Mitigating Membership Inference Attacks via Weighted Smoothing

Mingtian Tan
University of Virginia
wtd3gz@virginia.edu

Jun Sun, Xiaofei Xie
Singapore Management University
junsun@smu.edu.sg
xfxie@smu.edu.sg

Tianhao Wang
University of Virginia
tianhao@virginia.edu

ABSTRACT

Recent advancements in deep learning have spotlighted a crucial privacy vulnerability to membership inference attack (MIA), where adversaries can determine if specific data was present in a training set, thus potentially revealing sensitive information. In this paper, we introduce a technique, weighted smoothing (WS), to mitigate MIA risks. Our approach is anchored on the observation that training samples differ in their vulnerability to MIA, primarily based on their distance to clusters of similar samples. The intuition is clusters will make model predictions more confident and increase MIA risks. Thus WS strategically introduces noise to training samples, depending on whether they are near a cluster or isolated. We evaluate WS against MIAs on multiple benchmark datasets and model architectures, demonstrating its effectiveness. We publish code at https://github.com/BennyTMT/weighted-smoothing.

1 INTRODUCTION

In recent years, deep learning, grounded in neural networks, has made extraordinary strides in numerous fields. Despite these advancements, a pressing issue of model overfitting exists, presenting serious privacy risks for sensitive data embedded within the training set [9, 17, 18, 36, 37]. This vulnerability allows adversaries to determine the presence of specific data in the training set, an unintended disclosure of confidential information. A concrete example of this risk emerges when a user is identified within the training set of a disease analytic model. Such detection allows adversaries to infer that the user is or has been afflicted with a specific disease.

This type of security breach, known as a membership inference attack (MIA), was initially introduced by Homer et al. [23] in the context of genomic data. In a machine learning framework, Shokri et al. [36] were the first to formalize MIA as a binary classification task, where a given input sample is categorized as a member (i.e., part of the training set) or a non-member.

Various strategies have been advanced to mitigate MIA. Nasr et al. [31] introduce a regularization method, wherein a model is co-trained with an MIA model (i.e., a binary classifier for MIA), utilizing attack results as a regularizing loss function to diminish the real MIA's efficacy. Jia et al. [27] suggest a different approach, MemGuard, which masks confidence scores by injecting crafted noise into the prediction vector. Both defenses were subsequently challenged by a later MIA attack, the "privacy risk score" attack, proposed by Song and Mittal [38]. An alternative defense strategy [35, 41], using knowledge distillation [2, 22], involves substituting the original model with a simplified model. This method, while intuitively preventing the memorization of training data, requires data from the same distribution or more training costs (or both). Additionally, approaches for mitigating MIA may leverage the concept of differential privacy (DP). DP-based techniques function by strategically injecting noise throughout the training process. Despite offering theoretical guarantees against MIA [4, 26, 48], these methods frequently yield models with markedly reduced accuracy [1, 24].

In this work, we focus on MIA mitigation approaches that add perturbation to the training phase. We introduce a method termed weighted smoothing (hereafter WS) aimed at mitigating MIA while minimizing the impact on model accuracy. We start by observing a correlation between the distribution of training samples and the efficacy of MIA. Intuitively, a concentrated cluster of training samples at the class center escalates the risk of MIA for all of them. The high-level idea is that clustered samples collectively reinforce model memorization. As MIA fundamentally utilizes the model's confidence difference between members and non-members, the increased confidence for all these clustered samples makes themselves more vulnerable to MIA. Conversely, a dispersed distribution, spanning a larger space around the center, lowers the MIA risk. Guided by this insight, WS is formulated to introduce noise to the training samples. The extent of noise infused into a training sample is contingent on whether it is in a cluster or isolated, thereby ensuring a calibrated and limited amount of noise. For samples that clustered together, we add more noise to make the model less confident. As a by-product, this also improves model generality to some extent. For those isolated points, our WS adds less noise during training to ensure the model's accuracy is not overly compromised on those points.

To evaluate the efficacy of our proposed WS method, we deploy it against two sophisticated MIAs: a neural network-based MIA utilizing a shadow-training method [36], and a metric-based MIA employing 'privacy risk score' [38]. These MIAs are evaluated across diverse models, trained on established benchmark datasets: CIFAR-10, CIFAR-100 [13], CASIA-FACE [14], HAM10000 [43], Location, and Texas100 [36]. Three different model architectures, DenseNet [25], ResNet [21], and a deep neural network adopted from [27], are used in this assessment. Furthermore, an extensive evaluation involving the cutting-edge attack model by Carlini et al. [5] is conducted, employing both DenseNet and ResNet architectures within the CIFAR-10 dataset. Our WS method is benchmarked against the DP-based method by Abadi et al. [1].

ACSAC '23, December 5–9, 2023, Austin, TX, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

We refer readers to Section 6 for a detailed exposition of experimental outcomes. Concisely, our approach stands out by ensuring limited accuracy degradation while maintaining equivalent MIA mitigation levels. Specifically, in the context of a DenseNet model trained on CIFAR-10, CIFAR-100, CASIA-FACE, and HAM10000 datasets, our method mitigates MIAs. Remarkably, this is accomplished with an average accuracy reduction of a mere 0.95% on both metric-based [38] and neural-network-based [36] attacks, a significant improvement over the baseline MIA mitigation's accuracy loss of 16.11%.

Our contributions in this work are summarized as follows.

(1) We empirically establish that training samples exhibit heterogeneous levels of MIA risk, underscoring the significance of examining and understanding these risks in developing effective MIA mitigation strategies.
(2) In light of our insights into the disparate MIA susceptibilities across samples, we introduce the WS technique. This approach infuses weighted noise, proportional to a sample's MIA risk, into training steps of that sample. Empirical results demonstrate that WS offers robust MIA mitigation while preserving model accuracy.

The rest of this paper is organized as follows. We review related work in Section 2. In Section 3, we present our observations, and discuss our interpretation as well as the relationship between MIA and data distribution. In Section 4, we introduce our WS in detail. In Section 5 we introduce the experimental setup. In Section 6, we present details of our evaluation and experimental results. Lastly, we conclude in Section 7.

2 RELATED WORK

2.1 Membership Inference Attacks

Membership inference attacks (MIA) ascertain the likelihood of a sample belonging to the training dataset. Most MIAs leverage observed posterior [34, 36] or solely labels [10, 30] as the feature for the attack. State-of-the-art MIA include neural network-based [34, 36] and metric-based categories [38, 45]. The *neural network-based MIA* employs a binary classifier neural network (hereinafter referred to as the attacking neural network) for the attack, utilizing a sample's prediction vector as input to yield a membership classification outcome.

On the contrary, *metric-based MIA* works by calculating a scale based on the prediction vector, which is then compared with a threshold to decide whether the sample is a member or not. Song et al. [38] propose "privacy risk score", achieving impressive attack results. Additionally, Ye et al. [44] put forth a novel MIA method grounded in a theoretical game framework to decipher the privacy leakage of ML algorithms. And the cutting-edge metric-based MIA by Carlini et al. [5] enhances attack performance by estimating members' and non-members' prediction distribution and seeking an optimal threshold from the distribution gap. Considering both attack costs and attack effectiveness, we adopted "privacy risk score" as our primary evaluation method.

