Membership Inference Attack Using Self Influence Functions

Gilad Cohen
Tel Aviv University
giladco1@post.tau.ac.il

Raja Giryes
Tel Aviv University
raja@tauex.tau.ac.il

Abstract

Member inference (MI) attacks aim to determine if a specific data sample was used to train a machine learning model. Thus, MI is a major privacy threat to models trained on private sensitive data, such as medical records. In MI attacks one may consider the black-box settings, where the model's parameters and activations are hidden from the adversary, or the white-box case where they are available to the attacker. In this work, we focus on the latter and present a novel MI attack for it that employs influence functions, or more specifically the samples' self-influence scores, to perform MI prediction. The proposed method is evaluated on CIFAR-10, CIFAR-100, and Tiny ImageNet datasets using various architectures such as AlexNet, ResNet, and DenseNet. Our new attack method achieves new state-of-the-art (SOTA) results for MI even with limited adversarial knowledge, and is effective against MI defense methods such as data augmentation and differential privacy. Our code is available at https://github.com/giladcohen/sif_mi_attack.

1. Introduction

Machine learning (ML) algorithms have advanced tremendously over the past decade and have been commonly used for a variety of tasks, including privacy sensitive applications, such as medical imaging [3, 25], conversations [9], face recognition [40], and financial information [8]. Most of these models are trained using sensitive user data which can be leaked later by an adversary from the models' parameters [37].

Membership inference (MI) attacks aim to infer whether a specific sample was used to train a target ML model. This information can be detrimental if it falls to the wrong hands. For example, consider an ML model trained on blood tests of HIV patients, for predicting their reaction to a Covid-19 vaccine. If an adversary somehow obtains a patient's medical record, she can only observe the patient's blood reading and query the model for the predicted reaction, but she cannot deduce if the patient has HIV. However, if the adversary infers that the record was used to train the model, then she would know the patient has HIV. If this adversary is a health insurance company, it might increase the patient's insurance premium.

Many MI attacks make use of the class probability vector (or logits) at the output of the target model [22, 33, 37, 42], since deep neural networks (DNNs) often tend to exhibit over-confidence for samples from their training set [31], a phenomenon that is largely attributed to overfitting [42]. More recent studies do not assume access to model probability vectors and still achieve state-of-the-art (SOTA) MI accuracy by relying on the final predicted labels at the model output [5, 24].

MI attacks can operate under two threat model settings: *white-box* or *black-box*. The white-box setting assumes that the adversary has full information about the target model's architecture, parameters, activations, training process, and training data distribution. On the other hand, the black-box setting is more restrictive, allowing the adversary access only to the target model's input and output. All the aforementioned MI attacks use the black-box setting. Other works assumed a white-box setting and tried to exploit other information from the target model [20, 31, 32], however their white-box methods could not achieve a significant improvement in the MI prediction accuracy compared to black-box attacks.

Contribution. In this work we introduce a novel white-box MI attack that can be applied to any ML model. The core idea of our attack model is that training samples have a direct influence on the loss of test samples, but not vice versa. For quantifying this effect, we use influence functions [15], which determines how data points in the training set influence the target model's prediction for a given test sample. This measure quantifies how much a small upweighting of a specific training point in the target model's empirical error affects the loss of a test point. To speed up computation time, we utilize the self-influence function of a sample point on it own loss.

Given a sample point, we calculate its self-influence func-

tion (SIF[1]) score, and query the target model for its label prediction. These two values alone are sufficient to infer if the sample belongs to the training set. Our attack model makes use of only two parameters and thus exhibits fast inference time. We evaluate our MI attack on several datasets trained on various target models with different architectures, showing its advantage over current SOTA attacks. Moreover, we also consider the MI defense of training with data augmentations, which is a common practice in neural network training, and present an adaptive attack model that negates it. Specifically, we introduce the adaptive SIF (adaSIF), which takes into account also the used augmentations in its calculation.

2. Related work

Membership inference. Shokri et al. were the first to propose an MI attack against ML models [37]. Their attack model includes a bundle of "shadow models" which are trained to mimic the classification output vector of a black-box target model h, for training (*members*) and test (*non-members*) samples. These shadow models are then used to generate a shadow dataset. For a given shadow model S_h^k and a sample $(\mathbf{x}_i^k, y_i^k)$, where $\mathbf{x}_i^k$ is an input and y_i^k is its label, they predict the output vector $\mathbf{y} = S_h^k(\mathbf{x})$ and save the record $(y_i^k, \mathbf{y_i^k}, m_i)$, where m_i equals 1 if $(\mathbf{x}_i^k, y_i^k)$ is a member and 0 otherwise. The shadow dataset $\left\{(y_i^k, \mathbf{y_i^k}, m_i)\right\}_{1\le i\le n}^{1\le k\le p}$ obtained from n samples and p shadow models is utilized to train a binary classifier as an attack model for MI prediction.

The aforementioned attack requires training the S_h^k models on similar architecture as h, with samples distributed similarly to the training set of h. Salem et al. later showed that the exact architecture knowledge is not needed, and any sample distribution of a similar task (e.g., vision task) is sufficient [33]. Moreover, they achieved a comparable MI attack performance using a single shadow model.

Yeom et al. showed that overfitted target models are necessarily vulnerable to MI attacks [42], and proposed a simple baseline heuristic that predicts a sample $z = (x, y)$ to be a member if the target model prediction $\hat{y} = h(x)$ matches y, and a non-member otherwise. This baseline is named the "Gap attack" since its accuracy is correlated with the generalization error, which is the gap between h accuracy on the training data (A_{mem}) and the held out data (A_{out}):

$$\frac{1}{2} + \frac{1}{2}\big(A_{mem} - A_{out}\big), \quad \text{where } A_{mem}, A_{out} \in [0, 1].$$

As an attempt to mitigate MI attacks, several defenses were proposed to alter h output confidence vector [13, 27], however recent works presented SOTA MI attack performance on black-box models that only output hard labels, without accessing the class posterior probabilities [5, 24]. To that end, they applied a black-box adversarial attack [4, 21] on the input image x image until its label y was flipped, and inspected the L_2 distance $d = ||x - x'||_2$ where x' is the adversarial image. Next, they predicted the sample (x, y) to be a member if $d > \tau$ for some threshold τ.

