MIST: Defending Against Membership Inference Attacks Through Membership-Invariant Subspace Training

Jiacheng Li, Ninghui Li, and Bruno Ribeiro, *Purdue University*

https://www.usenix.org/conference/usenixsecurity24/presentation/li-jiacheng

This paper is included in the Proceedings of the 33rd USENIX Security Symposium.

August 14–16, 2024 • Philadelphia, PA, USA

978-1-939133-44-1

Open access to the Proceedings of the 33rd USENIX Security Symposium is sponsored by USENIX.

MIST: Defending Against Membership Inference Attacks Through Membership-Invariant Subspace Training

Jiacheng Li
Purdue University

Ninghui Li
Purdue University

Bruno Ribeiro
Purdue University

Abstract

In Member Inference (MI) attacks, the adversary tries to determine whether an instance is used to train a machine learning (ML) model. MI attacks are a major privacy concern when using private data to train ML models. Most MI attacks in the literature take advantage of the fact that ML models are trained to fit the training data well, and thus have very low loss on training instances. Most defenses against MI attacks therefore try to prevent the model from heavily overfitting the training data. Doing so, however, generally results in lower accuracy.

We observe that training instances have different degrees of vulnerability to MI attacks. Most instances will have low loss even when not included in training. For these instances, they are less vulnerable to MI attacks and the model can fit them well without concerns. An effective defense only needs to (possibly implicitly) identify instances that are vulnerable to MI attacks and avoids overfitting them. A major challenge is how to achieve such an effect in an efficient training process.

Leveraging two distinct recent advancements in representation learning: counterfactually-invariant representations and subspace learning methods, we introduce a novel Membership-Invariant Subspace Training (MIST) method to defend against MI attacks. MIST avoids overfitting the vulnerable instances without significant impact on other instances. We have conducted extensive experimental studies, comparing MIST with various other state-of-the-art (SOTA) MI defenses against several SOTA MI attacks. We find that MIST outperforms other defenses while resulting in minimal reduction in testing accuracy.

1 Introduction

Neural network-based machine learning models are now prevalent in our daily lives, from voice assistants [27], to image generation [35] and chatbots (e.g., ChatGPT-4 [32]). These large neural networks are powerful but also raise serious privacy concerns, such as whether personal data used to train these models are leaked by these models. One way to understand and address this privacy concern is to study membership inference (MI) attacks and defenses [31, 39]. In MI attacks, an adversary seeks to infer if a given instance was part of the training data. If the optimal MI attack is merely as good as random guessing, then there is no privacy leakage in the sense of information theory [24].

Most MI attacks in the literature are loss-based blackbox attacks, which exploit the fact that ML models tend to overfit the training data and have higher confidence on training instances. Several defenses were proposed to alleviate this effect by making the label of training instance "softer" so that the trained model would behave less confidently on training instances. Shokri et al. [39] propose to use temperature in softmax function to increase the entropy of the predictions. Li et al. [22] suggest to train with mix-up instances (linear combinations of two original training instances). Chen and Pattabiraman [7] propose two methods to reduce confidence on training instances: (1) change labels of training instances to soft labels, and (2) use a regularizer to penalize low entropy predictions in training. Forcing labels of training instances to be "softer", however, will inevitably incur some penalty of lower testing accuracy.

We observe that these defenses fail to take advantage of the differences among training instances. Some instances have low loss whether they are included in the training set or not, while other instances will incur high loss when they are not included in training [43]. Hereafter, we will refer to these specific cases as "distinctive". Distinctive instances are naturally more vulnerable to MI attacks. While this phenomenon has been exploited in MI attacks that set different loss thresholds for different instances [3, 43, 44], existing defenses fail to take advantage of this effect. By forcing all training instances to be "softer", these defenses incur accuracy loss that can be avoided.

In this paper, we propose a novel training approach that naturally differentiates among instances with different degree of distinctiveness. It can be viewed as making the label "softer" **only** for the distinctive instances, avoiding unnec-

essary penalty in testing accuracy. Our approach uses ideas similar to subspace learning [18], and uses a novel regularizer to train models to be counterfactually invariant to the membership of each instance in the training data D. We thus call our approach Membership-Invariant Subspace Training (MIST).

The algorithm behind MIST is detailed in Section 3.2 and here we give a brief description. MIST divides the training dataset into multiple subsets. While one epoch of standard SGD can be viewed as sequentially updating the model on these subsets one by one, each epoch in MIST has three steps. In the first step, we train on all subsets simultaneously, resulting in a subspace of multiple submodels (one for each subset). In the second step, for each such submodel M_i, we perform gradient updates to reduce the differences on instances used in computing M_i between predictions given by M_i and the average predictions given by other submodels. In the third and last step, we average the resulting submodels. The effect is that, if an instance used to train M_i is not distinctive, M_i's output on the instance would be similar to the average, resulting in little impact to the model. On the other hand, if an instance is distinctive, the gradient update step would pull the model towards the average of other submodels, which approximates the counterfactual scenario where this instance is not used in training.

Contributions.

1. We propose the **Membership-Invariant Subspace Training** (MIST) defense designed to specifically defend against blackbox membership inference attacks on the most vulnerable instances in the training data (which also helps defending against overall blackbox MI attacks). MIST uses a novel two-phase subspace learning procedure that trains models regularized to be counterfactually invariant to the membership of each instance in the training data D.

2. We compare our defenses against baselines and provide extensive experiments showing that our defense can significantly reduce the effectiveness of existing membership inference attacks. To the best of our knowledge, MIST is the first effective defense against LIRA and CANARY-style blackbox MI attacks that focuses on defending the most vulnerable instances in the training data, while improving the overall MI attack robustness.

Roadmap. The rest of this paper is organized as follows. In section 2, we introduce some machine learning background and summarize existing membership inference attacks and defenses in centralized setting. In section 3, we propose our new defense, the **Membership-Invariant Subspace Training** (MIST) defense. In section 4, we present our extensive experimental evaluations and compare our proposed defense with existing defenses. In section 5, we summarize the findings presented in this paper. In section 6, we discuss limitations of the proposed MIST defense and potential future directions.

2 Preliminary

In this section, we describe the MI attacks that we use in our experimental evaluation and defense mechanisms that we compare against. We focus on blackbox attacks, where the adversary queries the target model and uses the model prediction to infer membership. Other works in this area are discussed in Section A in the Appendix.

2.1 MI attacks in blackbox setting

We classify these attacks based on what they use to query the target model F^T.

2.1.1 Using the Target Instance

In these attacks, one uses the target instance x to query the model.

LOSS (Using loss with a global threshold). Yeom et al. [47] introduced an attack that predicts x is a member when the target model's loss on x is below the average training loss.

Class-NN (Training class-specific Neural Networks for MI). Shokri's attack [39] trains multiple neural network-based membership classifiers, one for each class. Training data are obtained using the shadow-mode technique, which is widely used in later attacks.

Using the **shadow-model** technique, one assumes that the adversary knows a dataset D^A, which contains the target instance and is from the same distribution as the dataset used to train F^T. The adversary creates k subsets $D_1, D_2, \ldots, D_k$ from D^A, and uses the same process used for training F^T to train k models, one from each D_i. These are called shadow models. For each instance x, some shadow models were trained using x, and others were trained without. The predictions of these models on instances in D^A provide training data for membership classifiers.

Modified Entropy. Song et al. [40] proposed to use a class-specific threshold on a modified entropy measure based on the model prediction to determine membership. This can be viewed as a simplified version of the attack in [39], and is very similar to using a class-specific loss threshold to determine membership.

LIRA. Carlini et al. [3] proposed an instance-specific threshold attack. For each instance, from the shadow models, one obtains a distribution for losses from models trained with x, and another distribution from models trained without x. A threshold can be chosen using likelihood ratio. Carlini et al. suggest to choose the threshold that optimizes for attack effectiveness at a very low false positive rate. We refer this attack as LIRA attack in later sections.

2.1.2 Using Perturbations of the Target Instance

Random perturbations. Jayaraman et al. [19] proposes an attack that generates multiple perturbed instances by adding Gaussian noise to x, and then query F^T using these perturbed instances and count how many times the prediction loss of them is higher than that of x. The instance x is predicted to be a member if the count is beyond a threshold. We refer this attack as the random-perturbation attack in later sections.

CANARY. The attack proposed in Wen et al. [44] also uses shadow models. For each target instance x, the shadow models are partitioned into two sets: those trained with x, and those trained without. One then computes a set of canaries, each generated via gradient descent starting from a slightly perturbed version of x, searching for an x' such that the difference between the average losses of x' from models in the two sets are as large as possible. One then use these canaries to query the target model and use the loss to predict membership. We call this attack the CANARY attack in later sections.

Adversarial perturbation label only. Choquette-Choo et al. [8] proposed two attacks for the situations where only the predicted label is provided and identified the adversarial perturbation attack as more effective. In this attack, one applies the adversarial example generation technique from [6] to generate x' that is close to x while have a different predicted label by F^T. The attack predicts membership if $||x'-x||_2$ is high.

2.2 Defenses against MI Attacks

Adversarial Regularization (Adv-reg). Nasr et al. [30] proposed a defense that uses similar ideas as GAN. The classifier is trained in conjunction with an MI attacker model. The optimization objective of the target classifier is to reduce the prediction loss while minimizing the MI attack accuracy.

