United We Defend: Collaborative Membership Inference Defenses in Federated Learning

Li Bai[1], Junxu Liu[1,2], Sen Zhang[1], Xinwei Zhang[1], Qingqing Ye[1], Haibo Hu[1,2] *
Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University[1]
PolyU Research Centre for Privacy and Security Technologies in Future Smart Systems[2]

Abstract

Membership inference attacks (MIAs), which determine whether a specific data point was included in the training set of a target model, have posed severe threats in federated learning (FL). Unfortunately, existing MIA defenses, typically applied independently to each client in FL, are ineffective against powerful trajectory-based MIAs that exploit temporal information throughout the training process to infer membership status. In this paper, we investigate a new FL defense scenario driven by heterogeneous privacy needs and privacy-utility trade-offs, where only a subset of clients are defended, as well as a collaborative defense mode where clients cooperate to mitigate membership privacy leakage. To this end, we introduce CoFedMID, a collaborative defense framework against MIAs in FL, which limits local model memorization of training samples and, through a defender coalition, enhances privacy protection and model utility. Specifically, CoFedMID consists of three modules: a class-guided partition module for selective local training samples, a utility-aware compensation module to recycle contributive samples and prevent their over-confidence, and an aggregation-neutral perturbation module that injects noise for cancellation at the coalition level into client updates. Extensive experiments on three datasets show that our defense framework significantly reduces the performance of seven MIAs while incurring only a small utility loss. These results are consistently verified across various defense settings.

1 Introduction

Federated learning (FL) [1] has emerged as a popular distributed machine learning paradigm in which multiple clients (e.g., mobile devices or institutions) collaboratively train a global model by exchanging local model updates (e.g., weights or gradients), rather than sharing their raw data. Despite direct data leakage being mitigated, indirect privacy risks from potential adversaries (e.g., the honest-but-curious server or clients) remain a concern. One prominent threat is membership inference attacks (MIAs), which aim to determine whether a specific data sample was part of a client's local training dataset [2, 3]. Compared to centralized settings, adversaries in FL often have access to detailed model information and historical model snapshots throughout the training process. This enables powerful trajectory-based MIAs against both local and global models [4–6], thereby making effective defenses considerably more challenging [6, 7].

Beyond the limitations imposed by those powerful attacks, current defense mechanisms, whether specifically designed for FL or adapted from centralized frameworks, typically enforce uniform and independent privacy safeguards across all clients. In practice, such one-size-fits-all strategies fail to reflect the diverse privacy needs of participants. Clients with low-risk or publicly available data may not require the same level of protection as those handling sensitive information [8–10], especially when stronger defenses come at the cost of reduced utility [11, 12]. Furthermore, existing research indicates that implementing defenses in isolation may yield weaker privacy guarantees compared to those achievable through collaborative approaches. For example, secure multi-party computation [4, 13] demonstrates that collaborative computation can provide stronger privacy protections than operating independently.

In this paper, we focus on a novel defense setting that differs from previous works in two key aspects: (1) *Partial Scenario:* only a subset of clients choose to implement defense mechanisms according to their actual privacy needs; and (2) *Collaborative Mode:* defended clients can organize into a group, referred to as a defender coalition, to collaboratively mitigate MIAs. Such collaboration is feasible as many real-world FL scenarios present conditions and opportunities that naturally enable it. For example, it can be formed by multiple devices owned by the same user in cross-device FL [14], or several institutions under a common administration in cross-silo FL [15, 16]. Nevertheless, this new defense setting introduces an additional vulnerability arising from the training divergence between defended and undefended

*Corresponding author: haibo.hu@polyu.edu.hk

clients. As shown in Figure 1(b), member samples from defended clients (the *member* line) can be easily distinguished from non-member samples from undefended clients (the *non-member-IFL* line), even though they are indistinguishable from non-members that are entirely external to the FL system (the *non-member-OFL* line).

To effectively mitigate membership privacy leakage for defended clients, it is essential to minimize behavioral discrepancies between their member samples and both types of non-member samples. Given that these behavioral differences will become more pronounced as local models increasingly memorize individual samples [17], our core idea is to limit the exposure of the training dataset, thereby constraining the extent of memorization in local models and, through federated aggregation, in the global model throughout the training process. To this end, we present **CoFedMID**, a Collaborative Federated Membership Inference Defense framework, which consists of three key modules as follows. First, to enforce the memorization constraint, we introduce a class-guided partition module that enables selective local training samples but ensures that the defender coalition captures the full data distribution to preserve model utility. Restricting the usage of training samples inevitably leads to utility degradation, especially as the coalition size increases. We then introduce a utility-aware compensation module, which reintroduces informative samples according to their contribution to model performance. Last but not least, to further enhance defense strength, CoFedMID incorporates an aggregation-neutral perturbation module to inject carefully designed random noise into client updates. It is worth noting that our framework employs a defender coalition to strengthen both privacy protection and model utility, without incurring any additional privacy cost beyond standard FL. Although this study focuses on a single coalition, our framework can be naturally extended to multiple ones.

In summary, we make the following key contributions:

- We present the first study to consider partial and collaborative defense settings in FL, beyond previous uniform and independent privacy-preserving approaches.
- We propose CoFedMID, an effective FL defense framework that mitigates local data memorization and enhances protection and utility through client collaboration.
- We design three modules to balance defense and utility: class-guided partition for selective training samples, utility-aware compensation to recycle contributive samples and prevent overconfidence, and aggregation-neutral perturbation to boost defense with noise injection.
- We evaluate CoFedMID against seven trajectory-based MIAs and six baseline defenses on three benchmark datasets, demonstrating its superior defense capability and high utility under various FL settings.

The paper is organized as follows. Section 2 and Section 3 provide the problem formulation and discuss the limitations of existing methods, respectively. Section 4 presents our CoFedMID framework, followed by experimental results in Section 5. Section 6 provides limitations and discussion, Section 7 reviews related work, and Section 8 concludes the paper.

2 Problem Formulation

2.1 Federated Learning

Federated Learning (FL) [1] is a distributed learning paradigm where multiple clients (e.g., mobile devices or institutions) collaboratively train a global model without sharing raw data. In this paper, we consider a typical supervised FL setup with a central server and a set of K clients, where each client $k \in [K]$ holds a private dataset D_k. The objective is to learn a globally shared model f with parameters $\tilde{\theta} \in \mathbb{R}^P$ by solving the following empirical risk problem

$$\tilde{\theta} = \min_{\theta \in \mathbb{R}^P} \left\{ \mathcal{L}(\theta) \triangleq \sum_{k=1}^{K} w_k \mathcal{L}_k(\theta; D_k) \right\},$$
$$\text{where } \mathcal{L}_k(\theta; D_k) \triangleq \frac{1}{|D_k|} \sum_{(\mathbf{x},\mathbf{y}) \in D_k} \ell(f_\theta(\mathbf{x}), \mathbf{y}).$$

Note that $\mathbf{y} \in [0,1]^N$ is a one-hot encoded label over a class space $\mathcal{S}$ of size N, where $\mathbf{y}_c = 1$ indicates the ground truth class. $f_\theta(\mathbf{x}) \in \mathbb{R}^N$ refers to the predicted confidence probabilities across the N classes and $\ell(\cdot,\cdot)$ denotes the loss function (e.g., cross-entropy). The most fundamental approach for solving the above FL optimization problem is federated averaging (FedAvg) [1], where the parameters of the global model are updated through iterative training rounds. During each round $t \in [T]$, each client initializes its local parameters with the current global parameters $\tilde{\theta}^t$, performs local optimization on its respective dataset D_k, and sends the updated local parameters θ_k^t to the server. The server then aggregates the received parameters using a weighted average, i.e., $\tilde{\theta}^{t+1} = \sum_{k=1}^{K} w_k \cdot \theta_k^t$.

2.2 Attack Formulation

Attack Objective. Similar to how it works in a centralized setting, MIA in FL aims to determine whether a given sample $(\mathbf{x}, \mathbf{y})$ was included in the target client's local training dataset D_{tar}. Formally, this can be formalized via a membership inference function:

$$\mathcal{A}(\theta, (\mathbf{x}, \mathbf{y})) = \begin{cases} 1, & \text{if } (\mathbf{x}, \mathbf{y}) \in D_{\text{tar}} \quad \text{(member)} \\ 0, & \textit{otherwise} \quad \text{(non-member)} \end{cases}$$

where $\mathcal{A}$ denotes an attack model that takes as input a sample $(\mathbf{x}, \mathbf{y})$ and the model parameters θ (e.g., local or global), and outputs a binary prediction indicating whether $(\mathbf{x}, \mathbf{y})$ is a member of D_{tar}.

Attacker Capability. Following prior work [4–6, 18], we assume that adversaries may be either recipients of the global model parameters $\tilde{\theta}^t$ (i.e., the other clients) or recipients of local model parameters from all participating clients $\{\theta_k^t\}_{k \in [K]}$

(i.e., the central server). Specifically, these adversaries are considered *honest-but-curious*, i.e., they adhere to the FL protocol but attempt to infer whether a particular sample is included in the *target client*'s local dataset. We refer to attacks that exploit the temporal evolution of these model snapshots to infer membership information as **trajectory-based MIAs**.

2.3 Defense Formulation

Uniform and Partial Scenarios. Without loss of generality, we design our framework from the client's perspective. Based on whether all clients participate in the defense, existing defense scenarios can be categorized as either *uniform* or *partial*. In uniform defenses, all clients are involved, whereas in partial defenses, only a subset of clients participates. The uniform defense scenario has been extensively explored in existing works [6, 19], while the partial defense scenario is overlooked. However, it is realistic and necessary in practice: clients with sensitive data require strong protection, while those holding low-risk or public data need few or no additional defenses [8–10]. Moreover, enforcing a uniform defense strategy across all clients may unnecessarily compromise the model performance [7, 20]. This arises from the inherent trade-off between privacy protection and model utility in most defense mechanisms [6, 19, 21, 22]. Unlike prior studies that work on a uniform defense setting, this work focuses on a partial defense scenario.

Independent and Collaborative Modes. As guided by the FL protocol, each client communicates only with the central server and does not exchange any information with other clients. Likewise, existing membership inference defenses are developed individually and independently in FL. We refer to this case as the independent mode. Beyond the independent mode, we explore a collaborative mode in which clients cooperate to defend against MIAs. We introduce the term *defender coalition* to denote a group of clients that collaboratively implement defense strategies, assuming that all clients within the coalition behave honestly and faithfully follow the designed defense framework. Such coalition-based cooperation may occur in real-world FL, e.g., multiple devices owned by a single user in cross-device FL [14], or institutions under shared administration in cross-silo FL [15, 16]. Similar to secure multi-party computation [13], this collaborative way can offer new opportunities to enhance privacy protection and maintain model utility in FL.

