Fortifying Federated Learning against Membership Inference Attacks via **C**lient-level **I**nput **P**erturbation

Yuchen Yang[§], Haolin Yuan[§], Bo Hui[§], Neil Gong[†], Neil Fendley[§‡], Philippe Burlina[‡], and Yinzhi Cao[§]
[§]Johns Hopkins University, [†]Duke University, [‡]Johns Hopkins Applied Physics Lab
{yc.yang, hyuan4, bo.hui, nfendle1, yinzhi.cao}@jhu.edu, neil.gong@duke.edu, philippe.burlina@jhuapl.edu

***Abstract*—Membership inference (MI) attacks are more diverse in a Federated Learning (FL) setting, because an adversary may be either an FL client, a server, or an external attacker. Existing defenses against MI attacks rely on perturbations to either the model's output predictions or the training process. However, output perturbations are ineffective in an FL setting, because a malicious server can access the model without output perturbation while training perturbations struggle to achieve a good utility. This paper proposes a novel defense, called CIP, to fortify FL against MI attacks via a client-level input perturbation during training and inference procedures. The key insight is to shift each client's local data distribution via a personalized perturbation to get a shifted model. CIP achieves a good balance between privacy and utility. Our evaluation shows that CIP causes accuracy to drop at most 0.7% while reducing attacks to random guessing.**

I. Introduction

Membership inference (MI) attacks [23], [29], [40], [41], [45], [55], [57] allow an adversary to infer if a given sample belongs to a target model's training dataset. For example, adversaries may be able to determine whether a medical image from an hospital was used to train a machine learning based diagnostic system, thus potentially violating patients' protected health information (PHI) and the Health Insurance Portability and Accountability Act (HIPAA). Under the setting of federated learning (FL) [13], [28], [32], [33], [42], [61], Nasr et al. [38] show that MI attacks are more severe as opposed to centralized learning especially when the server is potentially malicious. Such a malicious server poses a unique threat because the server has access to multiple local models during every iteration.

Existing defenses against membership inference attacks are mostly designed for centralized learning and can be broadly classified into two approaches, i.e., relying on either output or training perturbations. Although many of them can be extended to FL, none of them consider the complicated threat model and distributed nature of FL. On one hand, output perturbations, e.g., MemGuard [26], perturb the target model's output for a given input to conceal its membership status. However, such defenses—designed for centralized learning under a blackbox setting—are largely ineffective in FL where both server and client adversaries know the model and can obtain outputs without perturbation.

On the other hand, training perturbation based approaches tamper the training process of the target model via regularization [30], [37], [45], [57] or differential privacy [24], [44], [53]. For instance, adversarial regularization [37] models MI attacks as a regularization term to be used in the training of the target model. Differential privacy (DP), particularly DP-SGD [6], adds perturbations to the gradients in training process such that no single training sample has a significant impact on the learned target model. However, the trade-off between utility and privacy remains a challenging problem: when existing perturbation-based defenses achieve privacy by reducing MI attack accuracy to a certain degree, the target model's accuracy is substantially reduced so as to make the system useless. Using DP as an example. Jayaraman [25] et al. show that the model's accuracy on CIFAR-100 decreases by 50% with a fairly large ϵ value as 10.

In this paper, we propose a novel defense against MI attacks, called CIP (Client-level Input Perturbation), which is designed specifically for federated learning. The key insight is to shift each client's local data distribution via adding personalized perturbation to the local data at both training and inference time. Specifically, the perturbation of CIP is carefully designed with minimizing the training loss over the perturbed training data. In addition, CIP maximizes the training loss over the original training data with a controllable weight so that their outputs from the trained model assemble other non-members. At training time, CIP jointly optimizes an additive, personalized perturbation to training data and local model at each local FL client. Then, at inference time, CIP also adds the same perturbation to every input sample for each local FL model. Intuitively, CIP defends against MI attacks because neither a malicious server/client nor other external adversary can infer the original data distribution via MI attacks against the shifted local model or global model aggregated at the server.

One advantage of CIP is to preserve the FL utility via offsetting each client's local data distribution to better fit the global model via the personalized perturbation. Intuitively, such perturbation, shifts client data distribution and mitigates client heterogeneity via minimizing the training loss on returned global model, thus improving utility. CIP also integrates the existing learning model structure (e.g., ResNet and DenseNet, as a backbone) into a dual-channel architecture to better capture features of training data with the perturbation and further improve utility.

We evaluate CIP against MI attacks both theoretically and empirically. From the theoretical perspective, we demonstrate that the strongest MI attack to CIP is provably less effective when the attacker does not have access to the perturbation. Empirically, we also evaluate CIP on four benchmark datasets with six *adaptive* attacks: Our evaluation shows that CIP can reduce not only state-of-the-art but also adaptive MI attacks' efficacy (which proactively guess the input perturbation) to nearly random guessing without sacrificing model's accuracy. As a comparison, state-of-the-art defenses, including adversarial regularization, differential privacy, and MMD + Mixup, all reduce MI attacks to nearly random guessing but entail an unavoidable and substantial reduction of the model's accuracy.

To summarize, the contributions of our work are as follows:

- We propose an effective and novel defense, CIP, against MI attacks on federated learning models trained from private data samples with multiple sources. CIP preserves model utility for clients with the input perturbation while reducing the MI attack to random guessing.
- We formulate the generation of client-level input perturbation as an optimization problem and also propose a dual-channel neural network architecture to preserve the model's accuracy.
- We analytically prove we can combat the strongest adaptive attack via our defense. We also empirically compare CIP with state-of-the-art defenses on four datasets.

II. Overview

In this section, we give an overview of CIP. We first present the problem definition and the motivation of CIP and then describe the key ideas and threat model.

A. Problem Definition and Notations

Notations. Without loss of generality, we denote D as the training set with n training samples for learning a model, also called a target model. $z = (x, y)$ is a training sample, where $x \in \mathbb{R}^d$ is the training input and y its label. We denote the learning model's parameters as θ and the prediction function as f.

Membership inference attack. Given the target model parameters θ and a data sample $z = (x, y)$, a MI attack aims to infer whether z is in the training set D. In particular, an MI attack essentially computes the following probability:

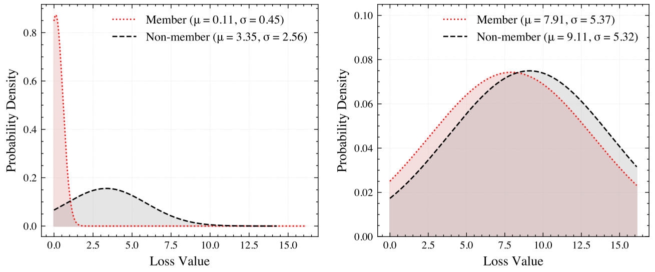
$$\Pr(m = 1 \mid \theta, z), \tag{1}$$

where m is a binary random variable that indicates the membership status of the data sample z. Specifically, $m = 1$ indicates that $z \in D$ and $m = 0$ indicates that $z \notin D$. For instance, an attacker infers z to be a member of D if and only if $\Pr(m = 1 \mid \theta, z) > 0.5$.

B. Motivation

In this subsection, we motivate our idea of perturbation using two examples: (i) a toy example on a pair of numbers following linear distribution, and (ii) a mini-scale experiment on a real-world dataset.

First, let us consider a toy example where all the training samples follow a strict linear distribution, i.e., $\theta^*(X) = 2X+1$, but all the testing samples deviate from, and are scattered around, this linear distribution. An MI attack is obviously possible here: an adversary can infer $z_i = (1, 3) \in D$ is a member with the MSE loss value $MSE(z_i, \theta^*) = 0$; by contrast, a testing sample $z_j = (1.5, 5)$ is a non-member with $MSE(z_j, \theta^*) = 1$.



(a) Loss distribution of original $\theta^*(X)$ (b) Loss distribution of shifted $\theta^*_B(X)$

Figure 1. Motivation of CIP using ResNet-50 on CIFAR-100 (1(a) shows that members and non-members have drastic different loss value distribution. Then, intuitively, 1(b) shows that CIP shifts the loss value distribution and makes them alike between members and non-members for the defense.)

We now explain why CIP defends against this MI attack. Let us assume that CIP shifts the training data distribution via $B(X) = ((1-\alpha)X + \alpha t)$ with $\alpha = 0.5$ and $t = 2$. Ideally, the model trained from perturbed data D is $\theta^*_B = \theta^* \circ B^{-1} = 4X - 3$ where the perturbation function B is invertible in this particular case. Therefore, we can obtain $MSE(z_i, \theta^*_B) = MSE(z_j, \theta^*_B) = 4$, making them non-separable.

There are two things worth noting here. On one hand, no adaptive attacks would be able to infer the original data distribution in this toy example as long as the personalized B is kept local at the FL client. The reason is that the server and other clients only know $\theta^*_B = 4X - 3$, which conceals the original data distribution. Furthermore, the functional space B is infinite and independent from θ^*_B, making it nearly impossible to make guesses. On the other hand, CIP preserves the utility of the original model: As long as θ^* is optimal, θ^*_B is as well given an invertible B.

