A Comprehensive Study of Privacy Risks in Curriculum Learning

Joann Qiongna Chen
San Diego State University
jchen27@sdsu.edu

Xinlei He
Hong Kong University of Science and Technology (Guangzhou)
xinleihe@hkust-gz.edu.cn

Zheng Li
CISPA Helmholtz Center for Information Security
zheng.li@cispa.de

Yang Zhang
CISPA Helmholtz Center for Information Security
zhang@cispa.de

Zhou Li
University of California, Irvine
zhou.li@uci.edu

Abstract

Training a machine learning model with data following a meaningful order, i.e., from easy to hard, has been proven to be effective in accelerating the training process and achieving better model performance. The key enabling technique is curriculum learning (CL), which has seen great success and has been deployed in areas like image and text classification. Yet, how CL affects the privacy of machine learning is unclear. Given that CL changes the way a model memorizes the training data, its influence on data privacy needs to be thoroughly evaluated. To fill this knowledge gap, we perform the first study and leverage membership inference attack (MIA) and attribute inference attack (AIA) as two vectors to quantify the privacy leakage caused by CL.

Our evaluation of 9 real-world datasets with attack methods (NN-based, metric-based, label-only MIA, and NN-based AIA) revealed new insights about CL. First, MIA becomes slightly more effective when CL is applied, but the impact is much more prominent to a subset of training samples ranked as difficult. Second, a model trained under CL is less vulnerable under AIA, compared to MIA. Third, the existing defense techniques like MemGuard and MixupMMD are not effective under CL. Finally, based on our insights into CL, we propose a new MIA, termed Diff-Cali, which exploits the difficulty scores for result calibration and is demonstrated to be effective against all CL methods and the normal training method. With this study, we hope to draw the community's attention to the unintended privacy risks of emerging machine-learning techniques and develop new attack benchmarks and defense solutions.

Keywords

Curriculum Learning, Membership Inference Attack

1 Introduction

Key to the success of machine learning (ML), especially deep learning (DL), is the advancement of algorithms, software, and hardware in training models on large-scale datasets. The traditional way to train a neural network (NN) is by feeding the training pipeline with random mini-batches in a sequence sampled from the training dataset. In other words, NN is forced to "remember" samples repeatedly in random order. On the other hand, human always learns the easy concepts first and then the hard ones, as guided by curricula. Given that NN is inspired by the human brain [69], curriculum learning (CL), which simulates human learning by ordering the training data with difficulty scores and repeating the order across training epochs, has been proposed [3]. With a "teacher" network, the difficult scores can be generated ahead of the samples and guide the training process. Previous studies have shown that CL can achieve both fast learning speed and high test accuracy [81, 89], and CL has been adopted in many application domains like computer vision [3, 15, 70, 80], natural language processing [3, 25, 52, 82, 101], and claiming prominent successes [89].

Despite the huge success of ML, the privacy issues of ML are becoming more and more concerning, given that the training data could contain a large amount of sensitive information. The two most notable privacy attacks are the membership inference attack (MIA) [38, 75] and the attribute inference attack (AIA) [78], where MIA aims to infer whether a given data sample is used to train the target model and AIA aims to infer the sensitive attribute of a data sample. Numerous attacks have emerged and have demonstrated that privacy threats are real (e.g., over 80% MIA accuracy against CIFAR100 [72]). Recent studies have also shown the data samples are not equally vulnerable under privacy attacks [94], and the attack effectiveness could differ across target classes [38], target individuals [55], and subgroups [7]. Yet, all previous works assume standard, stochastic training is employed by the target model. Hence, one interesting and important research problem is *how new training techniques impact privacy for the overall population and individual samples.* In this work, we specifically study the privacy risks of CL. We are particularly motivated because CL modifies the data order and repeatedly feeds the same samples, which differs from other learning techniques such as self-supervised learning [53]. In general, CL lets a model treat samples differently based on their difficulty levels[1], and we are interested in *whether CL introduces disparate impact on privacy of subgroups, aggravating "privacy unfairness" [99].* Furthermore, Shumailov et al. [76] studied the connection between data ordering and backdoor attacks, which indicates data ordering could have negative impacts. This further motivates us to investigate the privacy risks of CL.

Proceedings on Privacy Enhancing Technologies 2025(1), 613–631

https://doi.org/10.56553/popets-2025-0033

[1] The terms "difficulty level" and "difficulty score" are interchangeable.

Our Study. We take a quantitative approach to measure the privacy risks of CL. We selected two popular CL methods, bootstrapping [27] and transfer learning [91], as the evaluation objects, and constructed two other curriculum methods, named baseline curriculum and anti-curriculum, to understand the impact of data ordering and repeating, respectively. We selected 9 real-world, large-scale datasets (6 are image datasets and 3 are tabular datasets), trained target models with those CL methods and a normal method, and attacked the models with representative MIA and AIA methods.

Regarding MIA, our evaluation shows that the target models become slightly more vulnerable under CL. For example, the average attack accuracy (trained on ResNet-18 with transfer leaning) on our selected image datasets ranges from 0.01% to 2.46%. More importantly, we found CL has a much bigger impact on the samples within the difficult group compared to the easy group, with the biggest gap of 4.23% in terms of attack accuracy for CIFAR100 (ResNet-18 is the architecture). This observation sustains both image and non-image datasets. We found the reason is that the data order influences the learning process in a way that makes the model memorize difficult samples better, which is supported by measuring the memorization scores. Regarding AIA, we found CL does not increase the attack accuracy, which can be explained by the fact that the sensitive attribute to be inferred is not influenced by data ordering and repeating.

In addition to understanding the attacks, we also study existing defenses under the CL settings, including MemGuard [38], Mixup-MMD [48] and AdvReg [62]. The result shows that none of them can mitigate the threats from MIA, especially when CL is used to train the target models. Though DP-SGD [1] is another important defense, we found it cannot be applied to the CL settings, as CL breaks the DP guarantee due to data ordering and repeating.

Inspired by CL and a recent MIA that calibrates membership scores to achieve better attack accuracy [90], we consider the difficulty score as input for calibration and proposed a new MIA method, named Diff-Cali (difficulty calibrated MIA). Our attack not only brings the difficult samples to a more vulnerable stage but also achieves a higher true-positive rate at low false-positive rate regions. With this study, we hope to draw more attention to the privacy risks introduced by the new learning techniques and motivate the development of new protection mechanisms.

Contributions. The contributions of this work are summarized as follows.

- We take the first step to understanding the privacy risks introduced by CL.
- We conduct a comprehensive analysis to quantify the privacy risks and our results show CL introduces disparate impacts to samples separated by difficulty levels.
- We propose a new MIA that exploits the difficulty scores for better attack performance.

2 Preliminary

2.1 Curriculum Learning

Curriculum learning (CL) [3] is designed to emulate the concept of the human learning process. The general idea is to have a *curriculum* that imposes a structure on the training data so the "student" ML models can learn from the easier samples before the harder ones. As a result, training ML models under CL observes a shorter duration of convergence and higher testing accuracy [3, 24, 27, 91]. For example, Weinshall et al. proposed to use transfer learning to build the curriculum and achieved 0.5% to 3.5% higher accuracy than a model trained in the normal setting [91]. CL has gained significant interest from the ML community, powering real-world applications in many domains. Section 7 provides a more detailed survey.

Below, we formalize CL following the definition of Hacohen et al. [27]. Let $\mathcal{X} = \{X_i\}_{i=1}^{N} = \{(x_i, y_i)\}_{i=1}^{N}$ be the training dataset, where N is the number of samples, x_i is a data point, and y_i is the label of x_i. T is the ML model to be trained. Assuming Stochastic Gradient Descent (SGD) is used for optimization, and each training iteration takes a mini-batch of $\mathcal{X}$, and a sequence of M mini-batches $\mathcal{B}_1, \ldots, \mathcal{B}_M$ will be used for each epoch. The standard training procedure will sample $\mathcal{X}$ uniformly to generate the mini-batches. Instead, CL uses a *difficulty measurer* $f(\mathcal{X}, C)$ to generate difficult scores for $\mathcal{X}$, and a *training scheduler* sorts $\mathcal{X}$ by the difficult scores in an ascending order ahead of training. C is the curriculum, and we will elaborate on its common options in Section 4.1. A sequence of subsets $\mathcal{X}'_1, \ldots, \mathcal{X}'_M \subseteq \mathcal{X}$ are extracted from $\mathcal{X}$ after sorting, and the size of $\mathcal{X}'{}_i$ is determined by a *pacing function* $g(i)$. A mini-batch $\mathcal{B}_i$ is sampled uniformly from $\mathcal{X}'_i$. Algorithm 1 summarizes the process. Noticeably, slight changes can be applied (e.g., skip the step of mini-batch sampling), but they should not affect the conclusions drawn from this study.

Algorithm 1: Curriculum learning framework.

```
Input: Training dataset $\mathcal{X} = \{\mathbf{X_i}\}_{\mathbf{i=1}}^{\mathbf{N}}$, difficulty measurer
       $\mathbf{f(\mathcal{X}, C)}$, pacing function $\mathbf{g(i)}$, number of iterations $\mathbf{M}$,
       number of epochs $\mathbf{E}$, target model $\mathbf{T}$
$\mathcal{X} \leftarrow f(\mathcal{X}, C)$;
for $e \in 1, \ldots, E$ do
    for $i \in 1, \ldots, M$ do
        $\mathcal{X}'_i \leftarrow \mathcal{X}[1, \ldots, g(i)]$;
        $\mathcal{B}_i \leftarrow sample(\mathcal{X}'_i)$;
        $T \leftarrow train(T, \mathcal{B}_i)$
```

2.2 Privacy Risks in Machine Learning

Prior works have shown that the ML models could memorize sensitive information from the training data, which can be inferred by an adversary who keeps querying the model. Two major types of attacks are MIA [62, 63, 72, 75] and AIA [59, 78], which have been extensively studied. The detailed literature survey of privacy attacks and other attacks is left to Section 7.

Membership Inference Attack (MIA). Given a target model T and any adversary's external knowledge K, the goal of MIA is to determine whether a data sample x was used to train the model. Formally, we have:

$$\mathcal{A}_{MI} : x, T, K \mapsto 1 \text{ or } 0 \tag{1}$$

where T is the target model and K is the adversary's external knowledge, e.g., the distribution of the training data for T. 1 (0) denotes the sample is a member (non-member).

MIA can lead to serious privacy threats. For example, given a model trained on clinical records of cancer patients to determine the medicine dosage [38], the attacker can learn whether a person has cancer by applying MIA to the model. We follow previous work [11, 50, 72, 75, 79] and assume that the adversary only has black-box access to T, which means that the adversary can only query T with the data sample and obtain its corresponding output. Then, $\mathcal{A}_{MI}$ predicts membership with the output of T. Section 4.2 elaborates the details.

Attribute Inference Attack (AIA). Different from MIA, the goal of AIA is to infer attributes of a data sample that are not related to the target model's original classification task. A specific attack scenario is when AIA is used to infer some hidden sensitive attributes. For instance, a target model is trained to conduct gender classification, while AIA aims to infer the political view of a data sample. Such attribute is often hidden when training the target model. However, due to the intrinsic *over-learning* property of ML [78], a target model may try to capture attributes not directly relevant to its task. Note that AIA is different from property inference attack (PIA) [22] which infers a property about the entire dataset rather than a sample: e.g., PIA can tell whether a training dataset is gender-balanced.

Instead of having direct access to the sample, we follow previous work [59, 78] and consider the adversary only has its *representation* (e.g., embedding) generated by a target model T. Formally, AIA can be defined as:

$$\mathcal{A}_{AI} : h \mapsto s \tag{2}$$

where h is a sample's representation provided by T and s is the sample's sensitive attribute predicted by $\mathcal{A}_{AI}$.

Compared to MIA, the connection between AIA and CL might be less direct, but we are motivated to study this issue because CL makes the samples trained in the later batches introduce a greater impact on the trained model, and we suspect these samples are more vulnerable under AIA. Moreover, a recent study [35] suggests learning the underlying training distribution, which might not always be public, can boost AIA. In Appendix E, we elaborate the details of AIA.

3 Datasets and Target Models

In this work, we aim to quantify the privacy risks introduced by CL through the lens of MIA and AIA. To this end, we select popular datasets and models that are used for ML classification tasks. In our study, a total of 9 unique datasets are used, with 8 datasets used for MIAs and 3 datasets used for AIA. Among these datasets, 6 of them are image datasets, while the remaining 3 datasets consist of non-image data.

Datasets. Regarding MIA, we use the following 8 datasets, which are also adopted by previous work [32, 51, 60, 75]. They are CIFAR100 [44], Tiny ImageNet [47], Place100, Place 60 [100], SVHN [64], Purchase [75], Texas hospital stays [75] and Locations [95]. We focus on image datasets mainly (the first 5 datasets), but tabular datasets are also evaluated. Due to page limits, we defer the detailed description of the MIA datasets to Appendix A. Regarding the AIA datasets, we use Place100, Place60 and another dataset UTKFace [97]. We describe them in Appendix A as well.