2.2 Understanding MIA

Over-fitting is predominantly deemed the root cause for MIA [24, 34, 36, 38]. Broadly speaking, machine learning models strive to understand the statistical characteristics of the training dataset, inadvertently memorizing specific sample details in the event of over-fitting. Although certain metrics exist for evaluating over-fitting extent [36, 38, 45], they are not always reliable, as highlighted in Subsection 3.2. Furthermore, Truex et al. [42] explored the performance of MIA across diverse classes and the influence of differential privacy (DP) on imbalanced data, a topic closely intertwined with this research. Their analytical insights offer a unique perspective for MIA. This work, particularly the discussion regarding "close-to-the-center" samples, enhances their findings, and introduces WS, a proposed mitigation for MIA. In summation, a comprehensive and systematic evaluation of neural network over-fitting and its extent remains, to the best of current knowledge, an unresolved inquiry in the field.

2.3 Data Distribution Impacts Privacy

Multiple studies [3, 6, 7, 15, 16] share our objective, exploring the effects of data distribution imbalance on model performance. Some identify the challenges in memorizing "long-tailed" data, leading to generalization errors and consequent issues like diminished model utility in learning.

Carlini et al. [6, 7] find that DP training yields more fragile model performance within the long-tail subpopulation. The inability of DP to memorize the tail of the mixture distribution contributes to its limited utility. This paper delves into the heterogeneous mixture in the training set, relating to each sample's vulnerability in MIA, and extends this property to the discussion on "close-to-the-center" samples in DP training. We introduce a new concept to elucidate why such long-tail samples are likely to deviate from distribution in DP learning.

Recent work by Carlini et al. introduces the "onion effect" [8], a strategy that excludes the most vulnerable ”outlier” data from the training set. Despite this, MIA remains potent, affirming that the omission of long-tail data does not alleviate privacy leakage.

Feldman [15] suggests an intuitive approach to isolate long-tail samples within a class, noting an increased susceptibility to backdoor attacks [20] in these subpopulations.

2.4 Mitigating MIA

We categorize mitigations of MIA into four categories.

Confidence score masking is a technique striving to limit the information inadvertently disclosed through the prediction vector. This is executed by either solely disclosing predictions for the top few classes [36] or incorporating noise into the prediction vector [27]. Despite the simplicity of these methods, their efficacy is proven to be suboptimal. Studies [10, 30, 34] reveal the persistent vulnerability of these techniques to MIA. Rahimian et al. [32] suggest adding noise to *Logits* during the inference stage, which does not offer resistance against MIA attacker [38].

Regularization is a traditional strategy used to diminish over-fitting and consequently, it is a viable tool for MIA mitigation. Various regularization-centered methods, including ℓ_2 norm regularization, dropout [39], early stopping [46], and label smoothing [40], have been proposed to this end. Specifically, Nasr et al. [31] have crafted an adversarial regularization technique that incorporates membership inference gain into the loss function to curb MIA. Despite these efforts, the trade-off between privacy and utility with such a method is commonly deemed unsatisfactory [24]. Recognizing this dilemma, Li et al. [29] introduced *Mixup + MMD*, employing mix-up training [29, 49] to uphold the model’s accuracy. A limitation of this approach, however, is the necessity for additional data, a requirement often unattainable in real-world scenarios. Hence, the challenge of devising a regularization method that retains model accuracy while diminishing privacy leakage remains unresolved [24].

Knowledge distillation involves the transfer of knowledge from a teacher model to a student model by training the student model using predictions from the teacher model [2, 22]. This technique serves various purposes including reducing model size, defense against adversarial examples [19], and notably, mitigating MIA. In particular, Shejwalkar and Houmansadr [35] introduced distillation for membership privacy (DMP) for MIA mitigation. This approach involves the use of unlabeled reference data, to train the student model with labels generated by the original model. By restricting direct access to the original private dataset in the target model, DMP effectively mitigates MIA. However, a notable limitation of such methods is the necessity for additional data and the commonly observed reduced accuracy of the student model compared to the original model. Recently, Tang and Mittal [41] presented a self-distillation method for training models. Despite the requirement for additional computational resources for training student models, this approach offers notable defensive advantages. Note that the essence of MIA mitigation methods based on distillation and model smoothing are fundamentally different. Similarly, data condensation [?] is the process of condensing a larger dataset into a smaller set that retains the original data information, which is then used for model training. This approach is also significantly different from our method.

Perturbing the Training Phase. Intuitively, if we add noise to the training phase such that the membership of a training sample is protected, MIA can be mitigated theoretically. The guarantee of differential privacy (DP) [12] fits the goal of MIA mitigation. The potential of utilizing DP to counteract MIA was first highlighted by Shokri et al. [36]. Subsequent evaluations by Rahman et al. [33], and Jayaraman and Evans [26], underscore the unfavorable privacy-utility trade-off observed in diverse DP training scenarios. In this paper, we focus on mitigating MIA via model smoothing during the training phase.

3 OBSERVATIONAL STUDY

Over-fitting is acknowledged as a significant factor contributing to MIA [24, 34, 36, 38, 42]. Despite this recognition, a granular analysis of individual samples remains absent. This section examines the distribution of prediction vectors across various training samples.

3.1 Not All Samples Are Equal

Our initial observation highlights the unequal vulnerability of training set samples to MIA. This assertion is validated using the Mentr score, a metric demonstrating a strong correlation with a sample’s susceptibility to MIA [38].

$$\mathrm{Mentr}(x,\ell) = -(1-f_\ell(x,\theta))\log f_\ell(x,\theta) - \sum_{i\neq\ell} f_i(x,\theta)\log(1-f_i(x,\theta))$$

where $f(\theta)$ is a model parameterized with θ that outputs a prediction vector on any input x with true label ℓ, and $f_i(x,\theta)$ denotes the predicted likelihood for the i-th label. The intuition of Mentr is to score a sample higher if $f(\theta)$ is more confident (i.e., when $f_\ell(x,\theta)$ is approaching 1, the first part is close to 0, while the second part is large). It has been empirically shown that a sample’s Mentr value is closely correlated to its risk of being attacked by MIA [38].

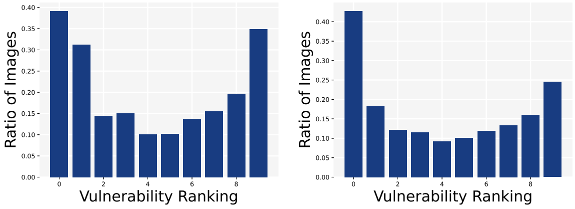


Figure 1: MIA Risk Distribution: Training samples are arranged in ascending order of MIA risk, from highest to lowest, and subsequently segmented into ten groups. Figures on the left and right correspond to data generated on CIFAR-10 and CASIA-FACE respectively.

We conduct an empirical study to evaluate the risk of MIA across various classes and samples, utilizing prominent network models such as DenseNet [25] and ResNet [21] trained on standard benchmark datasets, CIFAR-10 [13] and CASIA-FACE [14]. To minimize the influence of randomness, 50 models are trained for each dataset and network. Additional details regarding the experimental setup can be found in Section 5.

Figure 1 plots the percentage of samples that are always ranked within a certain range of percentiles. Here we can see that, among the 50 models, the Mentr values for each sample are pretty stable. For example, approximately 43% (resp. 37%) of identical training samples consistently rank among the top 10% of risky samples for the CASIA (resp. CIFAR) dataset. These samples are always more vulnerable to MIAs.