Defense Objective. Our defense framework is designed to achieve the following objectives:

- Strong membership privacy: Mitigate MIAs launched by both the server and clients.
- High model utility: Preserve model utility with performance comparable to that of undefended systems.
- Low overhead: Impose minimal communication and computational overhead compared with standard FL.
- Robustness: Remain effective against adaptive adversaries that attempt to bypass the defense.

3 Existing Defenses and Limitations

3.1 Existing Defenses

We now review existing defense mechanisms that are typically designed for the uniform scenario and operate independently.

Perturbation-based Defenses. Since local updates (i.e., weights or gradients) in FL can inadvertently reveal membership information to potential attackers, perturbation-based defenses are designed to mitigate this risk by obscuring these exchanged updates. Common strategies include employing model sparsification to limit exposed parameters [3, 23], injecting noise to mask original signals [6, 7, 24], and leveraging differential privacy techniques to provide formal privacy guarantees [12, 25].

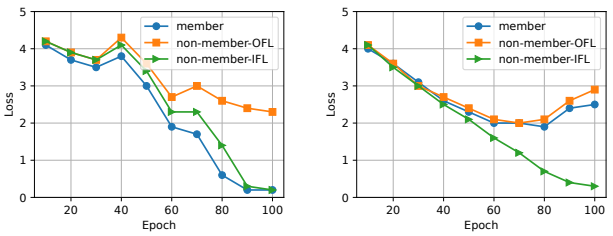
Overfitting-based Defenses. Since overfitting is a major contributor to membership leakage [26], overfitting-based defenses focus on narrowing the generalization gap between training and test samples to reduce the risk of MIAs. These defenses have been extensively studied in centralized settings, particularly for protecting against black-box attacks. Representative methods include regularization approaches (e.g., adversarial learning [27] and Mixup [21, 28]) and transfer learning techniques (e.g., knowledge distillation [29, 30]). Previous efforts to adopt centralized defenses in federated settings have failed to mitigate the threats posed by powerful trajectory-based MIAs [4, 6].

3.2 Limitations

We analyze the limitations of existing defenses based on two factors: ineffectiveness against emerging trajectory-based MIAs and inadequacy in handling complex non-IID data distributions.

Limitation 1: Existing defense mechanisms are often ineffective against emerging trajectory-based MIAs. In FL settings, attacks have evolved beyond exploiting single model updates [2, 3] to leveraging temporal information across multiple rounds, which significantly boosts the performance of advanced trajectory-based MIAs [4, 6]. This means that historical model snapshots throughout the training process become vulnerable to potential adversaries. Consequently, existing overfitting-based defenses, primarily focused on reducing overfitting at convergence and protecting a single model snapshot (e.g., the final one), are insufficient. Furthermore, perturbation-based defenses protect the entire training process by continuously adding empirical or theoretical perturbations to model updates, making it hard to balance protection strength and model utility.

Limitation 2: Existing defenses are weakened in the partial scenario due to the complexity of non-member sample distributions. Existing defenses typically consider a single



(a) No defense. (b) With HAMP defense [22].

Figure 1: Loss trajectories of members and non-members.

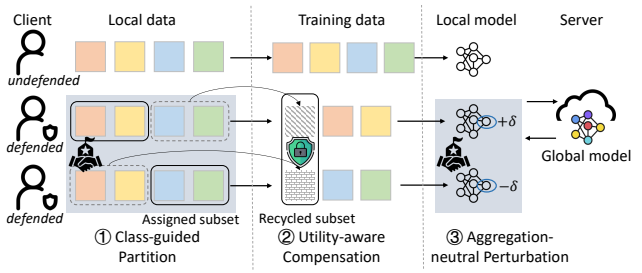


Figure 2: Overview of the proposed defense framework. It consists of three modules: ① class-guided partition, ② utility-aware compensation, and ③ aggregation-neutral perturbation. Both ① and ③ are performed in a collaborative mode.

type of non-member samples (i.e., entirely unseen by the target model) [6, 31] and overlook the distinctive characteristics of non-member distributions in FL, which can be divided into two distinct types: (1) **non-member-OFL**, denoting samples entirely *outside* the FL system (i.e., not included in any client's dataset); and (2) **non-member-IFL**, referring to samples belonging to other clients *within* the FL process but not to the target client. This heterogeneity among non-member samples introduces significant challenges for defense design, as their behavior (e.g., loss and confidence) can vary substantially after defense implementation. We present loss trajectories of member and non-member samples without any defense in Figure 1(a). Non-member-IFL samples are less vulnerable, as they are trained by other clients and exhibit similar model behavior to the target client, particularly under IID settings. However, this advantage diminishes in a partial defense scenario, where some clients actively defend while others do not. As illustrated in Figure 1(b), while the defense effectively reduces distinguishability between member and non-member-OFL samples, it unexpectedly amplifies the difference between member and non-member-IFL samples. Consequently, the divergence in training strategies between defended and undefended clients amplifies the distinction between member and non-member-IFL samples, which renders sensitive member data more vulnerable in partial defense scenarios.

4 Methodology

The ultimate goal of membership inference defenses is to ensure that member and non-member samples are statistically indistinguishable with respect to the target model's behavior [20]. However, the limitations discussed above underscore the inadequacy of existing approaches in effectively closing this gap. To address this issue, we introduce **CoFedMID** (Collaborative Federated Membership Inference Defense), a novel framework designed to defend against MIAs in the FL setting, with a specific focus on the more challenging partial defense scenario. It is worth noting that our framework can be seamlessly applied to the uniform defense setting where all clients participate, as discussed in Section 5.2. Unless otherwise specified, the term "client" in the following sections refers exclusively to participating members of the defender coalition.

Overview of CoFedMID. The core objective of CoFedMID is to offer effective defense capabilities while preserving the overall utility of the FL system. This is achieved by enabling each client to update their model parameters using a carefully selected subset of local training samples, followed by the injection of strategically designed random perturbations. Consequently, the extent of training data memorization is substantially reduced in local models and, through federated aggregation, in the global model at each training round. This approach effectively mitigates vulnerability to trajectory-based MIAs (addressing Limitation 1) and decreases the distinguishability between member samples and both types of non-member samples (addressing Limitation 2).

An overview of the framework is depicted in Figure 2. In summary, CoFedMID comprises three key modules: *class-guided partition*, *utility-aware compensation*, and *aggregation-neutral perturbation*. The first and third modules are executed collaboratively among clients, while the second one is performed independently by each client. We provide detailed descriptions of each module below.

Remark 1. *Note that the first and third modules rely on a coordinator to facilitate the collaboration process. Crucially, this coordinator neither accesses any client's private data nor model parameters, nor does it impose substantial computational overhead. In our implementation, this role is randomly assigned to an arbitrary participating client.*

4.1 Class-guided Partition

Design Intuition. The key insight underlying our method is that frequent exposure of training samples during the learning process can increase the likelihood of over-memorization [17, 26, 32, 33], leading to distinct model behaviors on training samples and elevating the risk of membership privacy leakage [34, 35]. As shown in Figure 1, the gap in average loss between member samples and both types of non-member

samples widens significantly after 60 rounds, regardless of whether defenses are applied. Inspired by this observation, our defense aims to **decrease the frequency of each training sample's usage to limit the target model's memorization.** Although similar strategies like early stopping [20, 36] have been investigated, they apply uniformly across all training samples and fail to account for sample-specific characteristics, resulting in suboptimal defense effectiveness.

We mitigate this issue by enabling clients to train their local models on a carefully selected subset of samples from specific classes during each training round. We propose a class-guided partition strategy that assigns minimally overlapping class subsets to clients, effectively limiting memorization while preserving model performance. This module operates through two subroutines: (1) *bounded class assignment*, which allocates class subsets to clients based on refined assignment principles, and (2) *decay-based assignment*, which gradually reduces the number of training samples each client uses as training progresses. Due to space constraints, we defer the pseudocode to Appendix A (Algorithm 1).

Bounded Class Assignment. We begin by partitioning the dataset classes among the coalition clients, where each client is assigned a unique subset of classes and restricts training to its corresponding local samples. To balance privacy preservation and model utility, this assignment should ensure that: 1) *minimum overlap*: class subsets assigned to different clients should exhibit minimal redundancy, thereby reducing redundant exposure of training samples and mitigating membership leakage risks; 2) *complete coverage*: the coalition as a whole should span the entire label space, ensuring that the learned model captures a comprehensive representation of the underlying data distribution. Formally, let $\mathcal{C} \subseteq [K]$ denote a defender coalition consisting of d clients, where each client $k \in \mathcal{C}$ is assigned a class subset $\mathcal{S}_k \subseteq \mathcal{S}$ with size m. The goal is to determine the minimally overlapping class subsets $\{\mathcal{S}_k\}_{k=1}^{d}$ w.r.t. coalition $\mathcal{C}$ such that:

$$\min_{\{\mathcal{S}_k\}_{k=1}^{d}} \sum_{1\le i<j\le d} |\mathcal{S}_i \cap \mathcal{S}_j| \quad \text{subject to: } \bigcup_{k=1}^{d} \mathcal{S}_k = \mathcal{S}, |\mathcal{S}_k| = m.$$

To address this assignment problem, we draw inspiration from the Johnson-type bound [37, 38], which characterizes the theoretical minimum overlap between fixed-size subsets. In our setting, this leads to a lower bound on the maximum pairwise class overlap among all clients. Formally, the minimum achievable value of the maximum class overlap between any two clients λ_{theo} is bounded as:

$$\lambda_{theo} = \max_{1\le i<j\le d} |\mathcal{S}_i \cap \mathcal{S}_j| \ge \left\lceil \frac{dm^2 - Nm}{N(d-1)} \right\rceil.$$

Motivated by this theoretical bound, we develop a heuristic algorithm that approximates near-optimal class-to-client assignments, allocating classes iteratively while controlling pairwise overlaps across the coalition. The assigned training subset D_k^{ass} for client k is then constructed by collecting all local samples whose labels belong to the allocated class subset $\mathcal{S}_k$ as follows:

$$D_k^{\text{ass}} = \{(\mathbf{x}, \mathbf{y}) \mid (\mathbf{x}, \mathbf{y}) \in D_k,\ \mathbf{y}_c \in \mathcal{S}_k\}.$$

Decay-based Assignment. Next, we dynamically adjust the class subset size m throughout the training process to further limit the exposure of training samples. Prior studies [39, 40] have demonstrated that, during the early stages of training, clients benefit from diverse data to effectively capture the underlying distribution. As the global model gradually converges and stabilizes, fewer samples are sufficient to preserve its performance. Building on this insight, we employ a linear decay function that reduces m from $m_{\max}$ to $m_{\min}$ at a rate of α over the total number of training rounds T, that is,

$$m^t = \max\left(m_{\min},\ m_{\max} - \gamma(t-1)\right) \text{ where } \gamma = \frac{m_{\max} - m_{\min}}{T-1}.$$

Here m^t denotes the class subset size at round $t \in [T]$. Nonlinear strategies, including exponential and polynomial decay, are also investigated in our experiments (see Section 5.3).