Mixup+MMD. Li et al. [22] proposed a defense that combines mixup data augmentation and MMD (Maximum Mean Discrepancy [12,14]) based regularization. Instead of training with original instances, mixup data augmentation uses linear combinations of two original instances to train the model. It was shown in [48] that this can improve target model's generalization. Li et al. [22] found that they also help to defend against MI attacks. Li et al. [22] also proposes to add a regularizer that is the MMD between the loss distribution of members and the loss distribution of a validation set not used in training. This helps make the loss distribution of members to be more similar to the loss distribution on non-members.

Distillation for membership privacy (DMP). Distillation uses labels generated by a teacher model to train a student model. It was proposed in [17] for the purpose of model compression. Shejwalkar et al. [38] proposed to use distillation to defend against MI attacks. One first trains a teacher model using the private training set, and then trains a student model using another unlabeled dataset from the same distribution as the private set. The intuition is that since the student model is not directly optimized over the private set, their membership may be protected. the authors also suggested to train a GAN using the private training set and draw samples from the trained GAN to train the student model, when no auxiliary unlabeled data is available.

SELENA. Tang et al. [41] proposed a framework named SELENA. One first generates multiple (overlapping) subsets from the training data, then trains one model from each subset. One then generates a new label for each training instance, using the average of predictions generated by models trained without using that instance. Finally, one trains a model using the training dataset using these new labels.

HAMP. Chen et al. [7] proposed a defense combining several ideas. First, labels for training instances are made smoother, by changing 1 to λ and each 0 to $\frac{1-\lambda}{k-1}$, where λ is a hyperparameter and k is the number of classes. Second, an entropy based regularizer is added in the optimization objective. Third, the model does not directly return its output on a queried instance x. Instead, one randomly generates another instance, reshuffle the prediction vector of the randomly generated instance based on the order of the probabilities of x and returns the reshuffled prediction vector. In essence, this last defense means returning only the order the classes in the prediction vector but not the actual values.

Mem-guard. Jia et al. [20] proposed the **Mem-guard** defense. In this defense, one trains an MI attack model in addition to the target classifier. When the target classifier is queried with an instance, the resulting prediction vector is not directly returned. Instead, one tries to find a perturbed version of the vector such that the perturbation is minimal and does not change the predicted label, and the MI attack model output (0.5,0.5) as its prediction vector.

Differential Privacy. Differential privacy (DP) [9–11] is a widely used privacy-preserving technique. DP based defense techniques, such as DP-SGD [1], add noise to the training process. This provides a theoretical upper-bound on the effectiveness of any MI attack against any instance. Unfortunately, achieving a meaningful theoretical guarantee (e.g., with a resulting $\varepsilon < 5$) requires the usage of very large noises. However, model trainer could use much smaller noises in DP-SGD. While doing this fails to provide a meaningful theoretical guarantee (the ε value would be too large), this can nonetheless provide empirical defense against MI attacks. In [7,22,41], extensive experiments have shown that several other defenses can provide better empirical privacy-utility tradeoff than DP-SGD.

3 Proposed Defense: Membership-Invariant Subspace Training

Almost all MI attacks take advantage of the fact that training aims to reduce the loss on member instances to zero. Many defenses thus try to deviate from this optimization objective. Unfortunately, this usually results in significant testing accuracy drop. Our key insight is that we can take advantage of the fact that relatively few number of instances are truly vulnerable to MI attacks. For example, under the LIRA attack at 0.001 False Positive Rate, less than 5% of members are identified. That is, for most instances, whether it is included in the training set or not will not drastically affect their loss. Intuitively, we only need to change the optimization objective for the vulnerable instances to avoid overfitting them. This avoids suffering from unnecessary testing accuracy reduction. The challenge is how to achieve this effect in the training process without incurring a very high runtime overhead.

3.1 Threat Model

We consider a blackbox attacker, who can query the target model and get its prediction, but does not use the internal state of the target model, such as model parameters. We also assume that the adversary knows the model architecture, the training recipe, and the distribution of training data, and is capable of training shadow models. This type of adversary is able to get the member prediction distribution and nonmember prediction distribution for a particular instance of interest by training multiple shadow models and then perform the LIRA attack or the CANARY attack, which are the strongest blackbox MI attacks in the literature.

3.2 Our Proposal

To effectively protect the membership of vulnerable instances without incurring accuracy reduction for less vulnerable instances, our proposed defense capitalizes on the combined strengths of two distinct recent advancements in representation learning: Counterfactually-invariant representations [21, 29, 34, 42] and subspace learning methods [18, 45].

A *counterfactual dataset* is a set of data samples whose distribution has been changed by an intervention. Inspired by differential privacy, given a training dataset D, we define $n = |D|$ environments $D_1, D_2, \cdots, D_n$ such that $D_M = D \setminus \{(x_M, y_M)\}$ for $1 \leq M \leq n$. That is, each D_M has the same distribution as the training data $p(x,y)$, except for the intervention that the M-th training example (x_M, y_M) has probability zero.

Let $F(\cdot;\theta)$ denote a classifier with parameters θ such that $F(x;\theta)$ gives a vector of predicted probabilities for the classes. Let $T_F(\cdot)$ denote the training algorithm such that $\theta_D = T_F(D)$ is the parameters trained using dataset D. Ideally, we want T_F to be counterfactually-invariant to the interventions, i.e., $\forall M \in \{1,\ldots,n\}$, $F(x;T_F(D))$ and $F(x;T_F(D_M)$ have the same classification behavior. Note that this objective is related to yet different from the objective of ε-differential privacy (DP) [9–11]. This objective is less strict than ε-DP when $\varepsilon = 0$, which requires $T_F(D)$ and $T_F(D_M)$ to have exactly the same distribution. Here we expect that $\theta_D = T_F(D)$ and $\theta_{D_M} = T_F(D_M)$ to have the same classification behavior, even though the parameters may be different.

We will denote our type of counterfactual invariance as *membership-invariance*. Employing membership-invariant classifiers as a defense against MI attacks appears both promising and "straightforward" at first glance. Closer inspection, however, reveals that the task of training a membership-invariant classifier (that generalizes well in test) to be immensely challenging, especially over large training datasets. Standard invariant representation training methods include invariant risk minimization [2], adversarial training, such as Ganin et al. [13], that uses a minimax game to ensure representations cannot predict the environments, and the Hilbert-Schmidt conditional independence criterion [33] used by Quinzan et al. [34] for counterfactually-invariant predictors). Using these methods for training membership-invariant classifiers would require constructing n leave-one-out datasets from D, each a new environment. This is infeasible for large datasets commonly used in state-of-the-art models.

The main technical challenge we need to overcome is thus: *How can we efficiently learn membership-invariant classifiers to defend against MI for all vulnerable instances?*

3.2.1 Cross-difference loss

We start by defining a regularization that we call *cross-difference loss* as follows:

$$\begin{aligned}&\mathrm{xdiff}(\theta_D, \theta_{D\setminus\{(x_M,y_M)\}}) \\ &\quad = \sup_{(x,y)\in\Omega} \|F(x;\theta_D)_y - F(x;\theta_{D\setminus\{(x_M,y_M)\}})_y\|_1,\end{aligned} \tag{1}$$

where Ω is the support of $p(x,y)$ and $\|\cdot\|_1$ is the L_1 norm. Equation (1) pushes the classifier $F(\cdot;\theta_D)$ trained with (x_M, y_M) to have the same output as the classifier $F(\cdot;\theta_{D\setminus\{(x_M,y_M)\}})$ trained without (x_M, y_M). Then, any classifier F that satisfies

$$\sum_{M\in\{1,\ldots,n\}} \mathrm{xdiff}(\theta_D, \theta_{D\setminus\{(x_M,y_M)\}}) = 0, \tag{2}$$

is membership-invariant. This is easy to verify since the condition in Equation (2) implies $F(x;\theta_{D\setminus\{(x_M,y_M)\}}) = F(x;\theta_D) = F(x;\theta_{D\setminus\{(x_{M'},y_{M'})\}})$, for all $M, M' \in \{1,\ldots,n\}$, that is, the classifier output is invariant to the environment in which it was trained.

3.2.2 Membership-invariance via Two-Phase Subspace Learning

The challenge of training a classifier that satisfies Equation (2) is the computational cost. To support the fast optimization

of F using the cross-difference loss in Equation (2), we will make use of the concept of subspace learning [45]. Subspace learning optimizes neuron weights in a subspace that contains diverse solutions that can be ensembled, approaching the ensemble performance of *independently* trained neural networks without the associated training cost. Originally designed by Wortsman et al. [45] to boost accuracy, calibration, and robustness to label noise, we will repurpose subspace learning for learning membership-invariant classifiers. In particular, our approach (MIST) leverages a version of subspace learning based on (virtual) federated learning: DART (Diversify-Aggregate-Repeat Training) [18].

Our core idea adapts DART and simultaneously trains multiple diverse models that are regularized with a computationally efficient approximation of the regularization implied by Equation (2). These diverse models within the subspace, in effect, regularize each other. They impose penalties on each other if discrepancies arise in their outputs when trained on their respective subsets of the training data. We denote the resulting method MIST and detail it in Algorithm 1; we also summarize the method below.

MIST. Training in MIST consists of E global epochs. We use $\overline{\theta}_e$ to denote (parameters of) the aggregated (i.e., global) model at epoch $e = 0,\ldots,E$. $\overline{\theta}_0$ is initialized with random parameters. The e-th epoch (where $1 \leq e \leq E$) consists of the following steps.