Second, since the toy example is ideal albeit illustrative, we now demonstrate our motivation beyond the toy example using a real-world training task using ResNet-50 on CIFAR-100 with 10,000 training data (members) and 10,000 testing data (non-members). Our perturbation is similar to $B(X)$ as described in the toy example, but adds another channel, i.e., $((1+\alpha)X - \alpha t)$, to better preserve original samples' features (More details can be found in Section III-A). Figure 1 shows the probability density on recorded loss values before and after applying CIP. Clearly, members and non-members are easily separable on the original θ^* in Figure 1(a) as opposed to the highly overlapped distributions on the shifted θ^*_B in Figure 1(b).

C. Threat model

Our threat model considers both internal and external adversaries of federated learning. We now describe both adversaries following Nasr et al. [38]:

- *Internal adversary.* An internal adversary could be either a malicious client or a malicious server. Because a server adversary is stronger than a client adversary and can perform all the attacks of a client, for simplicity, we use the malicious

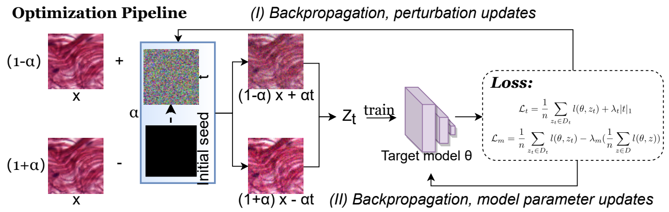

Optimization Pipeline
(I) Backpropagation, perturbation updates
(1-α)
x
+
(1+α)
x
-
α
t
Initial seed
(1-α) x + αt
(1+α) x - αt
Zt
train
Target model θ
Loss:
$\mathcal{L}_t = \frac{1}{n}\sum_{z_t \in D_t} l(\theta, z_t) + \lambda_t |t|_1$
$\mathcal{L}_m = \frac{1}{n}\sum_{z_t \in D_t} l(\theta, z_t) - \lambda_m(\frac{1}{n}\sum_{z \in D} l(\theta, z))$
(II) Backpropagation, model parameter updates

Figure 2. A visualization of optimization steps in CIP using image data as an example in defending against MI attacks. There are two general steps for CIP: (I) Generating perturbations, and (II) Learning the target model. The pipeline of non-image data, such as Purchase-50, is the same except that x is a vector and t is optimized starting from random noise with the same dimension as x.

server as worse case scenario and an upper bound of both malicious server and client.

- *External adversary.* An external adversary could be a malicious third party that is not involved in the training, but has white-box access to the final global model's parameters. That is, an external adversary observes the final global model's weights and makes an inference based on the observation.

III. Methodology

A. Overall Workflow

The key idea of CIP is to let each client generate a unique perturbation $t \in \mathbb{R}^d$ that can be applied locally on both training and testing data. t is personalized for each local client and should be kept as secret from other clients and adversaries. From the utility perspective, the perturbation is optimized by minimizing the training loss of the model in order for a shifted data distribution to better fit the model; the model's parameter distribution trained on data with the input perturbation is also shifted to cope with potential adversaries who query the model with original training or testing data and thus achieves higher privacy guarantee. Keeping perturbation locally is on par with recently proposed personalized federated learning works [43], [47], [58] which allow each client to use different local training strategy.

Next, we describe two important parts of the data perturbation process in CIP which depicts (i) how training and testing data samples are perturbed, and (ii) how perturbed data are used during training and inference time by CIP.

Data Perturbation. We describe next how to perturb training and testing data, i.e., incorporating the perturbation t with federated learning. During the training stage, each client C_i generates a unique t_i for its own training data and then trains a local model with each input x_i blended with t_i. Formally, we have:

$$x_{t_i} = \mathscr{B}(x_i, t_i) = ((1-\alpha)x_i + \alpha t_i,\ (1+\alpha)x_i - \alpha t_i), \quad (2)$$

where $\mathscr{B}$ is our blending function, α is called *blending parameter*, and the blended input x_t is a pair composed of $(1-\alpha)x + \alpha t$ and $(1+\alpha)x - \alpha t$. The blended input is then clipped within the range of x for further processing. Such a blended input includes more information about the original input x and the perturbation t, which enables a better privacy-utility trade-off. Then, the client communicates the local model to the server, and the server aggregates all local models and then sends the aggregation results back to each client. During testing and inference times, each client C_i also adds their own perturbation t_i to use the trained global model.

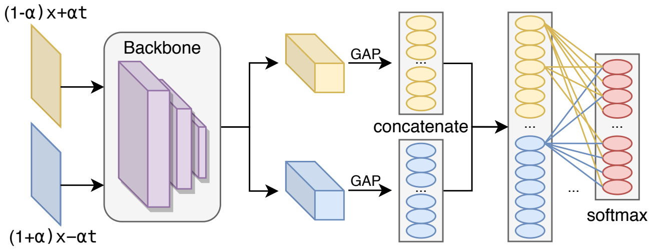


Figure 3. Our dual-channel neural network architecture. Specifically, both components of a blended input, i.e., $(1-\alpha)x + \alpha t$ and $(1+\alpha)x - \alpha t$, separately go through a backbone and then a global average pooling (GAP) layer. Then, the two GAP outputs are concatenated and processed by a fully connected layer. Note that the dual-channel architecture is model agnostic because the backbone can be replaced by any model architecture, e.g., ResNet or DenseNet.

Dual-channel Model. In this part, we describe how CIP accepts perturbed data via a novel dual-channel neural network architecture for improving utility. Figure 3 shows this architecture, which contains the following parts: a backbone network accepting two channels, a global average pooling (GAP) layer, a fully connected layer for concatenation, and a softmax layer. The backbone network may include any existing neural network structures, including ConvNets such as DenseNet, and ResNet, or transformers (e.g. vision transformer or Swin), which is then finally connected with an additional global average pooling (GAP) layer. The backbone takes two channels' inputs separately to produce two outputs and then the outputs from these two channels are concatenated as an input to the GAP layer. That is why the size of the GAP layer is twice as large as a normal GAP layer if it is connected with the backbone.

There are two advantages of using a dual-channel architecture. First, dual-channel inputs keep more features of original sample x while introducing the input perturbation t, which largely maintains the utility. Second, using one backbone means that both channels share the same model instead of placing two same-structured backbones, thus reducing the overall model's size. Then, the fully connected layer concatenates the outputs from the GAP layer and feeds the outputs to the softmax layer for prediction.

B. Formulating CIP as an Optimization Problem

In this subsection, we describe how to formulate CIP as an optimization problem via two steps: (i) generating perturbations and (ii) learning the target model. Note that for simplicity, we denote by $D_t = \{(x_t, y) \mid (x, y) \in D, x_t = \mathscr{B}(x, t)\}$ the training samples with the perturbation blended. We also denote by $z_t = (x_t, y)$ a data sample with the perturbation blended.

Main goals. We aim to achieve the following two goals for both steps of CIP:

- *Goal 1.* Protecting membership privacy of the original training samples D from adversaries.
- *Goal 2.* Preserving accuracy / utility of the target model for inputs blended with the perturbation for clients.

Intuitively, Goal 1 means that the target model should not memorize the original training samples in D, so as not to leak their membership status to adversaries. Goal 2 means that the target model can accurately classify an input when it is blended with the perturbation. In particular, we alternately perform the following two steps as shown in the top part of Figure 2:

1) Step I: Generating Perturbations: Step I is shown in the top part marked as "Backpropagation, perturbation updates" in Figure 2. Given a target model, we aim to generate a perturbation to achieve Goal 2, i.e., the perturbation minimizes the loss over the training samples D_t. Formally, we formulate generating such a perturbation via the following optimization problem:

$$\min_t \mathcal{L}_t \text{ where } \mathcal{L}_t = \frac{1}{n}\sum_{z_t \in D_t} l(\theta, z_t) + \lambda_t |t|_1, \tag{3}$$

whereby $l(\theta, z_t)$ is a loss function of the target model θ for a training sample z_t blended with the perturbation (the first sum term is the cross entropy), $|t|_1$ is used to regularize the magnitude of the perturbation, and λ_t is a hyperparameter to balance the two terms. In our implementation, we choose the cross entropy as the first loss term in Equation (3).

We generate the perturbation in Step I via directly treating perturbation t as variable. Specifically, we initialize the perturbation t as some random input. Then, we use the standard stochastic gradient descent (SGD) to solve the optimization problem in Equation (3) with respect to t.

2) Step II: Learning the Target Model: Step II is shown on the bottom part of Figure 2 marked as "Backpropagation, model parameter updates". Given a perturbation t, we aim to learn a target model to achieve both Goal 1 and Goal 2, i.e., the target model maximizes the loss over the original training samples D while minimizing the loss over the training samples with the perturbation blended D_t. We formulate learning the model as the following optimization problem:

$$\min_\theta \mathcal{L}_m \text{ where } \mathcal{L}_m = \frac{1}{n}\sum_{z_t \in D_t} l(\theta, z_t) - \frac{\lambda_m}{n}\sum_{z \in D} l(\theta, z), \tag{4}$$

whereby λ_m is a hyperparameter to balance the two cross entropy loss terms to avoid abnormally high loss on original data. We note that the standard training of the target model aims to minimize the loss term $\frac{1}{n}\sum_{z \in D} l(\theta, z)$ as opposed to maximizing the loss term in CIP.