The bounded class assignment is implemented by the coordinator, who knows the label-space IDs and assigns classes to members such that their union collectively covers the entire label space. Since our assignment centers on the coalition's overall class coverage, the coordinator does not require access to the specific class distribution of any individual member. Each coalition member uses assigned classes if they have corresponding samples; otherwise, the assignments are ignored. We highlight that the effectiveness of our class-guided partition module arises not only from its core objective of minimizing the exposure of each client's local training data, but also from its deliberate disruption of the implicit mapping between local datasets and their resulting model parameters. This is achieved through our randomized, adaptive class assignment procedure that increases uncertainty for potential adversaries attempting to infer whether a target sample belongs to a client, hence further amplifying the membership privacy protection.

4.2 Utility-aware Compensation

Design Intuition. While limiting the exposure of training samples to local models can enhance the defense against MIAs, it often comes at the cost of degraded model utility due to the limited data utilization. This trade-off becomes more pronounced as the size of the defender coalition grows. A larger number of participating clients results in a smaller portion of training data being utilized, thereby leading to a significant utility loss.

To restore model performance, we propose selectively using a subset of samples that were previously excluded by the participating client in the coalition, referred to as *recycled samples*, while carefully managing their risks of membership privacy leakage. To this end, we introduce a utility-aware

compensation module that consists of two components: (1) *sample recycling*, which identifies the recycled samples at specified training rounds that significantly improve the utility of the corresponding local models, and (2) *confidence regularization*, which constrains overconfidence on these recycled samples to control privacy risks. The complete procedure is provided in the Appendix Algorithm 2.

4.2.1 Sample Recycling

We now present a high-level overview of our sample recycling process. To effectively mitigate utility degradation, two pivotal considerations must be addressed: when to recycle and which samples to recycle. Recall that as the global model converges, the number of classes m assigned to each client gradually decreases, leading to a smaller portion of training samples being selected. Based on this observation, we recommend starting sample recycling at an intermediate stage t' rather than from the outset. A practical guideline for determining t' is to monitor m and trigger recycling once it falls below a predefined threshold.

For the first consideration, we propose selecting recycled samples according to their "compensation" for utility degradation, which can be quantified by the magnitude of *loss reduction* evaluated on the updated local model upon reusing these samples. This is inspired by curriculum learning techniques [39, 41], in which the model is trained progressively by first learning from easier samples with lower losses to achieve stability, followed by learning harder ones with larger losses to enhance generalization. Specifically, we introduce the notion of a *sample interval* which refers to a subset of samples exhibiting comparable levels of learning difficulty. At each training round $t \in [t', T]$, every client k in the coalition partitions its local dataset D_k into M disjoint sample intervals, denoted as $I_k = \{I_k^1, I_k^2, \ldots, I_k^M\}$[1]. Guided by a carefully designed recycling strategy, each client then selects an optimal interval I_k^j that maximizes utility compensation, and the set of recycled samples D_k^{rec} is formed as the set difference $I_k^j \setminus D_k^{\text{ass}}$.

The key technical challenges involved in implementing this approach are twofold: 1) constructing sample intervals that precisely capture the learning difficulty of training samples; and 2) establishing a robust strategy for selecting the optimal interval. We tackle these challenges through the following three steps, with Step 1 addressing the first challenge and Steps 2-3 addressing the second.

Step 1: Sample Interval Initialization. The construction of sample intervals starts by evaluating the per-sample loss values using the initial local parameters (i.e., the global model obtained from the preceding round). Considering the distribution of these losses varies across training rounds, we apply min-max normalization to ensure that the partitioning reflects the relative difficulty of samples in the current training context. Formally, given the model parameters $\tilde{\theta}$, we denote ℓ_i as the original loss value of the i-th training sample in D_k. Then the normalized loss value is calculated as $\tilde{\ell}_i = \frac{\ell_i - \ell_{\min}}{\ell_{\max} - \ell_{\min}}$, where $\ell_{\min}$ and $\ell_{\max}$ are the minimum and maximum values among the original losses, respectively. Next, we sort D_k by normalized losses in ascending order and divide it into M intervals of approximately equal size. Consequently, the sample intervals can be initialized as:

$$I_k^j = \left[\tilde{\ell}_{(q_{j-1})}, \tilde{\ell}_{(q_j)}\right), \quad \text{for } j = 1, 2, \ldots, M,$$

where $\tilde{\ell}_{(q_j)}$ denotes the q_j-th smallest normalized loss value, with $q_j = \lfloor (j/M) \cdot |D_k| \rfloor$, $q_0 = 0$, and $q_M = |D_k|$.

This initialization guarantees that each interval initially contains an equal number of samples, thereby eliminating quantity-induced bias and facilitating fair exploration in the early learning stages. Furthermore, the design allows the number of samples per interval to be dynamically adjusted over time, reducing the recycled sample size in later stages to limit additional exposure.

Step 2: Sample Interval Evaluation. Once the sample intervals are constructed, the next step is to develop a sample recycling strategy that selects the optimal sample interval. As previously noted, our intuition is that the contribution of a training sample to utility improvement can be quantified by the reduction in loss values observed between local models trained with and without that sample [42, 43]. We treat this loss reduction as a reward signal to guide the selection of sample intervals throughout the training process.

Specifically, given an arbitrary sample interval I_k^j and its corresponding set of recycled samples D_k^{rec}, we denote θ_k' as the initial local model and denote θ_k as the updated local model trained on $D_k^{\text{rec}} \cup D_k^{\text{ass}}$. Then the reward value r w.r.t. the interval is defined as the average loss reduction computed over a validation set D_k^{val}:

$$r = \frac{1}{|D_k^{\text{val}}|} \sum_{(\mathbf{x},\mathbf{y}) \in D_k^{\text{val}}} \left[\ell(f_{\theta_k'}(\mathbf{x}), \mathbf{y}) - \ell(f_{\theta_k}(\mathbf{x}), \mathbf{y})\right]. \quad (1)$$

We attribute this reduction to D_k^{rec} as our recycling strategy prioritizes high-loss samples, which are expected to contribute more to model performance than D_k^{ass} composed of samples with varying loss levels [44]. In line with previous studies [42, 43], we rescale the range of raw rewards to $[-1, 1]$ using the 20th and 80th percentiles of historical rewards. The normalized reward $\tilde{r}$ is defined as $\tilde{r} = \max(-1, \min(1, \frac{2(r - r_{20})}{r_{80} - r_{20}} - 1))$. This transformation smooths out extreme fluctuations and stabilizes the learning signal during interval selection.

Step 3: Sample Interval Selection. In this step, our goal is to select the optimal sample interval by maximizing the reward values defined above. This process inherently involves a core trade-off between *exploitation* (i.e., choosing intervals that have previously led to substantial loss reductions)

[1] For brevity, we simplify the notations by omitting the round identifier t, and will continue to do so throughout the remainder of this section.

and *exploration* (i.e., exploring alternative intervals that may yield greater reductions). We model this selection process as a multi-armed bandit (MAB) problem [45], where each "arm" represents a sample interval. We adopt the EXP3 [43, 46] algorithm, which balances exploration and exploitation by assigning a carefully designed weight to each arm based on its observed reward, enabling adaptation to non-stationary and noisy reward distributions during training. Upon receiving a normalized reward, the algorithm updates the weight of the selected arm based on the reward and its sampling probability. Moreover, we control recycling volume when an interval contains many less informative samples, especially in the later stages. When the candidate samples exceed a predefined threshold r_l, random subsampling is applied to prevent overexposure to easy examples and ensure training resources focus on more valuable data.

4.2.2 Confidence Regularization

Intuitively, sample recycling tends to select data points with high loss gaps—often those near decision boundaries or behaving as outliers [47, 48], which are consequently more vulnerable to MIAs. Thus, to mitigate this increased vulnerability, we regularize the model's confidence on recycled samples by encouraging smoother output distributions and reducing overconfident predictions. Following [22], for each recycled sample $(\mathbf{x}, \mathbf{y}) \in D_k^{\text{rec}}$, we construct a soft label $\tilde{\mathbf{y}} \in \mathbb{R}^N$ by assigning the predicted confidence $p_c = (f_{\theta_k}(\mathbf{x}))_c$ for the ground-truth class $\mathbf{y}_c$ while evenly distributing the residual probability mass $1 - p_c$ across the other $N-1$ classes. Together with an entropy-based constraint, the confidence-regularized loss function is formulated as follows:

$$\ell_{cr}(f_{\theta_k}(\mathbf{x}), \mathbf{y}) \triangleq \mathrm{KL}(\log f_{\theta_k}(\mathbf{x}) \,\|\, \tilde{\mathbf{y}}) - \mu \cdot \mathrm{Entropy}(f_{\theta_k}(\mathbf{x})),$$

where μ is a hyperparameter balancing the two terms. Together with the standard loss function $\ell(\cdot)$, the training objective for client k is to minimize:

$$\mathcal{L}_k(\theta_k; D_k) \triangleq \sum_{(\mathbf{x},\mathbf{y}) \in D_k^{\text{rec}} \cup D_k^{\text{ass}}} \ell(f_{\theta_k}(\mathbf{x}), \mathbf{y}) + \sum_{(\mathbf{x},\mathbf{y}) \in D_k^{\text{rec}}} \ell_{cr}(f_{\theta_k}(\mathbf{x}), \mathbf{y}).$$

In summary, we propose a utility-aware compensation algorithm designed to restore model performance by selectively reusing a minimal subset of previously excluded samples. We employ a multi-armed bandit framework to dynamically prioritize sample intervals that offer the greatest utility improvement and incorporate a specialized regularizer to mitigate privacy risks associated with the recycled samples. The above process uniformly treats all training samples, classes, and protected clients. Due to dynamic class assignment among different rounds, where no fixed client-class mapping exists, samples can originate from either assigned classes or serve as recycled data.

4.3 Aggregation-neutral Perturbation

Design Intuition. Unlike the aggregated global model, local models are more vulnerable as they directly expose client-specific information to potential attackers. Inspired by multi-party secure computation [13, 49], we introduce an aggregation-neutral perturbation module to further strengthen the overall defense capability by perturbing local models before aggregation. Specifically, this module injects carefully crafted noise into the local parameters of clients within the defender coalition $\mathcal{C}$, effectively masking private information while preserving the correctness of global aggregation. To achieve this, the injected noise is constructed to meet two key criteria: (1) its weighted sum across all collaborative clients is zero, and (2) it is applied locally to only a small subset of model parameters. The complete procedure is presented in the Appendix Algorithm 3.