Initialization: Data Partitioning. The training data D is split into C disjoint subsets D^c, $c = 1,\ldots,C$. We use $\overline{\theta}_{e-1}$ to initialize C (local) models.

Phase 1: Exploring diverse models (local training). For each $c \in \{1,\ldots,C\}$, we use D^c to update the c-th local model, by performing gradient updates $T_1 \geq 1$ times, where T_1 is a hyperparamenter. We use θ_e^c to denote (parameters of) the c-th local model after the gradient updates.

Phase 2: Finding models with small cross difference loss. In Phase 2, we aim to find models near the diverse models found in Phase 1 that have low cross-difference loss. For $c = 1,\ldots,C$, let $w_0^c = \theta_e^c$, we perform T_2 gradient update steps, using the following approximation of the cross-difference loss (described in Equation (1)) to regularize w_t^c in Algorithm 1:

$$\text{xdiff}_c(w_t^c) = \sum_{(x,y)\in D^c} \left\| F(x;w_t^c)_y - \frac{\sum_{i\neq c} F(x;\theta_e^i)_y}{C-1} \right\|_1 . \quad (3)$$

Minimizing Equation (3) will push our models towards satisfying Equation (2). Note that if an instance $(x,y) \in D^c$ is not vulnerable to MI attacks, the confidence from the c-th local model (trained using (x,y)) $F(x;w_t^c)_y$ will be similar to the average confidence from other local models, and the L_1 norm in Equation (2) is close to 0, resulting in little change to the model parameter. If, however, $(x,y) \in D^c$ is vulnerable to MI attacks, then the L_1 norm in Equation (2) will be large, resulting the c-th local model to be updated to fit less well on (x,y).

In Equation (3) we are not computing the L_1 loss over the entire support Ω of the training distribution $p(x,y)$ as in Equation (1) but, rather, restricting the cross-difference loss to the examples in D^c. This is a reasonable approximation since these examples are the ones used to directly optimize the c-th local model, and we thus focus on protecting their membership.

In Equation (3), we use L_1 norm. In Section 4.8, we compare the effect of using L_1, L_2 norms, and KL divergence in an ablation study.

Model aggregation. The (global) training epoch $e \geq 1$ ends by averaging the parameters of all C local models into a new parameter $\overline{\theta}_e$. In general, as in subspace learning, the averaging can be a convex combination of the model parameters. In our experiments we just use the arithmetic mean $\overline{\theta}_e = (1/C)\sum_{c=1}^{C} \theta_e^c$.

Returning. At the end of our MIST algorithm, we return $\overline{\theta_E}$ as the final trained model parameters.

Algorithm 1 (MIST) Membership-Invariant Subspace Training

Input: C: the number of local models; D: the training dataset; E: the number of epochs. Hyperparameters T_1, T_2, and λ (weight for cross difference loss).

Initialize $\overline{\theta}_0$ ▷ Model initialization
for $e = 1$ **to** E **do**
  Partition D into D^c, where $1 \leq c \leq C$
  for $c = 1$ **to** C **do** ▷ can be performed in parallel
    $\theta_e^c = \text{local\_training}(\overline{\theta}_{e-1}, D^c, T_1)$
  for $c = 1$ **to** C **do** ▷ can be performed in parallel
    $\theta_e^c = \text{xdifference\_update}(c, \theta^1, \ldots, \theta^C, D^c, \lambda, T_2)$
  $\overline{\theta}_e = \frac{\sum_{c=1}^{C} \theta_e^c}{C}$ ▷ Aggregate and average model weights
Output: $\overline{\theta_E}$

Function local_training($\overline{\theta}, D^c, T_1$): ▷ Phase 1
  $w_0^c = \overline{\theta}$
  for $t = 1$ **to** T_1 **do**
    gradient updates to w_t^c to minimize $\frac{1}{|D^c|}\mathcal{L}(w_{t-1}^c, D^c)$
Output: $w_{T_1}^c$

Function xdifference_update(c, $\theta^1,\ldots,\theta^C, D^c, \lambda, T_2$): ▷ Phase 2
  $w_0^c = \theta^c$
  for $t = 1$ **to** T_2 **do**
    gradient updates to w_t^c to minimize $\frac{\lambda}{|D^c|}\text{xdiff}_c(w_{t-1}^c)$
Output: $w_{T_2}^c$

3.2.3 Further improvements

MIST can be augmented with some existing MI defense approaches. For instance, the Mix-up data augmentation MI defense proposed by Li et al. [22] can be easily integrated

into MIST (Algorithm 1). Li et al. [22] shows that Mix-up data augmentation can reduce the generalization gap between train and test data, consequently making black-box MI attacks more difficult. Mix-up training uses linear interpolation of two different training instances to generate a mixed instance and train the classifier with the mixed instance. The generation of mixed instances can be described as follows:

$$\tilde{x} = \beta x_i + (1-\beta)x_j,$$
$$\tilde{y} = \beta y_i + (1-\beta)y_j, \quad (4)$$

where $\beta \sim \text{Beta}(\alpha,\alpha), \alpha \in (0,\infty)$. Here x_i and x_j in Equation (4) are instance feature vectors randomly drawn from the training set; y_i and y_j are one-hot label encodings corresponding to x_i and x_j. The new tuple $(\tilde{x},\tilde{y})$ is used in training.

4 Evaluation

In this section we present our extensive experimental results. We start by describing the details of our experimental setup. Next we discuss the computational cost of considered defenses. Then we dive into the detailed comparison between MIST and other defenses. After this, we present the evaluation of MIST against label-only attacks. In the following subsection, we show how to choose hyperparameters for our MIST defense. Lastly, we present ablation studies.

4.1 Experimental Setup

4.1.1 Datasets

Our evaluation uses the datasets most commonly used in MI attack literature [7, 20, 30, 31, 37, 39, 47].

CIFAR-10 and CIFAR-100. They contain 60,000 color images of size 32 × 32, divided into 50,000 for training and 10,000 for testing. In CIFAR-100, these images are divided into 100 classes, with 600 images for each class. In CIFAR-10, these 100 classes are grouped into 10 more coarse-grained classes; there are thus 6000 images for each class. We use AlexNet, ResNet-18 and DenseNet-BC(100,12) (abbreviated as DenseNet), which are the standard neural networks used in prior work [22, 31]. The model architectures and hyperparameters for training are described in Table 7 in the Appendix.

TEXAS-100. The records in the dataset contain information about inpatient stays in several health care facilities published by the TEXAS Department of State Health Services. We obtained the dataset from the authors of [39]. The dataset contains 67,330 records and 6,170 binary features. The records are clustered into 100 classes, each representing a different type of patient. We use the fully connected neural network from [31] as the target model.

PURCHASE-100. This dataset is based on the "acquire valued shopper" challenge from Kaggle. This dataset includes shopping records for several thousand individuals. We obtained the processed and simplified version of this dataset from the authors of [39]. Each data instance has 600 binary features. This dataset is clustered into 100 classes and the task is to predict the class for each customer. The dataset contains 197,324 data instances. We use the fully connected neural network from [31] as the target model.

LOCATION. This dataset contains the location "check-in" records of different people. It has 5,010 data records with 446 binary features, each of which corresponds to a certain location type and indicates whether the individual has visited that particular location. The goal is to predict the user's geo-social type. There are 30 classes in this dataset. We use the fully connected neural network from [31] as the target model.

4.1.2 Evaluated Attacks and defenses

Attacks. We consider the six attacks described in Section 2.1: LOSS [47], Modified Entropy [40], Class-NN [39], random perturbation attack from [19], the online version of the LIRA attack from [3] and the CANARY attack from [44]. Readers may want to review Section 2.1 on how these attacks work. The latter two are the state-of-the-art blackbox membership inference attack. We set the number of shadow models to be 100, which is common in the literature. We assume that the attackers are adaptive attackers, which means that the attackers already know MIST defense mechanism (including hyperparameters) and can train similar models in exactly the same setting.

Defenses. We compare with the following existing defenses: adversarial regularization (adv-reg) [30], Mem-guard [20], DMP [38], Mixup+MMD [22], SELENA [41], HAMP [7] and DP-SGD [1]. Readers may want to review Section 2.2 on how these defense methods work. As for our MIST defense, we use the following setting: for T_1, we set it to be the number so that each data instance in subset D^c can be visited just once and we set $T_2 = T_1$. The chosen λ for each model and dataset is also listed in Table 7. We first tune the C to achieve the highest validation accuracy possible then we tune the λ so that the test accuracy drop is less than 1%.