Target Models. We adopt three popular neural network architectures of different learning capacities as the target models' architectures for the image datasets. They are ResNet-18 [29], ResNet-34 [29] and MobileNet [73]. We adopt cross-entropy as the loss function and SGD as the optimizer. We train all models for 200 epochs with a batch size of 128 by default. The learning rate is set to 0.1[2]. For the non-image dataset Purchase and Location, we choose a 3-layer MLP with the same number of epochs and batch size. The number of neurons in the hidden layer is 256. For the Texas dataset, we use a 5-layer MLP with 512 neurons in the hidden layer because this dataset contains more features. To avoid fortuitous outcomes, all experiments are repeated five times with different random seeds, and the standard deviations are presented.

4 Methodology

In this section, we describe the curriculum designs experimented with by our study, the implementation of the basic MIA, our proposed MIA, and the defense techniques to be tested. The implementation of the basic AIA is described in Appendix E.

4.1 Curriculum Designs

We choose two popular curriculum learning (CL) methods, which are highlighted in surveys like Wang et al. [89] and have open-source implementations [26, 83], to train the target model. We expect our major observations (described in Section 5) are also applicable to other CL methods, like self-paced curriculum [40, 45], and automated curriculum [24], because they share similar high-level ideas (e.g., self-paced curriculum differs from bootstrapping only in that self-paced curriculum does not let the curriculum completely guide its learning process). Below we explain the two CL methods.

- **Bootstrapping [27].** The target model T is first trained without CL, then it serves as a difficulty measurer (f in Algorithm 1) to order the training samples by their loss.
- **Transfer learning [91].** Different from bootstrapping, a pre-trained model is used for the difficulty measurer. We use inception-v3 [84][3] as the pre-trained model to evaluate the image datasets. The evaluation on tabular datasets with transfer learning is skipped, as we did not find a widely used pre-trained model in such a setting.

To better assess the improvement brought by the above two CL methods and their vulnerabilities under attacks, we establish two other methods for comparison.

- **Baseline curriculum.** It uses a random curriculum that is irrelevant to the data samples' difficulty. This curriculum is then used across all training epochs. The normal training process is different in that a random order is drawn for every training epoch.
- **Anti-curriculum.** It shares the same difficulty measurer with bootstrapping but arranges the samples from difficult to easy, reversing the outcome of bootstrapping.

[2] This learning rate is empirically chosen and has a very limited effect on attack accuracy. For example, when using a learning rate of 0.001, the MIA accuracy is affected by less than 0.2% when attacking a ResNet-18 model trained on CIFAR100.

[3] It is a widely-used image recognition model that achieves over 78.1% accuracy on the ImageNet dataset [13].

For the pacing function g, we choose varied exponential pacing [27], exponentially increasing the fraction of data by steps (a step denotes the iterations with the same output of g). According to [27], different pacing functions perform similarly.

In summary, the four CL methods differ in the difficulty measurer and each CL method feeds training data using the same curriculum (or ordering) across all epochs. The baseline and anti-curriculum methods help us understand the contribution of data ranking and order fixing separately (e.g., anti-curriculum can be considered as using a wrong curriculum but still repeating the order across epochs as advised by CL).

As described in Section 2.1, CL can accelerate the training process to reach higher accuracy. We first validate this claim by evaluating the training performance and the testing accuracy and comparing them to the normal training method, which does not use any curriculum as guidance.

Table 1 validates the effectiveness of CL. At least one of the four CL methods can outperform the normal training by 0.06% to 4.42%, and the corresponding average training accuracy is given in Appendix B (Training Accuracy). The maximum standard deviation in Table 1 is 0.0221 while 32 out of 37 results have a standard deviation less than 0.01. This indicates the difference among various CL methods is statistically significant. It is worth noticing that bootstrapping and transfer learning always outperform normal training, and anti-curriculum performs the worst consistently. Interestingly, we observe that the baseline performs as well as the transfer learning curriculum for Place100 and Place60, which means the transfer learning curriculum does not suit these two datasets well. Figure 1 validates the major motivation of adopting CL, i.e., reaching higher accuracy while converging faster. Throughout most of the training, bootstrapping and transfer learning reach higher accuracy faster than all the other methods. At the same time, it takes the longest for the anti-curriculum to reach the same training accuracy compared to all other methods. This indicates that repeating a meaningful data order improves training. This observation aligns with the discovery from previous work [27, 93]. Finally, CL is expected to have a disparate impact on classification accuracy across samples. Besides the analysis in Section 5, we also use t-distributed stochastic neighbor embedding (t-SNE) to visualize this impact. More details including the visualization are in Appendix B (t-SNE Study).

4.2 Basic MIA

After providing a high-level overview of MIA in Section 2.2, we now delve into the details, focusing on the three well-known attacks: NN-based (Neural Network-based) [71, 75], metric-based [79], and label-only attacks [11, 50].

NN-based attack assumes a vector of *prediction posteriors* (e.g., confidence scores or loss) of all class labels can be returned by the target model T when querying T with a data sample x. It is also assumed that the adversary has a *shadow dataset* ($\mathcal{D}$) that has the same distribution and format as T's private training dataset. $\mathcal{D}$ is used to train a set of *shadow models* $\mathcal{S}$ that behave similarly as T (e.g., having the same architecture as T like previous work [72, 75, 79]). The attacker trains an *attack model* $\mathcal{A}_{MI}$ using $\mathcal{S}$. In particular, the attacker queries every shadow model $\mathcal{S}$ with the samples from its own training dataset and a disjoint testing dataset. The prediction posteriors of all samples and whether they are in training (denoted member) or testing (denoted non-member) are used as input to train $\mathcal{A}_{MI}$. Finally, the attacker queries T with a sample of interest x and uses the prediction posteriors as the input to $\mathcal{A}_{MI}$ to predict the membership status.

Compared to the NN-based attack, the model $\mathcal{A}_{MI}$ of metric-based attacks does not need to be trained. Instead, $\mathcal{A}_{MI}$ generates a privacy risk score from the output of T and compares it to class-specific thresholds.

For the label-only attack, it assumes only the prediction label instead of the prediction posteriors are returned from T. Still, the adversary can continuously add adversarial perturbations to the input sample x until its prediction label has been changed. The key insight is that the magnitude of the adversarial perturbation is larger for the member sample as T gives a more confident prediction. $\mathcal{D}$ and $\mathcal{S}$ can be used to select a threshold to separate the perturbation magnitudes of members and non-members.

MIA Models. Following the original setting of the NN-based attacks [75], we adopt a 3-layer MLP with 64 and 32 hidden neurons, and 2 output neurons, as our attack model $\mathcal{A}_{MI}$. We use cross-entropy as the loss function and Adam as the optimizer with a learning rate of 0.01. $\mathcal{A}_{MI}$ is trained for 100 epochs. For metric-based attacks, we follow the implementation of Song et al. [79] and consider 4 metrics, including correctness, confidence, entropy, and modified entropy. The associated attack methods are named metric-corr, metric-conf, metric-ent, and metric-ment. For label-only attacks, we leverage the implementation from ART [86].

Related research has shown that NN-based attacks often achieve better attack accuracy compared to metric-based and label-only attacks [32, 72, 75]. Thus, we use NN-based attack (specifically black-box-top3) for most of our evaluation in Section 5.

4.3 Our Proposed MIA

Given that CL orders training samples by difficulty, impacting the model, we investigate the potential enhancement of MIA when the target model is trained under CL. For this purpose, we propose a novel MIA method called *Diff-Cali* specifically tailored for CL. We first introduce calibrated MIA, which serves as inspiration for designing Diff-Cali, followed by the details of Diff-Cali.

Calibrated MIA. Recently, Watson et al. [90] proposed to use a calibrated membership score instead of the standard membership score (e.g., loss) to determine whether a sample is a member. Assume $s(T, x)$ is the original membership score, where T is the target model, and x is a sample. The calibrated membership score $s_{cal}(T, x)$ is defined as follows:

$$s_{cal}(T, x) = s(T, x) - \mathbb{E}_{\mathcal{S} \leftarrow \mathcal{A}(\mathcal{D})}[s(\mathcal{S}, x)] \tag{3}$$

where $\mathcal{S}$ are shadow models[4] that behave similarly as T, $\mathcal{D}$ is the shadow dataset, functions $s(T, x)$ and $s(\mathcal{S}, x)$ output the membership scores from target and shadow models, $\mathcal{A}$ randomly samples subsets of $\mathcal{D}$ to train $\mathcal{S}$, and $\mathbb{E}$ computes the expectation of $s(\mathcal{S}, x)$. Finally, $s_{cal}(T, x)$ is compared to a fixed threshold θ, and a sample is considered a member if $s_{cal}(T, x) \geq \theta$.

[4] $\mathcal{S}$ are named as reference models in [90], which resemble shadow models [75] as they are also trained on the same data distribution of T.

Method Dataset	Normal	Bootstrapping	Anti-curriculum	Baseline	Transfer Learning
Tiny ImageNet	0.3842 ± 0.0027	**0.4002** ± 0.0043	0.3776 ± 0.0036	0.3798 ± 0.0035	0.3803 ± 0.0043
CIFAR100	0.6081 ± 0.0053	**0.6232** ± 0.0078	0.5991 ± 0.0098	0.6099 ± 0.0045	0.6127 ± 0.0221
Place100	0.2992 ± 0.0054	**0.3159** ± 0.0059	0.2967 ± 0.0037	0.3088 ± 0.0060	0.3007 ± 0.0053
Place60	0.4756 ± 0.0041	**0.4903** ± 0.0040	0.4815 ± 0.0025	0.4847 ± 0.0071	0.4707 ± 0.0154
SVHN	0.9592 ± 0.0004	0.9598 ± 0.0006	0.9566 ± 0.0005	0.9593 ± 0.0006	**0.9599** ± 0.0006
Purchase	0.4931 ± 0.0055	**0.5324** ± 0.0037	0.4760 ± 0.0055	0.5289 ± 0.0043	-
Texas	0.4809 ± 0.0072	**0.4975** ± 0.0066	0.4606 ± 0.0101	0.4877 ± 0.0095	-
Location	0.5861 ± 0.0107	**0.5914** ± 0.0027	0.5563 ± 0.0156	0.5838 ± 0.0077	-

Table 1: Target model's average test accuracy on different datasets. ResNet-18 is used for all image datasets, and MLP for non-image datasets Purchase, Texas, and Location. Transfer learning CL does not apply to non-image datasets. The target model accuracy is relatively low except for SVHN because we use a subset of the original training data.

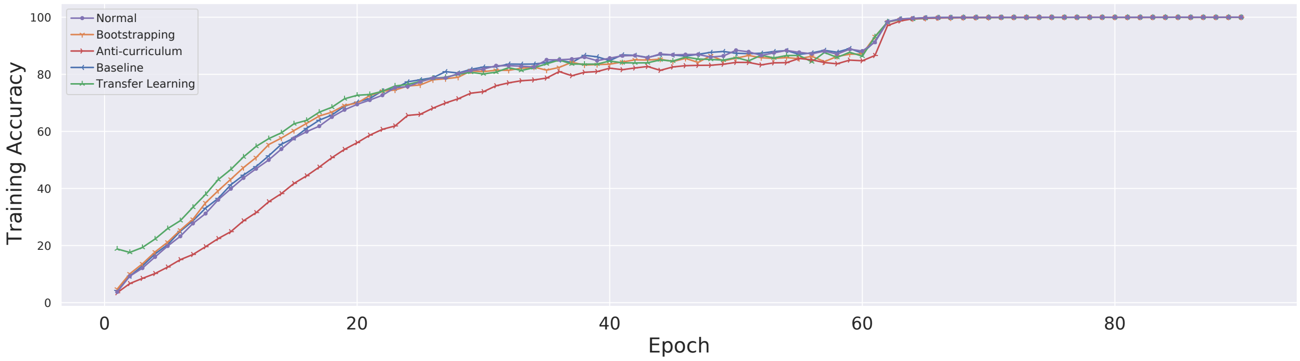


Figure 1: The training accuracy of different training methods with ResNet-18 on CIFAR100 along the increase of epochs (total of 90 epochs). Bootstrapping, transfer learning, and baseline reach higher accuracy faster and converge to a better result.

Previous MIA methods could have a high false positive rate (FPR) on non-members, often over-represented in the samples to be tested by the attacker. Equation 3 addresses this issue by using the *difference* between the target model and shadow models to derive the membership signal: if x is non-member to $\mathcal{S}$, it is also more likely non-member to T, therefore $s_{cal}(T, x)$ should be small. The evaluation results in [90] show the area under ROC curve (AUC) can be improved "by up to 0.10" (e.g., after calibrating the loss-based membership score with Equation 3).