Noise Assignment. To satisfy the first criterion, we design the client noises to be orthogonal to the aggregation weight vector in $\mathbb{R}^d$. We first generate a preliminary Gaussian noise $\delta'_k \sim \mathcal{N}(0, \sigma^2 I)$ for each client $k \in \mathcal{C}$ independently, where σ is the perturbation strength. Given the aggregation weights of all clients $\mathbf{w} = [w_1, \ldots, w_d]$, the scalar noise δ_k for client k is generated by projecting a base noise δ'_k onto the direction orthogonal to $\mathbf{w}$:

$$\delta_k = \delta'_k - \frac{w_k}{\|\mathbf{w}\|_2^2} \sum_{j=1}^{d} w_j \delta'_j, \quad \text{for } k = 1, \ldots, d.$$

This construction guarantees that the perturbations cancel out during aggregation, that is, $\sum_{k=1}^{d} w_k \delta_k = 0$.

Model Perturbation. To satisfy the second criterion, perturbations are selectively applied to a subset of model parameters (or gradients) so as to guarantee their defensive efficacy while incurring small disruption. Our approach perturbs only the final few layers while keeping earlier layers unchanged. This is because deeper layers (e.g., the classification layer) closer to the output exert a greater influence on predictions [50, 51]. Specifically, we begin by flattening all model parameters into a one-dimensional vector, arranged sequentially by layer. For a perturbation ratio $r_P \in (0, 1.0]$, each client perturbs the last $P' \triangleq \lfloor r_P \cdot P \rfloor$ parameters as

$$\theta_k^{(i)} \leftarrow \theta_k^{(i)} + \delta_k, \quad \text{where } i \in [P - P' + 1, P].$$

Since clients may not know their exact aggregation weights w_k, a practical approach is to approximate them by the size of each client's local dataset, assuming that aggregation weights are proportional to data volume. For FedAvg, we compute weights based on each client's local training data volume rather than assigned classes. This reduces communication overhead, as members do not need to report data size every round, and helps hide the defense coalition from an honest-but-curious server.

4.4 Algorithm Summary

We present the complete pseudocode of CoFedMID in the Appendix Algorithm 4. Prior to training, a randomly chosen client serves as the coordinator, allocating data classes and configuring noise perturbations for all defended members. Each client is then assigned a series of class subsets with controlled overlap, which is dynamically reduced each round following a decay schedule. Subsequently, during each training round, clients train on partial local data consisting of assigned and recycled samples, balancing utility preservation with privacy protection. Moreover, their model parameters are then perturbed with allocated neutral noise to obscure individual training behaviors. Meanwhile, clients outside the coalition conduct standard local training. The server ultimately aggregates all updated parameters via weighted averaging. Additionally, while our design focuses on a single defender coalition, it can be extended to multiple ones, with each employing the proposed defense framework independently.

Our framework operates within a coalition formed by clients collaborating on defense, and specifically, such a coalition can arise in two practical scenarios: (a) Coalition with prior relationships (our focus): Clients managed by the same supermaster (e.g., a user's phone and iPad participating in Gboard training via FL) or those that have interacted previously (e.g., devices joining the same IoT network) can autonomously form a trustworthy coalition [52]. The coordinator is randomly selected from these clients through a decentralized random election mechanism [53]. (b) Coalition without prior relationships (general case): Clients can request to join a coalition through a trusted third party [54], who acts as the coordinator and manages the coalition formation. Before FL begins, each coalition member reports its local data size to the coordinator. For coordination, scenario (a) can leverage stable pseudonymous identities (e.g., a client index [55]) while scenario (b) requires no client coordination at all. Consequently, in both scenarios, all subsequent communications between any client and the coordinator, including class assignments, data volumes, and scalar noise, do not access private data or model parameters.

5 Evaluation

5.1 Experiment Setup

Datasets and Models. Following previous works on MIA research [22, 56], our experiments focus on image classification and are conducted on three benchmark datasets: CIFAR10 [57], CIFAR100 [57], and TinyImageNet [58] on ResNet18 [59] and modified WideResNet-16-4 [19].

Federated Setting. Unless otherwise specified, we consider a horizontal FL system comprising $K = 10$ clients under an IID data distribution. We employ FedAvg [1], where each client performs one local training epoch per round for up to 100 training rounds. The aggregation weight for each client is set proportional to the size of their local dataset. We direct readers to Section 5.2 for extended scenarios involving a varying number of clients and non-IID data distributions.

Attack Algorithms. We implement seven trajectory-based attacks from both client-side and server-side perspectives, including Loss-Series (Loss-S) [26], Avg-Cosine (Avg-C) [23], FedMIA-I/II (FMIA-I/II) [6], FTA-C/L [5], and SeqMIA [18]. These attacks exploit temporal information, i.e., the changes in model updates or outputs across training rounds, to infer the membership status of query samples.

Baseline Defenses. We compare our framework with three typical FL defenses, including gradient sparsification (GradSparse) [60] and noise injection (GradNoise) [6, 7], DPSGD [19]. Additionally, we include comparisons with three defenses originally designed for centralized settings: Mixup [21, 61], Relaxloss [62], and HAMP [22].

CoFedMID Configurations. We set $m_{\max}$ to 10 for CIFAR10, 50 for CIFAR100, and 200 for TinyImageNet. Correspondingly, $m_{\min}$ is set to 2, 20, and 40, respectively (i.e., 20% of the total number of classes). Across all three datasets, the compensation module is triggered from round 10 (i.e., $t' = 10$), with M set to 10 and μ to 0.005 [22]. For the perturbation module, the perturbation strength and ratio are set to 0.1 and 0.2 for CIFAR100, with the ratio increased to 0.3 for TinyImageNet, and to 0.01 and 0.01 for CIFAR10, respectively.

Evaluation Metrics. Following [22, 56, 63], we adopt two common MIA evaluation metrics: AUC, the area under the ROC curve (0.5 indicates random guessing), and TPR (True Positive Rate) at a low FPR (False Positive Rate), which measures the ability of an attack to correctly identify members (lower is better). In our experiments, we use TF01 to denote TPR(%)FPR=0.1%.

All experiments were conducted on an Ubuntu system with NVIDIA RTX 3090 GPUs, and the average results are reported over five runs with different random seeds.

5.2 Defense Performance

This subsection mainly evaluates the defense performance of CoFedMID in two representative coalition settings: (1) *Pair*: a small coalition of two clients, and (2) *Half*: a large coalition comprising half of all clients. We then assess its effectiveness in a variety of settings, including non-IID data distributions, different numbers of clients, and a uniform defense scenario.

Performance in the Partial Defense Scenario. We begin by analyzing the minimal coalition required for collaborative defense in the pair-client setting. We present a comparison of all defense methods on CIFAR100 with ResNet18 using AUC and TF01 metrics in Tables 1 and 2. Additional results for other datasets and models are provided in Appendix B.3. Overall, CoFedMID consistently achieves superior or comparable defense performance across all attacks, yielding the lowest average values on both AUC and TF01 metrics. In con-

Table 1: AUC results of defense methods against different attacks on CIFAR100 with ResNet18. Lower values mean better defense. **Avg** is the average results over seven MIAs, and **ΔAcc** denotes the test accuracy change relative to the undefended FL (original accuracy). 'Pair' denotes a defender coalition of two clients, whereas 'Half' refers to a coalition comprising half of the clients in the FL system. The best average results are highlighted in **bold**.

Case	Defense	Loss-S	Avg-C	FMIA-I	FMIA-II	FTA-C	FTA-L	SeqMIA	Avg ↓	ΔAcc ↑
-	No Defense	0.65	0.78	0.72	0.82	0.71	0.80	0.89	0.77	(0.46)
Pair	GradSparse	0.69	0.81	0.80	0.86	0.76	0.82	0.94	0.78	-0.01
	GradNoise	0.71	0.81	0.72	0.83	0.77	0.83	0.90	0.79	+0.01
	DPSGD	0.54	0.54	0.50	0.53	0.50	0.54	0.81	0.57	-0.01
	Mixup	0.53	0.73	0.51	0.65	0.52	0.60	0.50	0.57	-0.01
	RelaxLoss	0.65	0.68	0.73	0.83	0.68	0.78	0.65	0.71	-0.01
	HAMP	0.61	0.65	0.73	0.62	0.51	0.54	0.55	0.55	-0.03
	CoFedMID	0.54	0.50	0.52	0.57	0.51	0.58	0.54	**0.52**	-0.01
Half	GradSparse	0.69	0.80	0.80	0.86	0.75	0.82	0.94	0.78	-0.01
	GradNoise	0.62	0.70	0.62	0.68	0.62	0.67	0.74	0.66	-0.18
	DPSGD	0.56	0.55	0.57	0.53	0.53	0.53	0.92	0.59	-0.05
	Mixup	0.52	0.66	0.50	0.64	0.54	0.57	0.80	0.57	-0.06
	RelaxLoss	0.66	0.66	0.72	0.83	0.66	0.77	0.69	0.71	-0.00
	HAMP	0.56	0.62	0.70	0.61	0.55	0.53	0.77	0.59	-0.10
	CoFedMID	0.50	0.50	0.54	0.51	0.52	0.56	0.59	**0.51**	-0.03

Table 2: TF01 results of defense methods against different attacks on CIFAR100 with ResNet18.

Case	Defense	Loss-S	Avg-C	FMIA-I	FMIA-II	FTA-C	FTA-L	SeqMIA	Avg ↓	ΔAcc ↑
-	No Defense	10.52	6.44	4.89	12.36	2.22	2.20	10.62	6.59	(0.46)
Pair	GradSparse	12.55	10.87	10.03	20.13	1.49	3.52	16.48	7.87	-0.01
	GradNoise	13.82	8.82	3.51	13.72	2.30	3.94	14.69	8.68	+0.01
	DPSGD	15.62	3.08	5.77	0.31	1.93	0.06	4.58	4.05	-0.01
	Mixup	12.00	4.44	13.64	16.72	2.84	0.89	0.15	5.81	-0.01
	RelaxLoss	14.47	1.67	3.50	10.82	3.96	2.26	8.49	5.33	-0.01
	HAMP	9.42	18.73	0.03	21.12	0.00	0.75	7.81	7.74	-0.03
	CoFedMID	2.28	2.08	0.97	3.12	2.18	0.29	0.11	**1.58**	-0.01
Half	GradSparse	13.23	9.26	13.08	19.18	8.29	2.23	13.80	11.87	-0.01
	GradNoise	10.65	5.21	6.36	12.50	3.83	1.14	10.48	7.45	-0.18
	DPSGD	16.93	3.10	1.82	0.20	2.34	0.00	15.73	5.59	-0.05
	Mixup	13.63	4.38	8.28	14.89	4.02	0.10	1.20	4.75	-0.06
	RelaxLoss	16.43	0.28	3.52	12.48	2.94	1.82	8.55	3.98	-0.00
	HAMP	11.18	13.58	0.00	15.00	6.33	0.20	13.15	5.46	-0.10
	CoFedMID	3.34	2.54	0.24	0.48	4.18	0.06	0.71	**1.65**	-0.03

trast, baseline methods may be effective against certain attacks but perform significantly worse against others. For example, although HAMP attains the second-best defense overall, its performance is notably inconsistent, exhibiting substantially poorer results than CoFedMID on Avg-C and FMIA-I, with AUC values approximately 0.15 and 0.25 higher, respectively. Furthermore, we provide a micro-level MIA analysis to evaluate our framework's uniform defense, examining its performance at the granularity of individual classes and clients in Appendix B.1.