4.1.3 Evaluation Metrics

Following existing literature [3, 31], we use a balanced evaluation set. For CIFAR-10 and CIFAR-100 datasets, we assume that the model trainer has 20,000 data instances to perform the model training. For PURCHASE dataset and TEXAS dataset, we assume the model trainer has 40,000 training data instances. For LOCATION dataset, we assume the model trainer has 4,000 training data. We mainly use PLR (positive likelihood ratio, which is the ratio between TPR and FPR where lower means a more effective defense) at a fixed low FPR as suggested in Carlini et al. [3] to evaluate different

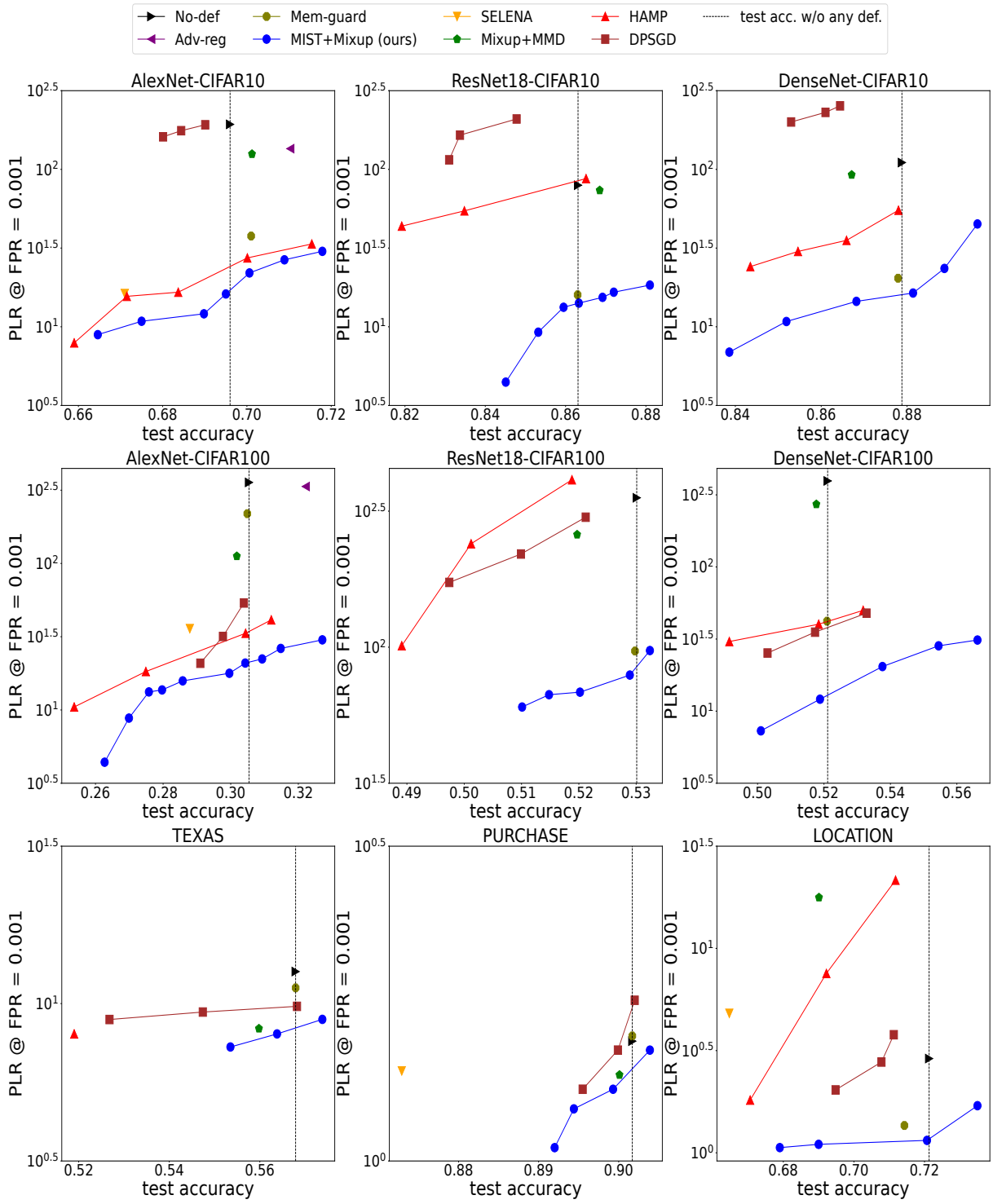

Figure 1: Comparing all defenses using the highest MI attack PLR @ 0.001 FPR among all evaluated attacks. **Defenses placed at the lower right corner are better** (high test accuracy and low PLR). Notice that for PURCHASE, TEXAS and LOCATION datasets the mixup data augmentation is not applied. We exclude some results when the test accuracy drop is larger than 5% for clearer comparison.

Defense name	Running time (s)	memory cost factor
SGD baseline	1195	1
Adv-reg	7008	1
DMP	2543	2
Mem-guard	36373	1
SELENA	20320	26
Mixup+MMD	3154	1
HAMP	1280	1
MIST+Mixup	2139	4

Table 1: Computational cost comparison for different defenses. For memory cost, we only count the number of models C to be trained.

attacks. For completeness, we also include AUC (area under curve) scores in the Appendix.

4.2 The computational cost (time and memory) of defenses

Here we assume that all the hyperparameters are already chosen, and compare the time that that model optimization needs to produce a final usable model and predictions for the test set. We run this comparison experiments on a server with one GTX-1080Ti GPU and we use CIFAR-100 on AlexNet as an example to compare. Table 1 shows our results, where we can see that MIST achieves the second fastest running time among all defenses (only slower than HAMP defense). Note that the SELENA defense requires significantly more time and space comparing to the other defenses, since it needs to train many extra models (25 with the code's default parameters) and store them. Moreover, the Mem-guard defense also requires significant time, because the noise for each prediction needs to be optimized separately. As for memory cost of our MIST defense, it depends on the number of models C (usually 2 to 4). The training time of MIST is about twice that of the baseline because at each epoch MIST performs two backprop steps instead of one to perform the regularization step. In conclusion, MIST requires moderate extra time and memory for model training where the best privacy-utility tradeoff is provided.

4.3 Defense effectiveness evaluation

Figure 1 presents the experimental results comparing MIST with other defenses on 5 datasets (with three model architectures for each of CIFAR-10 and CIFAR-100). We include only data points where the test accuracy drop is no more than than 5% for clearer comparison. In Figure 1, the x-axis gives test accuracy and the y-axis gives $\text{PLR}_{0.001}$ (PLR when FPR$=0.001$) in log-scale. We prefer methods to be in the lower-right region, as they can provide higher test accuracy under the same $\text{PLR}_{0.001}$. Note that our proposed defense method MIST+Mixup performs the best in all experiments.

When $\text{PLR}_{0.001} = 100$, it means that when we ensure that no more than 0.1% of non-members are falsely identified as members, if we identify 101 instances as members, we can expect that 100 are indeed members, and 1 of them is a false positive. When $\text{PLR}_{0.001} = 2$, it means that when we ensure that no more than 0.1% of non-members are falsely identified as members, if we identify 3 as members, we can expect 2 of them are indeed members. Due to page limit, we exclude the detailed numbers of experiments in this section and please refer to the appendix of the arxiv version of this paper if interested.

Effect of MIST+Mixup. We first look at how well our proposed defense (MIST+Mixup) prevents MI attacks. For CIFAR-10, our defense can reduce the highest PLR among all attacks from 127.63 to 17.48 (averaged over all models); for CIFAR-100 dataset, our defense can reduce the highest PLR from 368.94 to 29.97 (averaged over all models). For PURCHASE dataset, TEXAS dataset and LOCATION dataset, the highest PLR is reduced from 6.51 to 4.10, 1.55 to 1.08, 2.89 to 1.15 respectively. As for the average case attack effectiveness, our defense can also provide noticeable AUC drop comparing to the case without any defense. In summary, our new defense can provide significant attack effectiveness (including PLR and AUC) reduction for highly vulnerable cases (CIFAR-10 and CIFAR-100) and provide further attack effectiveness reduction on less vulnerable cases (PURCHASE, TEXAS and LOCATION), while incurring less than 1% test accuracy drop.

Comparison with adv-reg. We note that the adversarial regularization defense incurs significant test accuracy drop for all cases except when AlexNet is used (for CIFAR-10 or CIFAR-100). However, for these two cases, MIST+Mixup (blue curve) provides much lower PLR than adversarial regularization (purple dot) when similar test accuracy is achieved, as shown in Figure 1. In the adv-reg defense, the model is trained to defend against one attack model that evolves with it in adversarial training, and the resulting model can still be vulnerable to other attacks. Furthermore, the nature of adversarial training is such that one has to be very careful on tuning the hyperparameters because it is highly likely that the target model will have very low test accuracy.

Comparison with Mem-guard. The Mem-guard defense is a post-processing defense, which means the model utility (test accuracy in our case) is maintained. From Figure 1, we can conclude that MIST+Mixup (blue curve) provides lower PLR than Mem-guard (olive dot) when similar test accuracy is achieved. We note that due to computational resources and time constraint, we are not able to evaluate Mem-guard against the CANARY attack, since the CANARY attack involves instance optimization through multiple iterations and for each iteration the Mem-guard defense needs to calculate a new noise regarding each newly perturbed instance. If we

only compare the PLR of the LIRA attack, we can see that MIST+Mixup outperforms Mem-guard for all cases. Moreover, the PLR of the CANARY attack against MIST+Mixup is still lower than the PLR of the LIRA attack against Mem-guard for all CIFAR experiments. The Mem-guard defense uses one attack model to generate the noise to be added to each prediction, however this particular noise may not be effective against other attacks.

Comparison with DMP. In this comparison, we consider the case where no auxiliary unlabeled data is available because in practice collecting extra data is intractable for most privacy-concerning cases, for example medical images. More specifically, we train a GAN using the private training set and draw samples using the trained GAN to train the student model. However in our experiments, we see that the DMP defense also incurs significant test accuracy drop comparing to the no-defence case.

Comparison with Mixup+MMD. From Figure 1, we can conclude that MIST+Mixup (blue curve) provides lower PLR than Mixup+MMD (green dot) when similar test accuracy is achieved. We note that our proposed cross-difference loss is similar to the MMD (mean maximum discrepancy) loss proposed in Li et al. [22]. There are two key differences. One is the granularity, i.e., our proposed cross-difference loss is calculated on instance level, but the MMD loss is calculated on class level, where the loss is defined as the difference between averages of two classes. The other is that in [22], MMD loss is relative to a validation dataset. Here, we divide the dataset into multiple subsets and the cross-difference loss is relative to results from update from other subsets. As demonstrated in 4.8, training using divided subset can help the model generalize better (and less vulnerable to MI attacks) even without using the cross-difference loss.