Difficulty Calibrated MIA (Diff-Cali). Calibrated MIA compares $s_{cal}(T, x)$ of all samples to a fixed threshold θ, and we argue that θ can be *calibrated as well*. We observe that a CL curriculum re-orders the samples by their difficulty before the target model is trained, and such strategy changes how a sample is memorized and vulnerable under MIA (see Section 5.1 and Section 5.2). More specifically, we observe that CL makes the target model more vulnerable to MIA, and this impact is even more pronounced for difficult samples (Finding 1 in Section 5.1). Therefore, we can update θ according to the curriculum and make the attack model more accurate. We assume the attacker can generate a curriculum similar as the one used by the target model. For example, the attacker can use the publicly released pre-trained model to generate the curriculum. Alternatively, the attacker can train shadow models similar to the target model and build a curriculum according to loss from them.

We implement this idea for NN-based MIA. When the attack model $\mathcal{A}_{MI}$ outputs the prediction posteriors for an input x, the posterior of the label "member" is compared against θ, and x is predicted as a member when the posterior is larger. When training $\mathcal{A}_{MI}$, we adjust θ based on samples' difficulty level to improve the training accuracy, and the pseudo-code is shown in Algorithm 2. Specifically, in each epoch, the calibrated membership scores $s_{cal}(T, \mathcal{D})$ are generated for $\forall x \in \mathcal{D}$, and we use the loss to compute s (Line 2). Next, we try to find the threshold θ_0 (ranging from 0 to 0.1 based on our empirical study) that achieves the best accuracy in separating members and non-members from $\mathcal{D}$ (Line 3). After that, $\mathcal{A}_{MI}$ is updated by minimizing the training loss on $\mathcal{D}$ (Line 4) by adjusting the threshold with the following function:

$$g(x, C, \theta_0) = \frac{(|\mathcal{D}|-C(x))\ (\theta_0 - 0.0001)}{|\mathcal{D}|-1} + 0.0001 \qquad (4)$$

where $C(x)$ indicates the rank of sample x given by curriculum C. The rank for the easiest sample is 1, while the most difficult is $|\mathcal{D}|$. $g(x, C, \theta_0)$ is to assign a threshold θ from $[0.0001, \theta_0]$ (0.0001 is the initial threshold suggested by [90]) to each x based on its difficulty level (determined by a curriculum C), that is, calibrating threshold of each x based their difficulty level. The most difficult sample compares to 0.0001, the easiest one compares to θ_0, and others compare to θ that is ranged in $[0.0001, \theta_0]$. The more difficult x has a smaller threshold, meaning that we are lowering the bar for them to be predicted as members compared to the easy samples. During the testing phase, the threshold for a sample x is also adjusted with $g(x, C, \theta_0)$.

Algorithm 2: Training the attack model and adjusting threshold under Diff-Cali. "pred" is "prediction".

```
Input: Target model $\boldsymbol{T}$ , reference model $\boldsymbol{S}$, shadow dataset
       $\boldsymbol{\mathcal{D}}$, labels of shadow dataset $\boldsymbol{L}$, attack model $\boldsymbol{\mathcal{A}_{MI}}$,
       curriculum $\boldsymbol{C}$, number of epochs $\boldsymbol{E}$
for $e \in 1, \ldots, E$ do
    $s_{cal}\left(T, \mathcal{D}\right) = s\left(T, \mathcal{D}\right) - s\left(S, \mathcal{D}\right)$;
    $\theta_0 = \underset{\theta}{argmax}\ \mathrm{pred}(\mathcal{A}_{MI}, L, s_{cal}\left(T, \mathcal{D}\right))$;
    $\mathcal{A}_{MI} \longleftarrow train\left(\mathcal{A}_{MI}, s_{cal}\left(T, \mathcal{D}\right), g\left(x, C, \theta_0\right)\right)$;
```

Diff-Cali follows the direction of addressing the issue caused by over-represented non-members [5, 90]. On top of those works, Diff-Cali is customized under CL to amplify the effects of MIA. To demonstrate the benefit of Diff-Cali, we compare it with the score-based membership attack after difficulty calibration with default threshold in Cal [90].

Overall, Diff-Cali outperforms Cal by 4.0% to 9.9% of attack accuracy while maintaining the same AUC. Besides, Diff-Cali improves MIA's TPR at extremely low FPR, making the difficult sample more vulnerable. This focus (on the low FPR regime) is the setting with the most practical consequences, i.e., de-identifying even a few users contained in a sensitive dataset is far more significant than making an average-case statement like 'most people are not in the sensitive dataset' [5]. Moreover, we conclude that the knowledge of the actual curriculum being used is not required for the performance boost from introducing Diff-Cali (See Figure 4). The detailed evaluation of Diff-Cali across all metrics such as attack accuracy, confidence score, and TPR at low FPR are presented in Section 5.3.

Some recent works suggest to use class-specific thresholds [79], which are especially beneficial for unbalanced datasets. We did not adjust the threshold by classes because our thresholds have been fine-tuned with difficulty levels, and they are effective for both balanced and unbalanced datasets.

4.4 Defense Methods

Some defense methods have been proposed to reduce the success rate of privacy attacks, in particular, MIA. We are interested in how they perform under curriculum learning and our proposed attack. To this end, we select DP-SGD [1], MemGuard [38], MixupMMD [48] and AdvReg [62]. DP-SGD and MemGuard represent two directions in privacy protection, while MixupMMD and AdvReg are two more recent defense methods. Below, we explain the four defense methods.

DP-SGD. Differentially-Private Stochastic Gradient Descent (DP-SGD) modifies the stochastic gradient descent (SGD) algorithm and integrates (ϵ, δ)-DP [16] to provide provable privacy guarantee.

Definition 1. *((ϵ, δ)-DP) An algorithm $\mathcal{M}(\cdot)$ satisfies (ϵ, δ)- differential privacy ((ϵ, δ)-DP), if and only if for any pair of datasets V and V' that differs in only one element and for any possible output set O*

$$\Pr\left[\mathcal{M}(V) \in O\right] \leq e^{\epsilon} \Pr\left[\mathcal{M}(V') \in O\right] + \delta. \tag{5}$$

DP-SGD first randomly groups the samples by batches. Within a batch, after a per-sample gradient is computed, DP-SGD clips it to a maximum norm C and Gaussian noise is added to the gradient aggregated within the batch, with standard deviation δC. The output of the trained model will satisfy (ϵ, δ)-DP.

Because DP-SGD relies on random sampling, the DP guarantees in DP-SGD could be invalidated under CL, because the model will be trained with the same or public curriculum. Thus, we only show results of DP-SGD in normal training, and use it as a baseline to compare with the other methods.

MemGuard. Different from DP-SGD, MemGuard does not change the training process. At a high level, it obfuscates the predictions of the target model by adding noises to its output. It is designed to defend against MIA in particular, while DP-SGD deals with all sorts of privacy risks. Assuming an attack model $\mathcal{A}_{MI}$ has been trained with shadow training [75], and $\mathcal{A}_{MI}(T(x), y)$ outputs a confidence score ranging in $[0, 1]$, where $T(x)$ is the prediction of the target model and y is the label for x. A sample is considered a member if the score is larger than 0.5 and a non-member if smaller than 0.5. MemGuard has two phases. In Phase 1, it crafts adversarial noise and adds it to $T(x)$ to force $\mathcal{A}_{MI}(T(x), y)$ to be 0.5 to confuse the attacker, while the distance between the original prediction and the noisy prediction is minimized. In phase II, the adversary adds the noise to the original prediction with a certain probability of trade-off the utility and privacy.

MixupMMD. Li et al. [48] found a model vulnerability under MIA relates to the difference between the training and testing accuracy, and they proposed MixupMMD to intentionally reduce the training accuracy to validation accuracy. A new penalty, Maximum Mean Discrepancy (MMD), is used by the regularizer.

AdvReg. Nasr et al. [62] proposed to mitigate MIA by formulating the defense as a min-max optimization problem. Given a validation set that serves as "non-members", AdvReg introduces an adversarial classifier to infer the membership status using the posteriors generated from the target model. The optimization goal is to minimize the original classification loss and maximize the loss of the adversarial classifier.

5 Evaluation Results

In this section, we present the evaluation results of MIA when CL is applied to train the target model. We also attempt to explain the observations from the angle of data memorization and show the impact of CL on the existing defenses. We highlight our insights with text boxes. In Appendix E, we report the evaluation about AIA, but in general, CL is less vulnerable under AIA compared to MIA.

Evaluation Setup. To evaluate MIA, we split each dataset described in Section 3 into three disjoint parts: one for training the target model, one for training a shadow model, and one for testing both the target and shadow model.

Method Dataset	Normal	Bootstrapping	Anti-curriculum	Baseline	Transfer Learning
Tiny ImageNet	0.9193 ± 0.0000	0.9385 ± 0.0000	0.9116 ± 0.0001	0.9207 ± 0.0000	**0.9439** ± 0.0000
CIFAR100	0.8577 ± 0.0011	**0.8751** ± 0.0001	0.8376 ± 0.0001	0.8582 ± 0.0001	0.8718 ± 0.0001
Place100	0.9425 ± 0.0000	0.9549 ± 0.0001	0.9335 ± 0.0001	0.9416 ± 0.0001	**0.9617** ± 0.0001
Place60	0.8773 ± 0.0022	**0.8987** ± 0.0001	0.8625 ± 0.0001	0.8827 ± 0.0001	0.8902 ± 0.0001
SVHN	0.5570 ± 0.0000	**0.5605** ± 0.0002	0.5514 ± 0.0001	0.5599 ± 0.0003	0.5580 ± 0.0003
Purchase	**0.9524** ± 0.0016	0.9453 ± 0.0024	0.9118 ± 0.0122	0.9458 ± 0.0015	-
Texas	0.6749 ± 0.0092	**0.7068** ± 0.0139	0.5950 ± 0.0161	0.7039 ± 0.0122	-
Location	0.9153 ± 0.0066	**0.9194** ± 0.0048	0.8980 ± 0.0038	0.9169 ± 0.0038	-

Table 2: Accuracy of NN-based MIA on models trained on 8 datasets. Transfer learning CL does not apply to non-image dataset Purchase, Texas and Location.

To evaluate the defense methods, we split each dataset into five parts as some advanced methods need reference datasets for training. More details about the defenses can be found in Section 5.4. All experiments were *repeated 5 times* to minimize the fortuitous outcomes, and the mean value and standard deviation were reported.

Evaluation Metrics. First, we compute the attack accuracy, measured by the correct predictions (member/non-member) versus all predictions, to assess the effectiveness of MIA/AIA, and the classification accuracy of the target model to assess the impact of curriculum learning and defenses. Second, to better understand the attack results, we retrieve the confidence scores of members and non-members, respectively. Note that the confidence score indicates the likelihood of a sample being classified as a member or non-member. Third, we compute the true-positive rate (TPR) at the false-positive rate (FPR) of the attacks. As noted by Carlini et al. [5], attacks should emphasize the member guesses over non-member guesses, so they should be evaluated by considering TPR at low FPR. This cannot be precisely modeled by metrics like overall accuracy, precision, or recall.

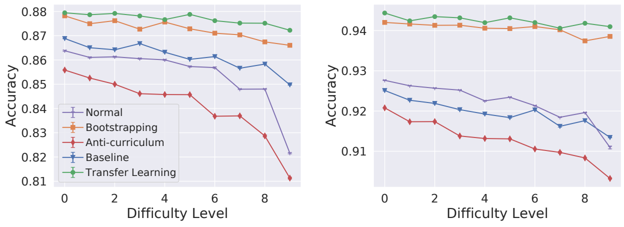
5.1 Evaluation of Basic MIA

We start with the experiments on the 5 image datasets (CIFAR100, Tiny ImageNet, Place100, Place60, and SVHN), using ResNet-18 as the target model architecture and later ResNet-34 and MobileNet for comparison. The evaluation of the tabular datasets (Purchase, Texas hospital stays, and Locations) is presented at the end. The attack models are described in Section 4.2.

MIA Accuracy. We found that models trained using meaningful CL methods (i.e., bootstrapping and transfer learning) are slightly **more vulnerable** to MIA. Table 2 shows the accuracy of NN-based black-box-top3 MIA [75] by datasets and CL methods. All experiments are repeated five times with different random seeds and the standard deviations are presented. Additionally, we run McNemar's test and verify that the difference among models trained with various curriculum methods are statistically significant (i.e., p-value < 0.05). The biggest attack accuracy improvement observed for image datasets is 2.46% (Tiny ImageNet with transfer learning) while the biggest improvement for non-image datasets is 3.20% (Texas with bootstrapping). Among different CL methods, bootstrapping and transfer learning are the most vulnerable, with an average of 1.29% and 1.44% improvement in the attack accuracy against the normal training, respectively. For baseline CL, the attack accuracy decreases for Place100, whereas a slight increase is observed for the attack accuracy on other datasets. For anti-curriculum CL, the attack accuracy decreases for all datasets. This result indicates both the data repeating (reflected by the results of baseline) and ordering (reflected by the results of bootstrapping and anti-curriculum) of CL (explained in Section 4.1) contribute to the vulnerability under MIA. The consistent performance of bootstrapping and anti-curriculum indicates that **data ordering plays a bigger role**.