We then analyze defense performance under a larger defender coalition, i.e., the Half case. A key observation is that most baseline methods experience significant accuracy degradation as the number of defended clients increases. For example, baselines with AUCs below 0.60 show accuracy drops of 0.05–0.10, substantially greater than those in the Pair case. In contrast, CoFedMID maintains defense performance comparable to the Pair setting, while incurring only a 0.03 reduction

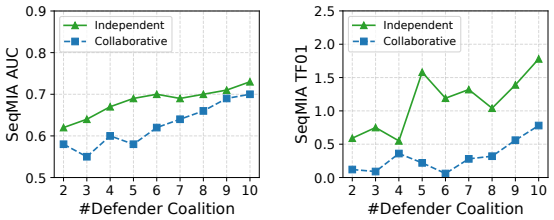


Figure 3: Performance of various coalitions in both independent and collaborative modes on CIFAR100.

in accuracy. This underscores the scalability and effectiveness of our framework in different partial scenarios.

Table 3: Comparison under the uniform defense scenario on CIFAR100 (AUC).

MIA	DPSGD	Mixup	HAMP	CoFedMID
FMIA-II	0.53	0.58	0.64	0.54
SeqMIA	0.98	0.98	0.99	0.69
ΔAcc	-0.16	-0.21	-0.14	-0.04

Performance in the Uniform Defense Scenario. Although our framework focuses on the partial defense scenario, it can be seamlessly extended to the uniform defense setting where all clients participate. To evaluate this, we compare the performance of CoFedMID with three representative baselines selected for their defense capabilities. In addition, we focus on two advanced attacks, FMIA-II and SeqMIA, due to their high attack strength and their reliance on distinct forms of temporal information. As shown in Table 3, our framework achieves the lowest attack performance, at the cost of only a slight reduction in utility.

Performance in Different Defense Modes. To demonstrate the advantages of our proposed collaborative defense mode, we conduct experiments across a comprehensive range of defender coalitions—from pairs of clients up to full participation (i.e., 2 to 10 clients). The defense performance of both independent and collaborative modes under different coalition sizes is shown in Figure 3. Overall, the collaborative mode consistently improves the performance of our framework on both attack metrics, with TF01 exhibiting notably lower values than in the independent setting.

Performance across Varying Total Clients. The preceding experiments were conducted on an FL system with 10 clients. Next, we assess the defense performance in FL systems with varying numbers of clients, ranging from 5 to 30, under both the Pair and Half cases. As shown in Figure 4, CoFedMID benefits from larger FL systems in the Pair case, achieving effective defense with lower utility degradation as more clients train. Furthermore, in the Half setting, increasing the number of clients does not have a significant impact on defense effectiveness or utility loss, resulting in results comparable to those in the 10-client scenario. Thus, these results demonstrate the scalability of our framework in FL systems of varying sizes.

Table 4: Performance under non-IID settings on CIFAR100 (AUC). None refers to no defense.

β	FMIA-II		SeqMIA	
	None	CoFedMID	None	CoFedMID
∞	0.83	0.56	0.92	0.50
10.0	0.85	0.58	0.97	0.54
1.0	0.91	0.64	0.98	0.52
0.5	0.96	0.66	0.99	0.58

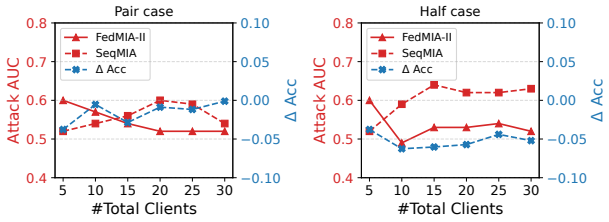


Figure 4: Performance of various total clients on CIFAR100.

Performance in Non-IID settings. We finally evaluate defense performance under different non-IID settings. To simulate these scenarios, we follow [64] and sample client data distributions from a Dirichlet distribution with parameter $\beta \in \{\infty, 10, 1.0, 0.5\}$, where smaller β values indicate more skewed distributions and ∞ represents the IID setting. We evaluate CoFedMID under various non-IID settings in the Pair case on CIFAR100, and present the results in Table 4. Our findings show that CoFedMID consistently reduces attack performance across varying degrees of data heterogeneity. While its effectiveness diminishes slightly under highly skewed distributions (e.g., $\beta = 0.5$) as attacks benefit from increased skewness, it still yields substantially lower AUCs and TF01s than the undefended case.

5.3 Detailed Analysis of CoFedMID

In this subsection, we present a detailed analysis of the three CoFedMID modules under different configurations on CIFAR100 and CIFAR10 datasets, with half of the clients forming the defender coalition.

Impact of Each Module. We first evaluate the individual contributions of the three CoFedMID modules to both defense performance and FL utility, with results for two representative attacks shown in Table 5. The results for the class-guided partition module (①) show that reducing the exposure of local training samples effectively decreases membership privacy risk compared to undefended FL, but this comes at the cost of

Table 5: Impact of each module on defense performance (AUC) and model utility. ①: class-guided partition module; ②: utility-aware compensation module; ③: aggregation-neural perturbation module.

Dataset	MIA	None	①	①+②	①+②+③
CIFAR100	FMIA-II	0.82	0.51	0.52	0.51
	SeqMIA	0.89	0.59	0.64	0.59
	ΔAcc	-	-0.04	-0.02	-0.02
CIFAR10	FMIA-II	0.62	0.53	0.55	0.54
	SeqMIA	0.88	0.54	0.64	0.60
	ΔAcc	-	-0.07	-0.03	-0.03

Table 6: Impact of bounded class assignment (AUC).

MIA	CIFAR100		CIFAR10	
	Random	Bounded	Random	Bounded
FMIA-II	0.53	0.51	0.52	0.53
SeqMIA	0.66	0.59	0.53	0.54
ΔAcc	-0.06	-0.04	-0.11	-0.07

noticeable utility loss. As a utility compensation, the module ② effectively mitigates this loss, particularly for CIFAR10. This difference arises from the data complexity: CIFAR10 holds a simpler data distribution, making the model more sensitive to additional training samples. Moreover, while combining ① and ② is effective against FMIA-II, incorporating the aggregation-level perturbation module (③) further mitigates SeqMIA that exclusively targets local models. Overall, these three modules contribute in complementary ways to balancing privacy protection against MIAs and model utility.

Next, we conduct an ablation study on various configurations within each module and analyze the effectiveness of their respective sub-modules.

Analysis of Label-guided Partition Module. Recall that this module is designed to assign class subsets to clients with minimal overlap, following a bounded, decay-based strategy. To assess its effectiveness, we implement it in isolation and evaluate the impact of different configurations.

1. Random vs. Bounded Class Assignment. We compare our bounded assignment strategy with random sampling, where each client is assigned a class subset selected uniformly at random. As shown in Table 6, the bounded strategy achieves comparable defense against MIAs while reducing utility loss, indicating that controlling inter-client class overlap preserves the overall class distribution and helps maintain FL performance.

2. Constant vs. Decay-based Class Assignment. We evaluate our decay-based strategy against a constant approach, where the number of assigned classes remains fixed throughout training. Table 7 shows that fixed class assignments suffer from a dilemma: assigning fewer classes generally improves defense but causes greater utility degradation. In contrast, our strategy achieves defense performance similar to that of $\lambda_{\min}$ and utility close to that of $\lambda_{\max}$.

Table 7: Impact of decay-based class assignment (AUC). Here, $\lambda_1 = \lambda_{\min}$, $\lambda_2 = \frac{\lambda_{\min}+\lambda_{\max}}{2}$, and $\lambda_3 = \lambda_{\max}$.

MIA	CIFAR100				CIFAR10			
	λ_1	λ_2	λ_3	Decay	λ_1	λ_2	λ_3	Decay
FMIA-II	0.51	0.54	0.56	0.49	0.48	0.50	0.54	0.54
SeqMIA	0.62	0.67	0.73	0.59	0.54	0.66	0.68	0.54
ΔAcc	-0.06	-0.01	-0.01	-0.03	-0.06	-0.05	-0.03	-0.03

Table 8: Comparison of different recycling strategies (AUC).

MIA	CIFAR100			CIFAR10		
	Rand	Seq	MAB	Rand	Seq	MAB
FMIA-II	0.52	0.52	0.52	0.55	0.56	0.55
SeqMIA	0.64	0.65	0.63	0.66	0.67	0.62
ΔAcc	-0.03	-0.03	-0.03	-0.08	-0.09	-0.04

Additionally, we compare linear and non-linear decay strategies and find that non-linear approaches, such as the exponential function, can further reduce the attack performance, as detailed in Appendix B.2.

Analysis of Utility-aware Compensation Module. The utility-aware compensation module reduces the utility loss from the class-guided partition module by selectively reintroducing a small subset of high-contribution samples. We evaluate its effectiveness against two heuristic baselines, examine the role of confidence regularization on these samples, and provide additional results for varying recycling ratios in Appendix B.2.

1. Comparison of Recycling Strategies. We compare our MAB-based sample recycling strategy (*MAB*) with two heuristic approaches. The random strategy (*Rand*) uniformly selects a fixed number of samples from the remaining local dataset, while the sequential strategy (*Seq*) reintroduces samples in order of their loss values, from low to high. Both baselines recycle a fixed ratio (i.e., 10%) of samples from local data in each round.

We present the comparison results of the three approaches in Table 8. While CIFAR100 exhibits little sensitivity to the choice of recycling strategy due to its diverse data distribution, our method achieves both lower attack performance on both datasets and reduced utility loss on CIFAR10. To further demonstrate the superiority of the MAB-based recycling strategy, we visualize the sampling scale over training in Figure 5, showing how it adaptively selects contributive samples in both early and later rounds. Compared to random and sequential methods (i.e., Static) that employ a fixed sampling

Table 9: Impact of confidence regularization (AUC).

MIA	CIFAR100		CIFAR10	
	w/o CR	w/CR	w/o CR	w/CR
FMIA-II	0.55	0.52	0.56	0.52
SeqMIA	0.80	0.59	0.73	0.61

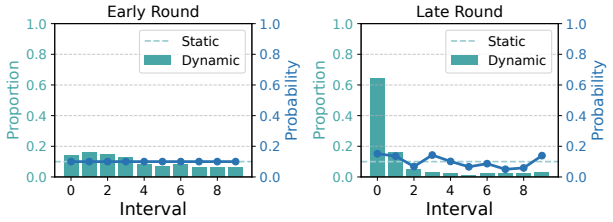


Figure 5: Proportions and probabilities of sample intervals in early and late training rounds.

size, our approach (i.e., Dynamic) consistently recycles fewer high-contribution samples, thereby reducing unnecessary exposure of local data and strengthening membership privacy protection.