C. Theoretical Adversarial Advantage Analysis

In this subsection, we theoretically analyze privacy provided by CIP via adaptive attacker's adversarial membership inference advantage. We first define the attacker's *adversarial advantage (Adv)* as below:

$$Adv(\theta, z) = \frac{\Pr(m = 1 \mid \theta, z)}{\Pr(m = 0 \mid \theta, z)}. \tag{5}$$

Intuitively, the adversarial advantage is a way to quantify the membership privacy for a data sample z: A larger adversarial advantage corresponds to less membership privacy.

We then have Theorem 1 based on the definition of adversarial advantage. The intuition taken from the theorem is that an adversary does not gain an additional adversarial advantage when guessing a perturbation that is different from the original one. More specifically, Theorem 1 shows the gap between these two adversarial advantages, as represented by ϵ, depends on the loss difference between two data samples with true and guessed input perturbation: A large difference in the loss indicates a small adversary advantage. In other words, for the strongest MI attacks, our defense achieves better membership privacy when an attacker does not know our secret trigger.

Theorem 1. *Given a target model θ, our true perturbation t, and a perturbation t' guessed by an attacker. Assume $l(\theta, z_t) \leq l(\theta, z_{t'})$. Then, the adversarial advantage $Adv(\theta, z_{t'})$ is bounded by the adversarial advantage $Adv(\theta, z_t)$. Formally, we have the following:*

$$Adv(\theta, z_{t'}) = \epsilon \cdot Adv(\theta, z_t), \tag{6}$$

where $\epsilon = e^{-\frac{1}{T}(l(\theta, z_{t'}) - l(\theta, z_t))} \leq 1$, and T is the temperature parameter.

Proof. We denote by a binary random variable m_t the membership status for z_t. By applying the Bayes' rule, we have:

$$\begin{aligned} \Pr(m_t = 1 \mid \theta, z_t) &= \frac{\Pr(m_t = 1, \theta \mid z_t)}{\Pr(\theta \mid z_t)} \\ &= \frac{\gamma_t \cdot \eta_t}{\gamma_t \cdot \eta_t + \beta_t \cdot (1 - \eta_t)}, \end{aligned} \tag{7}$$

where $\gamma_t = \Pr(\theta \mid m_t = 1, z_t)$, $\beta_t = \Pr(\theta \mid m_t = 0, z_t)$, and $\eta_t = \Pr(m_t = 1 \mid z_t)$. Therefore, the adversarial advantage $Adv(\theta, z_t)$ is as follows:

$$Adv(\theta, z_t) = \frac{\Pr(m_t = 1 \mid \theta, z_t)}{\Pr(m_t = 0 \mid \theta, z_t)} = \frac{\gamma_t \cdot \eta_t}{\beta_t \cdot (1 - \eta_t)}. \tag{8}$$

For $Adv(\theta, z_{t'})$, since when $m_t = 0$ (or $m_{t'} = 0$), θ does not depend on z_t (or $z_{t'}$), we have $\beta_{t'} = \beta_t$. Moreover, since η_t (or $\eta_{t'}$) essentially is the prior probability that z_t (or $z_{t'}$) is a member without observing anything from the target model, we have $\eta_{t'} = \eta_t$. Thus, we have:

$$\frac{Adv(\theta, z_{t'})}{Adv(\theta, z_t)} = \frac{\gamma_{t'}}{\gamma_t} = \frac{\Pr(\theta \mid m_{t'} = 1, z_{t'})}{\Pr(\theta \mid m_t = 1, z_t)}. \tag{9}$$

Following previous work [40], we assume that the target model θ follows a probability distribution determined by the training loss used to learn the target model. This is reasonable because the randomness presented in θ can be due to the training process, such as Bayesian posterior sampling, or occurs naturally, as is the case with Stochastic Gradient methods. Therefore, θ follows the following probability distribution:

$$\Pr(\theta \mid D_t) \propto e^{-\frac{n}{T}\mathcal{L}_m} \tag{10}$$

$$= e^{-\frac{1}{T}\sum_{z_t \in D_t} L(\theta, z_t)}, \tag{11}$$

where T is the temperature parameter and $L(\theta, z_t) = l(\theta, z_t) - \lambda_m l(\theta, z)$. Given the above probability distribution for the target model θ, we define the posterior over the parameters given a sample z_t and its memberships m_t:

$$\Pr(\theta \mid m_t = 1, z_t) = \frac{1}{c_t} \cdot E^{n-1}_{r \sim \mathcal{Z}_t}[e^{-\frac{1}{T}L(\theta,r)}] \cdot e^{-\frac{1}{T}L(\theta,z_t)} \tag{12}$$

where $\mathcal{Z}_t$ is the probability distribution of z_t, E^{n-1} is the expected value over $n-1$ samples (except z_t), and c_t is defined as follows ($c_{t'}$ is similar):

$$c_t = \int_\theta E^{n-1}_{r \sim \mathcal{Z}_t}[e^{-\frac{1}{T}L(\theta,r)}] \cdot e^{-\frac{1}{T}L(\theta,z_t)} d\theta. \tag{13}$$

If we consider θ to be the set of all possible model parameters $\theta = \{\theta_1, \theta_2, \theta_3, ..., \theta_i, ...\}$, because z_t and $z_{t'}$ are linear combinations of (x, t) and (x, t'), respectively, there exists a transformation $\mathbf{A}$ such that $\mathbf{A}(z_{t'}) = z_t$, $L(\theta_i, z_t) = L(\theta_i, \mathbf{A}(z_{t'}))$. In other words, there must exist $j, k \in [1, i]$ such that $L(\theta_j, z_t^j) = L(\theta_k, z_{t'}^k)$. Then the integral of $e^{-\frac{1}{T}L(\theta,z_t)}$ with respect to θ is equal to the integral of $e^{-\frac{1}{T}L(\theta,z_{t'})}$ with respect to θ, and thus $c_t = c_{t'}$.

Since $l(\theta, z_t) \leq l(\theta, z_{t'})$ (as $l(\theta, z_t)$ is minimized during training), we have:

$$\frac{Adv(\theta, z_{t'})}{Adv(\theta, z_t)} = \frac{c_t}{c_{t'}} \cdot e^{-\frac{1}{T}(L(\theta,z_{t'}) - L(\theta,z_t))} \tag{14}$$

$$= \frac{c_t}{c_{t'}} \cdot e^{-\frac{1}{T}(l(\theta,z_{t'}) - l(\theta,z_t))} \tag{15}$$

$$= \frac{c_t}{c_{t'}} \cdot \epsilon = \epsilon \leq 1 \tag{16}$$

□

IV. Experimental Setup

We implement CIP with 1,563 lines of Python 3.8 code based on TensorFlow 2.4.0. The implementation is open-source at this anonymous repository (https://github.com/yhhmia/CIP). All the experiments are performed using a GeForce RTX 2080 and Titan XP graphics cards (NVIDIA).

A. Dataset and Model Setting

In this section, we describe different learning models used by CIP and the evaluation. we describe our deep learning models set-up for both internal and external adversaries for five datasets: (i) CIFAR-100 (a popular benchmark dataset with 100 classes containing 600 images each), (ii) CIFAR-AUG (CIFAR-100 with data augmentation, i.e., each image being resized to 80×80, cropped to 64×64, and being flipped from left to right), (iii) CH-MNIST (a benchmark dataset [27] of 5,000 histological images of human colorectal cancer including 8 classes of tissues), and (iv) Purchase-50 (a dataset from Kaggle's "Aquired Valued Shoppers Challenge", which contains 20,000 data samples of purchase history of 50 shoppers: half for training and half for shadow model in which some member inference attacks uses to train the attack model).

Table I
[Internal Adversary Setup] Parameters of Legacy model (without defense) and CIP. (lr: learning rate; per.: perturbation; acc.: accuracy).

Model	Legacy Model Parameters				CIP Parameters			
	#clients	#Train iter	Train acc.	Test acc.	#Attacking iter	lr (per.)	λ_m	λ_t
ResNet	2	120	0.970	0.545	80, 100, 120	1e-2	1e-6	1e-8
	5	300	0.985	0.543	180, 240, 300	1e-2	1e-6	1e-8
	10	500	0.975	0.529	300, 400, 500	1e-2	1e-6	1e-8
	20	800	0.957	0.357	600, 700, 800	1e-2	1e-6	1e-8
	50	1500	0.924	0.328	1300, 1400, 1500	1e-2	1e-6	1e-8
DenseNet	2	300	0.943	0.565	180, 240, 300	1e-2	1e-6	1e-8
	5	600	0.921	0.587	400, 500, 600	1e-2	1e-6	1e-8
	10	1000	0.929	0.504	800, 900, 1000	1e-2	1e-6	1e-8
	20	1500	0.932	0.372	1300, 1400, 1500	1e-2	1e-6	1e-8
	50	3000	0.948	0.332	2800, 2900, 3000	1e-2	1e-6	1e-8
VGG	2	300	0.907	0.613	180, 240, 300	1e-2	1e-6	1e-8
	5	600	0.882	0.614	400, 500, 600	1e-2	1e-6	1e-8
	10	1000	0.947	0.541	800, 900, 1000	1e-2	1e-6	1e-8
	20	1500	0.982	0.471	1300, 1400, 1500	1e-2	1e-6	1e-8
	50	3000	0.966	0.424	2800, 2900, 3000	1e-2	1e-6	1e-8

Table II
[External Adversary Setup] Parameters of Legacy model (without defense) and CIP. (lr: learning rate; per.: perturbation; acc.: accuracy).