We extend our analysis to examine the variability of MIA risk across different classes. Figure 2 displays the average vulnerability value (expressed as $-\log[\mathrm{Mentr}(x,\ell)]$ for enhanced interpretability) for all samples within a class. We note significant variability in vulnerability across various classes. Moreover, this distribution of vulnerability remains consistent across model architectures, as evidenced by a Pearson’s correlation coefficient of 0.8612 for CASIA and 0.8110 for CIFAR-10.

Additionally, we notice that similar samples consistently exhibit a higher MIA risk within the same class. Figure 3 shows some representative images from a class with a high vulnerability value in the CASIA-FACE dataset. The top row shows 5 faces with high vulnerability values, and the bottom row shows 5 less vulnerable faces from the same class. We can observe that the faces in the top row look at the camera directly. On the contrary, faces in the bottom row have more diversity (e.g., often show a side face, lower head, or exaggerated facial expression). The same observation is

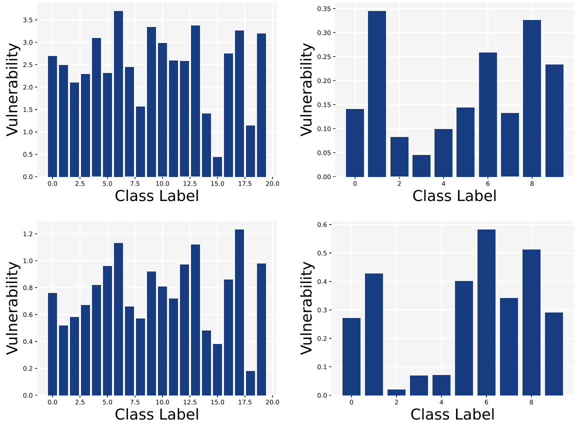


Figure 2: The first row shows the average vulnerability of each class in CASIA (20 classes) and CIFAR-10 (10 classes) on DenseNet, and the second row shows that of ResNet.

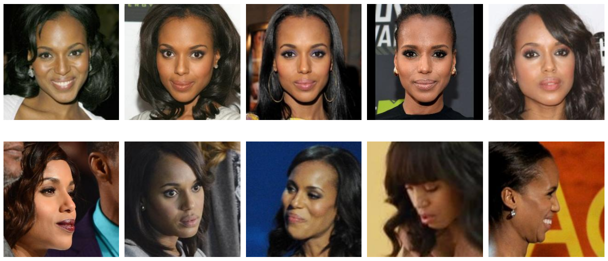


Figure 3: Examples of samples with similar features (top row) and samples that look different (bottom row) within a class.

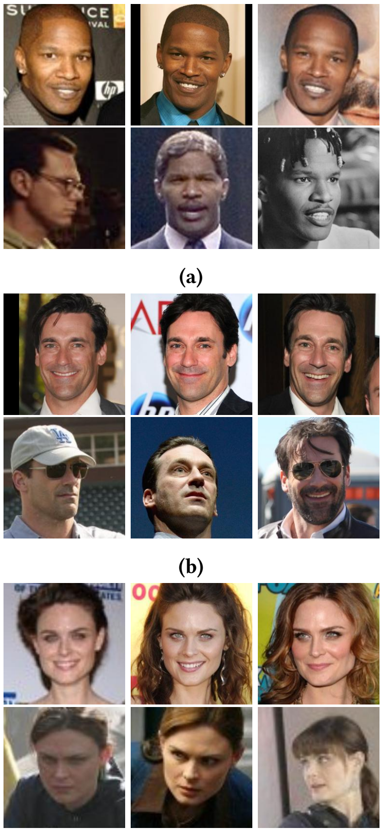
made with other classes in Figure 4. These similar samples, with lower Mentr value, are conceptually 'clustered', making them more susceptible to MIA.

3.2 Discussion

It is crucial to underscore the limitations of our analysis: (1) The use of Mentr as a vulnerability metric, limits our analysis to MIAs using Mentr. We claim that Mentr represents state-of-the-art, and we expect similar phenomenon with other metrics and even other neural network-based MIAs. (2) We exclusively focus on *training* samples. We anticipate a comparable impact will also show from the testing samples, if they follow the same distribution as training samples. (3) Our analysis is only applicable to *offline* MIAs and not applicable to the *online* setting [5], where MIA is specialized for each data sample. The intuition leveraged in the online MIA is the confidence *difference* between scenarios of whether the sample was used for training or not, rather than the *absolute confident* leveraged in offline MIA. The computation cost for online MIA is significantly higher; in this paper, we focus on offline MIA.

4 WEIGHTED SMOOTHING

In the previous section, the elevated risk of MIA attacks on similar, clustered samples was established. To mitigate this risk, it is important to diversify these samples. We choose to achieve this by infusing perturbations during the training phase. Moreover, the greater the assessed risk, the more perturbations we can add. As a by-product, demonstrated by subsequent experiments, this approach occasionally enhances the model's generality and test accuracy.



(a)

(b)

(c)

Figure 4: More examples in other classes of the CASIA dataset. Top row in each subfigure demonstrates similar samples having higher MIA risks, while bottom rows demonstrate dissimilar samples having lower MIA risks.

4.1 The WS Algorithm

Detailed elaboration of our method is provided in Algorithm 1. We first initialize a model by training on the entire dataset for a few epochs and obtain θ_α. Then, for each epoch t and a sample x, we add Gaussian noise $\mathcal{N}(0, \sigma^2 \mathrm{I})$ to prediction vector $f'(x, \theta_t)$, where σ represents its standard deviation, and I is the identity matrix, and follow the remaining steps in normal training.

There are various places to add perturbations during the model training process, like adding noise after any layer in the forward pass. We add Gaussian noise to the prediction, mainly because our tests, shown later in Figure 13 and Subsection 6.1, confirm that this works best. But the WS framework can be versatile and allow adding perturbations to other layers.

Algorithm 1: Weighted Smoothing in Prediction

```
Inputs: training set $\{x_1, \cdots, x_N\}$, labels $\{\ell_1, \cdots, \ell_N\}$,
  Gaussian noise parameter $\sigma$, training epochs $\alpha, T$;
train the model $f(\theta)$ normally for $\alpha$ epochs and get $\theta_\alpha$
for epoch $t$ from $\alpha$ to $T$do
    calculate $w_i = \mathsf{Mentr}(x_i, \ell_i)$ on model $f(\theta_t)$ for all $i$;
    for each class $c$ do
        compute mean $\mu_c$ and standard deviation $\sigma_c$ in class $c$
        normalize $w_i = 1 - (w_i - \mu_c)/\sigma_c$ for all $i$ where $\ell_i = c$
    for each batch do
        for each sample $x_i$ do
            feed-forward up to $f'(x_i, \theta_t)$
            add noise $w_i \cdot \mathcal{N}(0, \sigma^2 \mathrm{I})$ to $f'(x_i, \theta_t)$
        back-propagation to compute $\theta_t$
Return $\theta_T$;
```

In cases where a sample is more susceptible to MIA, the noise amount is increased. This is achieved by multiplying the Gaussian noise with an additional factor $1 - w$. Here, w is the normalized Mentr score of sample x, which within each class is adjusted to zero mean and unit variance.