Regarding the impact of datasets, we found more complex datasets (e.g., with more classes of labels) tend to have higher attack accuracy in general. For example, the average MIA accuracy is 94.39% for Tiny ImageNet (200 classes), 87.18% for CIFAR100 (100 classes), 96.17% for Place100 (100 classes), 89.02% for Place60 (60 classes), and 55.80% for SVHN (10 classes), all under transfer learning. The same effects have also been observed in other works [75].

Regarding the metric-based and label-only attacks, the result is similar to the NN-based attack, as suggested by the evaluation on CIFAR100, shown in Table 3. The only exception is metric-corr, which performs worse than other attacks with bootstrapping. This result can be explained by the assumption of metric-corr that the target model is trained to predict correctly on its training data, which may not generalize well on the test data. In the rest of the evaluation, we fix the attack model to black-box-top3, and the NN-based attack in the rest of the paper primarily refers to black-box-top3, unless indicated otherwise.



(a) CIFAR100 **(b) Tiny ImageNet**

Figure 2: MIA accuracy on CIFAR-100, Tiny ImageNet. ResNet-18 is used for target model training.

Method Attack	Normal	Bootstrapping	Anti-curriculum	Baseline	Transfer Learning
NN-based [75]	0.8577 ± 0.0011	**0.8751** ± 0.0001	0.8376 ± 0.0002	0.8582 ± 0.0001	0.8718 ± 0.0001
Metric-corr [79]	0.6920 ± 0.0000	0.6820 ± 0.0000	0.6905 ± 0.0000	**0.6930** ± 0.0000	0.6855 ± 0.0000
Metric-conf [79]	0.8600 ± 0.0000	**0.8810** ± 0.0000	0.8458 ± 0.0000	0.8553 ± 0.0000	0.8740 ± 0.0000
Metric-ent [79]	0.8490 ± 0.0000	**0.8750** ± 0.0000	0.8320 ± 0.0000	0.8435 ± 0.0000	0.8685 ± 0.0000
Metric-ment [79]	0.8620 ± 0.0000	**0.8820** ± 0.0000	0.8463 ± 0.0000	0.8568 ± 0.0000	0.8760 ± 0.0000
Label-only [86]	0.8200 ± 0.0082	**0.8263** ± 0.0082	0.7963 ± 0.0117	0.8050 ± 0.0045	0.8088 ± 0.0074
Cali [90]	0.7889 ± 0.0012	**0.8272** ± 0.0009	0.7532 ± 0.0004	0.7781 ± 0.0025	0.8148 ± 0.0013
Diff-Cali	0.8519 ± 0.0003	**0.8670** ± 0.0006	0.8382 ± 0.0006	0.8438 ± 0.0008	0.8614 ± 0.0006

Table 3: Average accuracy of NN-based, metric-based, label-only and Diff-Cali attacks on models trained on CIFAR100 with ResNet-18.

Figure 2 shows the attack accuracy of samples from different difficulty levels. More specifically, we construct the test dataset as half member samples and half non-member samples. Member samples are divided into different difficulty levels while non-member samples across each difficulty level are fixed. Figure 2 demonstrates that using a meaningful curriculum (i.e., bootstrapping and transfer learning) introduces a higher increase of attack accuracy on the difficult samples compared to the simple samples (e.g., 7% vs. 2.5% on CIFAR100). Hence, the impact of curriculum is more pronounced on difficult samples than on simple samples.

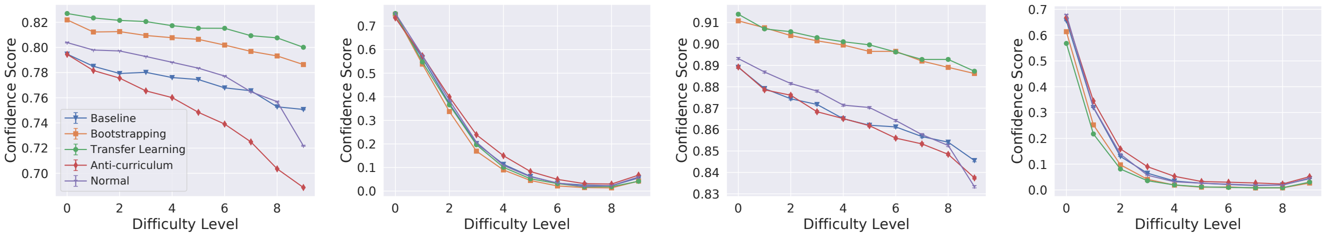
Confidence Score. Since the key contribution of CL is to factor in the samples' difficulty levels during the training procedure, here we evaluate how difficulty levels impact the samples' vulnerability individually. Intuitively, the difficult member samples should be harder to attack than the easy member samples. As we can see from Figure 3a, and Figure 3b, the confidence scores of difficult member samples are closer to the score distribution of non-member samples. On the other hand, difficult non-member samples could be easily attacked, as they have significantly lower confidence scores. However, since CL forces the model to learn the samples in a repetitive manner, we want to find out whether samples will be remembered by the model differently. To assess and quantify the possible privacy risk discrepancy caused by CL, we first arrange samples according to their difficulty level. Then, we use the confidence score and attack accuracy to analyze individual samples. Note that we train a separate model and use the sample loss given by this model as a guide to determine how difficult a sample is. This model is used solely for getting the difficulty levels of all samples and is different from the target model in our following evaluation.

Figure 3 depicts the attack model's confidence score by samples' difficulty levels, when CIFAR100 and Tiny ImageNet are tested. Though the difficult samples are not more vulnerable than the easy samples, **the gap in confidence scores is much narrower** (especially for the confidence score of members). Take the target model in CIFAR100 as an example, our attack model can recognize the most difficult member samples (scored as difficulty level 9) from this model with over 7.83% (absolute growth from 72.19% to 80.02%) more confidence, thanks to transfer learning (Figure 3a). Interestingly, for the most difficult member samples, it is even possible for anti-curriculum to have a higher confidence score compared to the normal training (Figure 3c). This observation indicates that enforcing difficult samples to the training process first does not necessarily make the model more likely to forget them. If we perceive feeding difficult samples first to a model as negative, the repetition of a curriculum can possibly compensate for such a negative effect, i.e., making the target model memorize the difficult samples better than a normal ML where these samples are presented at random times throughout training. In Appendix C, we show the confidence scores on the other image datasets, including SVHN (Figure 11), Place100 (Figure 12) and Place60 (Figure 13). The trend is similar.

TPR at Low FPR. In addition to the attack accuracy, we measured the relationship between TPR and low FPR, as explained in "Evaluation Metrics" (Section 5). Following Carlini et al. [5], we present the ROC curve for the attacks with both linear scaling and log scaling to emphasize the low-FPR regime. Figure 4a and Figure 4b demonstrate the ROC curve for NN-based attack. The results show that using curriculum increases ROC. The TPR of transfer learning and bootstrapping are generally higher than the others except at extremely low FPR ($< 10^{-4}$). This indicates CL introduces disparate impact to members and non-members for most samples. Moreover, the NN-based attack fails to achieve a TPR better than random chance at any FPR below 0.045, indicating potential for further improvement.

Loss Distribution. The previous evaluation presents a macro-level understanding of CL's impact on MIA. Here we present a micro-level analysis by examining the loss distribution between members and non-members in models trained with normal and CL methods. Due to the space limitation, here we only show the results of ResNet-18 trained on Tiny ImageNet in Figure 5 which shows a clearer discrepancy in terms of the loss distributions compared to other datasets. Note that the loss scores are normalized. As one can see, there is a clear difference between their loss distributions, e.g., bootstrapping makes the overall members' loss much lower and the members' loss distribution less overlapped with non-members', especially for those members with higher difficulty levels. In Section 5.2, we also reason this observation from the perspective of data memorization.

Target Model Architectures. To understand the impact of the architecture of the target model, we launched MIA against ResNet-34 and MobileNet and compared the results against ResNet-18. Table 4 demonstrates the average attack accuracy of MIA when target models are trained with ResNet-18, ResNet-34, and MobileNet,

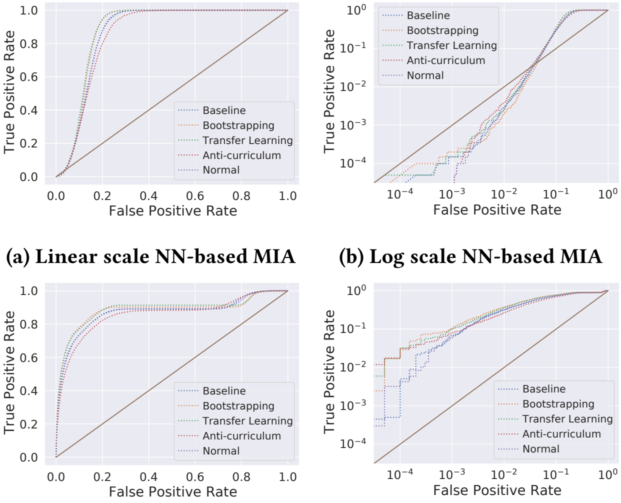


(a) Members of CIFAR100 (b) Non-members of CIFAR100 (c) Members of Tiny ImageNet (d) Non-members of Tiny ImageNet

Figure 3: Attack model's confidence score for both member and non-member samples on CIFAR-100 and Tiny ImageNet. ResNet-18 is used for target model training, and data samples are arranged according to their difficulty scores from bootstrapping.

Architecture \ Method	Normal	Bootstrapping	Anti-curriculum	Baseline	Transfer Learning
ResNet-18	0.8577 ± 0.0011	**0.8751** ± 0.0001	0.8376 ± 0.0002	0.8582 ± 0.0001	0.8718 ± 0.0001
ResNet-34	0.8564 ± 0.0001	**0.8746** ± 0.0003	0.8481 ± 0.0002	0.8559 ± 0.0002	0.8715 ± 0.0002
MobileNet	0.7979 ± 0.0001	0.8308 ± 0.0000	0.7763 ± 0.0002	0.8318 ± 0.0000	**0.8430** ± 0.0001

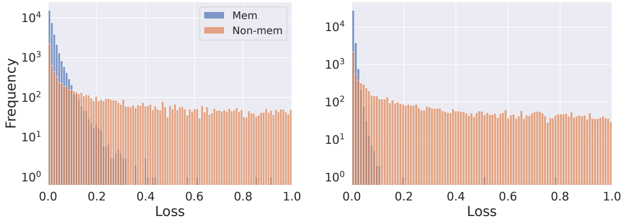
Table 4: The average accuracy of NN-based attacks on models trained on different network architectures with CIFAR100.



(a) Linear scale NN-based MIA (b) Log scale NN-based MIA

(c) Linear scale Diff-Cali (d) Log scale Diff-Cali

Figure 4: TPR/FPR of NN_based MIA and Diff-Cali under different training method trained with ResNet-18 on CIFAR100.

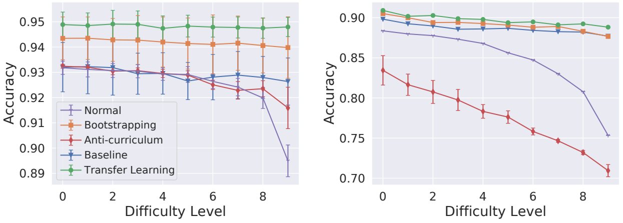


(a) Normal training (b) Bootstrapping

Figure 5: Loss distribution for models trained on Tiny ImageNet with ResNet-18.

respectively. It shows that they all share a similar trend of how CL affects MIA. Though MobileNet turns out to be less vulnerable (5.85% and 5.93% less attack accuracy compared to ResNet-34 and ResNet-18, respectively), bootstrapping, transfer learning, and baseline all increase the overall attack accuracy compared to normal training. Figure 6 demonstrates the results by difficulty levels on ResNet-34 and MobileNet when training with Tiny ImageNet, which can be viewed together with Figure 2b about ResNet-18. MobileNet again turns out to be less vulnerable (4% less attack accuracy compared to ResNet-34 and ResNet-18), bootstrapping, transfer learning, and baseline all increase the overall attack accuracy and narrow down the gap between difficult and easy samples. As such, the privacy concerns in CL cannot be addressed by changing the target models' architectures. This observation is consistent with other works [32, 50] about MIA vs. architectures. On a different note, we speculate that MobileNet is less vulnerable compared to ResNet due to its more limited learning capacity, which results in less over-fitting and memorization, making it more robust against MIA. We discuss the overfitting issue further in Section 6.