Dataset	Model	Legacy Model Parameters			CIP Parameters			
		#Train iter	Train acc.	Test acc.	lr (training)	lr (per.)	λ_m	λ_t
CIFAR-100	ResNet	40	0.998	0.323	5e-5	1e-3	1e-12	1e-6
CIFAR-AUG	ResNet	40	0.986	0.434	5e-5	1e-3	1e-12	1e-3
CH-MNIST	ResNet	70	0.993	0.899	5e-5	1e-3	1e-12	1e-6
Purchase-50	MLP	40	0.991	0.755	5e-5	1e-3	1e-12	1e-12

Next, we describe our deep learning models set-up for both internal and external adversaries. Following the approach of prior work [45], we use several models with different overfitting characteristics and robust levels, including some that are overfit (and with low accuracy), and some that are not (with high accuracy), and some with and without data augmentation. The batch size are set to be 32 for all cases.

- *Internal Adversary.* First, Table I shows the hyperparameters, and testing and training accuracies of target models. Specifically, we follow the averaging aggregation method used by prior works [28], [38], with the number of clients as 2,5,10,20 and 50. We denote communication rounds between client and server as training iteration shown in the table, and we set default value of local training epoch as 1. Our evaluation and model accuracies are consistent with prior work [38]: We use SGD optimizer to train the local model with decaying learning rate of 1e-3, 5e-4, and 1e-4. Second, Table I shows the hyperparameters of CIP. It is worth noting that the training iteration of the target models without CIP doubles compared with these with CIP. The attack iterations are in consistent with prior work [38].
- *External Adversary.* Table II shows the hyperparameters, and testing and training accuracies of target models. we adopt ResNet50 for CIFAR-100, CIFAR-AUG, CH-MNIST, and multilayer perceptron (MLP) model with three dense layers with size of 512, 256, and 128 for Purchase-50. Prior work [38] shows that the less the number of clients, the more vulnerable the target model is. To evaluate the worst-case scenario for our defense, we intentionally set the

number of clients as one to enhance external adversaries. Let us describe all the models on different datasets. The CIFAR-100 model is overfitted with a low testing accuracy of 0.323. We follow this for MI attacks as demonstrated in the literatures [23], [29], [40], [41], [45], [57]; The CH-MNIST model is well trained with a high testing accuracy (i.e., 0.899). The CIFAR-AUG model adopts data augmentation: The purpose is to show that CIP can be combined with other data augmentation techniques. The Purchase-50 model is to show the applicability of CIP on non-image datasets.

B. Membership Inference Attack Setting

In this subsection, we describe the settings of different MI attacks for both internal and external adversaries.

- *Internal Adversary.* State-of-the-art MI attacks [38] on internal adversary assume either the server or a client is malicious. We assume the server is malicious in the evaluation because it is a stronger threat model compared to a malicious client. There are two types of server attacks: passive and active. Following Nasr et.al [38], for passive attacks, we attack on several latest iterations, shown in the Table I; for active attack, we repeat gradient ascent for each epoch of the training and select 100 members and 100 nonmembers to test.
- *External Adversary.* External adversaries targets at final released global model. There are two types: output-based (Ob) and parameter-based (Pb). The former needs the model's output and the latter needs the model's parameters in addition to its outputs. We evaluate four state-of-the-art attacks: Ob-Label (an output-based attack [57] using label information), Ob-MALT (a Bayes Optimal output-based attack [40]), Ob-NN (an output-based attack using Neural Networks [41], [45]), Ob-BlindMI (the state-of-the-art output-based attack [23] using differential comparison), and Pb-bayes (the state-of-the-art parameter-based attack [29] using Bayes)

V. Evaluation

In this section, we evaluate CIP in answering the following four Research Questions (RQs).

- [RQ1] How does CIP compare with existing defenses in terms of testing accuracy and attack accuracy?
- [RQ2] How does CIP maintain the performance benefits brought by federated learning?
- [RQ3] How effective is CIP in defending against different variations of MI attacks?
- [RQ4] How effective is CIP in defending against adaptive adversaries?
- [RQ5] What are the performance and model size overheads of CIP?

A. RQ1: Comparison with Prior Defenses

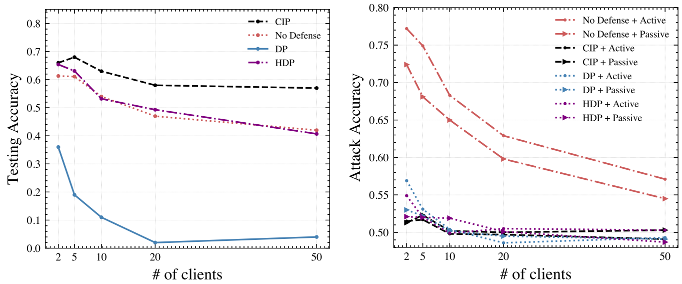
In this research question, we compare CIP with five state-of-the-art defenses:

- Differential Privacy (DP). We use an open-source implementation [4] of DP-Adam [6] with different ϵ values for the comparison.
- High-Accuracy Differential Privacy (HDP). We use an open-source implementation [2] of a recent improvement of DP [48] with different ϵ values for the comparison.
- Adversarial Regularization (AR). We adopt an open-source version [1] for the comparison and change the hyperparameter λ to control the privacy level.
- Mixup + MMD (MM). We implement the defense following Li et al. [30]. The tunable parameter controlling the weight of MMD loss is called μ.
- RelaxLoss (RL). We adopt an open-source version [3], [11] for the comparison and change the hyperparameter ω to control the privacy level.

The comparison is based on two metrics, testing accuracy and attack accuracy, under both internal and external adversaries as documented in Section II-C.

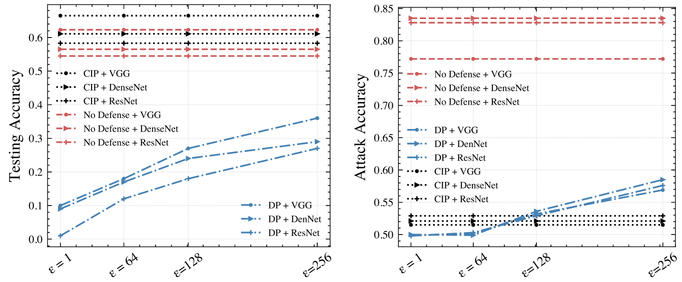
Comparison under Internal Adversaries. In this part, we compare CIP with differential privacy against internal adversaries. Specifically, we use local DP (LDP) because the alternative, i.e., central DP (CDP), does not defend against a malicious server as assumed in an internal adversary [38]. We set the $\delta = 1e - 5$ and evaluate different values of ϵ for DP. We did not include AR and MM because there are *no* implementations of either defense against internal adversaries in federated learning. We use CIFAR-100 in this experiment with 50,000 training data and 10,000 testing data, while attacking we regard each client's training data as members and use the same size of testing data as non-members. We follow Naseri et al. [36]'s non-i.i.d data distribution as our default setting, i.e., 20 random classes per client. We follow Milad et al. [38] to use the same size of training data for all clients, which is selected uniformly at random from the chosen classes of data sample. Our comparison is based on three perspectives: (i) different clients (ii) different model architectures, and (iii) different ϵ values.

First, we compare CIP with both DP and HDP with different number of clients. Figure 4(a) shows the testing accuracy of three approaches: CIP outperforms both DP and HDP. DP's testing accuracy is significantly lower than CIP and no defense even when ϵ is set to be 128. The performance of DP is getting worse as the number of clients increases, which drops to approximately 0.05 when FL has 20 or 50 clients. HDP's testing accuracy is on par with or sometimes higher than no defense when ϵ is also 128. The reason is the augmentation by additional training data of ImageNet. As a comparison, the performance of CIP also drops but is relatively stable. Interestingly, the testing accuracy with CIP is generally higher than without defenses. The reason is that CIP allows each client to optimize a local perturbation, which can shift the heterogeneous local data distribution to align with the others, thus the accuracy is even better than no defense. We will discuss more details about different data distributions in RQ2. We then look at the attack accuracies in Figure 4(b). The active and passive attack accuracies for CIP is close to random guessing



(a) Testing Accuracy vs. # of clients (b) Attack Accuracy vs. # of clients

Figure 4. [RQ1-Internal]: Comparison of CIP ($\alpha = 0.5$), DP, HDP, and no defense with different # of clients.



(a) Testing Accuracy vs. Different Model (b) Attack Accuracy vs. Different Model

Figure 5. [RQ1-Internal]: Comparison of CIP ($\alpha = 0.5$) and DP with varied models and two clients under varied ϵ.