Note that we adopted Mentr as our method for calculating weights because it is directly related to MIAs (and our goal is to mitigate MIAs). We chose to normalize Mentr values as weights because Mentr values often span a wide range, which are not directly suitable to be used as noise weights. Additionally, we choose to normalize within each class rather than across the entire training data to ensure that data from each class receives an equal degree of privacy protection (which may not be always necessary, and in such cases, we can do normalization across the whole training dataset).

4.2 Comparison with DP

This concept of noise incorporation aligns with the established privacy notion of differential privacy (DP) [1, 12]. The prevailing consensus within the DP community underscores the efficacy of adding perturbations during intermediate/gradient steps (as exemplified by the DP-SGD approach [1]). To the best of our knowledge, Du et al. [11] is the only work that advocates adding noise in the forward pass. Yet it focuses more on the larger models. Our strategy also involves noise integration during training. A notable distinction lies in that instead of unifying noise for all samples, in our approach, the amount of noise depends on each sample's individual MIA risk. This data-dependent approach inherently contradicts the idea of DP. Therefore, while we make a comparison with DP, we do not seek to derive a DP bound for it.

Specifically, given a model with parameter θ, let g_t represent the clipped gradient at time t. The DP gradient perturbation incorporates noise into the cumulative gradient of all samples,

$$\frac{1}{b}\sum_{i=0}^{b} g_t(x_i, \theta) + \mathcal{N}(0, \sigma^2 C^2 \mathrm{I})$$

where b to denote the size of a randomly sampled batch, and $\mathcal{N}(0, \sigma^2 C^2 \mathrm{I})$ to represent Gaussian noise with a per-sample clipping norm C. On the other hand, our WS integrates varying noise into the prediction, denoted by $f'(x_i, \theta_t)$ of each individual sample:

$$\frac{1}{b}\sum_{i=0}^{b} \left[f'(x_i, \theta_t) + w_i \mathcal{N}(0, \sigma^2 \mathrm{I}) \right]$$

It is evident that while DP introduces substantial noise to the collective gradient of the entire group, WS selectively injects noise to each sample, based on w_i, considering its MIA risk.

Intuitively, escalating noise magnitude potentially "pulls" the sample further from its initial position in the embedding space. DP-based methodologies indiscriminately apply uniform noise to every sample, potentially causing over-adjustment and consequently reducing model accuracy. Conversely, our strategy selectively "pulls" high MIA risk samples (i.e., those close to clusters), leaving low-risk samples largely unaffected, thereby preserving model accuracy.

5 EXPERIMENTAL SETUP

5.1 Datasets

Experiments are conducted utilizing four computer vision datasets: CIFAR-10, CIFAR-100, CASIA-Webface, and HAM10000, along with two tabular datasets: Location-30 and Texas-100. Many of recent works [27, 30, 34, 36, 37] use these datasets.

CIFAR-10, a standard dataset for image recognition tasks, encompasses 50,000 training images and 10,000 test images, each with a dimension of 32×32×3 pixels. It contains ten classes, each hosting approximately 6,000 images.

CIFAR-100 is composed of 100 classes, each containing 500 training images and 100 test images, totaling 50,000 and 10,000 images, respectively, identical in format to CIFAR-10.

CASIA-Face, harvested from the web, incorporates 494,414 facial images of 10,575 celebrities [14]. Each image, formatted as $250 \times 250 \times 3$, is categorized under the 20 most populated classes, summing up to a total of 12,440 images for our experiments.

HAM10000 is a repository of 10,000 dermatoscopic images, spanning a spectrum of skin conditions from benign keratoses and nevi to malignant melanomas [43]. This extensive image diversity aids in the crafting and training of machine-learning models targeting early skin cancer detection and diagnosis. We randomly selected 8,000 images for our evaluation.

Since NN-based MIA requires the same number of data as the training set when training the shadow models [36], there are no sufficient training data on HAM10000 to train the shadow models. Even if we only select 5,000 training samples, we still can't train shadow models that meet the requirements, because the data used to train each shadow model is often not exactly the same. Therefore, we only focus on metric-based MIA for this dataset.

Location-30, derived from a public mobile user's location "check-ins" dataset in the Foursquare social network[1], encompasses 5,010 user profiles, each delineated by 446 binary features. These features signify specific user visits to distinct locations. The dataset is clustered into 30 classes, denoting diverse geosocial types [36].

Texas-100, sourced from public Hospital Discharge Data documents, contains records detailing attributes such as external causes

[1] https://sites.google.com/site/yangdingqi/home/foursquare-dataset

Model	CIFAR-10	CIFAR-100	CASIA	HAM10000
DenseNet	60.76% (100.0%)	36.41% (100.0%)	72.17% (100.0%)	80.62% (100.0%)
ResNet	59.55% (100.0%)	34.07% (100.0%)	78.35% (100.0%)	80.92% (100.0%)

Table 1: Baseline validation (training) accuracy.

Model	Attack Method	CIFAR-10	CIFAR-100	CASIA	HAM10000
DenseNet	NN based	74.64%	75.90%	67.69%	-
	Metric based	80.98%	86.50%	72.26%	61.45%
ResNet	NN based	75.03%	82.65%	74.28%	-
	Metric based	86.52%	90.54%	80.72%	66.01%

Table 2: Baseline MIA success rates for neural network and metric-based attacks. The HAM10000 dataset is small, so we do not conduct neural network-based MIA on it.

of injury, diagnosis, undergone procedures, and generic information like gender, age, race, and hospital ID. Adhering to established methodologies in MIA [27, 36, 38], a simplified and preprocessed Texas dataset is employed, encompassing 67,330 samples, grouped into 100 classes, with each sample featuring 6,170 binary attributes [36].

5.2 Target Models

For the image datasets, we employ two widely recognized convolutional neural network architectures: DenseNet [25] and ResNet [21]. Table 1 delineates the baseline accuracy for both models across these datasets, presenting both validation and training accuracy for each entry. The different data formats within the CASIA dataset are accommodated by adjusting model hyper-parameters without altering the architecture. To maintain the focus on MIA effectiveness, techniques such as dropout [39] and regularization [31] are not utilized, and models are trained using 200 epochs without early stopping.

Concerning the tabular datasets, a neural network structure following the one outlined in [27] is adopted. The classification model consists of a fully connected neural network encompassing four hidden layers with 1024, 512, 256, and 128 neurons, respectively. Texas100 and Location30 use output layers with 100 and 30 neurons, respectively. The hidden layers employ ReLU activation function and group normalization, while the output layer utilizes the softmax function. The model undergoes training for 200 epochs at a learning rate of 0.075, utilizing Adam for optimization. The learning rate experiences a decay of 0.99 after each epoch.

5.3 Membership Inference Attacks

To assess our method against various MIA techniques, both neural network-based [28] and metric-based [38] strategies are considered. It is presumed that adversaries are potent, creating a shadow model identical to the target model and accessing the same data distribution as the training set. The shadow models' outputs serve to train the attacking model, ensuring data used for training shadow and target models remain disjoint.

Neural network-based MIA. Generally, the efficiency of neural network-based MIA significantly hinges on the shadow models' quantity and quality [36]. Consequently, 30, 50, and 40 shadow models (equally divided between DenseNet and ResNet) are trained on CIFAR-10, CIFAR-100, and CASIA-FACE, respectively, ensuring extensive coverage of the prediction distribution in alignment with [36]. The attack success rates are enumerated in Table 2. Note that due to the size of HAM10000, we do not conduct NN-based MIA on that dataset. For Location-30 and Texas-100, 40 and 80 shadow models are correspondingly trained.