Comparison with SELENA. We first note that the SELENA defense and its evaluation against shadow-model based attacks require extraordinarily large amount of computational resources. This is because the SELENA defense needs to train 25 (the default value) models and then train one distilled model from these 25 models. Moreover, if we want to evaluate the LIRA attack against the SELENA defense and the number of shadow models is 100, then we need to perform the SELENA training procedure for 100 times to generate 100 distilled models and in total we need to train 2600 models for all our models and datasets, which is unattainable in our experimental scenario where each model is not small. Thus, for our experiments we restrict SELENA defense evaluation to the following scenarios: CIFAR-10 with AlexNet, CIFAR-100 with AlexNet, PURCHASE, TEXAS and LOCATION. From Figure 1, we can conclude that MIST+Mixup defense (blue curve) provides lower PLR and higher test accuracy than SELENA defense (yellow dot). Particularly for the TEXAS dataset, the SELENA defense results in more than 25% test accuracy drop, thus we exclude this data point from Figure 1 for clearer comparison. One reason that our defense can outperform SELENA is that, during the training process, there is communication (regularizing each other and averaging parameters) between each submodels, however SELENA does not have communication between submodels. Moreover, this communication takes advantage of the proposed cross-difference loss, which helps to mitigate the overfitting and generalize better, thus is more favorable.

Comparison with HAMP. For the HAMP defense, we only evaluate the effectiveness of their training-time defense, since the test time defense can be integrated with any training time defenses and the test time defense is approximately equivalent to providing the predicted label only. In Figure 1, if we compare MIST+Mixup (blue curve) with HAMP (red curve), we can conclude that MIST+Mixup gives better privacy-utility tradeoff than HAMP. For the HAMP defense, our defense considers the MI difficulty of different individual instances and regularize more on vulnerable instances, however the HAMP defense treats all instances in the same fashion by applying an entropy-based regularizer and smoothed label. Moreover, by leveraging subspace learning, we can achieve better model generalization.

Comparison with DP-SGD. Note that for the DP-SGD defense we tune the hyperparameter so that the model trained with DP-SGD can achieve similar test accuracy and in this case, the privacy budget is huge. In Figure 1, if we compare MIST+Mixup (blue curve) with DP-SGD (brown curve), we can conclude that MIST gives better privacy-utility tradeoff than DP-SGD. The DP-SGD defense add random noise drawn from a given gaussian distribution, thus adding (expected) the same amount of noise for each instance, which implicitly treats each instance in the same way. However, our proposed MIST defense is able to treat each instance with different level of protection via the proposed cross-difference loss, thus can outperform the DP-SGD defense.

To summarize, our MIST defense can outperform all existing defenses for all evaluated cases. In addition, our defense also provides a "knob" (the parameters λ and T_2 of Phase 2) to further fine-tune the model, so practitioners can obtain different tradeoffs between test accuracy and MI attack robustness. This tradeoff flexibility is absent from some of the baseline defenses, unfortunately.

4.4 Evaluating against label only MIAs

Now we evaluate the effectiveness of MIST against label-only MIAs from Choquette-Choo et al. [8]. Particularly, we evaluate the more effective adversarial perturbation label-only MIA. The intuition of this label-only MIA is that the training samples need more perturbations to get their predicted class to be flipped. We use both the PLR at FPR$=0.1\%$ and the AUC score to evaluate the effectiveness of the distance attack. The results are shown in Table 2. It is easy to see that the label-

Defense	Attack PLR (FPR@0.1%) on CIFAR-10	Attack PLR (FPR@0.1%) on CIFAR-100	Attack AUC on CIFAR-10	Attack AUC on CIFAR-100
No Defense	1.10	0.90	0.641	0.826
Adv-reg	1.39	1.13	0.631	0.815
Mem-guard	1.10	0.90	0.641	0.826
Mixup+MMD	1.30	1.50	0.605	0.742
SELENA	1.21	1.11	0.542	0.545
HAMP	1.73	1.94	0.577	0.633
DP-SGD	1.40	1.03	0.645	0.661
MIST+Mixup (ours)	**1.05**	**0.70**	**0.536**	**0.540**

Table 2: Evaluation against label only MIA. CIFAR dataset, AlexNet. The DMP defense would result in training failure (model not converging), thus excluded from this table.

only MI attack is not effective in detecting most vulnerable instances (PLR close to 1 means that TPR is close to FPR). Moreover, we can conclude that our proposed defense can significantly reduce the AUC score of the distance attack and outperform all existing attacks in defending against this label only membership inference attack.

4.5 Evaluating against white-box MIAs

Carlini et al. [4] proposed a model extraction method that can accurately extract parameters for a 5-layer fully connected neural network with 1110 parameters. If an effective model extraction attack against deep DNNs is discovered, then black-box access to a target model becomes equivalent to whitebox access, and we need to evaluate defenses against whitebox attacks as well. In this subsection, we evaluate all defenses against the white-box MIA from Nasr et al. [31] to understand the effect of our proposed defense against white-box attacks. The results are shown in Tables 3 and 4. One can also see that our proposed defense can provide significantly lower PLR and AUC score comparing to other existing defenses, which shows that our proposed defense can provide benefits against both black-box attacks and white-box attacks. From the results we can also see that this whitebox attack is no more effective than state-of-the-art blackbox attacks such as LiRA. This fact may not be as counter-intuitive as it first seems. Exact model parameters are affected by many factors beyond the training dataset, including initial randomness in initialization, hyper-parameters, and randomness during training (e.g., due to dropout). A member instance affects the model parameters because the parameters are trained to minimize loss on the instance, thus the loss on the instance (which is used in black-box attacks) is the main "footprint" left the instance being a member.

4.6 Comparing against other baselines on the most vulnerable instances

In this subsection, we want to understand the impact of all defenses over the most vulnerable instances in the training set. For this experiment we consider AlexNet, CIFAR-100 and CIFAR-10 datasets. Our first step is finding the set of most vulnerable instances, which we will denote X_{vul}. The size of X_{vul} is set to be 1000 in this experiment. We train 100 models using AlexNet on CIFAR-100 (and CIFAR-10 datasets) without any defense and perform LIRA attack. Recall that the LIRA attack would produce two normal distributions of the loss (one for member case and one for non-member case) for each instance. For each instance, we calculate the non-overlapping area of these two normal distributions and choose the top 1000 instances with the largest non-overlapping area to construct the X_{vul}. Intuitively, the larger the non-overlapping area is, the more vulnerable the instance is against membership inference attacks.

Using the X_{vul} above we can evaluate two metrics: the test accuracy on X_{vul} and the highest attack PLR on X_{vul}. Moreover, we add the highest attack PLR on the whole training set for comparison and further illustration in Table 5. Note that for a fair comparison between different defenses (defenses evaluated in this subsection should result in less than 1% test accuracy drop) and clearer results presentation, we remove the SELENA defense and the DMP defense from this experiment, since these two defenses would result in significant test accuracy drop. In conclusion, the results in Table 5 show that our MIST defense can outperform all existing membership inference defenses in terms of attack effectiveness reduction on both the vulnerable instances X_{vul} and on the whole MI evaluation dataset X_{whole}.

In particular, a natural defense is the removal from training of the most vulnerable instances against membership inference attacks. The intuition is that, once the most vulnerable instances are removed, the remaining training instances should

Defense	No-def	MIST+Mix-up	Adv-reg	Memg-guard	Mix-up+MMD	SELENA	HAMP	DP-SGD
PLR(FPR@0.1%)	1.01	**0.00**	0.00	1.01	0.00	0.97	0.40	0.33
AUC score	0.734	**0.503**	0.706	0.741	0.512	0.541	0.579	0.584

Table 3: Evaluation against white-box MIA. CIFAR-10 dataset, AlexNet. The DMP defense would result in training failure (model not converging), thus excluded from this table.

Defense	No-def	MIST+Mix-up	Adv-reg	Memg-guard	Mix-up+MMD	SELENA	HAMP	DP-SGD
PLR(FPR@0.1%)	1.61	**0.00**	1.68	1.61	0.00	2.08	3.12	0.32
AUC score	0.871	**0.505**	0.866	0.871	0.542	0.541	0.663	0.667

Table 4: Evaluation against white-box MIA. CIFAR-100 dataset, AlexNet. The DMP defense would result in training failure (model not converging), thus excluded from this table.