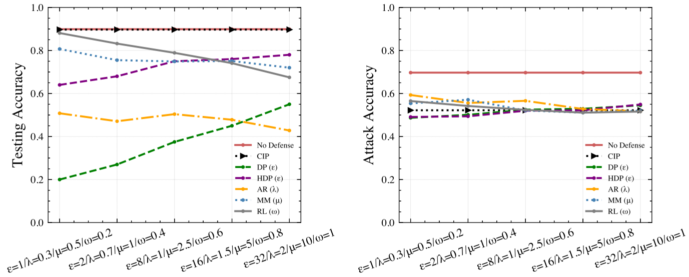
for any number of clients. As a comparison, the active attack accuracies for DP and HDP are still a little bit high when we have two clients because of the large ϵ value. This observation aligns with prior work [38]. Then, both attack accuracies for DP and HDP also drop as the number of clients increases. For no defense, both active and passive attacks accuracies of no defense are much higher than the ones of CIP.

Secondly, we compare CIP and DP using three model architectures, i.e., VGG, DenseNet, and ResNet. Figure 5(a) shows the testing accuracies. VGG performs the best in terms of testing accuracy, then is DenseNet and ResNet. This trend is the same for both CIP and DP. We then look at the attack accuracies in Figure 5(b). Still, VGG performs the best with the lowest attack accuracy, and the performances of ResNet and DenseNet are similar with ResNet being slightly more robust to attacks.

Thirdly, we observe the performance of DP as the value of ϵ increases. Let us start from the testing accuracy in Figure 5(a). The testing accuracies of three model architectures are all below 0.1 when ϵ equals to one; when ϵ increases to 256, the model's testing accuracies all increase to around 0.3, which is about half of CIP's testing accuracy. We then look at the attack accuracy in Figure 5(b). When ϵ is smaller than 64, the attack accuracy is very close to random guessing. Then, the attack accuracy increases as the ϵ value increases. It is very hard to find a balance of privacy and utility in DP for federated learning.

[RQ1] Take-away-Internal: CIP outperforms DP with a higher testing accuracy and a similar attack accuracy for different number of clients and different model architectures under an internal adversary.

Comparison under External Adversary. In this part, we compare CIP with state-of-the-art approaches against external adversary. We use CH-MNIST in this experiment as it has high testing accuracy with reasonable attacking accuracy. We use 2,500 balanced-class data as training data (members while attacking) and 2,500 balanced-class data as testing data (non-members while attacking). Note that to evaluate the defense ability of CIP in the worst case scenario, we consider the most vulnerable system with one client so that external adversaries can achieve higher attack accuracy without defense.



(a) Testing Accuracy vs. Different defenses (b) Attack Accuracy vs. Different defenses

Figure 6. [RQ1-External]: Comparison of CIP and state-of-the-art defenses (including DP, HDP, AR, RL, and MM) using CH-MNIST dataset.

- *Testing accuracy comparison.* In this part, we compare the model's performance in terms of the testing accuracy of all approaches. Figure 6(a) shows the testing accuracy of a model with no defense, CIP, local DP (LDP, or for short DP in this subsubsection), AR, and MM, as the privacy budget changes. Specifically, we select ϵ (for DP and HDP) as [1, 2, 8, 16, 32], λ (for AR) as [0.3, 0.7, 1, 1.5, 2], μ (for MM) as [0.5, 1, 2.5, 5, 10], ω (for RelaxLoss) as [0.5, 1, 2.5, 5, 10]. First, the model's accuracy is very close for CIP (with α value as 0.9 to provide better privacy) and no defense. (Note that the accuracy is a constant line because there is no privacy budget value for no defense.) That is, CIP can provide practical defense against MI attacks without sacrificing mode's performance. Second, the existing defenses are struggling in maintaining the utility while preserving the privacy. The testing accuracy of DP decreases as the ϵ decreases because a smaller ϵ provides better privacy. Even if with a relatively large $\epsilon = 32$, the accuracy drops for 40%. HDP improves over DP in terms of testing accuracy. Specifically, HDP brings around 20% to 40% accuracy improvement, especially with a small ϵ (e.g., 1 or 2), compared with the original DP. However, the testing accuracy still drops between 11% and 25% of HDP compared with CIP. Similarly, the testing accuracy drops are 40% to 50% of AR, 10% to 20% of MM, and 1% to 20% of RelaxLoss with different privacy budgets as a comparison with CIP.
- *Attack accuracy comparison.* In this part, we compare the attack accuracy of different approaches using Pb-Bayes, the state-of-the-art, strongest, whitebox attack. Figure 6 shows the attack accuracy of no defense, CIP, DP, AR, MM. First,

Table III
[RQ2] Accuracy of CIP, No Defense and Local Training with different data distributions of CIFAR-100 and five clients, which spans from non-i.i.d. to i.i.d.

Data Distribution (classes per client)	20 (non-i.i.d.)	40	60	80	100 (i.i.d.)
CIP (ours)	**0.683**	**0.676**	**0.672**	**0.670**	0.665
No Defense	0.611	0.635	0.653	0.668	**0.672**
Local Training*	0.674	0.616	0.525	0.483	0.439

*: Note that local training only learns a model with the number of classes that it has. That is why the accuracy for a non-i.i.d. setting is higher than that for an i.i.d. setting. For example, local training learns a 20-class classifier in a non-i.i.d. setting, and a 100-class classifier in an i.i.d. setting. This also demonstrates an advantage of collaborative training, which brings additional classes one client does not have.

the attack accuracy for models with no defense is high, being around 0.69. Then, all defenses bring down the attack accuracy to around 0.5. Second, the attacking accuracies against existing defenses increases and are higher than CIP when their privacy budget is set for better utility, i.e., $\epsilon \geq 8$ for DP, $\lambda \leq 2$ for AR, $\mu \leq 2.5$ for MM, which again indicates that they cannot find the trade-off between utility and privacy.

[RQ1] Take-away-External: On the testing accuracy perspective, CIP preserves model's accuracy while existing defenses, particularly DP, AR, and MM, decrease the model's accuracy from 10% to 70%. On the privacy perspective, CIP with $\alpha = 0.9$ can defend against white-box membership inference attack effectively as DP, AR, and MM do with appropriate privacy budget.

B. RQ2: Maintenance of FL's Performance

In this research question, we show that CIP still maintains the performance benefits brought by FL and most importantly outperforms training a local model without federated learning. We use CIFAR-100 in this experiment with 50,000 training data and 10,000 testing data divided equally by five clients. We set the data distribution from non-i.i.d to i.i.d, i.e., 20, 40, 60, 80, 100 classes per client. Note for both CIP and no defense federated learning, we evaluate each client's testing data on the aggregated global model (100-class classification problem no matter what data distribution) and get the average testing accuracy; for local training, we evaluate each client's testing data on her own local model (20-class classification if the data distribution is 20 classes per client.) and get the average testing accuracy. There is no aggregation during the training process, which is, each local model is trained on 10,000 local training data. We want to show the benefits that CIP brings to federated learning which can not be achieved by local training.

CIP vs. No defense. We first compare CIP with no defense as shown in the CIP and "No Defense" rows of Table III. As the data distribution becomes more non-i.i.d., i.e., the number of classes assigned to each client decreases, the testing accuracy of no defense decreases which is aligned with the previous work [32], [42] while that of CIP increases slightly. On the contrary, as the data distribution becomes more i.i.d., the testing accuracy of no defense increases while that of CIP drops slightly.

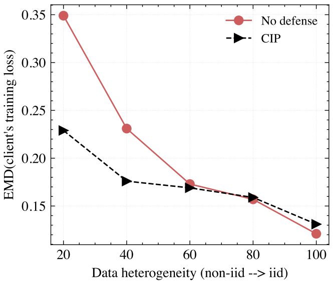


Figure 7. Earth mover distance (EMD) of client's training loss with different data heterogeneity. CIP shifts such distributions for non-i.i.d. data.

Note that the testing accuracy of CIP is higher than no defense with data distribution as 20, 40, 60, 80 classes per client. Even when the data distribution is 100 classes per client (i.i.d.), CIP gets a 0.007 accuracy drop than the model without any defenses. This improvement for non-i.i.d. data is comparable to the performance improvement of personalized federated learning [32], [42], [54]. Our perturbation is personalized for each FL client and not aggregated at the server. Therefore, such perturbation can shift data distribution of clients to be close to each other. As a comparison, in an i.i.d. scenario, such a personalized perturbation does not help much, i.e., CIP will not perturb data too much since each client's data distribution already aligns with each other and therefore there is only a slight accuracy drop comparing CIP and no defense.

To further explain why CIP improves FL's performance for non-i.i.d. data, we run an experiment of CIP on CIFAR-100 with ten clients and $\alpha = 0.3$. Specifically. we use the Earth Mover Distance (EMD) between the client's local training loss, on local data, during all training iterations to represent local data distribution differences between clients. Our calculation of EMD is as follows. We calculate each client-side, local model's training loss for each training round and then record the losses across all the training round as the distribution of training loss. Then, we calculate the Earth Mover Distance (EMD) of such distributions between all client pairs. The value in Figure V-B is the average EMD (training loss) between every two clients among ten clients. Compared with no defense, CIP can shift data distribution to reduce the distance between local training loss for heterogeneous data distribution (e.g., non-i.i.d.), thus improving FL's performance.