2. Impact of Confidence Regularization. We conduct an ablation study to evaluate the effectiveness of the confidence regularization (CR) approach, which is designed to prevent membership privacy leakage from recycled samples. As shown in Table 9, incorporating CR substantially reduces MIA success on both datasets, with SeqMIA showing roughly an order-of-magnitude drop, demonstrating its effectiveness in mitigating membership privacy risks.

Analysis of Aggregation-neutral Perturbation Module. We have shown the effectiveness of the aggregation-level perturbation module in defending against MIAs targeting local models (see Table 5). We further examine the impact of its hyperparameters, namely perturbation strength and ratio. As illustrated in Figure 6 for CIFAR100, the perturbation strength has a notable impact on defense performance, with the AUC decreasing by approximately 0.05 as the strength increases from 0.05 to 0.10. In the case of the perturbation ratio, its impact on defense effectiveness is relatively minor. For instance, the AUC changes by about 0.03 when the ratio increases from 0.10 to 0.25. As a practical guideline, it is preferable to focus on an appropriate perturbation strength and avoid an excessively small perturbation ratio.

5.4 Adaptive Attacks

An effective defense should demonstrate robustness against adaptive adversaries, who possess complete knowledge of the defense and can modify their attack strategies in response, especially when the existence of the coordinator or the coalition

Table 10: Defense performance against adaptive attacks (AUC).

Case	Loss-S	FMIA-II	FTA-C	SeqMIA
CIFAR100 Dataset				
None	0.66	0.68	0.60	0.70
Pair (Ours)	0.46	0.56	0.49	0.52
Half (Ours)	0.50	0.55	0.45	0.66
CIFAR10 Dataset				
None	0.62	0.55	0.50	0.80
Pair (Ours)	0.54	0.52	0.53	0.63
Half (Ours)	0.54	0.55	0.50	0.64

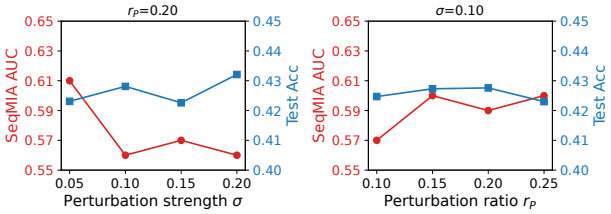


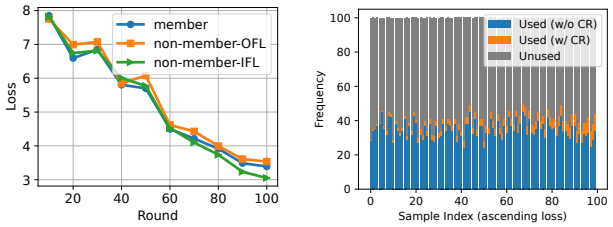
Figure 6: Impact of perturbation strength and ratio.

may be exposed. To evaluate the robustness of CoFedMID under this threat model, we design an adaptive attack tailored to challenge our defense mechanism. We assume the attacker can identify clients within the defender coalition, using auxiliary information such as device IPs, public institutional affiliations, or other metadata that reveal client identities or associations. Rather than performing MIAs on the local or global models of individual clients, the attacker directs its efforts toward *the aggregated model of the defender coalition*, considering this collective model as the attack target.

To evaluate adaptive attacks, we adopt four different attack methods to this threat model, replacing their respective target models with the aggregated model for each attack. We conduct experiments on both CIFAR10 and CIFAR100 datasets, with the resulting defense performance summarized in Table 10. We find that the aggregated model results in a noticeable reduction in attack performance compared to non-adaptive attacks in undefended settings. With respect to our framework, CoFedMID, it demonstrates strong defense capabilities against adaptive attacks. For the first three MIAs, CoFedMID maintains defense performance comparable to that observed under the non-adaptive attacks. Although SeqMIA exhibits an increased AUC under adaptive attacks as the aggregation-level perturbation module is bypassed, its TF01 remains below 2%, indicating a poor ability to distinguish between member and non-member samples. Overall, these results demonstrate that our defense framework is robust against adaptive attacks.

Table 11: Runtime of local training over (seconds).

Dataset	None	①	①+②	①+②+③
CIFAR100	1x (225.3)	0.78x	1.53x	1.52x
CIFAR10	1x (229.8)	0.87x	1.66x	1.69x
TinyImageNet	1x (1003.2)	0.57x	0.98x	1.03x
Average	-	0.73x	1.39x	1.41x



(a) Loss trajectory (b) Data usage and protection

Figure 7: Analysis of CoFedMID's effectiveness.

5.5 Overhead of CoFedMID

We assess the overhead of our defense framework for FL, including communication and computational costs.

Communication Cost. The communication cost is a critical concern in FL, particularly for limited network bandwidth. Our framework incurs extra communication overhead before training, as a leader client distributes class assignments and perturbation configurations to the coalition. Specifically, each client receives up to N class labels along with two scalar values for perturbation configurations per training round. Given T rounds, this totals $(N+2) \times T$ items. Assuming each item occupies 1 byte, the total communication overhead is approximately 20 KB for our default settings (e.g., $T = 100$ and $N = 200$ for TinyImageNet). Compared with model updates exchanged in FL, which are typically tens of megabytes per round, CoFedMID adds only minimal communication overhead while offering effective defense against MIAs.

Computational Cost. Effective defenses should not only provide strong protection but also remain efficient to deploy in practice. We evaluate the computational cost of our defense by measuring the client-side local training time across the entire FL process, as summarized in Table 11. As expected, the class-guided partition module (①) reduces runtime compared to undefended FL by using fewer training samples, the perturbation module (③) adds negligible overhead due to its efficient design, while the compensation module (②) incurs additional computational cost. Overall, the results show that CoFedMID incurs only about 1.4× the runtime of undefended FL on average, which remains acceptable for clients.

6 Discussion

The Reason Why CoFedMID Works. The above experiments demonstrate that our proposed framework is effective in defending against MIAs in FL both in uniform and partial scenarios. To further understand its effectiveness, we now analyze the underlying mechanisms of CoFedMID. We first show the average loss trajectories of a client's local model within the defender coalition in Figure 7(a). In conjunction with the loss trajectory for undefended FL in Figure 1(a), our framework elevates the overall sample loss and significantly reduces the loss gap between member samples and both types of non-member samples, thereby substantially lowering the attack performance in practice.

Moreover, our framework is designed to mitigate MIAs in FL by reducing the exposure of training samples. To investigate this, we randomly select a subset of training samples from CIFAR100 and record their usage in the local training process. As shown in Figure 7(b), most samples are trained only about half as many times as in the undefended setting (Used). In addition, each sample is protected by the additional defense mechanism over several rounds (Used (w/ CR)). This not only prevents a significant discrepancy from the training process of undefended clients but also narrows the distinguishability between member and non-member IFL samples. Furthermore, it indicates that our framework employs distinct strategies for different training samples. For instance, it allocates stronger protection to high-loss samples while providing less additional safeguarding to low-loss samples. This differentiated way enhances the overall effectiveness of our defense.

Limitations and Future Work. Despite the demonstrated superiority, CoFedMID has two limitations that warrant further investigation:

Privacy-Utility Trade-offs. Our framework cannot fully avoid the inherent trade-off between privacy and utility, particularly in large coalitions. As more clients participate, a greater portion of training data must be concealed, a more pronounced impact on model performance. Future research could mitigate this trade-off by adaptively determining the initial class subset size in the assignment strategy based on the coalition size.

Poisoning MIAs. Advanced MIAs combined with poisoning attacks represent an emerging research direction [65]. In this work, we focus on MIAs initiated by honest-but-curious clients or servers, and develop our defense framework under the assumption of an honest coordinator and coalition members. However, the distributed nature of FL makes it susceptible to poisoning attacks. Since defending against poisoning attacks is an orthogonal problem to membership inference defense, we leave mitigating these complex attack scenarios to future work.

Task Extension. Our framework requires explicit label information to construct the class-guided module, which makes it naturally effective for classification tasks. Consequently,

it does not directly apply to label-free settings such as unsupervised learning. Exploring extensions such as pseudo-labeling [66] is a promising direction for future work. Additionally, while the majority of MIA studies have concentrated on classification tasks—particularly those involving image data—future work could broaden the scope to include other data modalities, such as text and medical records.

7 Related Works

7.1 Membership Inference Attacks

MIAs are first introduced by Shokri et al. [67] in the context of centralized learning to determine whether a specific data record belonged to the training set of a target model. Since then, numerous approaches have been proposed, substantially enhancing inference performance across diverse model architectures and settings [31, 56, 63, 68].

Building upon these centralized works, research has extended MIAs to federated settings. Nasr et al. [2] first proposed MIAs in FL by exploiting per-sample gradients, intermediate features, and loss values during training. This work reveals significant privacy risks in model updates when local data is insufficiently protected. Li et al. [23] later developed a gradient-based attack leveraging cosine similarity between per-sample gradients across communication rounds.

Compared to the above update-based MIAs, trajectory-based approaches demonstrate stronger performance. These attacks capture richer temporal information by exploiting how certain statistics evolve throughout the training process. For instance, an attacker may monitor the trajectory of per-sample loss values, prediction confidence [4], or gradient norms [6] to distinguish member samples from non-members. FedMIA [6] adapts LiRA [63] for FL by using aggregated signals like loss values and gradient norms over multiple rounds to perform membership inference. Similarly, Chen et al. [5] propose a lightweight yet effective method that leverages the rate of change of per-sample loss or prediction confidence over multiple model snapshots, achieving strong attack performance.

7.2 Membership Inference Defenses

A variety of defenses against MIAs have been extensively studied in both centralized and federated contexts. Centralized defenses include output perturbation to reduce confidence gaps [22, 67, 69], regularization techniques like data augmentation and adversarial training to mitigate overfitting [27, 70], knowledge distillation to transfer knowledge to protected models [29, 30], and differential privacy to provide formal privacy guarantees [19, 71–74]. These methods mainly reduce the difference between member and non-member data by preventing the target model from overfitting.

However, due to the decentralized nature of FL, defensive mechanisms differ substantially, typically relying on perturbation-based approaches. Partial sharing reduces the attack surface by selectively suppressing or filtering model updates during training, such as gradient compression [3, 23] and weight pruning [75], to limit exposure of sensitive information. Nevertheless, this attack-agnostic approach provides limited defense performance [3]. Differential privacy remains a key defense in FL by adding calibrated noise to gradients to obscure individual contributions and reduce membership leakage. The effectiveness of local and central DP against white-box attacks is demonstrated in [12], often at the cost of model utility [76, 77]. The work [24] offers effective defense via model perturbation but requires a trusted server. An orthogonal line of defense employs cryptographic techniques [78, 79] to protect membership privacy. For instance, SecAgg [78] uses secret sharing and masking to defend against honest-but-curious servers. While these methods secure model updates during transmission, they incur substantial computational and communication overhead. Furthermore, they cannot prevent MIAs via the global model, as clients still have access to the decrypted global model [4].