Sablayrolles et al. explored MI attacks in a white-box setting [32]. They showed that optimal membership inference only depends on the loss function, and thus claimed that white-box attacks cannot perform better than black-box attacks. Rezaei and Liu also assumed white-box setting and utilized hidden layers activations and gradient norms in their attack models, and observed only a marginal improvement compared to the black-box attack baseline [31].

Leino and Fredrikson constructed white-box MI attacks that can be calibrated for its output confidences [20] (the member/non-member classes) and showed that they can obtain higher precision than a black-box attack. However, tuning the MI attack for precision greatly reduced their recall score. Our work shows that white-box information can assist the adversary and perform SOTA MI, without sacrificing the member recall or the accuracy on the non-member class.

Nasr et al. utilized a white-box attack that trains a DNN attack model on features collected from all the target model layers, for both the forward pass (activations) and backward pass (gradients) [26]. Their approach surpassed the performance of a baseline black-box. We show that our attack method achieves even superior results on CIFAR-100 [17] using their target model training setup.

Defense against MI attacks. Multiple defenses were proposed against MI attacks for ML models. Regularization defenses reduce the overfitting of target models, lowering the gap between train and test accuracy. Such defenses are data augmentation [14], L_2 weight regularization [37], and early stopping [38]. Based on knowledge distillation [11], Shejwalkar & Houmansadr [36] proposed the Distillation for Membership Privacy (DMP) defense which first trains a teacher model and uses it to predict an unlabeled reference dataset. Next, only the predictions with the lowest entropy are selected to train the final target model. Such training records are classified more easily and thus reduce membership information leakage. Differential Privacy (DP) in DNNs [1] avoids overfitting of the model parameters by clipping the gradients and adding Gaussian noise to them in the backward pass during training. DP has been shown to mitigate MI attacks on ML models [5, 30].

Influence functions. Koh and Liang proposed to interpret the predictions of an ML model by tracing them through its learning algorithm and training data [15]. They quantify the influence a train sample z_{train} has on a specific loss value of a test sample z_{test}. Aside from interpretability, this measure had been shown to improve classifier training [35], defend against adversarial attacks [6], and fix mislabeled training data [16].

[1] SIF refers both to the self-influence function score and the attack model that is based on it interchangeably, depending on the context.

The disadvantage of influence functions is that their computation is computationally demanding. To mitigate that, we use the self-influence measure, which calculates the influence an example has on itself and has been used to fix erroneous training labels [29, 34]. It allows us to perform the MI attack in a computationally efficient manner.

3. Method

In order to describe our approach, we start by formally defining influence functions in general and their derived self-influence functions (SIF) that we use in the paper. Next, we introduce our proposed SIF attack model for neural networks that have been trained without data augmentation. Lastly, we modify our approach to attack target models that are trained with data augmentation.

We study a classification task from an input space $\mathcal{X}$ (e.g., images) to an output space $\mathcal{Y}$ (e.g., labels). For a sample point $z = (x, y) \in \mathcal{X} \times \mathcal{Y}$ and model parameters θ, we denote the loss by $L(z, \theta)$. Let $\{z_1, ..., z_n\}$ be a training set of size n, and let $\frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$ be the empirical risk. The empirical risk minimizer is defined by $\hat{\theta} \overset{\text{def}}{=} \underset{\theta}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta)$. We assume that the empirical risk has first and second gradients and it is strictly convex in θ.

3.1. Influence functions

We study the change in model parameters due to upweighting a specific training sample z by a small ϵ in the training process. Upweighting z adjusts the model parameters to be $\hat{\theta}_{\epsilon,z} \overset{\text{def}}{=} \underset{\theta}{\operatorname{argmin}} \frac{1}{n}\sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$. Cook and Weisberg [7] showed that the influence function of upweighting z on the model parameters $\hat{\theta}$ is given by

$$I_{up,params}(z) \overset{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}), \quad (1)$$

where $H_{\hat{\theta}} = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta^2 L(z_i, \hat{\theta})$ is the Hessian.

Influence functions interpret an ML model by indicating which of the training samples assisted it to make its prediction, and which training samples were destructive, i.e., inhibited the model from its prediction. Koh and Liang [15] proposed to measure the influence a train sample z_{train} has on the loss of a test sample z, using the term:

$$\begin{aligned} I_{up,loss}(z_{train}, z) &\overset{\text{def}}{=} \left.\frac{dL\left(z, \hat{\theta}_{\epsilon,z_{train}}\right)}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_\theta L(z, \hat{\theta})^T \left.\frac{d\hat{\theta}_{\epsilon,z_{train}}}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_\theta L(z, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(z_{train}, \hat{\theta}). \end{aligned} \quad (2)$$

The influence function $I_{up,loss}(z_{train}, z)$ measures how much the test loss $L(z, \hat{\theta})$ would change if we were to "upweight" the training sample z_{train} in the empirical risk. The influence function is composed of three components: the gradient of the training sample z_{train}, the gradient of the test sample z, and the "similarity" of these samples with respect to the model perspective that is expressed by the term $H_{\hat{\theta}}^{-1}$, which is a positive definite matrix. In the influence functions formulation, larger gradients and similarity are correlated to larger influence.