CIP vs. Local training. We then compare CIP with local training with client collaboration under FL. The purpose is to show that CIP can improve the model's performance compared to a local model trained with local dataset alone. It is worth noting that the benefits brought by collaborative training like FL are not only performance but also the number of classes especially under a non-i.i.d. setting where other clients may have labelled classes that one client does not have.

The CIP and "Local Training" rows of Table III show the testing accuracies. Compared with CIP, local training's accuracy is always lower than CIP. The reasons are as follows. (i) CIP can benefit from federated learning in which the global model is aggregated from all clients' local model, so the size of training data can be regarded as 50,000 while the local training is only 10,000. (ii) CIP can shift the data distribution to align with each other under non-i.i.d. distribution, and therefore CIP can still outperform local training even under a non-i.i.d. setting with 20 classes per client. Another thing worth noting is that the model's testing accuracy of local training is the highest at 20 classes per client and becomes lower when the number of classes increases. The reason is that the 20-class classification problem is much easier than the 100-class classification problem.

> **[RQ2] Take-away:** The performance of CIP is on par with, and sometimes (i.e., under non-i.i.d. setting) even better than, FL without any defense. CIP maintains the performance benefits brought by a collaborative learning like FL because the performance of CIP is always better than local training (i.e., without contribution from other clients).

C. RQ3: State-of-the-art MI attacks

In this research question, we evaluate CIP against five different state-of-the-art MI attacks as documented in Section IV-B. We adopt four different datasets against external adversaries. We show CIP can decrease the attacking accuracy effectively while maintaining the testing accuracy. We focus on an external adversary with the same setting described in Section V-A.

Attack Accuracy. We start from attack accuracy and show the accuracies of different attacks in Figure 8 (with Figure 8(a) on CIFAR-100, Figure 8(b) on CIFAR-AUG, Figure 8(c) on CH-MNIST, and Figure 8(d) on Purchase-50) as the blending parameter α increases. First, the attack performance decreases as the α value increases. This is because the larger the α is, the more perturbations that CIP brings to the original image is. In practice (e.g., in RQ1), we use a α as 0.9 to provide a privacy protection of the model.

Second, let us compare different datasets. The attack accuracy on CIFAR-100 is the highest and the accuracies on both CIFAR-AUG, CH-MNIST, and Purchase-50 are quite similar. The reason is that the model on CIFAR-100 is extremely overfitted, but the other three are less overfitted as shown in Section IV-A. This aligns with observations from prior works [23], [40], [41], [45], [57] and also exists in the presence of CIP.

Third, let us compare different attacks against CIP. We start by comparing parameter-based with output-based attacks against CIP. Pb-Bayes is the most powerful attack against CIP, the reason is that Pb-Bayes has access to the model's parameters in addition to the outputs. We then look at three different output-based attacks. The performances of three attacks are similar to Ob-BlindMI.

Attack Precision, Recall, and F1-score. We also describe and compare different MI attacks' precision, recall and F1-score against CIP ($\alpha = 0.7$) in Table IV. We report several observations here. First, CIP is generally effective in reducing both attacks' recall and precision, i.e., recall value below 0.5 and precision value around 0.5. Second, CIP performs better in reducing recall than precision, and the reason is that CIP leads the attacker to misclassify the training data without perturbation as testing data, which results in a high false-positive rate. The precision and recall are generally balanced for Ob-NN, but extremely unbalanced for Ob-MALT, which leads to a generally low F1-score for Ob-MALT. Third, the attack F1-score against CIP is generally on par with, and lower than, the attack accuracy, which indicates that attack accuracy is a good metrics for the evaluation of defense. In other words, CIP is highly effective in reducing the attack recall against Ob-MALT, which leads to the unbalance.

Table IV
[RQ3] Precision, Recall, F1-Score and Accuracy of different attacks against CIP ($\alpha = 0.7$).

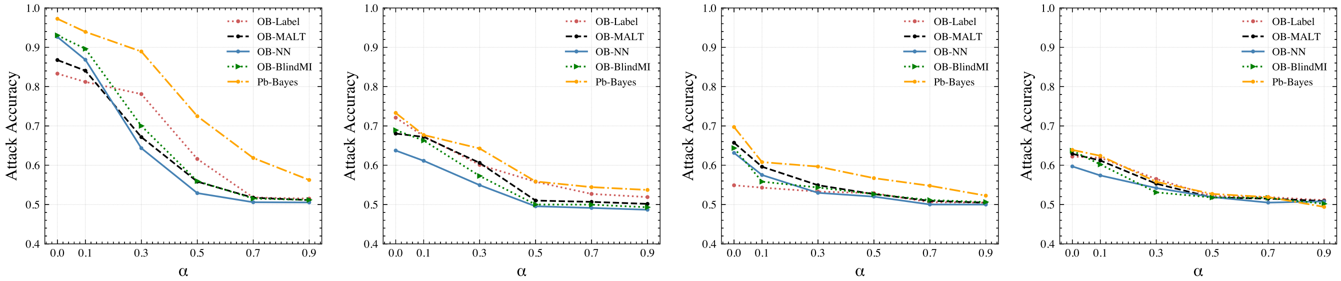
Dataset	Attack	Precision	Recall	F1-Score	Accuracy
CIFAR-100	Ob-Label	0.539	0.256	0.347	0.518
	Ob-MALT	0.598	0.105	0.178	0.517
	Ob-NN	0.509	0.326	0.397	0.506
	Ob-BlindMI	0.515	0.468	0.491	0.515
	Pb-Bayes	0.686	0.447	0.541	0.621
CIFAR-AUG	Ob-Label	0.537	0.388	0.450	0.527
	Ob-MALT	0.522	0.159	0.244	0.506
	Ob-NN	0.484	0.259	0.373	0.491
	Ob-BlindMI	0.474	0.022	0.041	0.499
	Pb-Bayes	0.615	0.235	0.341	0.544
CH-MNIST	Ob-Label	0.506	0.451	0.477	0.506
	Ob-MALT	0.523	0.215	0.305	0.509
	Ob-NN	0.497	0.373	0.426	0.498
	Ob-BlindMI	0.523	0.263	0.350	0.511
	Pb-Bayes	0.588	0.317	0.412	0.548
Purchase-50	Ob-Label	0.524	0.234	0.324	0.511
	Ob-MALT	0.534	0.237	0.328	0.515
	Ob-NN	0.506	0.408	0.451	0.505
	Ob-BlindMI	0.524	0.371	0.434	0.517
	Pb-Bayes	0.528	0.357	0.426	0.519

Table V
[RQ3] Testing accuracy of CIP with different α on different datasets.

Dataset	α					
	0 (No defense)	0.1	0.3	0.5	0.7	0.9
CIFAR-100	0.323	0.335	0.328	0.327	0.323	0.316
CIFAR-AUG	0.434	0.474	0.457	0.436	0.422	0.398
CH-MNIST	0.899	0.921	0.904	0.905	0.903	0.892
Purchase-50	0.755	0.768	0.757	0.754	0.755	0.741

Testing Accuracy. We then illustrate that CIP can maintain testing accuracy comparing with no defense in Table V. We observe that CIP will not decrease the testing accuracy if appropriate α is adopted, e.g., $\alpha \leq 0.5$. However, if a larger α, e.g., $\alpha \geq 0.7$, is applied for the purpose of stronger defense, the accuracy decrease is 1.6% on average. It is worth noting that ideally if the model can fully recover the original samples like a simple linear model, α should have no impact on the model's accuracy. At the same time, such recovery is not always ensured in practice, and therefore a larger α increases the weight of perturbation and decreases that of original sample, thus hurting testing accuracy in some extreme scenarios such as $\alpha = 0.9$.

> **[RQ3] Take-away:** CIP is effective against various state-of-the-art MI attacks in reducing attack accuracy, precision, recall and F-1 Score, while maintaining the testing accuracy. We commonly use $\alpha = 0.9$ in practice for strong defense.



(a) CIFAR-100 (b) CIFAR-AUG (c) CH-MNIST (d) Purchase-50

Figure 8. [RQ3]: State-of-the-art MI attack accuracy against CIP on different datasets.

Table VI
[RQ4-Adaptive-Optimization-1] Attack accuracy (internal-passive/external) against different datasets with probing the target model to optimize t.

Dataset	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
CIFAR-100	0.950/0.948	0.901/0.892	0.769/0.746	0.698/0.649	0.642/0.606
CIFAR-AUG	0.702/0.681	0.669/0.662	0.625/0.618	0.603/0.586	0.578/0.564
CH-MNIST	0.653/0.658	0.639/0.631	0.622/0.617	0.608/0.596	0.570/0.573
Purchase-50	0.624/0.614	0.609/0.597	0.556/0.545	0.539/0.536	0.541/0.533

D. RQ4: Adaptive Adversaries

In this research question, we evaluate the effectiveness of CIP in defending against six different adaptive adversaries trying to guess client's perturbation. We classify these six adversaries into two categories: optimization-based and knowledge-based. The former (Optimization-1, 2) allows the adversaries to probe the target model to optimize a t or optimize the target model directly in a malicious way. The latter (Knowledge-1, 2, 3, 4) assumes adversaries have prior knowledge of CIP's mechanism or training data to launch a one-time attack. We choose Pb-Bayes as the external one since it is the strongest as we can seen in previous RQs. We set the number of clients as one as the most vulnerable target model.