Metric-based MIA. Employing Song et al.'s method [38], members and non-members are distinguished based on a sample's Mentr score. A sole shadow model is trained to pinpoint a threshold that maximizes attack accuracy on the shadow dataset, following the original work. Another cutting-edge metric-based MIA by Carlini et al. [5] enhances attack performance by estimating members' and non-members' prediction distribution and seeking an optimal threshold from the distribution gap.

Success rates of Song et al. are displayed in Table 2, highlighting anticipated elevated success rates across all image datasets and models [38]. Carlini et al.'s success rates on two models are presented in Table 3 (first row, WoDef), showing relatively diminished rates compared to Song et al., attributed to our use of the offline attack setting. Offline setting requires 256 shadow models, and it requires a significant increase in online setting to find a more precise Gaussian distribution for each sample. Thus we mainly apply Mentr in our evaluation and also use Carlini et al. [5] in a few cases.

Dataset Settings in MIA. Initially, samples are allocated for various datasets as target training sets: 5,000 for CIFAR-10, 20,000 for CIFAR-100, 6,220 for CASIA, 8,000 for HAM10000, 1,000 for Location-30, and 5,000 for Texas-100. An equivalent quantity is earmarked as the target model testing set (i.e., target non-members), with the exception of CIFAR-100, which, due to dataset size constraints, is limited to 10,000 samples. For CIFAR-10, a training sample size of 5,000 is adopted, aligning with established research practices [29, 34, 36, 38]. This choice is informed by observations that an inflated sample size, for instance, 40,000, tends to depress MIA accuracy to around 55%, rendering both MIA attack and defense more difficult. Subsequently, shadow models are trained from the remaining dataset, mirroring the conditions observed with CIFAR-100, Location-30, and Texas-100 datasets. For the CASIA dataset, 6,220 images are utilized due to category-specific image limitations. The top 40 categories, averaging approximately 350 images each, guide the allocation of 311 images to each category within the training set to ensure adequate training data for the shadow model.

For the enhancement of attack efficacy, it is assumed that the training/testing set of the shadow model mirrors the size of the target model. In instances of abundant data, as with CIFAR-10, random sampling is employed for shadow model datasets.

5.4 MIA Mitigation

Multiple MIA mitigation strategies have been explored in the literature [1, 27, 31, 35, 36, 40, 49]. This work primarily benchmarks against various privacy methods delineated in [1], collectively termed the DP method henceforth. Implementation is conducted utilizing PyTorch Opacus [47], with δ configured at 2×10^{-5}. The optimization technique employed is SGD, and other default hyper-parameter settings [47]. Note that since WS does not provide a formal guarantee, for a fair comparison, we do not try to derive a privacy guarantee for DP. The goal of DP is to provide a baseline that adds homogeneous noise, compared to WS that adds data-dependent noise.

Within the scope of our WS, the Adam optimization algorithm is harnessed. Training spans 220, 180, 180, 120 epochs for four image datasets and 200 for both Tabular datasets. Weight initialization involves preliminary training of the model for 2, and 1 epoch, for image and tabular datasets, prior to the application of smoothing. More hyper-parameter settings, such as initial learning rates along with exponential decay rates, are given in our repository[2].

5.5 Metrics

Following Jayaraman and Evans [26], we measure the performance of MIA mitigation methods using two metrics, i.e., *privacy leakage*, defined as the success rate of MIA minus 0.5, and *accuracy loss*, defined as the accuracy difference normalized by the accuracy of the original model.

6 EVALUATION

6.1 Effectiveness of WS

6.1.1 WS *Makes* Mentr *Distributions Overlap.*

At a high level, our proposed approach involves adjusting the distribution of prediction vectors for training samples to closely align with that of the testing/validation samples. This adjustment renders the two distributions nearly indistinguishable to MIA, including both neural network-based [36] and metric-based [38] attacks, thereby effectively mitigating such attacks. To visually comprehend the action of WS in altering the distribution, Figure 5 delineates the contrast in the distribution of the Mentr value between training and testing samples of a DenseNet model trained on the CIFAR-10, CIFAR-100, and CASIA-FACE datasets in the three rows, respectively, with (right column) and without (left column) the application of WS. The x-axis represents the Mentr value (expressed in negative log scale) and the y-axis denotes the probability of such values. In the absence of mitigation methods, two distributions exhibit considerable separation, intuitively signifying substantial vulnerability to MIA. The incorporation of WS noticeably reduces this separation, aligning the two distributions more closely.

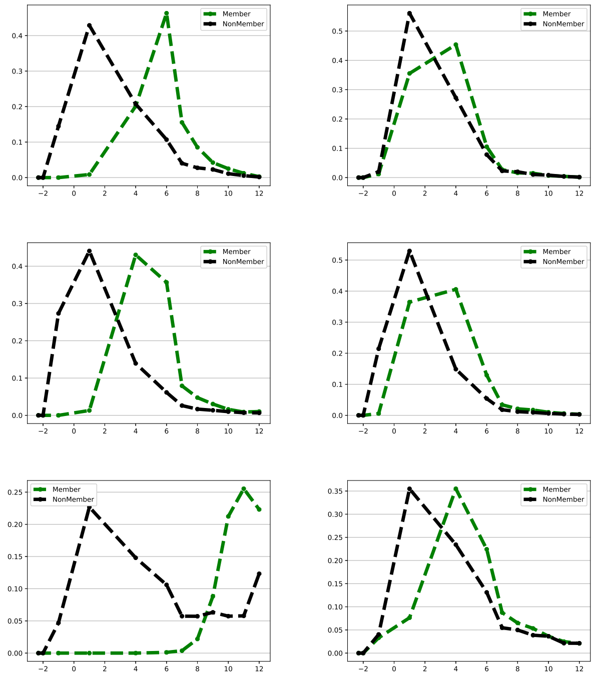


Figure 5: Distribution of vulnerability scores (negative log Mentr**) for members and non-members (DenseNet on CIFAR-10, CIFAR-100, and CASIA-FACE in three rows, respectively). Left is through normal training and an MIA attack can differentiate members and non-members easily. When the model is trained with** WS**, the two distributions overlap a lot and thus MIA is mitigated.**