3.2. SIF values

Our goal is to build an attack model that is a binary classifier that predicts whether a sample was used to train the target model or not. The hypothesis that underlines our approach is that if an image has been used to train an ML target model, then it would have a large influence measure on test images' loss with the same label. If so, in order to infer whether a specific image is a member (used in training), we need to examine its influence measure (Eq. (2)) on other images with the same label.

Given an unseen sample $z = (x, y)$ (either member or non-member) and a set of samples known to be non-members $\{z_1, \ldots, z_m\}$ with the same label y, a rigorous influence function analysis requires applying Eq. (2) to every pair (z, z_i) for $1 \le i \le m$, and inspecting the m obtained influence measures. Alas, the expression $I_{up,loss}(z, z_i)$ requires calculating a Hessian vector product (HVP) and thus it is not scalable for large datasets due to the large computational cost. To make our attack model practical with low computational time, we propose a faster approach that utilizes only the influence of a sample z on itself by merely calculating its SIF measure:

$$I_{SIF}(z) = -\nabla_\theta L(z, \hat{\theta})^T H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}). \quad (3)$$

This measure stands for the influence a single sample point has on its own loss. We calculate $I_{SIF}(z)$ and classify z as member if it satisfies the conditions: (i) $I_{SIF}(z) \in [\tau_{min}, \tau_{max}]$ and (ii) $y = \hat{y}$, where $\hat{y}$ is the prediction of the target model and τ_{min}, τ_{max} are some thresholds. If any of (i) or (ii) is violated, then we classify z as non-member. The pseudo code of our attack model training (τ_{min}, τ_{max}) and inference is detailed in Section 3.3.

Notice that our framework operates in the white-box setting, requiring access to the model's parameters, activations, and to its first/second order gradients. Therefore, it is not a label-only attack and cannot be applied to black-box models.

3.3. SIF MI attack model

Our attack model fits only two parameters, τ_1 and τ_2 which denote the SIF value range a sample can be considered as a "member". For every sample in the training set $\mathcal{D}_{mem}^{train}$ or $\mathcal{D}_{non-mem}^{train}$ (defined in Section 3.2), we collect the I_{SIF} measure (Eq. (3)) together with a variable m that indicates

Target Model	$\mathcal{M}$-1	$\mathcal{M}$-2	$\mathcal{M}$-3	$\mathcal{M}$-4	$\mathcal{M}$-5	$\mathcal{M}$-6	$\mathcal{M}$-7
$\lvert\mathcal{D}_{mem}\rvert$	100	1000	5000	10000	15000	20000	25000

Table 1. The number of the training set size $\lvert\mathcal{D}_{mem}\rvert$ for each of the target models.

if the target model h predicted the same class as the ground truth label. These values are then used to calculate τ_1 and τ_2.

We aim to find an interval (τ_1, τ_2) that best encapsulates only the members, i.e., we want to have that most of the members' SIF values are inside (τ_1, τ_2) and most of the non-members' SIF values are outside this range. For every candidate pair τ_1, τ_2 we calculate the balanced accuracy as defined in Eq. (5). The optimal threshold pair is selected based on a maximization of the balanced accuracy on the training set.

In inference time, given a target model h and data sample $z = (x, y)$, we calculate the SIF value s and query h for its class prediction. If both conditions are met: (i) $s \in (\tau_1, \tau_2)$ and (ii) $y = \hat{y}$ (where $\hat{y} = h(x; \theta)$), then we predict z as a member. Otherwise, z is predicted as a non-member.

Detailed pseudo codes of the target model fitting and inference appear in the supp. mat.

3.4. Adaptive attack to augmentations

The SIF attack model assumes that a given sample z either belongs to the training set (member) or not (non-member). Alas, most computer vision training schemes employ data augmentation. Thus, the target model might have been introduced to some transformations of the image x, instead of the original image. Training with augmentation can be considered as a defense against our SIF-based method since Eq. (3) assumes that a data point z remains unchanged in the training process. Thus, we propose an adaptation to SIF (Eq. (3)) to better estimate the influence of a training sample z on itself, assuming that z is augmented during training.

Calculating the Hessian and its inverse for a DNN is too expensive due to the millions of parameters involved. Note that for n training points and $\theta \in \mathbb{R}^p$, this calculation has a complexity of $\mathrm{O}\left(\mathrm{np}^2 + \mathrm{p}^3\right)$. To overcome this problem, we avoid the explicit calculation of H_θ^{-1} and use HVPs with stochastic estimation, as proposed by [15]. Specifically, we approximate the vector $\mathrm{s}(z) = H_\theta^{-1} \nabla_\theta L(z, \theta)$ using a stochastic estimation method proposed by [2] and then rewrite Eq. (3) as:

$$I_{SIF}(z) = -\mathrm{s}(z) \cdot \nabla_\theta L(z, \theta).$$

With this approximation at hand, we turn to describe our adaptive attack to augmentations, adaSIF. Let $z = \left(x, y\right)$ denote the original training sample and I be a random data augmentation operator sampled from the family of training augmentation distribution $\mathcal{T}$ $\left(I \sim \mathcal{T}\right)$. Then, we define the adaptive self-influence measure of z on Eq. (3) as:

$$I_{adaSIF}(z) = -\mathrm{s}(z) \cdot \mathbb{E}_{I \sim \mathcal{T}} \Big[\nabla_\theta L\big(I(x), y, \theta\big) \Big]. \quad (4)$$

Note that in adaSIF, we average the influence of different augmentations of z on itself. For calculating the term $\mathbb{E}_{I \sim \mathcal{T}} \Big[\nabla_\theta L\big(I(x), y, \theta\big) \Big]$, we followed the same implementation of [15], but instead of sampling the training set samples (the goal in [15] was to check the influence of the training examples on z), we sampled different augmentations of z, $I(z)$, as our goal is to check the influence of the augmentations on z. We compared adaSIF with a naive ensemble of SIF measures calculated on data augmentations assemble and found that adaSIF is slightly better in most cases. More details on adaSIF and the naive ensemble are in the supp. mat.