8 Conclusion

In this work, we explore a new and realistic scenario in which a group of clients collaborates to form a defender coalition against MIAs in FL. Towards this end, we propose CoFedMID, a novel defense framework designed to mitigate membership privacy leakage in such settings. Extensive experiments show that our approach consistently surpasses existing methods from both federated and centralized settings. In addition to our primary scenario, CoFedMID also outperforms existing defenses in the previously studied uniform and independent scenarios. We underscore the value of client collaboration for defense and hope that our work will inspire further investigation into advanced mitigation strategies in FL.

Ethical Considerations

In this study, we propose an effective defense against membership inference attacks in federated learning and analyze its potential impacts. To ensure ethical research practices, we discuss the following points.

Safety Considerations for Experiments. All experiments were conducted exclusively on publicly available benchmark datasets within a simulated federated learning environment. These datasets contain no personally identifiable or sensitive information, and all clients and servers were instantiated in a controlled virtual setting, without involving any real users, devices, or organizations.

Positive Impacts of Proposed Defenses. Our defense provides protection for clients' local data against membership inference attacks. By mitigating the risk of privacy leakage to honest-but-curious servers and clients outside the coalition,

the defense primarily benefits FL clients, who are the main recipients of its protective effects. In real-world scenarios, industries such as healthcare, finance, and mobile applications that rely on federated learning could see improved privacy for users' sensitive data if proposed defenses are deployed.

Negative Impacts of Proposed Defenses. Despite these benefits, there are potential negative consequences that must be pointed out: (1) Protected clients may overestimate the defense's effectiveness if coalition members misbehave and collude. (2) While the defense offers partial protection for sensitive users, it does not cover clients outside the coalition, leaving their data potentially vulnerable to attackers. (3) Attackers may adapt strategies to bypass the defense or exploit insights from it. (4) Behavioral shifts among clients may occur, as organizations and users might over-rely on the protection mechanism, potentially altering the overall risk profile and unintentionally amplifying the effectiveness of attacks.

Open Science

To support reproducibility and replicability, we provide the research artifacts of our defense framework, including source code, scripts, and detailed instructions, to facilitate the reproduction of our experimental results. They are available at Zenodo, which include:

- **Datasets:** We use three publicly available datasets (i.e., CIFAR10, CIFAR100, TinyImageNet) and provide instructions for downloading them.
- **Source Code:** We provide the core implementation of the proposed defense method as well as baseline algorithms.
- **Scripts:** We include two scripts for running experiments on CIFAR100 and ResNet18 to evaluate the defense performance of CoFedMID and the baselines.
- **Evaluation Demo:** We provide a Jupyter notebook for statistical analysis and visualization of the experimental results.

All artifacts, along with detailed instructions and full version, are also accessible at Github.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No: 92270123, 62372122, 62502416, and U25A20430), and the Research Grants Council (Grant No: 15208923, 25207224, and 15207725), Hong Kong SAR, China.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282, 2017.

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

[3] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE symposium on security and privacy (SP)*, pages 691–706. IEEE, 2019.

[4] Yuhao Gu, Yuebin Bai, and Shubin Xu. Cs-mia: Membership inference attack based on prediction confidence series in federated learning. *Journal of Information Security and Applications*, 67:103201, 2022.

[5] Hongyan Chang, Brandon Edwards, Anindya S Paul, and Reza Shokri. Efficient privacy auditing in federated learning. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 307–323, 2024.

[6] Gongxi Zhu, Donghao Li, Hanlin Gu, Yuan Yao, Lixin Fan, and Yuxing Han. Fedmia: An effective membership inference attack exploiting" all for one" principle in federated learning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 20643–20653, 2025.

[7] Li Bai, Haibo Hu, Qingqing Ye, Haoyang Li, Leixia Wang, and Jianliang Xu. Membership inference attacks and defenses in federated learning: A survey. *ACM Computing Surveys*, 57(4):1–35, 2024.

[8] Zach Jorgensen, Ting Yu, and Graham Cormode. Conservative or liberal? personalized differential privacy. In *2015 IEEE 31St international conference on data engineering*, pages 1023–1034. IEEE, 2015.

[9] Hamid Ebadi, David Sands, and Gerardo Schneider. Differential privacy: Now it's getting personal. *Acm Sigplan Notices*, 50(1):69–81, 2015.

[10] Junxu Liu, Jian Lou, Li Xiong, Jinfei Liu, and Xiaofeng Meng. Cross-silo federated learning with record-level personalized differential privacy. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 303–317, 2024.

[11] Huadi Zheng, Haibo Hu, and Ziyang Han. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intelligent Systems*, 35(4):5–14, 2020.

[12] Mohammad Naseri, Jamie Hayes, and Emiliano De Cristofaro. Local and central differential privacy for robustness and privacy in federated learning. In *Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2022.

[13] Oded Goldreich. Secure multi-party computation. *Manuscript. Preliminary version*, 78(110):1–108, 1998.

[14] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[15] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE signal processing magazine*, 37(3):50–60, 2020.

[16] Micah J Sheller, Brandon Edwards, G Anthony Reina, Jason Martin, Sarthak Pati, Aikaterini Kotrotsou, Mikhail Milchenko, Weilin Xu, Daniel Marcus, Rivka R Colen, et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. *Scientific reports*, 10(1):12598, 2020.

[17] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*, 2017.

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

[19] Michael Aerni, Jie Zhang, and Florian Tramèr. Evaluations of machine learning privacy defenses are misleading. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1271–1284, 2024.

[20] Li Hu, Anli Yan, Hongyang Yan, Jin Li, Teng Huang, Yingying Zhang, Changyu Dong, and Chunsheng Yang. Defenses to membership inference attacks: A survey. *ACM Computing Surveys*, 56(4):1–34, 2023.

[21] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[22] Zitao Chen and Karthik Pattabiraman. Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction. In *31st Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2024.

[23] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Effective passive membership inference attacks in federated learning against overparameterized models. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] Xue Yang, Yan Feng, Weijun Fang, Jun Shao, Xiaohu Tang, Shu-Tao Xia, and Rongxing Lu. An accuracy-lossless perturbation method for defending privacy attacks in federated learning. In *Proceedings of the ACM Web Conference 2022*, pages 732–742, 2022.

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

[26] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282. IEEE, 2018.

[27] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 634–646, 2018.

[28] Zongqi Chen, Hongwei Li, Meng Hao, and Guowen Xu. Enhanced mixup training: A defense method against membership inference attack. In *International Conference on Information Security Practice and Experience*, pages 32–45. Springer, 2021.

[29] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9549–9557, 2021.

[30] Xinyu Tang, Saeed Mahloujifar, Liwei Song, Virat Shejwalkar, Milad Nasr, Amir Houmansadr, and Prateek Mittal. Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In *31st USENIX security symposium (USENIX security 22)*, pages 1433–1450, 2022.

[31] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.

[32] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. Machine learning models that remember

too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*, pages 587–601, 2017.

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

[34] Jason W Rocks and Pankaj Mehta. Memorizing without overfitting: Bias, variance, and interpolation in overparameterized models. *Physical review research*, 4(1):013201, 2022.

[35] Runqi Lin, Chaojian Yu, Bo Han, and Tongliang Liu. On the over-memorization during natural, robust and catastrophic overfitting. In *The Twelfth International Conference on Learning Representations*. OpenReview.net, 2024.

[36] Tao Zhang, Tianqing Zhu, Kun Gao, Wanlei Zhou, and Philip S. Yu. Balancing learning model privacy, fairness, and accuracy with early stopping criteria. *IEEE Transactions on Neural Networks and Learning Systems*, 34(9):5557–5569, 2023.

[37] Peter Frankl and Vojtěch Rödl. Forbidden intersections. *Transactions of the American Mathematical Society*, 300(1):259–286, 1987.

[38] Calvin Beideman and Jeremiah Blocki. Set families with low pairwise intersection. *arXiv preprint arXiv:1404.4622*, 2014.

[39] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.

[40] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[41] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):4555–4576, 2021.

[42] Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *international conference on machine learning*, pages 1311–1320. Pmlr, 2017.

[43] Yao Tang, Zhihui Xie, Zichuan Lin, Deheng Ye, and Shuai Li. Learning versatile skills with curriculum masking. *Advances in Neural Information Processing Systems*, 37:65562–65582, 2024.

[44] Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International conference on machine learning*, pages 2525–2534. PMLR, 2018.

[45] Aditya Mahajan and Demosthenis Teneketzis. Multi-armed bandit problems. In *Foundations and applications of sensor management*, pages 121–151. Springer, 2008.

[46] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[47] Zheng Li and Yang Zhang. Membership leakage in label-only exposures. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 880–895, 2021.

[48] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.

[49] Vaikkunth Mugunthan, Antigoni Polychroniadou, David Byrd, and Tucker Hybinette Balch. Smpai: Secure multi-party computation for federated learning. In *Proceedings of the NeurIPS 2019 Workshop on Robust AI in Financial Services*, volume 21, 2019.

[50] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[51] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[52] Sarhad Arisdakessian, Omar Abdel Wahab, Azzam Mourad, and Hadi Otrok. Coalitional federated learning: Improving communication and training on non-iid data with selfish clients. *IEEE Transactions on Services Computing*, 16(4):2462–2476, 2023.

[53] Ning Zhang, Qian Ma, Wuxing Mao, and Xu Chen. Coalitional fl: Coalition formation and selection in federated learning with heterogeneous data. *IEEE Transactions on Mobile Computing*, 23(11):10494–10508, 2024.

[54] Sen Cui, Jian Liang, Weishen Pan, Kun Chen, Changshui Zhang, and Fei Wang. Collaboration equilibrium in federated learning. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 241–251, 2022.

[55] Dana Keeler, Chelsea Komlo, Emily Lepert, Shannon Veitch, and Xi He. Dprio: Efficient differential privacy with high utility for prio. *Proceedings on Privacy Enhancing Technologies*, 2023(3):375–390, 2023.

[56] Rui Wen, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against in-context learning. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 3481–3495, 2024.

[57] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[58] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge, 2015.

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

[60] Otkrist Gupta and Ramesh Raskar. Distributed learning of deep neural network over multiple agents. *Journal of Network and Computer Applications*, 116:1–8, 2018.

[61] Hanlin Gu, Jiahuan Luo, Yan Kang, Lixin Fan, and Qiang Yang. Fedpass: Privacy-preserving vertical federated deep learning with adaptive obfuscation. *arXiv preprint arXiv:2301.12623*, 2023.