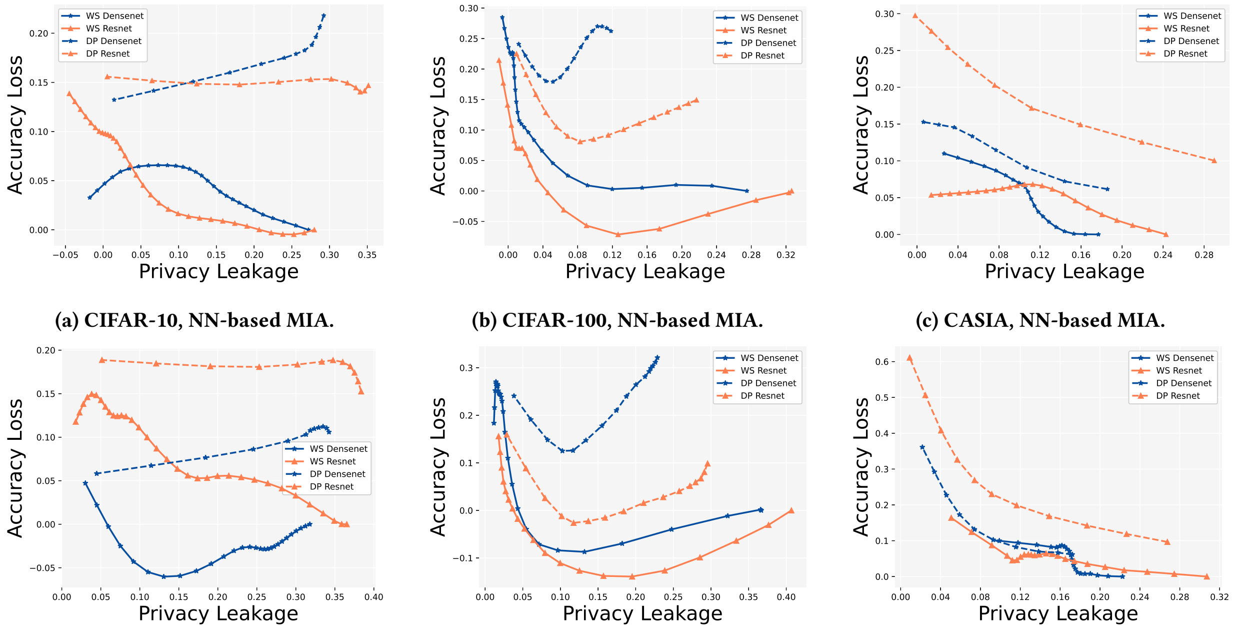
		DenseNet	ResNet
WoDef	Accuracy	60.74%	60.14%
	ASR	73.24%	73.16%
DP	Accuracy	50.82%	50.02%
	ASR	56.44%	54.58%
WS	Accuracy	52.47%	56.65%
	ASR	55.66%	55.21%

Table 3: Model prediction accuracy and attack success rate (ASR) measured by Carlini et al. [5] for models without defense (WoDef), WS **and DP-SGD on CIFAR-10. For** WS **and DP, for comparison, noise was selected so that ASR is close to** 55%**.**

6.1.2 *Overall Comparison between DP and* WS*.*

To systematically evaluate the privacy-accuracy trade-off, both WS and DP with different noise scales are employed for comparison. Results are

[2] https://github.com/BennyTMT/weighted-smoothing



(a) CIFAR-10, NN-based MIA.

(b) CIFAR-100, NN-based MIA.

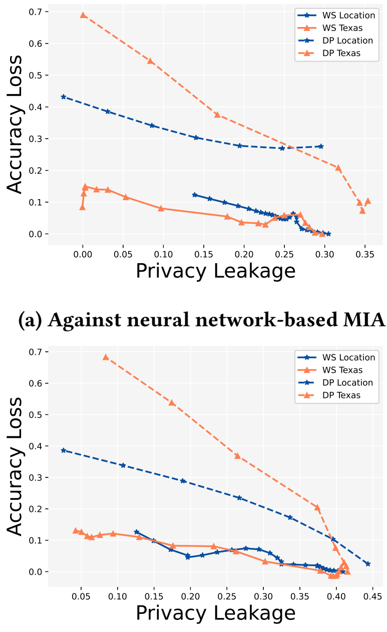
(c) CASIA, NN-based MIA.

(d) CIFAR-10, metric-based MIA.

(e) CIFAR-100, metric-based MIA.

(f) CASIA, metric-based MIA.

Figure 6: Accuracy loss over privacy leakage for different models and MIA attacks.



(a) Against neural network-based MIA.

(b) Against metric-based MIA.

Figure 7: Location30 and Texas100 accuracy loss over privacy leakage for different models and MIA attacks. Each subfigure plots both datasets trained with the same model.

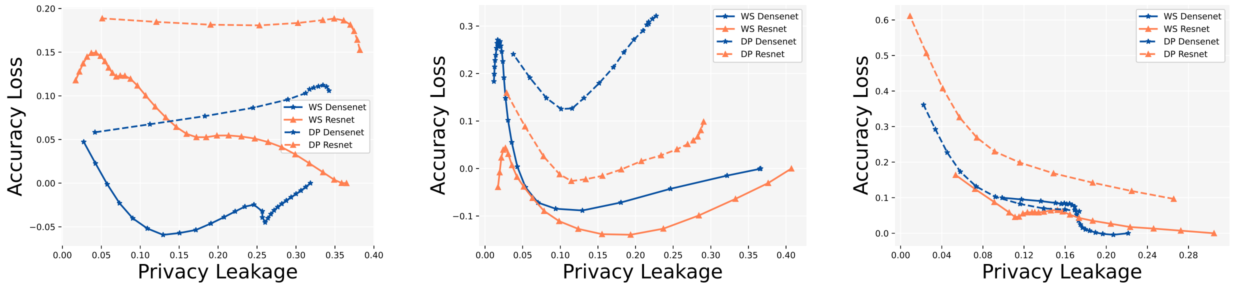
illustrated in Figure 6 and Figure 7. In those figures, each point represents an independent training with different noise scales (and neighboring points are connected); the x-axis represents privacy leakage, computed using the success rate of the corresponding MIA method minus 0.5, and the y-axis as the normalized accuracy loss (using the metrics mentioned in Subsection 5.5). We also conduct corresponding experiments on metric-based MIA based on *confidence score* [34, 36], i.e., determining the threshold based on the maximum value of the prediction vector. Results are shown in Figure 8 and Figure 9.

Comparing solid lines (our method) and dashed lines (the DP baseline), our approach almost always achieves a better privacy-utility tradeoff (in the figures, the solid lines are always lower than the dashed lines), with the only exception in CASIA trained with Densenet (a few points of our method is slightly below that of DP baseline). For the HAM10000 dataset, as shown in Figure 10, both WS and DP effectively mitigate metric-based MIA, but WS does not show a clear advantage.

6.1.3 *Detailed Results.*

We've also looked into the accuracy loss specific to each class. As indicated in Figure 11a and Figure 11b, the accuracy for each class without any MIA defense is higher than when we use WS. Our goal with WS is to bring the metric-based MIA accuracy close to random guessing, which is around 52.10% and 52.8%.

We further scrutinize the interrelation between escalating noise levels and the consequential accuracy loss, as graphically represented in Figure 11c. Noted trends are the byproduct of multiple factors, including dataset complexity, model intricacy, and the impact of noise. Even under identical noise conditions, there are observable



(a) CIFAR-10. **(b) CIFAR-100.** **(c) CASIA.**

Figure 8: Accuracy loss over privacy leakage on metric-based MIAs using confidence scores.

fluctuations in the accuracy, with standard deviations being 0.0146 and 0.0088 for Densenet and Resnet, respectively. Overall, we see an accuracy decline with the amplification of noise levels.

Extending our exploration, a comprehensive analysis across each class for diverse model architectures is undertaken. This examination revealed a variance in MIA performance for individual classes across different models, as illustrated in Figure 12. The MIA accuracy demonstrates disparity across models, a trend that holds irrespective of the application of our WS defense. Intriguingly, the correlation coefficient for MIA accuracy across each CIFAR10 class between DenseNet and ResNet is recorded at a minimal 0.2913 without defense. This observation intimates that a class perceived as secure against MIA within one model may not retain this security within an alternate model.

6.1.4 High-level summary. In light of the preceding results, one takeaway is that a mitigation approach introduces a balance between accuracy loss and privacy enhancement. Our WS improves the Pareto curve of this tradeoff.