4. Experimental setup

Here we list the seven target models we used for evaluating our work, provide technical details on how they were trained, and describe the dataset split done to fit our attack model and present the balanced accuracy metric used to compare between all attack models. The hardware apparatus used in our experiments is detailed in the supp. mat.

4.1. Target model and implementation details

Since overfitted machine learning models are more susceptible to membership leakage [33,37,39], we trained seven different target models $\mathcal{M}$-1, ..., $\mathcal{M}$-7, where each model differs only by the training set size. A similar target model setup was also utilized in previous works [24,41]. The sizes of the target models are summarized in Table 1. The Tiny ImageNet [19] dataset was not evaluated on $\mathcal{M}$-1 since it has 200 labels which exceed $\mathcal{M}$-1 training set size.

We split the full official training set of CIFAR-10, CIFAR-100, and Tiny ImageNet into *training* and *validation* sets. The *training* size is set by Table 1 and *validation* was set to 5% of the official training set. Three DNN architectures were used in our experiments to train the target models: Resnet18 [10], AlexNet [18], and DenseNet [12]. We applied ReLU activations for all models and optimized the cross entropy loss while decaying the learning rate using the *validation* set's accuracy score, for 400 epochs, batch size 100, with L_2 weight regularization of 0.0001, using a stochastic gradient decent optimizer with momentum 0.9 and Nesterov updates. For the data augmentation adaptive attack in Section 5.4 we trained the target models with random crop and horizontal flipping.

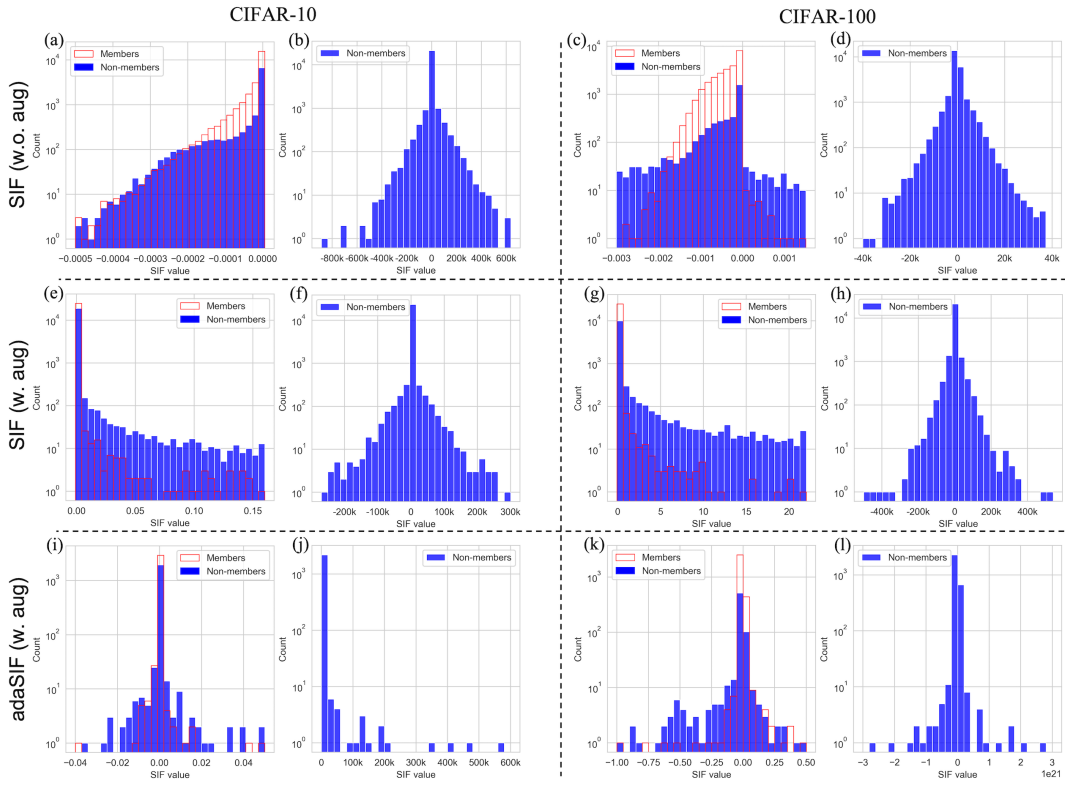

Figure 1. SIF and adaSIF values distribution for CIFAR-10 and CIFAR-100 images within the training set (*members*) and images outside it (*non-members*) for target model $\mathcal{M}$-7 trained on Resnet18. The top and middle rows correspond to SIF values (Eq. (3)) obtained from models trained without and with data augmentation, respectively. The bottom row corresponds to adaSIF values (Eq. (4)) obtained from a model trained with data augmentations. All members reside in a short range near 0; non-members seldom share the same distribution in this range, and they can attain extreme values. This information is exploited by our attack. (e) and (g) show that data augmentations expands the SIF values for members and mitigate our vanilla SIF attack. (i) and (k) show that applying adaSIF can recover the short range property for the members and therefore facilitate the attack.

We applied early stopping by selecting the model checkpoint with the best (highest) accuracy on the validation set. For the DP training in Section 5.5 we used DP-RMSProp [23] on $\mathcal{M}$-7. The full DNN *training*, *validation*, and official test accuracies of the target models are reported in the supp. mat.