Dataset-Model	Defense	Acc. on X_{vul}	Test Acc. on X_{test}	PLR (FPR@1%) on X_{vul}	PLR (FPR@1%) on X_{whole}
CIFAR-100, AlexNet	No Defense	1.00	0.30	73.89	25.65
CIFAR-100, AlexNet	Adv-reg	0.77	0.32	83.31	8.18
CIFAR-100, AlexNet	Mem-guard	0.95	0.30	81.98	32.19
CIFAR-100, AlexNet	Mixup+MMD	0.37	0.30	52.61	23.13
CIFAR-100, AlexNet	HAMP	0.30	0.31	41.45	17.00
CIFAR-100, AlexNet	DP-SGD	0.21	0.30	18.76	7.23
CIFAR-100, AlexNet	Data Removal	0.08	0.28	N/A	26.53
CIFAR-100, AlexNet	MIST+Mixup (ours)	0.13	0.30	**16.10**	**6.76**
CIFAR-10, AlexNet	No Defense	1.00	0.70	31.68	18.78
CIFAR-10, AlexNet	Adv-reg	0.78	0.71	27.71	43.91
CIFAR-10, AlexNet	Mem-guard	0.95	0.70	31.74	6.64
CIFAR-10, AlexNet	Mixup+MMD	0.62	0.70	30.80	9.79
CIFAR-10, AlexNet	HAMP	0.51	0.71	16.24	5.42
CIFAR-10, AlexNet	DP-SGD	0.58	0.69	28.95	5.63
CIFAR-10, AlexNet	Data Removal	0.30	0.68	N/A	19.11
CIFAR-10, AlexNet	MIST+Mixup (ours)	0.41	0.70	**13.15**	**5.26**

Table 5: Evaluation against label only MIA. CIFAR dataset, AlexNet. The DMP defense would result in training failure (model not converging), thus excluded from this table. The SELENA defense is also removed for a fair comparison. X_{vul} stands for the vulnerable instance set (of size 1000). X_{test} stands for the whole test set. X_{whole} stands for the whole evaluation set that we use in previous experiments, which includes $20,000$ training instances. For No Defense case, the X_{vul} is included in training, thus the test accuracy on X_{vul} is not applicable. For the PLR metric on X_{vul}, since this is not part of the training set for the data removal defense, this metric is not applicable.

be more robust against MI attacks. Unfortunately, our experiments show that this defense is surprisingly ineffective, with re-identification metrics close to the "No Defense" scenario, which is also detailed in Table 5.

The difference between MIST and the data removal defense is worth a closer look. For the test accuracy on X_{vul}, the performance of the data removal defense is significantly worse than MIST defense. For the PLR at 1% FPR on X_{vul}, since the whole X_{vul} is removed for the data removal defense, this PLR metric is not applicable to the data removal defense and there is no privacy threat against X_{vul}. However, for the PLR at 1% FPR on X_{whole}, the data removal defense performs much worse than MIST defense and close to the "No Defense" scenario, which makes the data removal defense similar to having no defense.

Let's take CIFAR-100, AlexNet as one example. Intuitively, when FPR=1%, the PLR is 25.65 for no-defense case, which means that $0.01 \times 25.65 \times 20000 = 5130$ instances have been detected as members. If we remove the X_{vul} from training, then the expected PLR at FPR=1% should be $\frac{4130}{5130} * 25.65 = 20.65$, however, the resulted PLR is now 26.53. This result implies that even though the vulnerable instances X_{vul} are removed from the training set, now other instances became vulnerable against MI attacks. One intuitive explanation is that, since X_{vul} is mostly constructed by outliers (since outliers are harder to fit and thus easier to be attacked by membership inference attacks), removing them from training would make some other instances to become outliers, just like peeling an onion. This finding is aligned with the recent findings of Carnili et al. [5], which show that removing vulnerable instances from the training set allows other instances to become outliers and thus vulnerable.

To summarize, the data removal defense seems to be ineffective. Moreover, we show that MIST, our proposed defense, can provide the most reduction in attack effectiveness on the most vulnerable set, while achieving the best average case test accuracy and (both average case and extreme case) privacy tradeoff of all baselines.

4.7 Choosing hyperparameters

Now we discuss how to determine the two most crucial hyperparameters of our proposed defense: the number of models C and the weight of the cross-difference loss. In this paper, we use the following method to determine these two hyperparameters: we choose the number of models C to achieve the highest validation accuracy possible and then choose the weight of the cross-difference loss to provide the strongest defense while making sure the test accuracy drop is less than 1% comparing to the base case without any defense. In Figure 5 in the Appendix , we show that the customized subspace learning (without the cross-difference loss) can help the model generalize better, thus we would like to use the model with the highest validation accuracy. Moreover, the mixup data augmentation can also help to further improve the validation accuracy. Hence, subspace learning and mixup combined get higher test accuracy boost compared to standard SGD, which then gives us room to increase the regularization of the cross difference loss without lower test accuracy when compared against standard SGD.

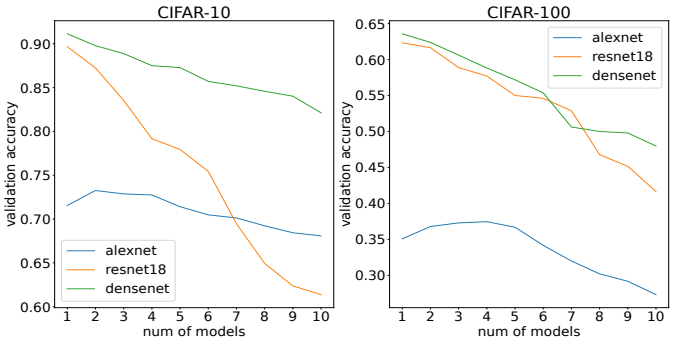

Figure 2: Validation accuracy v.s. number of models C. Mixup data augmentation is applied.

In Figure 2, we show the validation accuracy curve v.s. the number of models C for all models and datasets where mixup data augmentation is applicable. Recall that we need to choose the number of models C that achieves the highest validation accuracy. However, we also need to have at least two models, so that our cross-difference loss is applicable. Thus, where mixup data augmentation is applicable, for all models and datasets except CIFAR-100 and AlexNet, the number of models C should be 2. For the CIFAR-100 and AlexNet, the number of models C should be four. For the case where the mixup data augmentation is not applicable (TEXAS dataset and PURCHASE dataset), we choose to use 4 models for TEXAS dataset and 4 models for PURCHASE dataset based on Figure 5 in the Appendix. In practice, we recommend developers to generate this validation accuracy curve for their own datasets to determine the number of models C. If the computational resources are limited, our ad-hoc advice is to use 2 to 4 models since this range was suitable for all our datasets.

As for the choice of the hyperparameter λ of Phase 2 (the cross-difference loss minimization), we choose λ as to provide the strongest defense while making sure the test accuracy drop is less than 1%. In practice, the training does not need to follow this procedure, as λ may be utilized as a knob to tune a privacy-utility tradeoff, which allow practitioners to determine this tradeoff based on their needs.

4.8 Ablation Study

The positive impact of subspace learning and cross-difference loss. In this ablation study, we use AlexNet on CIFAR-10 and CIFAR-100 dataset. For CIFAR-10 dataset,

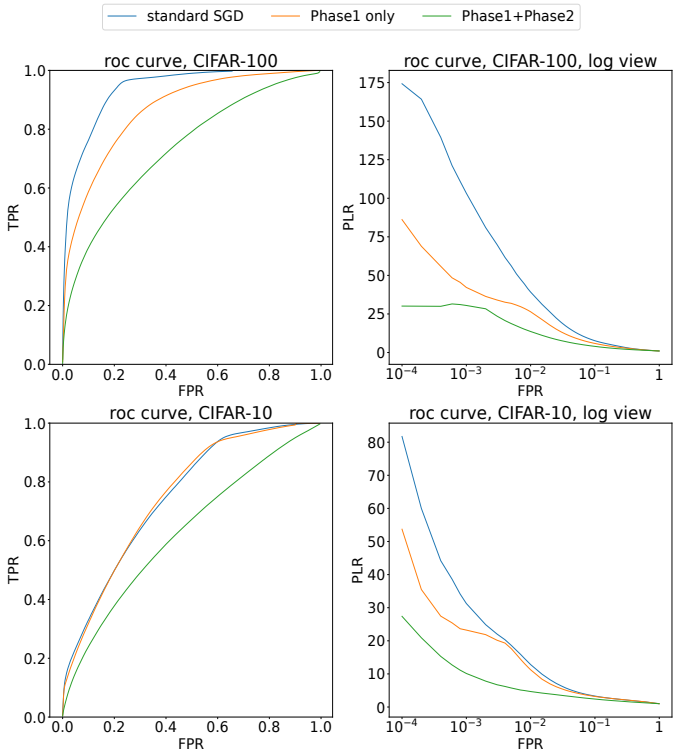

Figure 3: The clear positive impact of Phase 1 and Phase 2. Impact measured on ROC curve and PLR at low FPR region. LIRA attack is evaluated to produce the ROC curve and PLR curve. CIFAR-100 and CIFAR-10 using AlexNet.

we use 3 models and for CIFAR-100 dataset we use 7 models. These two values are chosen because the highest test accuracy (higher than the standard SGD case with one model) can be achieve with this choice when no cross-difference loss and mix-up data augmentation is applied . For the λ of the cross difference loss, we choose the λ so that the maximum defense can be provided while the test accuracy drop is less than 1% in this experiment. We compare the attack effectiveness against model trained without defense, trained with Phase 1 only and trained with both Phases 1 and 2. The results are shown in Figure 3. We can conclude that the only performing Phase 1 of our MIST algorithm (ignoring Phase 2) can still be used as a defense which can preserve utility and decrease attack effectiveness at the same time with appropriate number of models C. We also repeat this experiment on other models and datasets and we observe similar phenomenon. Moreover, adding Phase 2 (cross-difference loss minimization) can provide further robustness against the LIRA attack.

