*, 34: 7232–7241.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Y.; Chen, C.; Zheng, X.; and Zhang, J. 2023. Federated unlearning via active forgetting. *arXiv preprint arXiv:2307.03363*.

Liu, G.; Ma, X.; Yang, Y.; Wang, C.; and Liu, J. 2021. Federaser: Enabling efficient client-level data removal from federated learning models. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*, 1–10. IEEE.

Liu, Y.; Xu, L.; Yuan, X.; Wang, C.; and Li, B. 2022a. The right to be forgotten in federated learning: An efficient realization with rapid retraining. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, 1749–1758. IEEE.

Liu, Y.; Zhao, Z.; Backes, M.; and Zhang, Y. 2022b. Membership inference attacks by exploiting loss trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2085–2098.

Liu, Z.; Dou, G.; Chien, E.; Zhang, C.; Tian, Y.; and Zhu, Z. 2024a. Breaking the trilemma of privacy, utility, and efficiency via controllable machine unlearning. In *Proceedings of the ACM Web Conference 2024*, 1260–1271.

Liu, Z.; Jiang, Y.; Jiang, W.; Guo, J.; Zhao, J.; and Lam, K.-Y. 2024b. Guaranteeing data privacy in federated unlearning with dynamic user participation. *IEEE Transactions on Dependable and Secure Computing*.

Liu, Z.; Jiang, Y.; Shen, J.; Peng, M.; Lam, K.-Y.; Yuan, X.; and Liu, X. 2024c. A survey on federated unlearning: Challenges, methods, and future directions. *ACM Computing Surveys*, 57(1): 1–38.

Liu, Z.; Wang, T.; Huai, M.; and Miao, C. 2024d. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14115–14123.

Long, Y.; Wang, L.; Bu, D.; Bindschaedler, V.; Wang, X.; Tang, H.; Gunter, C. A.; and Chen, K. 2020. A pragmatic approach to membership inferences on machine learning models. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, 521–534. IEEE.

Luo, B.; Li, X.; Wang, S.; Huang, J.; and Tassiulas, L. 2021. Cost-effective federated learning design. In *IEEE INFOCOM 2021-IEEE conference on computer communications*, 1–10. IEEE.

Marchant, N. G.; Rubinstein, B. I.; and Alfeld, S. 2022. Hard to forget: Poisoning attacks on certified machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 7691–7700.

Mehta, R.; Pal, S.; Singh, V.; and Ravi, S. N. 2022. Deep unlearning via randomized conditionally independent hessians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10422–10431.

Nasr, M.; Songi, S.; Thakurta, A.; Papernot, N.; and Carlin, N. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*, 866–882. IEEE.

Neel, S.; Roth, A.; and Sharifi-Malvajerdi, S. 2021. Descent-to-delete: Gradient-based methods for machine unlearning. In *Algorithmic Learning Theory*, 931–962. PMLR.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; Ng, A. Y.; et al. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, 7. Granada.

Nguyen, N.-B.; Chandrasegaran, K.; Abdollahzadeh, M.; and Cheung, N.-M. 2023. Re-thinking model inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16384–16393.

Nguyen, Q. P.; Low, B. K. H.; and Jaillet, P. 2020. Variational bayesian unlearning. *Advances in Neural Information Processing Systems*, 33: 16025–16036.

Qian, W.; Zhao, C.; Le, W.; Ma, M.; and Huai, M. 2023. Towards understanding and enhancing robustness of deep learning models against malicious unlearning attacks. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1932–1942.

Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; and Backes, M. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*.

Sekhari, A.; Acharya, J.; Kamath, G.; and Suresh, A. T. 2021. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34: 18075–18086.

Sheng, X.; Bao, W.; and Ge, L. 2024. Robust federated unlearning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 2034–2044.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 3–18. IEEE.

Su, N.; and Li, B. 2023. Asynchronous federated unlearning. In *IEEE INFOCOM 2023-IEEE conference on computer communications*, 1–10. IEEE.

Tao, Y.; Wang, C. L.; Pan, M.; Yu, D.; Cheng, X.; and Wang, D. 2024. Communication Efficient and Provable Federated Unlearning. In *50th International Conference on Very Large Data Bases, VLDB 2024*.