Here, we identify several special cases, where both WS and DP are employed to reduce the attack success rate of either neural network-based MIA or metric-based MIA to a predetermined desired level (i.e., increasing the noise level of WS and DP until the MIA method's success rate approximates the desired level), and we measure the accuracy loss. For example, for Densenet, the accuracy reduction is mere 0.95% on both metric-based [38] and neural-network-based [36] attacks, and a significant improvement over the baseline MIA mitigation's accuracy loss of 16.11%. The results are presented in Table 4.

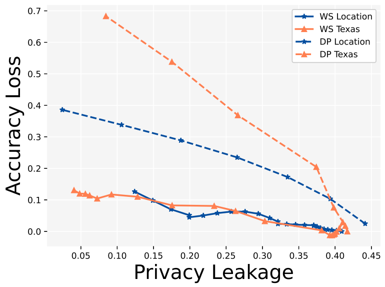


Figure 9: Accuracy loss over privacy leakage for the tabular datasets on metric-based MIAs using confidence scores.

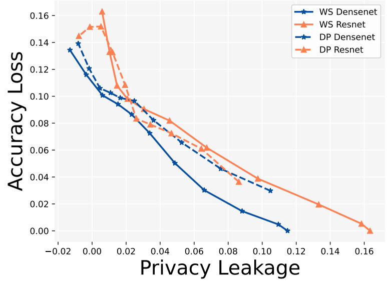
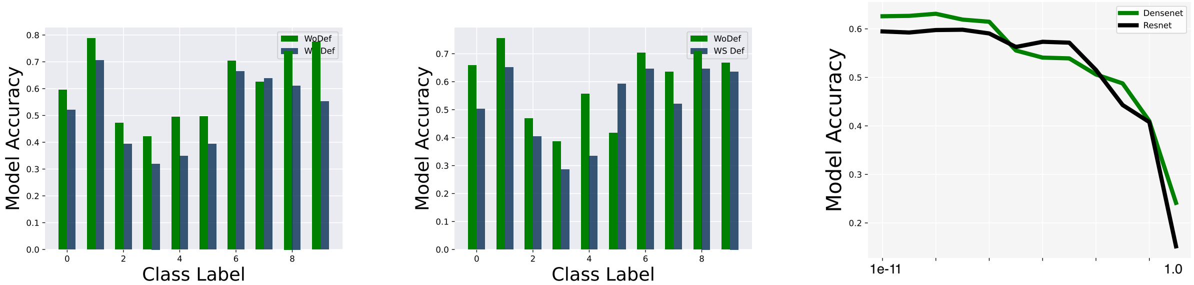


Figure 10: The trade-off of HAM10000 dataset in Densenet and Resnet models on metric-based MIA.

DataSet		AL	SR	AL	SR
		Metric-based		**NN-based**	
CIFAR-10	WS	0.98%	4.71%	5.44%	0.00%
	DP	15.17%	4.47%	15.17%	1.46%
CIFAR-100	WS	-2.74%	3.21%	-18.92%	1.43%
	DP	22.80%	3.82%	8.95%	1.45%
CASIA-face	WS	9.98%	9.89%	10.99%	2.64%
	DP	10.39%	6.66%	24.23%	1.29%
		Metric-based		**NN-based**	
CIFAR-10	WS	8.54%	6.20%	3.22%	3.37%
	DP	18.86%	5.14%	18.86%	0.55%
CIFAR-100	WS	-10.92%	3.91%	-14.61%	1.99%
	DP	22.49%	2.86%	4.42 %	2.54%
CASIA-face	WS	11.24%	9.56%	5.24%	1.85%
	DP	23.51%	7.46%	17.88%	1.82%

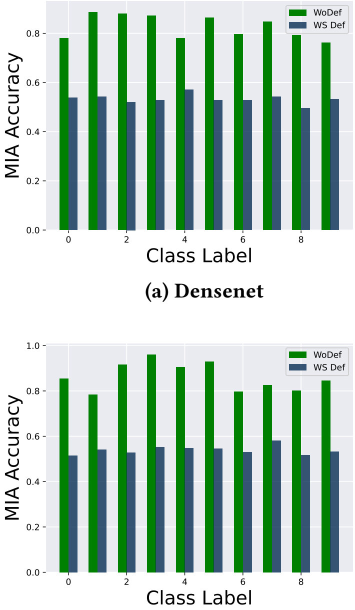
Table 4: Accuracy loss (AL) and success rate (SR) of MIA. The top half is DenseNet and the bottom half is ResNet.

Observational analysis reveals that for both neural network-based MIA and metric-based MIA, the collective cost of employing DP vastly overshadows that of WS. This underscores the substantial accuracy compromise DP incurs for analogous privacy fortification. Concurrently, the recent assault postulated by Carlini et al. [5] is



(a) Class Accuracy of in Densenet. **(b) Class Accuracy of in Resnet.** **(c) Overall accuracy with noise scale.**

Figure 11: Detailed measurement of each classes in CIFAR-10 about WS.



(a) Densenet

(b) Resnet

Figure 12: Detailed metric-based MIA accuracy for each class.

evinced to be more susceptible compared to the previously mentioned attacks.

Subsequent experimentation insinuates the relative ease of safeguarding against neural network-based MIA compared to its metric-based counterpart. Specifically employing WS, the accuracy losses for CASIA-FACE, CIFAR-100, and CIFAR-10 for neural network-based MIA stand at 8.11%, −16.76%, and 4.33% respectively. In contrast, the corresponding losses for metric-based MIA are recorded at 10.61%, −6.83%, and 4.76%. Analogously, employing DP exhibits a more significant average accuracy loss against metric-based MIA as opposed to neural network-based MIA. This observation highlights the augmented challenge presented by the defense against the metric-based MIA delineated in [38]. Notably, in instances showcased in Table 4, a negative accuracy loss denotes a slight enhancement in model accuracy.

6.2 Ablation Study

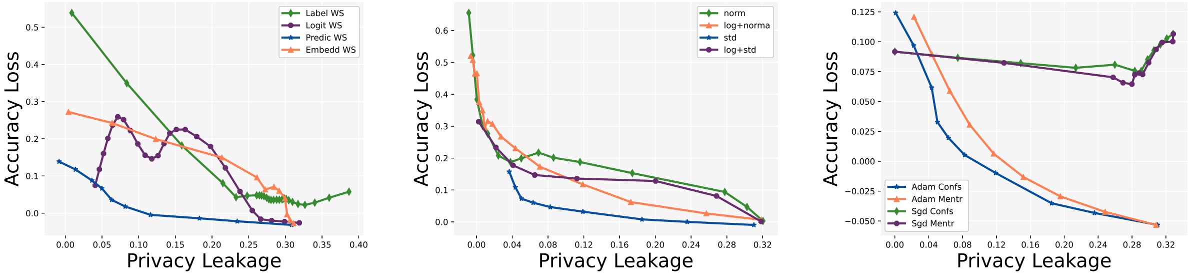
Following the discussion in Section 4, employing smoothing at various positions may yield comparable outcomes. Despite this, the evaluation is extended across four smoothing locations (layers of a model), namely Prediction, Logit, Label, and Embedding. Here, Logit is the vector prior to the Softmax operation, Embedding is the vector preceding the fully connected layer, and Label denotes the one-hot ground truth. As illustrated in Figure 13a, most scenarios observe better privacy-utility trade-off when smoothing is applied to the prediction. Additionally, in Figure 13b, we also evaluated different normalization weight methods on WS, including the weight standardization (std) outlined in Algorithm 1, standardization post logarithm operation (log+std), normalization (norm), and normalization following a logarithm operation (log+norm). Evidenced in Figure 13b, standardization emerges as the most balanced approach in the context of model smoothing. Note that the difference between standardization and normalization is that they subtract the minimum value or mean, respectively, and then divide by the maximum value or standard deviation, respectively.