4.2. Attack model training and evaluation

To train and evaluate our SIF attack model, we split each dataset into $\mathcal{D}_{mem}$ and $\mathcal{D}_{non-mem}$ subsets. The former is the *training* set defined in Section 4.1, whereas the latter holds only images that are outside the *training* and *validation* sets. $\mathcal{D}_{mem}$ and $\mathcal{D}_{non-mem}$ were further divided to $\mathcal{D}^{train}_{mem}$, $\mathcal{D}^{train}_{non-mem}$, and $\mathcal{D}^{test}_{mem}$, $\mathcal{D}^{test}_{non-mem}$, where the first two subsets were used to fit the attack models and the last two subsets were used to evaluate membership inference by the attack models. For simplicity, we matched the test set size to the training set size. More explicitly we set $\left|\mathcal{D}^{train}_{mem}\right| = \left|\mathcal{D}^{test}_{mem}\right| = \left|\mathcal{D}^{train}_{non-mem}\right| = \left|\mathcal{D}^{test}_{non-mem}\right|$.

The attack model's thresholds τ_{min}, τ_{max} (Section 3.2) are chosen to optimize the Balanced accuracy in Eq. (5) on $\mathcal{D}^{train}_{mem}$ and $\mathcal{D}^{train}_{non-mem}$, similarly to [24]. The threshold choosing algorithm is provided in the supp. mat. Since the SIF attack requires setting two thresholds, we choose to evaluate our MI attack using balanced accuracy as done in [5, 31] instead of the AUC of the ROC curve. We denote $N_1 = \left|\mathcal{D}^{test}_{mem}\right|$, and $N_2 = \left|\mathcal{D}^{test}_{non-mem}\right|$. Our MI test samples are denoted as $\mathcal{D}^{test}_{mem} = \left\{(x^1_m, y^1_m), ..., (x^{N_1}_m, y^{N_1}_m)\right\}$ and $\mathcal{D}^{test}_{non-mem} = \left\{(x^1_{nm}, y^1_{nm}), ..., (x^{N_2}_{nm}, y^{N_2}_{nm})\right\}$, where x denotes an image and y_m, y_{nm} labels denote member (1), non-member (0) labels, respectively. The balanced accuracy of an attack model is then defined by:

$$\text{Balanced Acc} = \frac{1}{N_1 + N_2}\left[\sum_{i=1}^{N1} \hat{y}^i_m + \sum_{i=1}^{N2}\left(1 - \hat{y}^i_{nm}\right)\right], \tag{5}$$

where $\hat{y}^i_m$ and $\hat{y}^i_{nm}$ are the attack model's predictions for y^i_m and y^i_{nm}, respectively.

For the baseline comparison, in our experiments we used the Gap, Black-box, and Boundary distance MI attacks im-

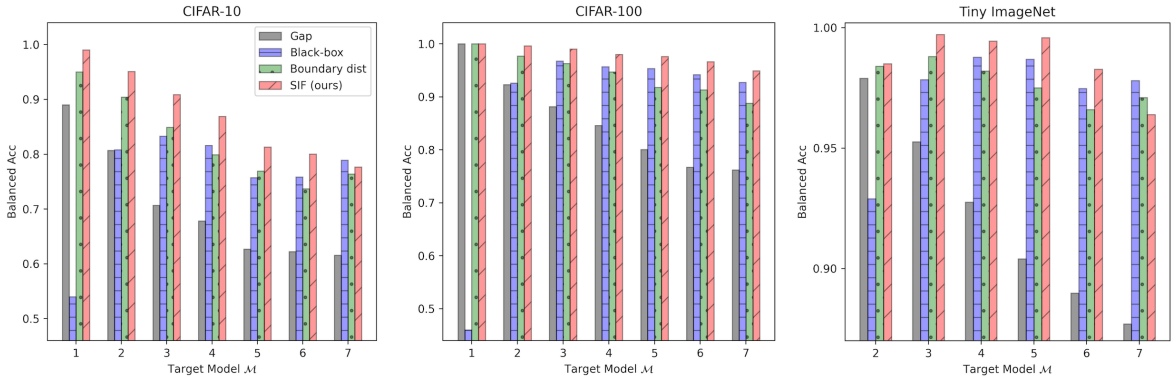

Figure 2. Comparison of our SIF attack with some baseline MI attacks: Gap, Black-box, and Boundary distance. The x-axis indicates the attacked target model and the y-axis shows the balanced attack accuracy (Eq. (5)). Our SIF method surpasses previous SOTA attacks for most target models.

plementation from ART [28]. The Boundary distance attack was implemented with the HopSkipJump adversarial attack [4]. Due to the very long computation time of the Boundary distance attack and our adaSIF attack (Section 3.4), we limited the size of the fitting and evaluation subsets to 1000 and 5000, respectively, for these attack models only.

5. Results

We start by presenting histograms for SIF and adaSIF values to motivate the use of our method. We then evaluate the performance of our SIF MI attack and compare it to current SOTA attack methods. Next, we conduct ablation studies aimed at improving the adaSIF attack with minimal run time. Next, we test our adaSIF attack on target models trained with data augmentation or differential privacy, which are defenses that aim to mitigate our vanilla SIF attack. Lastly, we show the fitting and inference time for all the attack models used in this paper.

5.1. SIF distribution of membership

To better understand how our attack works, we show in Figure 1 the SIF values (Eq. (3)) distribution for members and non-members of CIFAR-10 and CIFAR-100, calculated on the target model $\mathcal{M}$-7 trained on Resnet18. The top row shows SIF values on a model trained without data augmentation. We observe that members are distributed solely within a short interval around 0, whereas non-members can attain very large absolute values, and their distribution in the aforementioned interval seldom matches the members' distribution. This shows that member samples have negligible influence scores on themselves, while non-member samples have a large impact on their test loss. Our SIF attack exploits that property and sets thresholds τ_1 and τ_2 to encapsulate most of the members.

The middle row shows the same SIF values when calculated on a model trained with data augmentation. In this case the non-members still exhibit extreme values, but the members' range spans to a larger interval (see Figure 1(e) and Figure 1(g) compared to Figure 1(a) and Figure 1(c), respectively). Thus, data augmentation can be considered as a defense against SIF since it requires setting an expanded range $[\tau_2, \tau_1]$ which hampers our attack.