Wang, F.; Li, B.; and Li, B. 2023. Federated unlearning and its privacy threats. *IEEE Network*, 38(2): 294–300.

Wang, J.; Guo, S.; Xie, X.; and Qi, H. 2022. Federated unlearning via class-discriminative pruning. In *Proceedings of the ACM web conference 2022*, 622–632.

Wang, K.-C.; Fu, Y.; Li, K.; Khisti, A.; Zemel, R.; and Makhzani, A. 2021. Variational model inversion attacks. *Advances in Neural Information Processing Systems*, 34: 9706–9719.

Wang, W.; Ma, Q.; Zhang, Z.; Liu, Y.; Liu, Z.; and Fang, M. 2025. Poisoning Attacks and Defenses to Federated Unlearning. In *Companion Proceedings of the ACM on Web Conference 2025*, 1365–1369.

Wang, W.; Tian, Z.; Zhang, C.; and Yu, S. 2024a. Machine unlearning: A comprehensive survey. *arXiv preprint arXiv:2405.07406*.

Wang, Z.; Gao, X.; Wang, C.; Cheng, P.; and Chen, J. 2024b. Efficient vertical federated unlearning via fast retraining. *ACM Transactions on Internet Technology*, 24(2): 1–22.

Wu, C.; Zhu, S.; and Mitra, P. 2023. Unlearning Backdoor Attacks in Federated Learning. In *ICLR 2023 Workshop on Backdoor Attacks and Defenses in Machine Learning*.

Wu, L.; Guo, S.; Wang, J.; Hong, Z.; Zhang, J.; and Ding, Y. 2022. Federated unlearning: Guarantee the right of clients to forget. *IEEE Network*, 36(5): 129–135.

Xiong, Z.; Li, W.; Li, Y.; and Cai, Z. 2023. Exact-fun: an exact and efficient federated unlearning approach. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1439–1444. IEEE.

Xue, L.; Hu, S.; Zhao, R.; Zhang, L. Y.; Hu, S.; Sun, L.; and Yao, D. 2024. Revisiting gradient pruning: A dual realization for defending against gradient attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 6404–6412.

Yan, H.; Li, X.; Guo, Z.; Li, H.; Li, F.; and Lin, X. 2022. ARCANE: An Efficient Architecture for Exact Machine Unlearning. In *IJCAI*, volume 6, 19.

Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2015. Modeling User Activity Preference by Leveraging User Spatial Temporal Characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.

Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, 268–282. IEEE.

Yuan, W.; Yin, H.; Wu, F.; Zhang, S.; He, T.; and Wang, H. 2023. Federated unlearning for on-device recommendation. In *Proceedings of the sixteenth ACM international conference on web search and data mining*, 393–401.

Zhang, L.; Zhu, T.; Zhang, H.; Xiong, P.; and Zhou, W. 2023a. Fedrecovery: Differentially private machine unlearning for federated learning frameworks. *IEEE Transactions on Information Forensics and Security*, 18: 4732–4746.

Zhang, R.; Guo, S.; Wang, J.; Xie, X.; and Tao, D. 2023b. A Survey on Gradient Inversion: Attacks, Defenses and Future Directions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, 5678–685.

Zhang, Y.; Jia, R.; Pei, H.; Wang, W.; Li, B.; and Song, D. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 253–261.

Zhao, C.; Qian, W.; Ying, R.; and Huai, M. 2023a. Static and sequential malicious attacks in the context of selective forgetting. *Advances in Neural Information Processing Systems*, 36: 74966–74979.

Zhao, Y.; Wang, P.; Qi, H.; Huang, J.; Wei, Z.; and Zhang, Q. 2023b. Federated unlearning with momentum degradation. *IEEE Internet of Things Journal*, 11(5): 8860–8870.

Zhu, L.; Liu, Z.; and Han, S. 2019. Deep leakage from gradients. *Advances in neural information processing systems*, 32.

Zhu, X.; Li, G.; and Hu, W. 2023. Heterogeneous federated knowledge graph embedding learning and unlearning. In *Proceedings of the ACM web conference 2023*, 2444–2454.