1) Optimizaiton-based Attacks: We describe two adaptive attacks below, which are based on optimization.

[Optimization-1] Passive Observe + Probe + t Optimization. The first adversary passively observes the model, probes the target model to obtain a shadow dataset, and then optimizes a perturbation to maximize the target model's accuracy on the data set. Such an adversary could be either internal or external. Specifically, the external adversary queries the final target model with 2,000,000 times for CIFAR-100/CIFAR-AUG and 50,000 times for CH-MNIST corresponding to the size of attack datasets. Similarly, the internal adversary queries the local model from the last fifth rounds with the same number of times as the external.

Table VI shows attack accuracy as the α value increases. First, the attack accuracy decreases as the α increases. Second, internal adversary achieve higher attack accuracy, 0.02 on average, compared to external adversary due to the accessing to internal training process. Third, for each α value, we do observe that this adaptive attack improves the attack accuracy a little bit, e.g., 0.01 to 0.08. However, when α is large enough, e.g., 0.9 in our deployment, the overall attack accuracy is still very low, i.e., being close to random guessing for all three datasets.

Table VII
[RQ4-Adaptive-Optimization-2] Internal active attack accuracy on different datasets.

Dataset	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
CIFAR-100	0.758	0.672	0.608	584	0.547
CIFAR-AUG	0.602	0.565	0.533	0.531	0.519
CH-MNIST	0.540	0.535	0.521	0.519	0.505
Purchase-50	0.522	0.520	0.515	0.516	0.511

Table VIII
[RQ4-Adaptive-Knowledge-1] Attack accuracy on different datasets with public seed, α and shadow t

Dataset	SSIM(client's seed, adversary's seed)				
	0.1	0.3	0.5	0.7	1.0
CIFAR-100	0.575	0.586	0.607	0.618	0.624
CIFAR-AUG	0.542	0.551	0.550	0.562	0.569
CH-MNIST	0.532	0.534	0.549	0.566	0.571
Purchase-50	0.518	0.521	0.525	0.534	0.538

[Optimization-2] Active Alteration + Optimization. The second adversary actively alters the target model uploaded to the server. Specifically, the adversary trains the model in the direction of reducing the loss value on a target dataset and sends the altered model back to the client. Then, in the next round, the adversary queries the updated model with the target dataset and classifies those with a larger loss as members. This attack is possible because CIP increases the loss on original training data. The adversary starts attack from the last fifth rounds.

Table VII shows the attack accuracy as the α value increases. The results are close to random guessing with $\alpha = 0.5$. The reason is we set the a small hyperparameter λ to limit the loss increase on original training data, the loss increase is not as significant as to divide members and non-members.

> **[RQ4] Take-away-optimization:** It does not give an adversary advantages for MI attacks for either actively optimizing the target model (obtained from a victim client) on target samples or active probing and optimizing t based on probing results.

2) Knowledge-based Attacks: We describe four adaptive attacks below, which are based on additional knowledge of either the defense or the model.

Table IX
[RQ4-ADAPTIVE-KNOWLEDGE-2] ATTACK ACCURACY ON DIFFERENT DATASETS WITH SHADOW t AND PARTIAL TRAINING DATA

Dataset	% of known training samples			
	20%	40%	60%	80%
CIFAR-100	0.583	0.584	0.572	0.575
CIFAR-AUG	0.533	0.531	0.536	0.535
CH-MNIST	0.532	0.525	0.537	0.539
Purchase-50	0.528	0.519	0.517	0.524

Table X
[RQ4-ADAPTIVE-KNOWLEDGE-4] INVERSE MEMBERSHIP INFERENCE ATTACK ACCURACY ON DIFFERENT DATASET WITH DIFFERENT α

Dataset	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	$\alpha = 0.7$	$\alpha = 0.9$
CIFAR-100	0.159	0.328	0.442	0.483	0.489
CIFAR-AUG	0.328	0.394	0.490	0.494	0.498
CH-MNIST	0.414	0.451	0.474	0.491	0.495
Purchase-50	0.387	0.447	0.482	0.485	0.491

[Knowledge-1] Public Seed + α + Shadow t. This adversary knows the initial seed and α and then generates an adaptive t' from shadow training data.That is, the adversary starts from a public seed, which is the same as CIP adopts, optimizes a t' from random shadow training dataset.

Table VIII shows the attack accuracy of an external attack given α is 0.7 when we change the structural similarity index measure (SSIM) [5] to calculate the similarity between the attacker's and the initial seeds. There are three things worth noting here. First, when the SSIM between two seeds increases, the attack accuracy increases as well. That means, when the attacker knows the exact seed or is getting close to the exact seed, the adaptive attack is more effective. Second, even if the attacker knows the exact secret perturbation seed, the adaptive attack accuracy is still much lower than the one of the state-of-the-art MI attack. The reason is that the attacker does not know the training data and the generated t and t' still differ a lot with the same seed.

[Knowledge-2] Shadow t with Partial Training Data. This adversary knows part of the training data and generates a perturbation t' and a model with parameters θ from that part of training data and random initial seed.

Table IX shows the accuracy of external adaptive attack upon the training data that is unknown to the adversary. Interestingly, the accuracy of the attack does not change much as the percentage of training data that is known to the adversary. The reason is that the training data does not provide more information than what the adversary obtains from the target model [29]. More importantly, knowing part of training data does not give insights on other parts of training data that is unknown.

[Knowledge-3] A substitute t' from a malicious FL client under an i.i.d. setting. This adversary is a malicious client and tries to use its own perturbation t' to launch a membership inference attack against the target data that is supposed to use perturbation t. We use the i.i.d. setting on the CIFAR-100 dataset as described in Section V-B (because t will be quite different for non-i.i.d. settings) for the experiment.

Here are our results. The testing accuracy on target data with substitute t' is 0.695, which is similar to real t, 0.666, and demonstrate the effectiveness of t' and the trained model. Then, the attack accuracy is 0.535 even though the malicious client can achieve a good testing accuracy with t'. It is because t' is not trained with the original training data, which makes outputs of members and non-members with t' non-separable. The reason can also be reflected in the training accuracies, which are 0.991 for t and 0.722 for t'. That is, the accuracy gap between training and testing accuracy—which leads to membership inference attacks—is only 0.027 for t' as opposed to 0.325 for t. Note that the SSIM between t and t' is also large, which is 0.665 under an i.i.d. setting.

[Knowledge-4] Inverse membership inference attack. This adversary learns about the mechanism of CIP, knowing that we intentionally increase the loss of original training data, and designs an adaptive attack that classifies data with abnormally high loss as the member. This attack can be regarded as the inverse version of external adversary Ob-MALT, which classify the data with high loss as non-member and low loss as member.

Table X shows the attack accuracy, which is close to randomly guessing. That is, this adaptive attack is clearly not effective against CIP. We think the reason is that we choose a small λ_m, i.e., $1e-12$, to adjust the weight of increasing loss, which will not lead to a abnormally high loss value. Instead, CIP will just make the model's outputs of original samples look like other non-members. It is worth noting that the attack accuracy increases while the α increases. The reason is that the larger α introduces more perturbation, which leads to a higher loss on original training data than a smaller α while training the target model.

[RQ4] Take-away-knowledge: It does not give the adversary any attack advantages over privacy if they know some parameters of the defense.

E. RQ5: Overhead

In this research question, we evaluate CIP in terms of the number of model parameters and training epochs needed to converge. Table XI shows both numbers of federate learning with the number of client as five. First, the number of parameters of CIP is a little higher (+0.87%) than the one of conventional FL models without defense. Note that although CIP adopts a dual channel, the backbone is shared with no increase on the number of parameters. That is, the parameter number increase comes from the concatenated dense layers as shown in Figure 3.

Second, the number of epochs of CIP is half of the one of conventional FL models. The reason is that CIP has two-step optimizations with the perturbation and the model. We believe that this two-step optimization will help the convergence.

[RQ5] Take-away: CIP introduces minimum overhead (i.e., 0.87%) in terms of model size and actually reduces the number of epochs to converge because of the two-step optimization.

Table XI
[RQ5-OVERHEAD]THE NUMBER OF PARAMETERS AND THE NUMBER OF EPOCHS TO CONVERGE OF CIP AND CONVENTIONAL MODELS WITHOUT DEFENSE (# OF CLIENT EQUALS TO FIVE).

	Model type	ResNet	DenseNet	VGG	Avgerage Δ
Params	No defense	23,792,612	14,765,988	7,140,004	+ 0.87%
	CIP	23,997,412	14,817,188	7,242,404	
Epochs	No defense	300	600	600	- 50.0%
	CIP	150	300	300	

VI. Related Work

Machine learning is vulnerable to different privacy attacks including model inversion [16], [17], membership inference [45], property inference [7], [18], as well as model and hyperparameter stealing [49], [52]. Our work studies the defense against membership inference (MI) attacks. We describe related work on MI defenses and attacks.