The bottom row shows adaSIF values (Eq (4)) calculated on the same data augmented target model that is used in the middle row. We observe that adaSIF restores the short range characteristic for the members, and therefore negates the effect of the data augmentation on the target model defense.

5.2. Comparison of MI attacks

Figure 2 shows the attack power (balanced accuracy) of the four inspected attacks: Gap (black), Black-box (blue), Boundary distance (green), and SIF (red), on three popular classification tasks: CIFAR-10, CIFAR-100, and Tiny ImageNet. We compare the attack scores calculated on seven different Resnet18 target models (Table 1), where each model was trained on a different number of samples. Our SIF attack achieves higher MI accuracy than the baselines for most of the target models. Table 2 summarizes the attack scores for all the MI methods presented in Figure 2, and also details both the member and non-member accuracy. We observe that SIF almost always achieves perfect accuracy ($\sim$ 1.0) for the members, which is crucial for a reliable membership inference. We run the same comparison for AlexNet and DenseNet in the supp. mat and show that SIF achieves new SOTA for these architectures as well. A detailed analysis with precision and recall values is also presented in the supp. mat.

5.3. Ablation studies

Adversarial knowledge. Throughout this paper, we train our SIF and adaSIF attacks with thousands of data samples as explained in Section 4.2. Alas, such data knowledge might not be available to the adversary. Therefore, we repeat our

Dataset	Target Model	Gap			Black-box			Boundary dist			SIF (ours)		
		Member	Non-mem	Balanced	Member	Non-mem	Balanced	Member	Non-mem	Balanced	Member	Non-mem	Balanced
CIFAR-10	$\mathcal{M}$-1	1.000	0.780	0.890	0.600	0.480	0.540	0.980	0.920	0.950	1.000	0.980	**0.990**
	$\mathcal{M}$-2	1.000	0.614	0.807	1.000	0.616	0.808	0.994	0.814	0.904	0.996	0.906	**0.951**
	$\mathcal{M}$-3	1.000	0.414	0.707	1.000	0.666	0.833	0.946	0.752	0.849	1.000	0.818	**0.909**
	$\mathcal{M}$-4	1.000	0.356	0.678	0.986	0.646	0.816	0.914	0.684	0.799	0.989	0.749	**0.869**
	$\mathcal{M}$-5	1.000	0.254	0.627	1.000	0.515	0.757	0.950	0.588	0.769	0.987	0.639	**0.813**
	$\mathcal{M}$-6	1.000	0.244	0.622	0.886	0.631	0.758	0.970	0.504	0.737	0.976	0.624	**0.800**
	$\mathcal{M}$-7	1.000	0.231	0.616	1.000	0.578	**0.789**	0.910	0.618	0.764	1.000	0.553	0.777
CIFAR-100	$\mathcal{M}$-1	1.000	1.000	**1.000**	0.140	0.780	0.460	1.000	1.000	**1.000**	1.000	1.000	**1.000**
	$\mathcal{M}$-2	1.000	0.846	0.923	1.000	0.852	0.926	1.000	0.954	0.977	0.998	0.994	**0.996**
	$\mathcal{M}$-3	1.000	0.763	0.882	1.000	0.935	0.967	0.988	0.938	0.963	0.999	0.982	**0.990**
	$\mathcal{M}$-4	1.000	0.692	0.846	1.000	0.913	0.957	0.998	0.896	0.947	1.000	0.960	**0.980**
	$\mathcal{M}$-5	1.000	0.601	0.801	1.000	0.907	0.953	0.974	0.862	0.918	1.000	0.953	**0.976**
	$\mathcal{M}$-6	1.000	0.535	0.767	0.993	0.891	0.942	0.986	0.840	0.913	1.000	0.932	**0.966**
	$\mathcal{M}$-7	1.000	0.524	0.762	0.993	0.861	0.927	0.978	0.798	0.888	0.999	0.900	**0.949**
Tiny ImageNet	$\mathcal{M}$-2	0.996	0.962	0.979	0.944	0.914	0.929	0.992	0.976	0.984	0.992	0.978	**0.985**
	$\mathcal{M}$-3	1.000	0.905	0.953	1.000	0.957	0.978	1.000	0.976	0.988	1.000	0.994	**0.997**
	$\mathcal{M}$-4	1.000	0.855	0.928	1.000	0.976	0.988	1.000	0.964	0.982	1.000	0.989	**0.994**
	$\mathcal{M}$-5	1.000	0.808	0.904	1.000	0.974	0.987	0.992	0.958	0.975	1.000	0.992	**0.996**
	$\mathcal{M}$-6	1.000	0.780	0.890	0.999	0.950	0.975	0.988	0.944	0.966	1.000	0.966	**0.983**
	$\mathcal{M}$-7	1.000	0.754	0.877	0.994	0.962	**0.978**	0.996	0.946	0.971	1.000	0.928	0.964

Table 2. Comparison of accuracies for various MI attack methods: Gap, Black-box, Boundary distance, and SIF. We detail for every attack the accuracy on the members, the non-members, and the balanced accuracy.

evaluation with merely 10 training data points and show that our attacks obtain marginally lower performance compared to the vanilla training, demonstrating the strength of our methods in realistic setups (see supp. mat).

Adaptive attack. To better understand the impact of the different terms on our method, we performed several ablation experiments. The adaSIF attack described in Section 3.4 requires a proper approximation of the $s(z)$ term in Eq. (4); this approximation is controlled by two parameters: (i) r, the number of iterations used to estimate $s(z)$; and (ii) the recursion depth d, i.e., the number of augmentations performed during one iteration of $s(z)$ calculation. Increasing either parameter prolongs the attack's inference time so we aim for the smallest values of r, d for a successful adaptive attack.