Different cross-difference loss implementations. In this ablation study, we experiment on different implementations for the cross-difference loss. More specifically, we use three different implementations: KL-divergence, Equation (3) and L_2 version of Equation (3). We run experiments using AlexNet

Loss function	Test accuracy	PLR at 0.1% FPR
L1	0.3049	17.43
L2	0.2997	17.54
KL-divergence	0.3003	19.72

Table 6: Comparing different implementations of the MIST loss. CIFAR-100 dataset on AlexNet.

on CIFAR-100 dataset and we choose the optimal weights that we can find for each loss function so that the test accuracy drop is less than 1% comparing to the case without any defense. The results are shown in Table 6, where we can conclude that the L_1 variant slightly outperforms the other two variants and the KL-divergence variant performs the worst. One possible reason for the KL-divergence variant being worse than L_1 is because the KL-divergence considers the distance between two prediction vectors, while the state-of-the-art attacks only focus on the probability of the correct label. Moreover, we also observe that this KL-divergence variant makes the model training unstable, meaning that sometimes the model fails to converge, thus we recommend the use of L_1.

5 Conclusion

In this work we introduce Membership-Invariant Subspace Training (MIST), a method for training classifiers that acts as a defense designed to specifically defend against black-box membership inference attacks on the most vulnerable instances in the training data (which also helps defending against overall blackbox MI attacks). MIST uses a novel two-phase subspace learning procedure that trains models regularized to be *membership invariant* (counterfactually invariant to the membership of each instance in the training data D). Through our proposed two-phase subspace learning method, and a new proposed loss (*cross-difference loss*), we can efficiently regularize neural network training towards membership-invariant classifiers. Our experiments empirically show that our trained classifiers are significantly less vulnerable against blackbox membership inference attacks than baselines, especially on those most vulnerable training examples.

6 Limitation and Future Works

The main limitation of the proposed MIST defense is the computational overhead of maintaining multiple models. This is because at each epoch, the MIST defense requires maintaining multiple copies of the central model, performing local model updates, update local models based on predictions from other models, and averaging the model weights. More specifically, if the training set is divided into C subsets, this results

in a memory overhead by a factor C. One possible mitigation strategy is to perform the training for each submodel in a sequential way and only load one model into memory. However, this solution would result in a time overhead of a factor C. Another interesting future direction is adapting the MIST to regression and other non-classification tasks. Finally, there may be other, yet unknown, ways to apply the cross-difference loss that may reduce overhead and improve performance (better privacy-utility tradeoff).

7 Acknowledgements

This work was funded in part by the National Science Foundation (NSF) Awards CAREER IIS-1943364, CCF1918483, CNS-2247794, and IIS-2229876, Amazon Research Award, AnalytiXIN, the Wabash Heartland Innovation Network (WHIN), Ford, and CISCO. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.

References

[1] Martín 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. ACM, 2016.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

[3] 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.

[4] Nicholas Carlini, Matthew Jagielski, and Ilya Mironov. Cryptanalytic extraction of neural network models. In *Annual international cryptology conference*, pages 189–218. Springer, 2020.

[5] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramer. The privacy onion effect: Memorization is relative. *Advances in Neural Information Processing Systems*, 35:13263–13276, 2022.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017.

[7] Zitao Chen and Karthik Pattabiraman. Overconfidence is a dangerous thing: Mitigating membership inference attacks by enforcing less confident prediction. *arXiv preprint arXiv:2307.01610*, 2023.

[8] 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.

[9] Cynthia Dwork. Differential privacy. In *ICALP*, pages 1–12, 2006.

[10] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[11] 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.

[12] Robert Fortet and Edith Mourier. Convergence de la répartition empirique vers la répartition théorique. *Annales scientifiques de l'École Normale Supérieure*, 70(3):267–285, 1953.

[13] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *The journal of machine learning research*, 17(1):2096–2030, 2016.

[14] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[15] Xinlei He and Yang Zhang. Quantifying and mitigating privacy risks of contrastive learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 845–863, 2021.

[16] Seira Hidano, Takao Murakami, and Yusuke Kawamoto. Transmia: membership inference attacks using transfer shadow training. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10. IEEE, 2021.

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[18] Samyak Jain, Sravanti Addepalli, Pawan Kumar Sahu, Priyam Dey, and R Venkatesh Babu. Dart: Diversify-aggregate-repeat training improves generalization of neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16048–16059, 2023.

[19] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. Revisiting membership inference under realistic assumptions. *Proceedings on Privacy Enhancing Technologies*, 2021(2), 2021.

[20] 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.

[21] Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *Advances in Neural Information Processing Systems*, 35:20782–20794, 2022.

[22] 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.

[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] Ninghui Li, Wahbeh Qardaji, Dong Su, Yi Wu, and Weining Yang. Membership privacy: A unifying framework for privacy definitions. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 889–900, 2013.

[25] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. Encodermi: Membership inference against pre-trained encoders in contrastive learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2081–2095, 2021.

[26] Yunhui Long, Vincent Bindschaedler, and Carl A Gunter. Towards measuring membership privacy. *arXiv preprint arXiv:1712.09136*, 2017.

[27] Gustavo López, Luis Quesada, and Luis A Guerrero. Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In *Advances in Human Factors and Systems Interaction: Proceedings of the AHFE 2017 International Conference on Human Factors and Systems Interaction, July 17- 21, 2017, The Westin Bonaventure Hotel, Los Angeles, California, USA 8*, pages 241–250. Springer, 2018.

[28] 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*, pages 691–706. IEEE, 2019.

[29] S Chandra Mouli and Bruno Ribeiro. Asymmetry learning for counterfactually-invariant classification in ood tasks. In *International Conference on Learning Representations*, 2022.

[30] 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. ACM, 2018.

[31] 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*, pages 739–753. IEEE, 2019.

[32] OpenAI. Gpt-4 technical report, 2023.

[33] Junhyung Park and Krikamol Muandet. A measure-theoretic approach to kernel conditional mean embeddings. *Advances in neural information processing systems*, 33:21247–21259, 2020.

[34] Francesco Quinzan, Cecilia Casolo, Krikamol Muandet, Niki Kilbertus, and Yucen Luo. Learning counterfactually invariant predictors. *arXiv preprint arXiv:2207.09768*, 2022.

[35] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[36] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. Updates-leak: Data set inference and reconstruction attacks in online learning. In *29th USENIX Security Symposium*, pages 1291–1308, 2020.

[37] 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 *25th Annual Network and Distributed System Security Symposium (NDSS)*, 2019.

[38] Virat Shejwalkar and Amir Houmansadr. Membership privacy for machine learning models through knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

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

[40] Liwei Song, Reza Shokri, and Prateek Mittal. Membership inference attacks against adversarially robust deep learning models. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 50–56. IEEE, 2019.

[41] 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.

[42] Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. *Advances in neural information processing systems*, 34:16196–16208, 2021.

[43] Lauren Watson, Chuan Guo, Graham Cormode, and Alex Sablayrolles. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440*, 2021.

[44] Yuxin Wen, Arpit Bansal, Hamid Kazemi, Eitan Borgnia, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Canary in a coalmine: Better membership inference with ensembled adversarial queries. *arXiv preprint arXiv:2210.10750*, 2022.

[45] Mitchell Wortsman, Maxwell C Horton, Carlos Guestrin, Ali Farhadi, and Mohammad Rastegari. Learning neural network subspaces. In *International Conference on Machine Learning*, pages 11217–11227. PMLR, 2021.

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

[47] 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.

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

[49] Ligeng Zhu and Song Han. Deep leakage from gradients. In *Federated Learning*, pages 17–31. Springer, 2020.

A Further Related Work

Other MI attacks that are similar to LIRA but weaker. Several MI attacks are similar to LIRA and are shown in [3] to be less effective in LIRA; we thus do not consider them in the experiments. We list these attacks below.

Long et al. [26] proposed to train instance-specific MI classifiers. For each instance x, there will be two sets of models: trained with x and trained without x. Then, the average prediction of x from the member set and the average prediction of x from the non-member set is calculated. x is predicted to be a member if the target model's prediction of x is more similar to the average prediction of the member set (in the sense of KL divergence).

Ye et al. [46] followed the same shadow model procedure as [43] to produce a collection of loss for one instance when this instance is not used in the training. Next, a one-sided hypothesis testing is proposed to predict membership. The advantage of this hypothesis testing is that the attacker could select a false positive rate. However, the possible false positive rate range is limited by the number of shadow models.

Watson et al. [43] propose to use the shadow models to set a per-instance loss threshold by the average of loss of the instance on shadow models that are not trained using this example.

Other membership inference defenses. In [39], it was proposed to reduce the information given by the prediction vectors, such as providing only the top-k probabilities and using high temperature in softmax. This has limited effectiveness as the top-k probabilities give enough information needed by the best attacks, and high temperatures change only the absolute values of confidences, but not the fact that confidences for members tend to be higher than that for non-members. Moreover, this defense is not effective against LIRA, thus we do not consider this defense in our experiments.

Membership inference attacks in other settings. Melis et al. [28] identified membership leakage when using FL for training a word embedding function, which is a deterministic function that maps each word to a high-dimensional vector. Given a training batch (composed of sentences), the gradients are all 0's for the words that do not appear in this batch, and thus the set of words used in the training sentences can be inferred. The attack assumes that the participants update the central server after each mini-batch, as opposed to updating after each training epoch.

Nasr et al. [31] use gradient updates as feature vectors and train an auto-encoder to generate a single-number embedding to predict membership. This attack was also applied in the white-box setting, for which it performs worse than attacks that directly use gradient norm to predict membership, which show only small improvement over blackbox attacks. Another interesting idea in [31] is that a malicious server can actively improve MI attacks, by applying a gradient ascent with respect

to the target instance. If the instance is a member, then the gradients tend to be larger in order to compensate for the malicious gradient ascent.