MI Attacks. We discuss MI attacks in this part. First, existing MI attacks can be roughly categorized as either *output* or *parameter*-based depending on their required information. An output-based attack [23], [40], [41], [45], [55], [55], [57] uses the target model's output, e.g., from the last Softmax layer of a neural network, to predict membership; as a comparison, a parameter-based attack [29], [38] takes the target model's parameters as input so that it can compute not only the target model's output but also its gradient for a given data sample. Both categories of attacks can be applied in the so-called *whitebox* setting [29], [38], which assumes that the adversary has full, whitebox access to the target model either published by the owner for easy distribution or leaked in a cyber attack. Note that MI attacks were also explored in other learning settings, such as generative adversarial networks [21] and genome-based study [22].

Defenses against MI attacks. Existing defenses fall into two categories, training perturbation and output perturbation: (i) Training perturbation defenses alter the training process to make the target model secure against MI attacks. For instance, regularization was proposed to regularize the training process of the target model to reduce model overfitting and subsequently mitigate MI attacks [10], [12], [30], [37], [45], [57].In particular, Shokri et al. [45] proposed to use the standard ℓ_2-norm regularization term in training. Adversarial regularization [37] models an MI attack's success as a regularization term and adds it to the loss function when learning the target model. Aside from regularization, differential privacy [14], [24], [36], [44], [53] bounds the attacker's capability in inferring whether a data sample was used to train the target model. (ii) Output perturbation defenses change the target model's output to become less informative about the input's membership status. For instance, MemGuard [26] adds a carefully crafted perturbation to the target model's output and turns it into an adversarial example for the attacker's classifier that is used to infer membership.

Privacy-preserving FL with Secure Aggregation. Recent researches have studied the combination of FL and secure aggregation [8], [9], [19], [46] sometimes in combination with homomorphic encryption [20], [60] to ensure privacy. Bonawitz et al. [9] enables clients to encrypt their updates so that the central parameter server can only access the sum of the updates. Mohassel et al. [35] allows clients to distribute their training data to two non-colluding servers. Then, the servers train a global model on the encrypted data using multi-party computation. However, secure aggregation only guarantees that the aggregation process does not leak privacy. A client or server can still perform membership inference attacks to the aggregated global model. Moreover, homomorphic encryption and secure aggregation are often of low efficiency [9] due to heavy-weight encryption, while perturbation-based defenses are relatively lightweight.

Federated learning and MI attacks. Federated learning [28], [31], [50], [51], [56], [59] is a collaborative learning that allows multiple clients to train a model without explicitly sharing data with each other. Specifically, a centralized server coordinates multiple clients to train and exchange local models for a global one. However, FL is vulnerable to multiple privacy attacks [15], [34], [39], [62]. Nasr et al. [38] show that malicious clients or servers can launch MI attacks, both active and passive ones, against federated learning models to cause privacy violations. Recent work [43], [47], [58], [59] propose personalized federated learning to let clients use a different local model or learning strategy to address the accuracy decrease caused by data heterogeneity.

VII. Conclusion

In conclusion, we propose a defense, called CIP, against membership inference attacks and use a different perspective, letting the clients in an FL system shift the local input distribution in a utility-preserving way using a client-level input perturbation. We demonstrate both theoretically and empirically that CIP can defend against both state-of-the-art and adaptive attackers (e.g., those who follows the same procedure to generate an adaptive access credential to infer membership). Furthermore, CIP maintains the benefits brought by a collaborative learning like FL by introducing new classes to other clients in a non-i.i.d. setting and improving local model's accuracy in an i.i.d. setting.

Acknowledgment

We would like to thank anonymous reviewers for their helpful comments and feedback. This work was supported in part by Johns Hopkins University Institute for Assured Autonomy (IAA) with grants 80052272 and 80052273, and National Science Foundation (NSF) under grants CNS-21-31859, CNS-21-12562, CNS-19-37786, and CNS-18-54000. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF or JHU-IAA.

References

[1] [github] adversarial regularization. https://github.com/NNToan-apcs/python-DP-DL.

[2] [github] handcrafted-dp. https://github.com/ftramer/Handcrafted-DP.

[3] [github] relaxloss. https://github.com/DingfanChen/RelaxLoss.

[4] [github] tensorflow privacy. https://github.com/tensorflow/privacy.

[5] [wikipedia]structural similarity. https://en.wikipedia.org/wiki/Structural_similarity.

[6] 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.

[7] Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3), 2015.

[8] James Henry Bell, Kallista A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, 2020.

[9] 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.

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

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

[12] Christopher A. Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1964–1974. PMLR, 18–24 Jul 2021.

[13] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[14] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[15] Liam H. Fowl, Jonas Geiping, Wojciech Czaja, Micah Goldblum, and Tom Goldstein. Robbing the fed: Directly obtaining private data in federated learning with modified models. In *International Conference on Learning Representations*, September 2021.

[16] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, 2015.

[17] Matthew Fredrikson, Eric Lantz, Somesh Jha, Simon Lin, David Page, and Thomas Ristenpart. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, 2014.

[18] Karan Ganju, Qi Wang, Wei Yang, Carl A. Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *CCS*, 2018.

[19] Xiaojie Guo, Zheli Liu, Jin Li, Jiqiang Gao, Boyu Hou, Changyu Dong, and Thar Baker. Verifl: Communication-efficient and fast verifiable aggregation for federated learning. *IEEE Transactions on Information Forensics and Security*, 16:1736–1751, 2021.

[20] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677*, 2017.

[21] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019(1):133–152, 2019.

[22] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8), 2008.

[23] Bo Hui, Yuchen Yang, Haolin Yuan, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Practical blind membership inference attack via differential comparisons. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'21)*, February 2021.

[24] Roger Iyengar, Joseph P Near, Dawn Song, Om Thakkar, Abhradeep Thakurta, and Lun Wang. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 299–316. IEEE, 2019.

[25] Bargav Jayaraman and David Evans. Evaluating differentially private machine learning in practice. In *28th USENIX Security Symposium (USENIX Security 19)*, pages 1895–1912, Santa Clara, CA, August 2019. USENIX Association.

[26] 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.

[27] Jakob Nikolas Kather, Cleo-Aron Weis, Francesco Bianconi, Susanne M Melchers, Lothar R Schad, Timo Gaiser, Alexander Marx, and Frank Gerrit Z”ollner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.

[28] Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

[29] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated white-box membership inference. In *29th USENIX Security Symposium (USENIX Security 20)*, pages 1605–1622, 2020.

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

[31] 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.

[32] Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data, 2020.

[33] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[34] 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, 2019.

[35] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 19–38, 2017.

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

[37] 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.

[38] 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.

[39] Dario Pasquini, Danilo Francati, and Giuseppe Ateniese. Eluding secure aggregation in federated learning via model inconsistency. In Heng Yin, Angelos Stavrou, Cas Cremers, and Elaine Shi, editors, *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 2429–2443. ACM, 2022.

[40] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567, 2019.

[41] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference

attacks and defenses on machine learning models. In *Network and Distributed Systems Security Symposium 2019*. Internet Society, 2019.

[42] F. Sattler, S. Wiedemann, K. R. Müller, and W. Samek. Robust and communication-efficient federated learning from non-i.i.d. data. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3400–3413, 2020.

[43] Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *Proceedings of the 38th International Conference on Machine Learning (ICML), PMLR 139*, 2021.

[44] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[45] 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.

[46] Jinhyun So, Basak Guler, and A. Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. Cryptology ePrint Archive, Paper 2020/167, 2020. https://eprint.iacr.org/2020/167.

[47] Canh T Dinh, Nguyen Tran, and Tuan Dung Nguyen. Personalized federated learning with moreau envelopes. *Advances in Neural Information Processing Systems*, 33, 2020.

[48] Florian Tramer and Dan Boneh. Differentially private learning needs better features (or much more data). In *International Conference on Learning Representations*, 2021.

[49] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX Security Symposium*, pages 601–618, 2016.

[50] Nguyen H. Tran, Wei Bao, Albert Zomaya, Minh N. H. Nguyen, and Choong Seon Hong. Federated learning over wireless networks: Optimization model design and analysis. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 1387–1395, 2019.

[51] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security*, pages 1–11, 2019.

[52] Binghui Wang and Neil Zhenqiang Gong. Stealing hyperparameters in machine learning. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 36–52. IEEE, 2018.

[53] Di Wang and Jinhui Xu. Differentially private empirical risk minimization with smooth non-convex loss functions: A non-stationary view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1182–1189, 2019.

[54] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li. Optimizing federated learning on non-iid data with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1698–1707. IEEE, 2020.

[55] Lan Zhang Xiaoyong Yuan. Membership inference attacks and defenses in neural network pruning. In *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA, August 2022. USENIX Association.

[56] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

[57] 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.

[58] Felix Yu, Ankit Singh Rawat, Aditya Menon, and Sanjiv Kumar. Federated learning with only positive labels. In *International Conference on Machine Learning*, pages 10946–10956. PMLR, 2020.

[59] Haolin Yuan, Bo Hui, Yuchen Yang, Philippe Burlina, Neil Zhenqiang Gong, and Yinzhi Cao. Addressing heterogeneity in federated learning via distributional transformation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2022.

[60] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.

[61] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data, 2018.

[62] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.