[62] Dingfan Chen, Ning Yu, and Mario Fritz. Relaxloss: Defending membership inference attacks without losing utility. In *International Conference on Learning Representations (ICLR)*, 2022.

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

[64] Tzu-Ming Harry Hsu, Hang Qi, and Matthew Brown. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv preprint arXiv:1909.06335*, 2019.

[65] Wenjin Mo, Zhiyuan Li, Minghong Fang, and Mingwei Fang. Find a scapegoat: Poisoning membership inference attack and defense to federated learning. *arXiv preprint arXiv:2507.00423*, 2025.

[66] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

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

[68] Yixin Wu, Ning Yu, Zheng Li, Michael Backes, and Yang Zhang. Membership inference attacks against text-to-image generation models. *arXiv preprint arXiv:2210.00968*, 2022.

[69] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. Memguard: Defending against black-box membership inference attacks via adversarial examples. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 259–274, 2019.

[70] Yigitcan Kaya and Tudor Dumitras. When does data augmentation help with membership inference attacks? In *International conference on machine learning*, pages 5345–5355. PMLR, 2021.

[71] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX security symposium (USENIX security 19)*, pages 1895–1912, 2019.

[72] Huadi Zheng, Haibo Hu, and Ziyang Han. Preserving user privacy for machine learning: Local differential privacy or federated machine learning? *IEEE Intelligent Systems*, 35(4):5–14, 2020.

[73] Yuemin Zhang, Qingqing Ye, and Haibo Hu. Federated heavy hitter analytics with local differential privacy. *Proceedings of the ACM on Management of Data*, 3(1):1–27, 2025.

[74] Lixu Wang, Chenxi Liu, Junfeng Guo, Qingqing Ye, Heng Huang, Haibo Hu, and Wei Dong. Federated continuous category discovery and learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2429–2439, 2025.

[75] Dimitris Stripelis, Umang Gupta, Nikhil Dhinagar, Greg Ver Steeg, Paul M Thompson, and José Luis Ambite. Towards sparsified federated neuroimaging models via weight pruning. In *International Workshop on Distributed, Collaborative, and Federated Learning*, pages 141–151. Springer, 2022.

[76] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, and Xuyun Zhang. Source inference attacks in federated learning. In *2021 IEEE International Conference on Data Mining (ICDM)*, pages 1102–1107. IEEE, 2021.

[77] Anshuman Suri, Pallika Kanani, Virendra J Marathe, and Daniel W Peterson. Subject membership inference attacks in federated learning. *arXiv preprint arXiv:2206.03317*, 2022.

[78] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[79] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE transactions on information forensics and security*, 13(5):1333–1345, 2017.

A Algorithms

In this section, we provide the pseudocode of the three core modules: (i) **Class-guided partition** (Algorithm 1), which limits the exposure of individual datasets by distributing the overall data distribution across clients, each handling only a local subset; (ii) **Utility-aware compensation** (Algorithm 2), which mitigates utility loss by selectively increasing exposure in a performance-preserving manner; and (iii) **Aggregation-neutral perturbation** (Algorithm 3), which enhances defense by enabling defenders to collaboratively inject noises that cancel out during aggregation. Furthermore, we present the pseudocode of the overall framework that integrates these modules in Algorithm 4.

Algorithm 1: Class-guided Partition

Input: Number of classes N; coalition size d; maximum classes per client $m_{\max}$; minimum classes per client $m_{\min}$; total rounds T
Output: Subsets $\{\mathcal{S}_1^t,\ldots,\mathcal{S}_d^t\}_{t=1}^T$

for $t \leftarrow 1$ **to** T **do**
  $m^t \leftarrow \texttt{Decay}(m_{\max}, m_{\min}, t)$;
  Initialize overlap bound $\lambda \leftarrow \lambda_{theo}(m^t)$;
  repeat
    Initialize $\mathcal{S}_1^t,\ldots,\mathcal{S}_d^t \leftarrow \emptyset$, success $\leftarrow$ `True`;
    for $i \leftarrow 1$ **to** d **do**
      if not *ClassAssign*(N,d,m^t,λ) **then**
        `// Overlap constraint violated`
        success $\leftarrow$ `False`; $\lambda \leftarrow \lambda+1$; **break**
  until *success*;
return $\{\mathcal{S}_1^t,\ldots,\mathcal{S}_d^t\}_{t=1}^T$

Algorithm 2: Utility-Aware Compensation

Input: Local training set D_k, validation set D_{val}; Total rounds T, number of intervals M; Initialization round t_0, maximum recycling ratio r_l
Output: Local model θ_k

Initialize interval weights $w_j \leftarrow 1$ for all $j = 1,\ldots,M$;
Initialize sample intervals $\{I_j\}_{j=1}^M$ at round t_0;
for $t \leftarrow t_0+1$ **to** T **do** `// Recycling begins`
  Receive global model $\tilde{\theta}$ and update local model θ_k;
  Obtain D_k^{ass} from the first module;
  Sample interval index $j \sim \pi_w$;
  `// Collect recycled samples from` I_j
  $D_k^{\text{rec}} \leftarrow \{(x_i,y_i) \in D_k \setminus D_k^{\text{ass}} \mid (x_i,y_i) \in I_j\}$;
  $D_k^{\text{rec}} \leftarrow \text{RandomSubset}(D_k^{\text{rec}}, \lfloor r_l \cdot |D_k| \rfloor)$;
  `// Local update`
  Train θ_k on $D_k^{\text{ass}} \cup D_k^{\text{rec}}$ using the combined loss;
  `// Reward estimation and EXP3 update`
  Evaluate validation losses of $\tilde{\theta}$ and θ_k on D_{val};
  Compute normalized reward $\tilde{r}_j$;
  Update weight and sampling probability accordingly;
return θ_k

Table 12: TF01 values (%) of SeqMIA on CIFAR10.

	Avg	c0	c1	c2	c3	c4	c5	c6	c7	c8	c9
Client0	0.40	0.12	0.04	0.00	0.00	0.00	0.00	0.27	0.00	0.44	0.12
Client4	0.27	0.00	0.08	0.46	0.10	0.39	0.01	0.81	1.27	0.06	0.16

Table 13: Comparison of different decay strategies on CIFAR100 (TF01).

Decay	Loss-S	FMIA-II	FTA-C	SeqMIA	Avg	Test acc
Linear	2.62	0.11	3.6	0.64	1.74	0.43
Cosine	4.14	0.02	2.18	1.56	1.98	0.44
Exp	2.51	0.06	1.18	0.9	1.16	0.43
Poly	1.68	0.01	3.94	1.01	1.66	0.43

B Additional Experimental Results

B.1 Micro-level MIA Analysis

Our defense treats all training samples, classes, and protected clients uniformly. Samples may be from assigned classes or reused as recycled samples, as shown in Figure 7(b), and there is no fixed mapping between classes and clients (Algorithm 1). Table 12 shows at most 1.27% TF01 difference across clients and classes.

Table 14: Impact of recycling threshold on CIFAR100 under different coalition sizes (AUC).

	r_l	0.05	0.10	0.20	0.30
Pair	SeqMIA	0.53	0.58	0.59	0.62
	Test Acc	0.47	0.46	0.46	0.47
Half	SeqMIA	0.66	0.69	0.68	0.72
	Test Acc	0.44	0.44	0.45	0.45

B.2 Other Ablation Studies

Impact of Different Decay Functions. We conduct experiments to evaluate the impact of different decay functions, including linear, cosine, exponential, and polynomial, on defense performance and FL utility. Table 13 reports the comparison of these strategies on CIFAR100 in terms of multiple attack metrics and test accuracy. The results show that non-linear approaches, such as exponential and polynomial functions, can further mitigate attack performance without incurring additional utility loss.

Impact of Recycling Threshold. To prevent excessive recycling of samples, we introduce a recycling threshold that limits the maximum number of samples recycled in each round. We investigate how varying this ratio affects both defense effectiveness and model utility. As shown in Table 14, a lower recycling ratio consistently improves defense performance, while having little impact on the FL system's utility.

B.3 More Defense Results

Due to space limitations, additional comparison results between CoFedMID and the baselines in the Pair and Half settings on CIFAR10 and TinyImageNet with ResNet18 are available in the full version. It also includes: 1) a proof statement of the theoretical lower bound on the overlap between fixed-size subsets; 2) detailed descriptions of the datasets; 3) comprehensive information on defense baselines; 4) detailed attack algorithms.

Algorithm 3: Aggregation-Neutral Perturbation

Input: Model parameters $\{\theta_k\}_{k=1}^d$ of defender coalition; Aggregation weights $\{w_k\}_{k=1}^d$; Perturbation strength σ; Perturbation ratio r_p

Output: Perturbed model parameters $\{\theta_k\}_{k=1}^d$

for $k = 1$ **to** d **do**
  Sample $\delta'_k \sim \mathcal{N}(0, \sigma^2 I)$;
for $k = 1$ **to** d **do**
  Compute projection: $\delta_k = \delta'_k - \frac{w_k}{\|w\|_2^2} \cdot \sum_{j=1}^d w_j \delta'_j$;
  Let θ_k have P parameters and define $P' = \lfloor r_p \cdot P \rfloor$;
  for $i = P - P' + 1$ **to** P **do**
    Inject perturbation: $\theta_k^{(i)} \leftarrow \theta_k^{(i)} + \delta_k$;
return $\{\theta_k\}_{k=1}^d$

Algorithm 4: Federated Learning with CoFedMID

Input: Number of clients K, defender coalition $\mathcal{D}$, total rounds T

Output: Global model $\tilde{\theta}$

`// Preparation by the coordinator`
Initialize class subsets $\{\mathcal{S}_1^t, \ldots, \mathcal{S}_d^t\}_{t=1}^T$;
Precompute perturbation noises $\{\delta_k\}_{k \in \mathcal{D}}$;
`// FL process`
for $t \leftarrow 1$ **to** T **do**
  Server samples a subset $\mathcal{K}$ from K clients;
  `// Client-side local update`
  foreach $k \in \mathcal{K}$ **do**
    Initialize local model $\theta_k \leftarrow \tilde{\theta}$;
    if $k \in \mathcal{D}$ **then**
      `// Coalition client training`
      $D_k^{\text{ass}} \leftarrow \texttt{Partition}(D_k, \mathcal{S}_k^t)$;
      $D_k^{\text{rec}} \leftarrow \texttt{Compensation}(D_k, D_k^{\text{ass}})$;
      Train θ_k on $D_k^{\text{ass}} \cup D_k^{\text{rec}}$;
      Perturb θ_k with δ_k;
    else
      `// Standard client training`
      Update θ_k on D_k using ℓ_{ce};
    Send θ_k to server;
  `// Server-side model aggregation`
  $\tilde{\theta} \leftarrow \sum_{k \in \mathcal{K}} \frac{|D_k|}{\sum_{j \in \mathcal{K}} |D_j|} \cdot \theta_k$;
return $\tilde{\theta}$