Figure 3(a) shows the effect of d on the balanced accuracy of our adaptive adaSIF attack, for CIFAR-10, CIFAR-100, and Tiny ImageNet trained on the target model $\mathcal{M}$-7 with r set to 1. The width of each line corresponds to the measured standard deviation of five experiments. We set adaSIF with $d = 8$ since it achieves a good balanced accuracy with high confidence (narrow interval).

MI Attack	CIFAR-10	CIFAR-100	Tiny ImageNet
Gap	0.5221	**0.5276**	0.5150
Black-box	0.5053	0.5006	0.5000
Boundary dist	0.5140	0.5238	0.5122
SIF	**0.5228**	**0.5276**	**0.5152**

Table 3. MI attack performance on target models $\mathcal{M}$-7 trained with differential privacy.

Next, we inspect the effect of r on the balanced accuracy with d set to 8. Figure 3(b) shows that r has a marginal impact on the balanced accuracy for CIFAR-100 and Tiny ImageNet and some improvement for CIFAR-10. We therefore select $d = 8$ and $r = 8$ for our adaSIF method. Yet, one may gain very similar results using our attack by using $r = 1$, which reduces the computational time by a factor of 8.

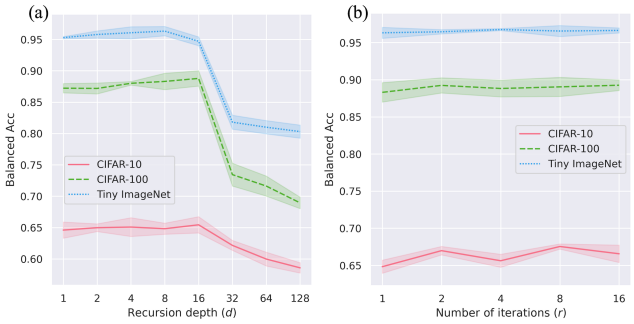

Figure 3. Ablation study on the recursion depth (d) and number of iterations (r) used to estimate $s(z)$ in Eq. (4). The balanced accuracy of our adaSIF attack was calculated for target model $\mathcal{M}$-7 as a function of: (a) d where $r = 1$ and (b) r where $d = 8$. Both d and r are shown in logarithmic scale.

5.4. Data augmentation adaptive attack

We repeat the same comparison of MI attacks in Section 5.2, where the target models are trained with data augmentation. Figure 4 shows the balanced accuracy (Eq. 5) of the attacks: Gap, Black-box, Boundary distance, SIF, and our adaptive adaSIF, on CIFAR-10, CIFAR-100, and Tiny ImageNet, for the different target models trained on Resnet18. As expected, our vanilla SIF attack efficacy is attenuated and surpassed by a baseline in most cases. On the other hand, adaSIF boosts our SIF attack to a new SOTA (red bar) for all datasets. Similar results for AlexNet and DenseNet are shown in the supp. mat.

In another experiment, we trained target models for

Architecture	Gap		Black-box		Boundary dist		SIF (ours)		adaSIF (ours)	
	Fitting	Inference	Fitting	Inference	Fitting	Inference	Fitting	Inference	Fitting	Inference
Resnet18	-	0.19 ms	165.28 s	0.22 ms	7.87 hr	22.51 s	4.67 hr	0.66 s	5.31 hr	16.11 s
AlexNet	-	0.10 ms	169.35 s	0.14 ms	6.68 hr	20.40 s	4.19 hr	0.62 s	4.27 hr	12.20 s
DenseNet	-	0.13 ms	169.26 s	0.20 ms	8.41 hr	25.57 s	4.72 hr	0.68 s	7.42 hr	16.39 s

Table 4. Time required for fine-tuning ("fitting") an attack model on its training points and running a single membership inference ("inference"). Our SIF attack takes a considerable amount of time to fit, but its inference is shorter than the Boundary distance attack.

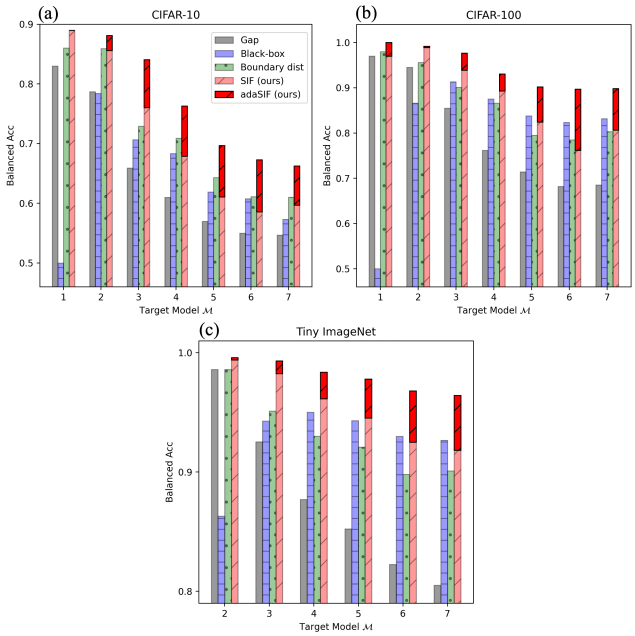

Figure 4. Similar comparison of MI attack as in Figure 2 when the target models are trained with data augmentation. The vanilla SIF value from Eq. (3) performs on par to previous SOTA. When implementing the adaptive attack from Eq. (4) (red bar) we surpass all previous attacks by a large margin.

CIFAR-100 with 50000 training samples, similarly as [26]. We show that our adaSIF attack surpasses their reported white-box MI attack accuracy (see supp. mat).