Figure 13c shows the comparison between DP-SGD and DP-ADAM. The evaluation of DP-ADAM on the CIFAR10 dataset, employing both Densenet [25] and Resnet [21], reveals worse performance compared to DP-SGD.

To validate the performance of WS from different measurements, and its applicability beyond Mentr, WS is assessed utilizing another metric-based MIA, using entropy as the metric [34]. The obtained results, as shown in Figure 15, exhibit consistent performance across both Densenet and Resnet models on the CIFAR10 dataset.

7 CONCLUSION

In this work, our observations highlight the heterogeneous behavior of samples within deep neural networks. Specifically, samples located "close-to-the-center" demonstrate heightened vulnerability to membership inference attacks. In light of these findings, we introduce WS, a novel approach that judiciously infuses noise into training samples during the training phase. Our empirical evaluations reveal that, in contrast to existing differential privacy-based methods, WS adeptly minimizes the risk of MIA, while maintaining appreciable model accuracy.



(a) Position of Noise

(b) Weight of Noise

(c) Optimizer

Figure 13: Ablation Study of WS **in CIFAR-10**

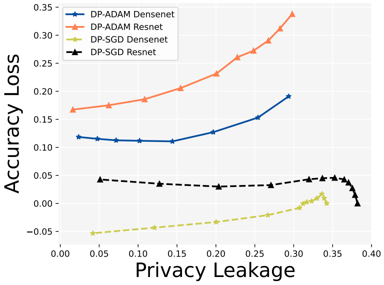


Figure 14: Comparison between DP-ADAM and DP-SGD in CIFAR-10 with Densenet and Resnet. In our settings, DP-ADAM performs worse than DP-SGD.

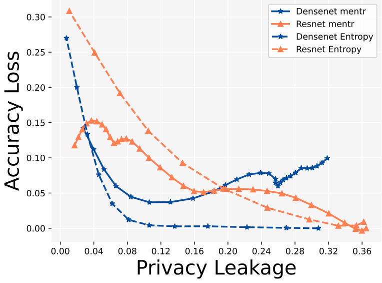


Figure 15: Comparison between metric-based MIA using Mentr **and entropy. We use** WS **on CIFAR-10.**

ACKNOWLEDGMENTS

We thank our shepherd and anonymous reviewers for their valuable suggestions. Part of the work was done when Mingtian was in SMU. Tianhao was partly supported by NSF CNS-2220433 and OAC-2319988.

REFERENCES

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

[2] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? *Advances in neural information processing systems* 27 (2014).

[3] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. 2019. Differential privacy has disparate impact on model accuracy. *Advances in neural information processing systems* 32 (2019).

[4] Mark Bun and Thomas Steinke. 2016. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*. Springer, 635–658.

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

[6] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. 2018. Prototypical examples in deep learning: Metrics, characteristics, and utility. (2018).

[7] Nicholas Carlini, Ulfar Erlingsson, and Nicolas Papernot. 2019. Distribution density, tails, and outliers in machine learning: Metrics and applications. *arXiv preprint arXiv:1910.13427* (2019).

[8] Nicholas Carlini, Matthew Jagielski, Nicolas Papernot, Andreas Terzis, Florian Tramer, and Chiyuan Zhang. 2022. The privacy onion effect: Memorization is relative. *arXiv preprint arXiv:2206.10469* (2022).

[9] Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. 2021. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*. 2633–2650.

[10] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *International Conference on Machine Learning*. PMLR, 1964–1974.

[11] Minxin Du, Xiang Yue, Sherman Chow, Tianhao Wang, Chenyu Huang, and Huan Sun. 2023. DP-Forward: Fine-tuning and Inference on Language Models with Differential Privacy in Forward Pass. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security.*

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

[13] Krizhevsky et al. 2013. CIFAR-10 and CIFAR-100 dataset and analysis. . https://www.cs.toronto.edu/~kriz/cifar.html. [Online; accessed 15-March-2022].

[14] Yi et al. 2021. CASIA-WebFace dataset introduction and source. https://paperswithcode.com/dataset/casia-webface. [Online; accessed 5-March-2022].

[15] Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. 954–959.

[16] Vitaly Feldman and Chiyuan Zhang. 2020. What neural networks memorize and why: Discovering the long tail via influence estimation. *Advances in Neural Information Processing Systems* 33 (2020), 2881–2891.

[17] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*. 1322–1333.

[18] Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. 2018. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. 619–633.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[20] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. 2019. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access* 7 (2019), 47230–47244.

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

[22] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* 2, 7 (2015).

[23] 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. 2008. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics* 4, 8 (2008), e1000167.

[24] Hongsheng Hu, Zoran Salcic, Gillian Dobbie, and Xuyun Zhang. 2021. Membership Inference Attacks on Machine Learning: A Survey. *arXiv preprint arXiv:2103.07853* (2021).

[25] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4700–4708.

[26] Bargav Jayaraman and David Evans. 2019. Evaluating differentially private machine learning in practice. In *28th {USENIX} Security Symposium ({USENIX} Security 19)*. 1895–1912.

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

[28] Bogdan Kulynych and Mohammad Yaghini. 2018. mia: A library for running membership inference attacks against ML models. https://doi.org/10.5281/zenodo.1433744

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

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

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

[32] Shadi Rahimian, Tribhuvanesh Orekondy, and Mario Fritz. 2020. Sampling attacks: Amplification of membership inference attacks by repeated queries. *arXiv preprint arXiv:2009.00395* (2020).

[33] Md Atiqur Rahman, Tanzila Rahman, Robert Laganière, Noman Mohammed, and Yang Wang. 2018. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.* 11, 1 (2018), 61–79.

[34] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2018. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[35] Virat Shejwalkar and Amir Houmansadr. 2019. Membership privacy for machine learning models through knowledge transfer. *arXiv preprint arXiv:1906.06589* (2019).

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

[37] Congzheng Song, Thomas Ristenpart, and Vitaly Shmatikov. 2017. Machine learning models that remember too much. In *Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security*. 587–601.

[38] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *30th {USENIX} Security Symposium ({USENIX} Security 21)*.

[39] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[40] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

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

[42] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Wenqi Wei, and Lei Yu. 2019. Effects of differential privacy and data skewness on membership inference vulnerability. In *2019 First IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 82–91.

[43] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. 2018. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data* 5, 1 (2018), 1–9.

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

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

[46] Xue Ying. 2019. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, Vol. 1168. IOP Publishing, 022022.

[47] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, Graham Cormode, and Ilya Mironov. 2021. Opacus: User-Friendly Differential Privacy Library in PyTorch. *arXiv preprint arXiv:2109.12298* (2021).

[48] Lei Yu, Ling Liu, Calton Pu, Mehmet Emre Gursoy, and Stacey Truex. 2019. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 332–349.

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