Non-image Datasets. As shown in Table 2, most experiments remain to have the same trend they are showing in image datasets. For Purchase, however, attack accuracy on normal training is 0.71% higher than bootstrapping for example. This shows that CL does not always empower MIA more. In Figure 14 of Appendix C, we show the confidence score of members and non-members on Purchase, and the result is similar to the image datasets, where the impact of CL is more prominent on difficult samples.



(a) ResNet-34

(b) MobileNet

Figure 6: MIA accuracy for target model trained on Tiny ImageNet with ResNet-34 and MobileNet, respectively.

In the meantime, we found the changes caused by different CL methods are more drastic on the non-image datasets, compared to the image datasets. For example, Texas has a more prominent attack accuracy drop (8.0%) on anti-curriculum. The non-image datasets are relatively simple, containing only binary features after preprocessing, hence they are more likely to be impacted by CL. Table 1 also shows the target model accuracy varies more for the non-image datasets under CL.

Finding 1: CL makes the target model more vulnerable to MIA, and the impact of CL on difficult samples is more pronounced than on simple samples.

Finding 2: Both data ordering and data repeating make a model more vulnerable under MIA, while data ordering plays a bigger role in influencing the vulnerability of a model under MIA.

5.2 Analysis with Data Memorization

The previous experiments show that the impact of CL on difficult samples is more pronounced than on simple samples. Here, we attempt to explain this observation with a more principled analysis. Recent works [17, 18] suggest the effectiveness of MIA could be tied to how well the target model *memorizes* individual data sample. The notion of memorization is formally defined as [17]:

$$\mathbf{mem}(\mathcal{A}, \mathcal{D}, i) := \Pr_{T\sim A(D)}\left[T\left(x_i\right) = y_i\right] - \Pr_{T\sim A\left(D^{\setminus i}\right)}\left[T\left(x_i\right) = y_i\right] \quad (6)$$

where $\mathcal{A}$ denotes the training algorithm, $\mathcal{D}$ denotes the training dataset, T is the trained model, (x_i, y_i) denotes one sample with its ground-truth label, and $\mathcal{D}^{\setminus i}$ denotes $\mathcal{D}$ with i-th sample removed. The model is likely to memorize the data sample if adding (x_i, y_i) to training significantly changes the model's prediction on y_i. Though Equation 6 models the memorization of a single data sample, we can easily extend it to quantify the memorization of multiple samples at once.

Specifically, we evaluate ResNet-18 trained with CIFAR100. We first leave out 800 most difficult data samples (4% of all samples) and train a model without these data via bootstrapping ("not seen"). Then, we train the model under CL according to data memorization: the curriculum makes the 800 data samples either be seen at the beginning ("first seen"), end ("last seen"), or random places ("random") of each training epoch. Figure 7 depicts the prediction probability of the true labels of the 4 scenarios. Data memorization under CL can be assessed by comparing "first seen", "last seen", and "random" to "not seen", following the idea of Equation 6. We observe that other than "not seen", the other three scenarios memorize the difficult samples fairly well (higher prediction probability of the true class). It turns out that data ordering has a strong impact on data memorization, e.g., "last seen" provides the strongest memorization compared to "first seen" and "random". The impact on difficult samples is more pronounced under CL because they are memorized better after data ordering. Another concept often considered to be connected to memorization is data valuation. In Appendix D, we elaborate on the topic of data Shapley and study if our observation in this section can be explained from the angle of data valuation.

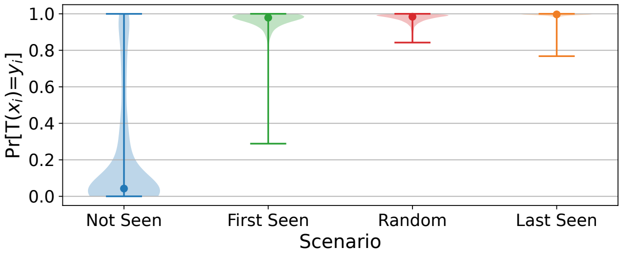
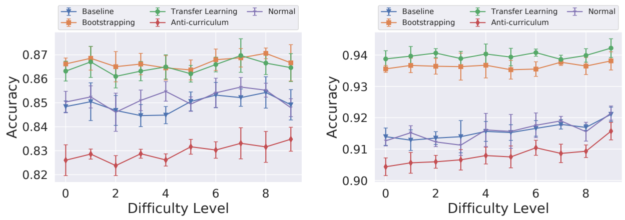


Figure 7: Memorization: violin plots of prediction probability of 800 most difficult samples, according to bootstrapping CL. The horizontal bars of each violin represent the minimum and maximum of the prediction probability.

Finding 3: CL forces the model to memorize the difficult samples harder, which makes them more vulnerable.

5.3 Evaluation of Diff-Cali

In order to fully utilize the information of difficulty levels exposed by CL, we propose Diff-Cali as described in Section 4.3. Overall, the NN-based attack still has a slightly better attack accuracy compared to Diff-Cali, but Diff-Cali has higher confidence scores for difficult samples and has better TPR at the low FPR regime.



(a) CIFAR100

(b) Tiny ImageNet

Figure 8: Diff-Cali's accuracy for models trained on CIFAR100 and Tiny ImageNet with ResNet-18.

Attack Accuracy. Table 3 presents the accuracy of Diff-Cali, which is about 1% lower compared to NN-based attack on all CL methods. Figure 8 depicts the attack accuracy on CIFAR100 and Tiny ImageNet. Though Diff-Cali achieves slightly lower accuracy (a difference of less than 1.44%) compared to NN-based attack, with adaptive calibration, we are able to make **the difficult samples more vulnerable**: For example, the attack accuracy of difficulty level at 9 and 0 are 86.47% and 86.32% for transfer learning under CIFAR100. The most difficult samples now can be predicted 2.64% and 2.35% more accurately for normal and anti-curriculum ML, respectively. Overall, Diff-Cali is able to overcome the privacy risk discrepancy of different samples through calibration and results in better attack accuracy for difficult samples for normal ML and anti-curriculum ML.

Confidence Score. Like the evaluation of basic MIA, we show the confidence scores of samples according to their difficulty level in Figure 15 and Figure 16 of Appendix C. Overall, we are able to achieve confidence scores greater than 0.7807 (normal) for CIFAR100 and 0.8678 (normal) for Tiny ImageNet for all member samples, whereas the minimum member confidence score from NN-based is 0.6889 for CIFAR100 and 0.8333 for Tiny ImageNet (Figure 3). In short, we are able to improve the normal training confidence score for all members by 3.29% for CIFAR100 and 3.45% for Tiny ImageNet. Similarly, we reduce the confidence score of non-members (note that a lower confidence score means less chance to be misclassified as non-members) by 0.0414 for CIFAR100 and 0.1751 for Tiny ImageNet. Unlike previous NN-based attack, the accuracy of Diff-Cali does not share a similar trend as the confidence score because the final prediction of the membership status of Diff-Cali is not based on the confidence score solely.

TPR at Low FPR. In Figure 4, we show that Diff-Cali can achieve much higher TPR at low FPR ($< 10^{-4}$). We present the ROC curve for the attacks with both linear scaling and log scaling to emphasize the low-FPR regime. Figure 4c and Figure 4d demonstrate the ROC curve for Diff-Cali. The results show that using curriculum increases ROC (Figure 4a, Figure 4c). We observe that our proposed Diff-Cali performs better at low FPR. More specifically, Figure 4b shows that NN-based attack fails to achieve a TPR better than random chance at any FPR below 0.045 while Diff-Cali can be better than random guessing at all times.

> Finding 4: Diff-Cali improves MIA performance in terms of TPR at low FPR, making the difficult samples not only more vulnerable compared to other attacks but also more vulnerable than the simple samples.

5.4 Evaluation of Defense

We evaluate how the defenses including DP-SGD, MemGuard, MixupMMD, and AdvReg perform under the normal setting or CL. Table 5 shows the attack accuracy on ResNet-18 which is trained with CIFAR100. Because MixupMMD and AdvReg require reference datasets for defense deployment, we equally divided CIFAR100 into 5 parts for fair comparison among all the defense techniques. More specifically, all target models in Table 5 are trained with only 12,000 data points, which also explains why the accuracies are lower.

Regarding DP-SGD, ϵ and δ in our evaluation are 124,496 and $1e-5$. We have a large ϵ because we have 200 epochs of training and ResNet-18 contains a large number of parameters. We did not change these settings for a fair comparison with other defense techniques. Previous studies have used large ϵ for DP-SGD in order to achieve good model accuracy [34, 46]. Based on a recent work [4], we are able to make ϵ 10 times smaller after proper parameter tuning while achieving similar target accuracy. The ϵ can be brought down even first with a large batch size. Pulling tricks of DP-SGD based on the above recent work can further boost the tradeoff. Note that we still use small batch size for DP-SGD evaluation though that results in large ϵ. This is because we want to keep parameters across all target models the same for a fair MIA evaluation, and we have limited computing resources for handling large batch numbers.

Because of the conflicting requirement of DP-SGD and CL, we only present the result of DP-SGD under the normal setting. DP-SGD is able to curb the MIA accuracy from 90.3% to 50.8%, which is close to random guess (i.e., member or non-member), though at the cost of a significant drop in the target model's classification accuracy (from 48.0% to 17.4%). This observation is consistent with previous works [46, 48]. We also found DP-SGD is effective against Diff-Cali (e.g., attack accuracy for normal is dropped to 53.67%).

For MemGuard, due to its design, NN-based MIA accuracy is fixed to 50% when the defender knows what MIA method is performed by the attacker, reaching the same level as DP-SGD. In the meantime, the classification task of the target model is not impacted by MemGuard. However, it is not effective against label-only attacks, as it does not change the label. Our evaluation shows that the label-only attack accuracy can still reach up to 84.5% even with MemGuard deployed. MixupMMD decreases the MIA accuracy (e.g., 91.4% to 83.1% for bootstrapping) but it is much higher than DP-SGD. Interestingly, it increases the target model accuracy (e.g., from 51.4% to 54.4% for bootstrapping), which might be attributed to its new regularizer. AdvReg can also increase target accuracy (e.g., 51.4% to 54.2% for bootstrapping) but like MixupMMD it is not effective in mitigating MIA (e.g., MIA accuracy is even increased from 91.4% to 91.6% for bootstrapping). This observation concurs with a previous work [79].

Overall, there is still room for improvement in defenses. Potential future work can follow the direction of preserving certain properties brought by an ML technique (e.g., fast convergence and higher final performance by CL) and mitigating privacy risks generically.

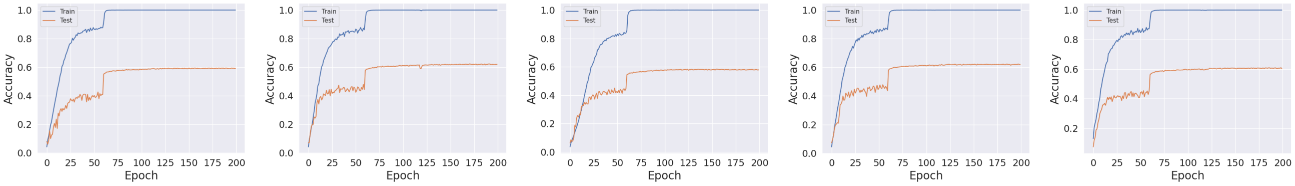
> Finding 5: Except DP-SGD, none of the studied defenses can significantly drop the MIA accuracy. DP-SGD cannot deliver the DP guarantee under CL.

6 Discussion

Limitations. 1) The research on ML privacy has been growing strong in recent years, and numerous attacks, variations, and defenses have emerged. Admittedly, not all attack methods (e.g., adaptive attack [79] and LiRA [5]) and defense techniques (e.g., PATE [67]) have been examined. Though LiRA is more effective than the basic MIA attacks we experimented, it requires multiple shadow models while all other attacks on our paper need one. To fairly compare with LiRA, the current datasets need to be divided into much smaller subsets, which will lead to worse performance of all target models and shadow models. Thus, we did not examine LiRA in this work. However, we believe our key conclusions (e.g., the difficult samples become more vulnerable when trained with

	None		DP-SGD		MemGuard			MixupMMD		AdvReg	
	Target	MIA	Target	MIA	Target	MIA	Label-only	Target	MIA	Target	MIA
Normal	48.0	90.3	17.4	50.8_±0.07	48.0	50.0	83.0	54.1	81.6_±0.02	51.2	89.2_±0.01
Bootstrapping	51.4	91.4_±0.03	-	-	51.4	50.0	84.5	54.4	83.1_±0.02	54.2	91.6_±0.02
Transfer	48.9	91.3_±0.03	-	-	48.9	50.0	84.5	55.7	76.1_±0.03	50.4	92.8_±0.04
Baseline	50.0	91.5_±0.02	-	-	50.0	50.0	84.0	55.0	84.4_±0.02	53.0	91.6_±0.01
Anti-curriculum	49.3	89.5_±0.02	-	-	49.3	50.0	81.3	52.6	79.1_±0.02	52.1	87.3

Table 5: The average accuracy of MIA (± standard deviation) on target model trained on CIFAR100 with ResNet-18 and different defense methods. All numbers are in percentage, entry without ± STD means the STD is less than 0.01%.