Li et al. [23] proposed a new membership inference attack against the federated learning setting. They observed that the cosine similarity between gradients of each data instance and the parameter updates sent by each client has very different distribution for members and non-members. Thus, the cosine similarity is used as the metric to predict members. They also proposed to use the gradient difference between the gradients of each data instance and the parameter updates to predict membership. Their results show that the new proposed attacks can achieve higher TPR at low FPR than the attack proposed in [31].

Song et al. [40] evaluated how adversarial training would affect the privacy leakage. Experiments showed that adversarial training would boost the performance of instance loss membership inference attack and the testing accuracy of the target model will be decreased.

Liu et al. [25] evaluated the privacy leakage of pre-trained models which are trained using unsupervised contrastive learning strategy. The authors proposed a specific membership inference attack against pre-trained models. Given one instance, this new attack generates many perturbed versions of this instance and gather all the embeddings of these perturbed versions using the pre-trained model. The intuition is if one instance is used in the training of this pre-trained model, then the embeddings of its perturbed versions are generally closer to each other. He et al. [15] utilized the same contrastive learning idea and proposed to fine-tune the pre-trained model to get the final target model. Experiments showed that using pre-trained model as feature extractor would reduce privacy leakage, comparing to training models from scratch.

Hidano et al. [16] evaluated MI attack under the setting where the attacker can know the parameters of some shallow layers. In the experiment, the author assumed that the attacker can get all but the last layer and the parameters of these layers are used to initialize shadow models to facilitate the class-vector attack. With these known parameters, the class-vector attack can outperform its black-box version.

Other privacy-threatening attacks. Salem et al. [36] studied the possible information leakage of an update set when the machine learning model is updated using the update set. To detect the information leakage, the authors proposed two different attacks: label inference attack and instance reconstruction attack. These two attacks can be applied to both single instance update set case and multi-instances update set case, with only black-box access to the machine learning model.

Zhu et al. [49] presented a gradient-based instance reconstruction attack. If the gradients of one specific instance are revealed to one adversary who can access the trained model, then the adversary is able to reconstruct the specific instance with high fidelity. One random instance is gradually optimized

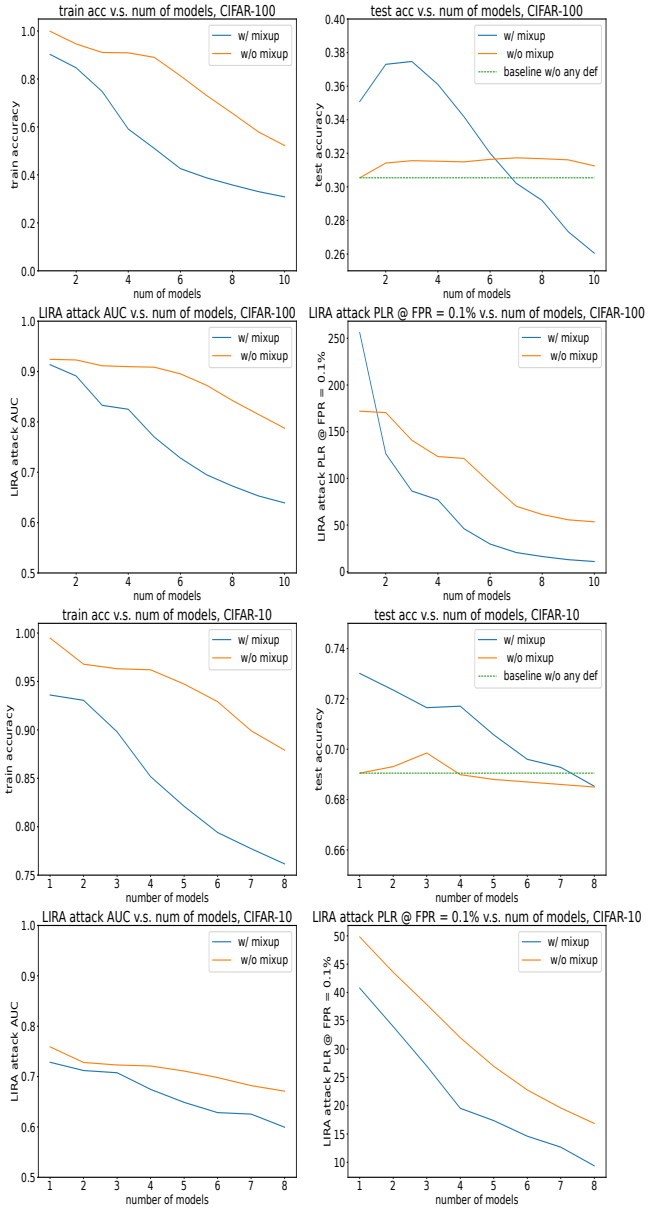

Figure 4: The impact of mixup data augmentation on training accuracy, testing accuracy and attack effectiveness when the number of models is varied. AlexNet, CIFAR-10 and CIFAR-100 datasets.

by matching its gradients with the provided gradients of the specific instance.

B Additional Ablation Study

The positive impact of mixup data augmentation. The experiment is performed using CIFAR-10 and CIFAR-100

dataset	model	lr	epochs	schedule	batch size	number of models	lambda
CIFAR-10	AlexNet	0.05	120	[100]	100	2	3.5
CIFAR-10	ResNet-18	0.10	300	[200,250]	100	2	3.5
CIFAR-10	DenseNet_BC(100,12)	0.10	500	[300,350]	100	2	6.0
CIFAR-100	AlexNet	0.05	120	[100]	100	4	12.0
CIFAR-100	ResNet-18	0.10	300	[200,250]	100	2	13.0
CIFAR-100	DenseNet_BC(100,12)	0.10	500	[300,350]	100	2	13.0
PURCHASE-100	PURCHASE-100	0.10	100	None	100	4	40.0
TEXAS-100	TEXAS-100	0.10	100	None	100	4	25.0
LOCATION	LOCATION	0.10	100	None	100	4	14.0

Table 7: Training recipe for different models. Learning rate is adjusted to 0.1x when current epoch is in schedule.

dataset on AlexNet. We perform experiments while varying the number of models K from 1 to 8 for CIFAR-10 (10 for CIFAR-100). D, which contains 20,000 training instances, is divided into K equal-size, disjoint partitions, and each model will be trained using one partition for one epoch, before we average the model. For each number of models choice, we experiment with two cases: with mixup and without mixup. The results are shown in Figure 4. From Figure 4, for CIFAR-100 dataset, the first thing we can observe is that the training accuracy is decreased and the testing accuracy is increased when mixup is applied for the same number of models. Furthermore, by adding the mixup data augmentation, the LIRA attack AUC and LIRA attack PLR at low FPR are both reduced when the same number of models are created. This validates the effectiveness of the mixup data augmentation. However, we also noticed that the testing accuracy is always decreasing while creating more than one models and when the number of models is seven, the testing accuracy is very similar to the testing accuracy when there is no defense. Thus, there is a privacy-utility tradeoff that should be considered by the model trainer and the model trainer should choose the number of models based on their own needs. In addition, in order to regularize the trained model with the cross-difference loss, there should be at least two models. For CIFAR-10 dataset, we also notice that the training accuracy is decreased and the testing accuracy is increased when mixup is applied for the same number of models, which again verifies the effectiveness of the mixup data augmentation. The only difference is that, the testing accuracy increases and then decreases while having more models. Therefore, we suggest the model trainers to generate similar figures like Figure 4 and choose the number of models based on their needs.

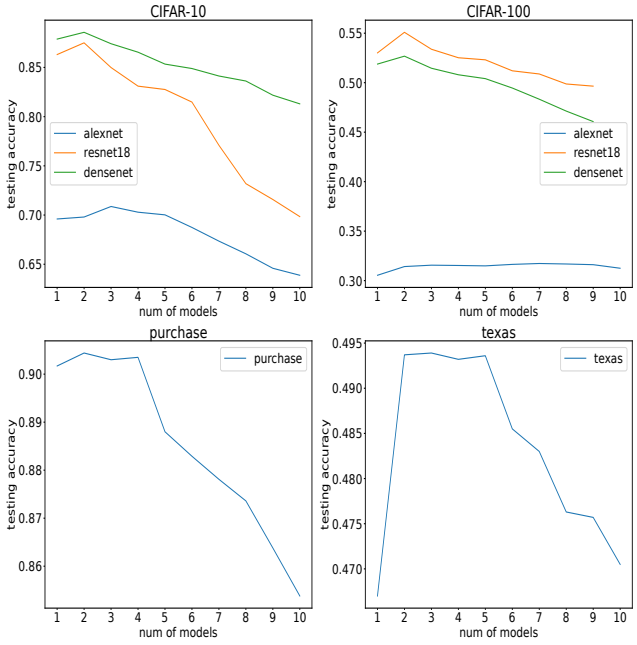

Figure 5: Testing accuracy v.s. number of models for all datasets and models considered in this paper. No mixup data augmentation. The main observation is that testing accuracy can be improved by adding a few models. However, adding too many models would cost testing accuracy.

C Additional Experimental Details and Results

Hyperparameters. In Table 7, we present all the hyperparameters used in our evaluation. Due to page limit, we exclude the detailed numbers for all defenses evaluation and extra additional experiments. Please refer to the arxiv version of this paper if interested.