(a) Normal **(b) Bootstrapping** **(c) Anti-curriculum** **(d) Baseline** **(e) Transfer Learning**

Figure 9: The training and test accuracy over 200 epochs for target model ResNet-18 on CIFAR-100.

Method Epoch (Network)	Normal	Bootstrapping	Anti-curriculum	Baseline	Transfer Learning
100 (ResNet-18)	0.8390 ± 0.0001	0.8468 ± 0.0001	0.8153 ± 0.0002	0.8461 ± 0.0000	0.8670 ± 0.0000
200 (ResNet-18)	0.8577 ± 0.0011	0.8751 ± 0.0001	0.8376 ± 0.0002	0.8582 ± 0.0001	0.8718 ± 0.0001
100 (ResNet-34)	0.8356 ± 0.0001	0.8466 ± 0.0001	0.8310 ± 0.0000	0.8502 ± 0.0000	0.8577 ± 0.0001
200 (ResNet-34)	0.8564 ± 0.0001	0.8746 ± 0.0003	0.8481 ± 0.0002	0.8559 ± 0.0002	0.8715 ± 0.0002
100 (MobileNet)	0.6475 ± 0.0002	0.6764 ± 0.0002	0.6012 ± 0.0002	0.6695 ± 0.0002	0.6744 ± 0.0002
200 (MobileNet)	0.7979 ± 0.0001	0.8308 ± 0.0000	0.7763 ± 0.0002	0.8318 ± 0.0000	0.8430 ± 0.0001

Table 6: The average accuracy (± standard deviation) of NN-based attacks on target models on CIFAR100 trained with different epochs and network architectures.

CL) hold generically, due to the fundamental designs of the curriculum. 2) Overfitting can impact a target model's memorization of the training data, which in turn affects membership leakage, as discussed later in this section. Early stopping is a well-known method to limit such memorization and may help mitigate membership leakage. We acknowledge that this technique is not utilized in the paper. 3) We mainly evaluated the privacy attack on image and tabular datasets, with widely used models like ResNet and MLP. The two popular CL methods including bootstrapping and transfer learning are tested. Admittedly, not all data types (e.g., text and speech), models and CL methods are covered. Particularly, the newer model structures, such as the transformer-based model (e.g., Vision Transformer [14]), could result in larger privacy leakage, due to their better model capacity, and we leave the investigation as a future work. 4) Not all ML privacy attacks are tested, such as model inversion attacks [21, 96], as we suspect they are less likely to be impacted by CL. In the end, we want to mention that our motivation and efforts are comparable to other works that study the privacy of special ML settings like contrastive learning [32]. 5) We provided a few ways to calculate the difficulty score, such as bootstrapping and transfer learning, which rely on only one model. However, there are more sophisticated methods to measure difficulty scores that might give CL an even larger boost. For example, using difficulty measurements from an ensemble of models, such as MobileNet and ResNet. 6) Due to the conflicting requirements of CL and DP-SGD, we did not test the original DP-SGD on models trained under CL. We have not found a study that combines them but we believe such a study would be interesting and necessary.

Evaluation Metrics. For privacy attacks like MIA, whether and how it is effective is determined by the evaluation metrics. Attack accuracy is the one adopted in the beginning and is still widely used today, but recent studies have suggested metrics have to be carefully selected to fully understand the results. Following Carlini et al. [5], we adopt TPR at low FPR as another metric. We also view the results under confidence scores to shed light on the divergent impacts of CL into samples, which reveal new insights that are not captured by other metrics. Other metrics like precision/recall [5] and disparate vulnerability [94] can be considered and we believe this research direction still needs new input.

Overfitting. We acknowledge that overfitting can affect a target model's memorization of the training data, thereby making MIA easier. To further study this topic and its impact, we use CIFAR100 as an example and train target models for both 100 and 200 epochs. We then compared their overfitting levels (measured by the difference between training and test accuracy) and their corresponding MIA accuracy. Figure 9 shows the training and test accuracy over 200 epochs, which demonstrate the overfitting levels over time are comparable among different CL methods. Table 6 shows the MIA accuracy for target models trained for 100 and 200 epochs using different network architectures. This demonstrates that both overfitting and CL strategies can increase a model's vulnerability to MIA. Building on this, we will investigate how these factors contribute to model vulnerabilities in future work. Specifically, we will examine the convergence process, focusing on techniques like data augmentation and regularization to mitigate overfitting, while evaluating the impact of CL strategies on model vulnerability. Furthermore, *early stopping* can serve as a potential mitigation for membership leakage, as indicated in previous research work [79].

7 Related Work

Curriculum Learning (CL). The idea of CL was first introduced by Bengio et. al [3]. Researchers have then developed many new designs such as predefined CL [41], self-paced CL [40], CL by transfer learning [91] and other automated CL [24]. CL is proved to be effective in the domain of reinforcement learning [19, 20, 58, 61], computer vision [3, 15, 70, 80], natural language processing [3, 25, 52, 82, 101], speech [6, 56, 98], etc. Note that the concept of self-paced[45] learning can often be confused with CL bootstrapping. They share a similar idea of using an iterative procedure to assign higher weights to training examples that have lower costs with respect to their chosen hypothesis. Bootstrapping differs in that the difficulty score is generated based on the model accuracy rather than a hypothesis [27].

Membership Inference Attack (MIA). Section 4.2 has surveyed some representative works about MIA. Here we describe other notable works. On top of the original MIA [75], Salem et al. [72] proposed three more powerful attacks by relaxing the assumptions made by Shokri et al. [75]. Nasr et al. [63] investigated privacy risks in centralized and federated learning scenarios under both black-box and white-box settings. Recent works show that MIA can be further enhanced by adopting flexible thresholds [36], calibrated difficulty level [90], and loss trajectory [54]. Besides the general ML settings, recent works examined special settings like contrastive learning [32, 51], Generative Adversarial Networks (GAN) [8, 10, 33], and Graph Neural Networks (GNN) [30, 31, 92]. However, none of them investigated curriculum learning, and we aim to fill this knowledge gap. To mitigate MIA, researchers have proposed a few defensive mechanisms, like DP-SGD [1], MemGuard [38], Mixup-MMD [48], and AdvReg [62], as described in Section 4.4. PATE [67] uses teacher models to supervise the training of the student model and adds Laplacian noise to the teacher models' output. Salem et al. [72] leverage model stacking and dropout to reduce overfitting.

Attribute Inference Attack (AIA). AIA presents another notable threat to ML privacy. Appendix E surveyed the key works under AIA. In addition, He et al. [32] show that AIA is more vulnerable to models trained by contrastive learning. Recently, Song et al. [77] show that AIA is also effective against language models. Jayaraman et al. propose a new white-box AIA method that achieves better accuracy [35]. We focus on the black-box setting.

Other Attacks Against ML Models. MIA and AIA can be considered as attacks on the data privacy of ML. Model privacy, integrity, and availability have also been investigated, resulting in numerous studies. Model stealing aims to learn the parameters [42, 43, 66, 74, 85] or hyperparameters [65, 88] of a target model, and model inversion, whose goal is to recover the training dataset [21, 96]. There also exists some works focus on protecting a model's ownership [2, 9, 12, 37, 49, 57, 68, 87] to defend against model stealing attacks and other attacks like network pruning and fine-tuning.

8 Conclusion

In this work, we perform the first quantitative study to understand how curriculum learning (CL) , a widely used technique that accelerates model training, and affects the privacy of the trained model. Specifically, we trained target models under 6 image datasets and 3 tabular datasets and performed membership inference attacks (MIA) and attribute inference attacks (AIA) against them to assess the privacy risk in CL. Our results show that the target model becomes slightly more vulnerable to MIA but not so under AIA. We also found MIA has a significantly larger impact on samples with high difficulty levels. By exploiting the leakage from difficulty levels, we design a new MIA, termed Diff-Cali, which achieves similar overall accuracy with much better TPR at low FPR and can infer difficulty samples from normal ML more accurately. Finally, we evaluate the existing defenses like MemGuard, MixupMMD, and AdvReg in the CL setting, and our results show that none of them are effective when the model is trained under CL. With this study, we hope to draw attention to potential future work that preserves certain properties introduced by advances ML techniques (e.g., fast convergence and higher final performance by CL) while mitigating privacy risks generically.

Acknowledgments

This work is partially funded by NSF CNS-2220434 and the European Health and Digital Executive Agency (HADEA) within the project "Understanding the individual host response against Hepatitis D Virus to develop a personalized approach for the management of hepatitis D" (D-Solve) (grant agreement number 101057917).

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] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. In *USENIX Security Symposium (USENIX Security)*. USENIX, 1615–1631.

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

[4] Zhiqi Bu, Yu-Xiang Wang, Sheng Zha, and George Karypis. 2022. Automatic clipping: Differentially private deep learning made easier and stronger. *arXiv preprint arXiv:2206.07136* (2022).

[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] Antoine Caubrière, Natalia Tomashenko, Antoine Laurent, Emmanuel Morin, Nathalie Camelin, and Yannick Estève. 2019. Curriculum-based transfer learning for an effective end-to-end spoken language understanding and domain portability. *arXiv preprint arXiv:1906.07601* (2019).

[7] Hongyan Chang and Reza Shokri. 2021. On the privacy risks of algorithmic fairness. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 292–303.

[8] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. 2020. GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 343–362.

[9] Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. 2022. Copy, Right? A Testing Framework for Copyright Protection of Deep Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.

[10] Junjie Chen, Wendy Hui Wang, Hongchang Gao, and Xinghua Shi. 2021. PAR-GAN: Improving the Generalization of Generative Adversarial Networks Against Membership Inference Attacks. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*. ACM, 127–137.

[11] Christopher A. Choquette Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. 2021. Label-Only Membership Inference Attacks. In *International Conference on Machine Learning (ICML)*. PMLR, 1964–1974.

[12] Tianshuo Cong, Xinlei He, and Yang Zhang. 2022. SSLGuard: A Watermarking Scheme for Self-supervised Learning Pre-trained Encoders. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 579–593.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. 2020. Curriculum deepsdf. In *European Conference on Computer Vision*. Springer, 51–67.

[16] 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.

[17] 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.

[18] 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.

[19] Francesco Foglino, Matteo Leonetti, Simone Sagratella, and Ruggiero Seccia. 2019. A gray-box approach for curriculum learning. In *World Congress on Global Optimization*. Springer, 720–729.

[20] Pierre Fournier, Cédric Colas, Mohamed Chetouani, and Olivier Sigaud. 2019. CLIC: Curriculum Learning and Imitation for object Control in non-rewarding environments. *IEEE Transactions on Cognitive and Developmental Systems* (2019).

[21] 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.

[22] 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 *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 619–633.

[23] Amirata Ghorbani and James Zou. 2019. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. PMLR, 2242–2251.

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

[25] Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 7839–7846.

[26] Guy Hacohen. 2019. https://github.com/GuyHacohen/curriculum_learning.

[27] Guy Hacohen and Daphna Weinshall. 2019. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*. PMLR, 2535–2544.

[28] Zayd Hammoudeh and Daniel Lowd. 2022. Training Data Influence Analysis and Estimation: A Survey. *arXiv preprint arXiv:2212.04612* (2022).

[29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778.

[30] Xinlei He, Jinyuan Jia, Michael Backes, Neil Zhenqiang Gong, and Yang Zhang. 2021. Stealing Links from Graph Neural Networks. In *USENIX Security Symposium (USENIX Security)*. USENIX, 2669–2686.

[31] Xinlei He, Rui Wen, Yixin Wu, Michael Backes, Yun Shen, and Yang Zhang. 2021. Node-Level Membership Inference Attacks Against Graph Neural Networks. *CoRR abs/2102.05429* (2021).

[32] Xinlei He and Yang Zhang. 2021. Quantifying and Mitigating Privacy Risks of Contrastive Learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 845–863.

[33] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. 2019. Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models. *Privacy Enhancing Technologies Symposium* (2019).

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

[35] Bargav Jayaraman and David Evans. 2022. Are Attribute Inference Attacks Just Imputation?. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 1569–1582.

[36] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. 2020. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881* (2020).

[37] Hengrui Jia, Christopher A. Choquette-Choo, Varun Chandrasekaran, and Nicolas Papernot. 2021. Entangled Watermarks as a Defense against Model Extraction. In *USENIX Security Symposium (USENIX Security)*. USENIX, 1937–1954.

[38] 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 *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 259–274.

[39] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. 2021. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification?. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8239–8247.

[40] Lu Jiang, Deyu Meng, Qian Zhao, Shiguang Shan, and Alexander G Hauptmann. 2015. Self-paced curriculum learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.

[41] Lu Jiang, Zhengyuan Zhou, Thomas Leung, Li-Jia Li, and Li Fei-Fei. 2018. Mentornet: Learning data-driven curriculum for very deep neural networks on corrupted labels. In *International Conference on Machine Learning*. PMLR, 2304–2313.

[42] Sanjay Kariyappa, Atul Prakash, and Moinuddin K. Qureshi. 2021. MAZE: Data-Free Model Stealing Attack Using Zeroth-Order Gradient Estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 13814–13823.

[43] Kalpesh Krishna, Gaurav Singh Tomar, Ankur P. Parikh, Nicolas Papernot, and Mohit Iyyer. 2020. Thieves on Sesame Street! Model Extraction of BERT-based APIs. In *International Conference on Learning Representations (ICLR)*.

[44] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. Learning multiple layers of features from tiny images. (2009).

[45] M Pawan Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-Paced Learning for Latent Variable Models.. In *NIPS*, Vol. 1. 2.

[46] Alexey Kurakin, Steve Chien, Shuang Song, Roxana Geambasu, Andreas Terzis, and Abhradeep Thakurta. 2022. Toward training at imagenet scale with differential privacy. *arXiv preprint arXiv:2201.12328* (2022).

[47] Ya Le and Xuan Yang. 2015. Tiny imagenet visual recognition challenge. *CS 231N* 7, 7 (2015), 3.

[48] 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.

[49] Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. 2019. How to Prove Your Model Belongs to You: A Blind-Watermark based Framework to Protect Intellectual Property of DNN. In *Annual Computer Security Applications Conference (ACSAC)*. ACM, 126–137.

[50] Zheng Li and Yang Zhang. 2021. Membership Leakage in Label-Only Exposures. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 880–895.

[51] Hongbin Liu, Jinyuan Jia, Wenjie Qu, and Neil Zhenqiang Gong. 2021. EncoderMI: Membership Inference against Pre-trained Encoders in Contrastive Learning. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM.

[52] Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Task-level curriculum learning for non-autoregressive neural machine translation. *arXiv preprint arXiv:2007.08772* (2020).

[53] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Li Mian, Zhaoyu Wang, Jing Zhang, and Jie Tang. 2021. Self-supervised learning: Generative or contrastive. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (2021), 857–876.

[54] Yiyong Liu, Zhengyu Zhao, Michael Backes, and Yang Zhang. 2022. Membership Inference Attacks by Exploiting Loss Trajectory. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 2085–2098.

[55] Yunhui Long, Vincent Bindschaedler, Lei Wang, Diyue Bu, Xiaofeng Wang, Haixu Tang, Carl A. Gunter, and Kai Chen. 2018. Understanding Membership Inferences on Well-Generalized Learning Models. *CoRR abs/1802.04889* (2018).

[56] Reza Lotfian and Carlos Busso. 2019. Curriculum learning for speech emotion recognition from crowdsourced labels. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27, 4 (2019), 815–826.

[57] Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. 2022. SoK: How Robust is Image Classification Deep Neural Network Watermarking?. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE.

[58] Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2019. Teacher–student curriculum learning. *IEEE transactions on neural networks and learning systems* 31, 9 (2019), 3732–3740.

[59] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting Unintended Feature Leakage in Collaborative Learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 497–512.

[60] Fatemehsadat Mireshghallah, Mohammadkazem Taram, Praneeth Vepakomma, Abhishek Singh, Ramesh Raskar, and Hadi Esmaeilzadeh. 2020. Privacy in deep learning: A survey. *arXiv preprint arXiv:2004.12254* (2020).

[61] Sanmit Narvekar and Peter Stone. 2018. Learning curriculum policies for reinforcement learning. *arXiv preprint arXiv:1812.00285* (2018).

[62] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2018. Machine Learning with Membership Privacy using Adversarial Regularization. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 634–646.

[63] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1021–1035.

[64] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. 2011. Reading digits in natural images with unsupervised feature learning. (2011).

[65] Seong Joon Oh, Max Augustin, Bernt Schiele, and Mario Fritz. 2018. Towards Reverse-Engineering Black-Box Neural Networks. In *International Conference on Learning Representations (ICLR)*.

[66] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. 2019. Knockoff Nets: Stealing Functionality of Black-Box Models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4954–4963.

[67] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. 2018. Scalable Private Learning with PATE. In *International Conference on Learning Representations (ICLR)*.

[68] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar. 2018. DeepSigns: A Generic Watermarking Framework for IP Protection of Deep Learning Models. *CoRR abs/1804.00750* (2018).

[69] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature* 323, 6088 (1986), 533–536.

[70] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 2019. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7374–7383.

[71] Ahmed Salem, Apratim Bhattacharya, Michael Backes, Mario Fritz, and Yang Zhang. 2020. Updates-Leak: Data Set Inference and Reconstruction Attacks in Online Learning. In *USENIX Security Symposium (USENIX Security)*. USENIX, 1291–1308.

[72] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. 2019. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. In *Network and Distributed System Security Symposium (NDSS)*. Internet Society.

[73] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 4510–4520.

[74] Yun Shen, Xinlei He, Yufei Han, and Yang Zhang. 2022. Model Stealing Attacks Against Inductive Graph Neural Networks. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 1175–1192.

[75] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership Inference Attacks Against Machine Learning Models. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 3–18.

[76] Ilia Shumailov, Zakhar Shumaylov, Dmitry Kazhdan, Yiren Zhao, Nicolas Papernot, Murat A Erdogdu, and Ross J Anderson. 2021. Manipulating sgd with data ordering attacks. *Advances in Neural Information Processing Systems* 34 (2021), 18021–18032.

[77] Congzheng Song and Ananth Raghunathan. 2020. Information Leakage in Embedding Models. In *ACM SIGSAC Conference on Computer and Communications Security (CCS)*. ACM, 377–390.

[78] Congzheng Song and Vitaly Shmatikov. 2020. Overlearning Reveals Sensitive Attributes. In *International Conference on Learning Representations (ICLR)*.

[79] Liwei Song and Prateek Mittal. 2021. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium (USENIX Security)*. USENIX.

[80] Petru Soviany, Claudiu Ardei, Radu Tudor Ionescu, and Marius Leordeanu. 2020. Image difficulty curriculum for generative adversarial networks (CuGAN). In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 3463–3472.

[81] Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2021. Curriculum learning: A survey. *arXiv preprint arXiv:2101.10382* (2021).

[82] Valentin I Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2009. Baby Steps: How "Less is More" in unsupervised dependency parsing. (2009).

[83] Ritwick Sundar. 2020. https://github.com/rsundar96/curriculum-learning-acceleration.

[84] 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.

[85] Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, and Thomas Ristenpart. 2016. Stealing Machine Learning Models via Prediction APIs. In *USENIX Security Symposium (USENIX Security)*. USENIX, 601–618.

[86] Trusted-AI. 2023. https://github.com/Trusted-AI/adversarial-robustness-toolbox.

[87] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. 2017. Embedding Watermarks into Deep Neural Networks. In *International Conference on Multimedia Retrieval (ICMR)*. ACM, 269–277.

[88] Binghui Wang and Neil Zhenqiang Gong. 2018. Stealing Hyperparameters in Machine Learning. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 36–52.

[89] Xin Wang, Yudong Chen, and Wenwu Zhu. 2021. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[90] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2022. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *International Conference on Learning Representations (ICLR)*.

[91] Daphna Weinshall, Gad Cohen, and Dan Amir. 2018. Curriculum learning by transfer learning: Theory and experiments with deep networks. In *International Conference on Machine Learning*. PMLR, 5238–5246.

[92] Fan Wu, Yunhui Long, Ce Zhang, and Bo Li. 2022. LinkTeller: Recovering Private Edges from Graph Neural Networks via Influence Analysis. In *IEEE Symposium on Security and Privacy (S&P)*. IEEE, 2005–2024.

[93] Xiaoxia Wu, Ethan Dyer, and Behnam Neyshabur. 2021. When Do Curricula Work?. In *International Conference on Learning Representations*. https://openreview.net/forum?id=tW4QEInpni

[94] Mohammad Yaghini, Bogdan Kulynych, and Carmela Troncoso. 2019. Disparate Vulnerability: on the Unfairness of Privacy Attacks Against Machine Learning. *CoRR abs/1906.00389* (2019).

[95] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 1–23.

[96] Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. 2020. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 253–261.

[97] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age progression/regression by conditional adversarial autoencoder. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 5810–5818.

[98] Siqi Zheng, Gang Liu, Hongbin Suo, and Yun Lei. 2019. Autoencoder-based semi-supervised curriculum learning for out-of-domain speaker verification. *System* 3 (2019), 98.

[99] Da Zhong, Haipei Sun, Jun Xu, Neil Gong, and Wendy Hui Wang. 2022. Understanding disparate effects of membership inference attacks and their countermeasures. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. 959–974.

[100] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence* 40, 6 (2017), 1452–1464.

[101] Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 6934–6944.

A Datasets

MIA Datasets. We use the following 8 datasets for MIA evaluation, which are also adopted by previous work [32, 51, 60, 75] to study MIA. They are CIFAR100 [44], Tiny ImageNet [47], Place100, Place 60[100], SVHN [64], Purchase[75], Texas hospital stays[75] and Locations [95]. We focus on image datasets mainly (the first 5 datasets), but tabular datasets are also evaluated. Below are the detailed descriptions for the datasets.

- **CIFAR100 [44].** This dataset consists of 60,000 colored images in 100 classes, with 600 images per class. The size of each image is 32×32.
- **Tiny ImageNet [47].** This is a subset of the ImageNet dataset[13]. It contains 100,000 colored images of 200 classes (500 for each class). The size of each image is 64×64.
- **Place100.** This dataset is a subset of Places365[100] dataset, which is composed of more than 1.8 million images with 365 scene categories. Place100 is generated by randomly selecting 100 scene categories with 600 random images per category.
- **Place60.** This dataset is similar to Place100, except that it has 60 classes containing 1,000 images each.
- **SVHN [64].** The Street View House Numbers (SVHN) dataset is a real-world image dataset containing over 600,000 digit images. This dataset includes images of house numbers taken from Google Street View images. The size of each image is 32×32.
- **Purchase.** This is a tabular dataset about purchase styles. Following Shokri et al. [75], we leverage the Purchase-100 dataset (abbreviated as Purchase) and use 10,000 records for training. The dataset itself contains 197,324 records from 100 classes, where each record has 600 binary features.
- **Texas hospital stays.** This dataset contains information about inpatient stays in several health facilities. Following Shokri et al. [75], our task is to predict a patient's main procedure. After pre-processing, the resulting dataset has 67,330 records and 6,170 binary features.
- **Locations [95] .** The original dataset was released by Foursquare about its mobile users' location "check-ins", which has 11,592 users and 1,136,481 check-in records. Our task is to predict the user's geo-social type (128 in total). Here we use the version pre-processed by Shokri et al. [75], which has 446 binary features.

AIA Datasets. Datasets with multiple attributes are required for AIA. To this end, we adapt Place100 and Place60 used as MIA datasets to AIA setting as they both contain multiple attribute labels. More specifically, the model for Place100 outputs whether a sample is an indoor scene, while the sensitive attribute is the category of the scene, which contains 100 labels. Place60 has the total number of categories as 60. In addition to Place100 and Place60, we introduce UTKFace [97] specifically for AIA study.

- **UTKFace [97].** This is a large-scale facial dataset, which consists of over 20,000 face images with annotations of age, gender, and ethnicity. In our evaluation, we set gender classification as the the task for target model, and the sensitive attribute to be inferred is ethnicity, which includes 5 classes.

B More Results of CL

Training Accuracy. Training accuracy corresponding to datasets in Table 1 are listed in Table 7. All numbers are in percentage.

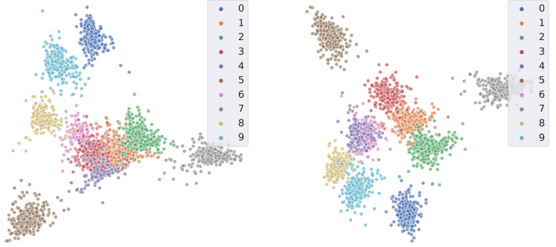
t-SNE Study. To investigate the disparate impact CL has on the classification accuracy across samples. we use t-distributed stochastic neighbor embedding (t-SNE) to visualize the classification tasks carried out by bootstrapping and normal ML on the most difficult batch of data of SVHN. Figure 10 shows all samples within the difficult batch, and it turns out bootstrapping can separate samples from group "1", "2" and "3" better than normal training.

C More Confidence Scores of MIA

Here we present the confidence scores of different MIA evaluation results to supplement Section 5. In particular, Figure 11, Figure 12, and Figure 13 present the results about the three image datasets including SVHN, Place100 and Place60. Figure 14 presents the results about the tabular dataset Purchase. Figure 15 and Figure 16 present the results of Diff-Cali on Tiny ImageNet and CIFAR100.

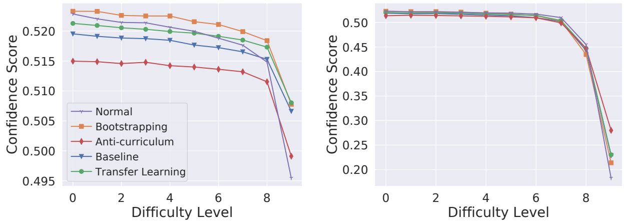
Method \ Dataset	Tiny ImageNet	CIFAR100	Place100	Place60	SVHN	Purchase	Texas	Location
Normal	100.0	100.0	100.0	100.0	100.0	100.0	96.770	100.0
Bootstrapping	100.0	100.0	100.0	99.996	100.0	100.0	94.030	100.0
Transfer	100.0	99.997	100.0	99.972	100.0	/	/	/
Baseline	100.0	99.993	100.0	100.0	100.0	99.990	95.600	100.0
Anti-curriculum	99.963	100.0	100.0	99.918	100.0	100.0	97.410	100.0

Table 7: The average training accuracy of datasets in Table 1. Image datasets are trained on ResNet-18 while non-image datasets are trained on MLP. Numbers are all in percentage. We observe that all training accuracies are nearly 100%. Note that for non-image datasets, we skip the transfer method as there is no commonly used pre-trained model for the tabular dataset.



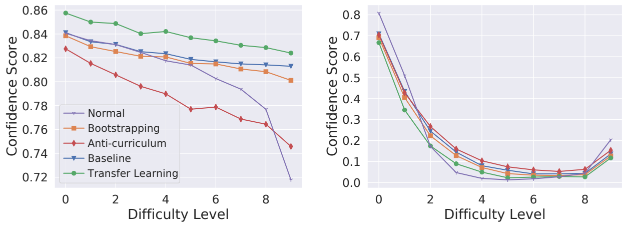
(a) Normal training **(b) Bootstrapping**

Figure 10: t-SNE of the classification results on the difficult batch of SVHN.



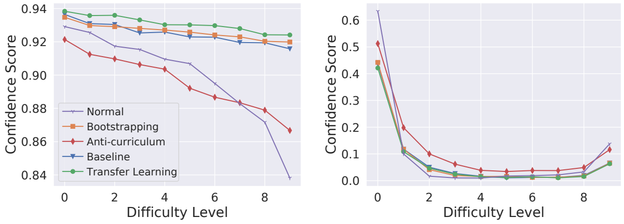
(a) SVHN member **(b) SVHN non-member**

Figure 11: Attack model's confidence score for both member and non-member samples on SVHN. ResNet-18 is used for target model training, and data samples are arranged according to their difficulty scores from bootstrapping.



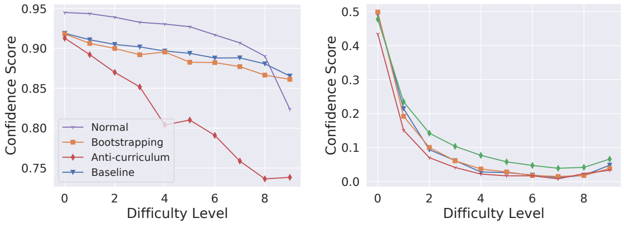
(a) Place60 member **(b) Place60 non-member**

Figure 13: Attack model's confidence score for both member and non-member samples on Place60. ResNet-18 is used for target model training, and data samples are arranged according to their difficulty scores from bootstrapping.



(a) Place100 member **(b) Place100 non-member**

Figure 12: Attack model's confidence score for both member and non-member samples on Place100. ResNet-18 is used for target model training, and data samples are arranged according to their difficulty scores from bootstrapping.

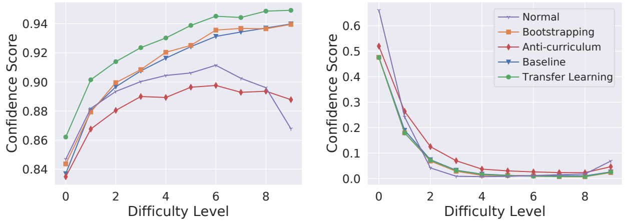


(a) Purchase member **(b) Purchase non-member**

Figure 14: Attack model's confidence score for both member and non-member samples on Purchase. MLP is used for target model training, and data samples are arranged according to their difficulty scores from bootstrapping.

D Difficulty Level vs. Shapley Value

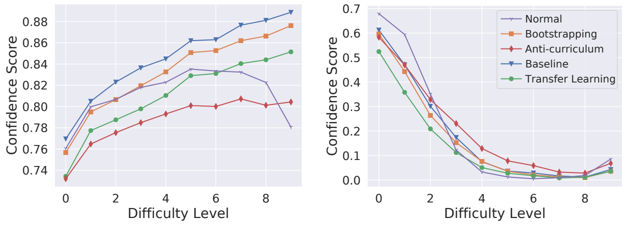
In Section 5.2, we show there is a strong tie between data memorization with difficulty level, which explains why the difficult samples are more vulnerable under CL. On the other hand, samples of different difficulty levels could provide different values to the model, so we are also interested in whether this observation Section 5.2 can be explained from the angle of data valuation. Specifically, we choose Shapley value [23] as the metric, as it has the "strongest theoretical foundation" in data valuation research [28]. In essence, the data with high Shapley values are ones that on average contribute



(a) Member

(b) Non-member

Figure 15: Diff-Cali's member and non-member confidence score for models trained on Tiny ImageNet with ResNet-18.



(a) Member

(b) Non-member

Figure 16: Diff-Cali's member and non-member confidence score for models trained on CIFAR100 with ResNet-18.

significantly to a model's prediction performance. We follow most of the experiment steps in this section and only change how the samples are selected for "not seen" (i.e., selected based on their Shapley values rather than difficulty levels).

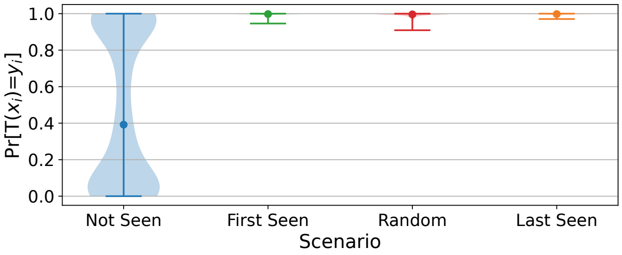


Figure 17: Shapley: violin plots of prediction probability of 800 most valuable samples according to KNN-Shapley.

KNN-Shapley. Calculating Shapley values is intractable for a DNN model that is trained on a large dataset, as it requires a model to be retrained for 2^n times, where n is the number of data points, to assess the contribution of one data point versus all possible subsets of the training set [28]. To address this scalability issue, Jia et al. [39] proposed KNN-Shapley, which uses a lightweight KNN *surrogate* model to reduce the overhead of model retraining. The time complexity is reduced to $O(nlogn)$ and still, a good approximation of Shapley values can be obtained. As such, we use KNN-Shapley to calculate the Data Shapley values.

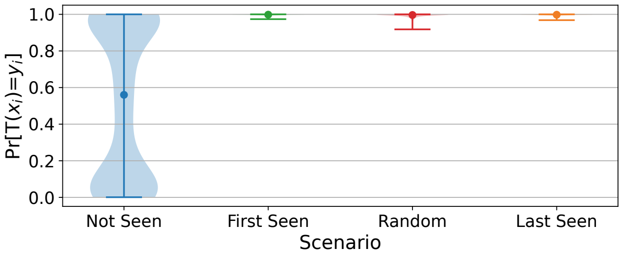


Figure 18: Reverse Shapley: violin plots of prediction probability of 800 least valuable samples according to KNN-Shapley.

Figure 17 and Figure 18 show the prediction probability of true label with 800 most and least valuable data samples according to KNN-Shapley. From the results of "not seen", we observe that the least valuable data have higher prediction accuracy on average (51%), meaning that their absence in training has less impact compared to the more valuable data as presented in Figure 17. Similarly, feeding the least valuable data first or at last to the training does not affect the prediction much.

Then, we compare the impact of difficulty level and Shapley value on data memorization, from Figure 7 and Figure 17. Though both show that the absence of the most difficult or valuable data leads to poor prediction and seeing these data lastly benefits more than seeing them first during training, these changes are much more drastic for difficult samples (Figure 7) than the valuable samples (Figure 17). For example, the median prediction probability of the "not seen" difficult samples and valuable samples are 39.19% and 56.01%. As such, the data reordering of CL makes the difficult samples more vulnerable, but not so for the valuable samples.

E AIA

In this section, we describe our setup of AIA and the evaluation result.

Basic AIA method. Song et al. proposed an inference-time attack and model-repurposing attack [78] for AIA, and here we focus on the first attack and follow the same setting as this work. We consider the model evaluation to be partitioned [78] or the model is trained under federated learning [59]. The target model T is split into two parts, i.e., an encoder and a classifier, and the adversary has black-box access to the encoder E. The attacker has an auxiliary dataset D containing pairs of (x, s) where s is the sensitive attribute. The embeddings h can be generated by querying E, i.e., $h = E(x), \forall x \in D$. All pairs of (h, s) will be used to train the attack model $\mathcal{A}_{AI}$ and later used to predict the values of s in the target model T.

AIA Model. Our $\mathcal{A}_{AI}$ is a 3-layer MLP with 128 hidden neurons in each hidden layer. We use cross-entropy as the loss function and SGD as the optimizer with a learning rate of 0.01. The attack model is trained for 100 epochs. The dimension of each sample's embedding (i.e., second to the last layer's output) is 512 for ResNet-18, 512 for ResNet-34, and 1024 for MobileNet. To train the target model T, we use the label for the original classification (e.g., gender). To train $\mathcal{A}_{AI}$, we use the label from another field (e.g., race).

Method \ Dataset	Place100	Place60	UTKFace
Normal	**0.107**±0.003	**0.173**±0.002	**0.528**±0.005
Bootstrapping	0.092±0.003	0.168±0.004	0.515±0.006
Transfer Learning	0.104±0.001	0.150±0.005	0.512±0.006
Baseline Curriculum	0.079±0.004	0.143±0.001	0.506 ±0.008
Anti-Curriculum	0.033±0.001	0.128±0.005	0.517±0.007

Table 8: Average accuracy of AIA (± standard deviation) on model trained with different methods. ResNet-18 is the target model architecture.

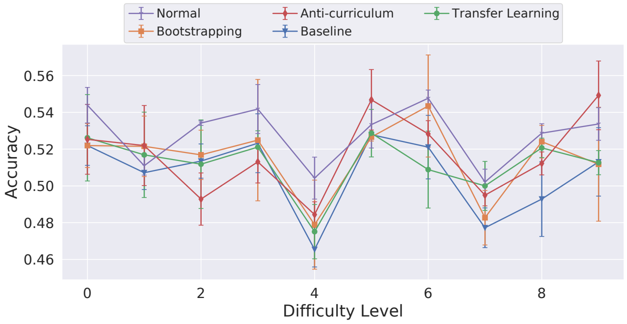


Figure 19: Attribute inference attack accuracy on UTKFace

Evaluation of AIA. We split the AIA datasets in the same way as the MIA evaluation as described in Section 5 (Evaluation Setup). We evaluate the 4 CL methods and normal training under the AIA setting as described above. Table 8 demonstrates the overall attack accuracy. Generally, our results indicate that CL does not make the target model more vulnerable. This somehow contradicts a study [32] showing that a model is more vulnerable under AIA when trained under special settings, i.e., contrastive learning. Interestingly, the normal training yields the highest average attack accuracy (e.g., 0.107 for Place100), even compared to anti-curriculum. UTKFace has a much higher attack accuracy because the baseline accuracy (random guessing based on majority class labels) of UTKFace is already quite high (42.1%). Our further investigation also shows that the attack accuracy is about the same for samples in different groups of difficulty levels (Figure 19). We speculate that this is because the attributes of a sample themselves are already very complex and hard to learn. Besides, the difficulty score (e.g., bootstrapping) is calculated based on the original ML task, which emphasizes the specific attribute the original ML task tries to learn. That means the data ranking is effective only for the attribute chosen for the classification task but does not influence the sensitive attribute that one intends to infer.

Finding 6: The model trained under CL is less vulnerable under AIA compared to MIA.