5.5. Attacking differential privacy

We repeat the same comparison of MI attacks in Section 5.2, where the target models are trained using DP-RMSProp with $\epsilon = 50$. Table 3 shows the balanced accuracy (Eq. 5) of SIF attack compared to the previous MI attack baselines, for target models $\mathcal{M}$-7 trained on Resnet18. SIF marginally surpasses or on par with previous MI attacks. Note that although in this case, differential privacy provides a good defense against all attacks, training with very large ϵ as done here also degrades the performance significantly, which is a remarkable drawback of this defense approach.

5.6. Computational cost

Computation time is particularly an issue for calculating the HVP values for the influence function in large datasets [15]. Table 4 shows a comparison of the fitting and inference time for our SIF attack, adaptive adaSIF attack (with $d = 8$, $r = 8$), and other baseline attacks used in our experiments, on Tiny ImageNet trained using the $\mathcal{M}$-7 target model. "Fitting" indicates the time used to fine-tune the attack model's parameters, and "inference" indicates the average cost time for membership inference on a single data point.

The Gap attack model has no parameters and thus does not need fitting. In addition, Gap and Black-box attacks have negligible inference time. Since the Boundary distance and adaSIF attacks are very slow, we fitted and evaluated them on 1000 random samples from $\mathcal{D}^{train}_{mem} \cup \mathcal{D}^{train}_{non-mem}$ and on 5000 random samples from $\mathcal{D}^{test}_{mem} \cup \mathcal{D}^{test}_{non-mem}$, respectively. The vanilla SIF was fitted and evaluated on all the attack model's samples, similarly to Gap and Black-box. We observe that SIF and adaSIF take less time to fit than the Boundary distance, and their inference cost is also lower, particularly for the SIF attack which runs a single MI attack in less than a second.

6. Conclusions

In this paper we addressed the task of membership inference, which is the prediction of whether a data sample was used to train a model or not. We showed that the self-influence function in Eq. (3) is an excellent indicator of membership inference. The aforementioned SIF values combined with the target model's prediction were used to achieve new SOTA MI attack performance for CIFAR-10, CIFAR-100, and Tiny ImageNet, on Resnet18, AlexNet, and Densenet, for various target models (Table 1).

Furthermore, we showed that our SIF attack can be adjusted to address the common MI defense of training the target model with data augmentation. This refined adaSIF attack surpasses all other baselines by a large margin, for every dataset, architecture and target model listed above, while requiring an inference time of 15 seconds. We also showed that our attacks can be fitted using only 10 members, making them feasible in realistic setups.

One possible direction to form a more sophisticated defense method against our SIF and adaSIF attacks is to "shift" the members distributions towards the non-members. Yet, this involves Hessian estimation which makes such a method computationally demanding.

References

[1] Martín Abadi, Andy Chu, Ian J Goodfellow, H B McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep Learning with Differential Privacy. *ACM SIGSAC*, 2016. 2

[2] Naman Agarwal, Brian Bullins, and Elad Hazan. Second Order Stochastic Optimization in Linear Time. *ArXiv*, abs/1602.0, 2016. 4

[3] Erdi Çallı, Ecem Sogancioglu, Bram van Ginneken, Kicky G van Leeuwen, and Keelin Murphy. Deep learning for chest X-ray analysis: A survey. *Medical Image Analysis*, 72:102125, 2021. 1

[4] Jianbo Chen and Michael I Jordan. HopSkipJumpAttack: A Query-Efficient Decision-Based Attack. *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294, 2020. 2, 6

[5] Christopher A Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-Only Membership Inference Attacks. In *ICML*, 2021. 1, 2, 5

[6] Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *CVPR*, 2020. 2

[7] R Dennis Cook and Sanford Weisberg. Residuals and Influence in Regression. 1982. 3

[8] Thitimanan Damrongsakmethee and Victor-Emil Neagoe. Data Mining and Machine Learning for Financial Analysis. *Indian journal of science and technology*, 10:1–7, 2017. 1

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv*, abs/1810.0, 2019. 1

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *CVPR*, pages 770–778, 2016. 4

[11] Geoffrey E Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the Knowledge in a Neural Network. *ArXiv*, abs/1503.0, 2015. 2

[12] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely Connected Convolutional Networks. *CVPR*, pages 2261–2269, 2017. 4

[13] Jinyuan Jia, Ahmed Salem, Michael Backes, Yang Zhang, and Neil Zhenqiang Gong. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. *ACM SIGSAC*, 2019. 2

[14] Yigitcan Kaya and Tudor Dumitras. When Does Data Augmentation Help With Membership Inference Attacks? In *ICML*, 2021. 2

[15] Pang Wei Koh and Percy Liang. Understanding Black-box Predictions via Influence Functions. In *ICML*, volume 70, pages 1885–1894, 2017. 1, 2, 3, 4, 8

[16] Shuming Kong, Yanyan Shen, and Linpeng Huang. Resolving Training Biases via Influence-based Data Relabeling. In *ICLR*, 2022. 2

[17] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 2

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *NeurIPS*, pages 1–9, 2012. 4

[19] Ya Le and X Yang. Tiny ImageNet Visual Recognition Challenge. 2015. 4

[20] Klas Leino and Matt Fredrikson. Stolen Memories: Leveraging Model Memorization for Calibrated White-Box Membership Inference. In *USENIX Security Symposium*, pages 1605–1622, 2020. 1, 2

[21] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. QEBA: Query-Efficient Boundary-Based Blackbox Attack. *CVPR*, pages 1218–1227, 2020. 2

[22] Jiacheng Li, Ninghui Li, and Bruno Ribeiro. Membership Inference Attacks and Defenses in Supervised Learning via Generalization Gap. *ArXiv*, abs/2002.1, 2020. 1

[23] Tian Li, Manzil Zaheer, Sashank J Reddi, and Virginia Smith. Private Adaptive Optimization with Side Information. In *ICML*, 2022. 5

[24] Zheng Li and Yang Zhang. Label-Leaks: Membership Inference Attack with Label. *CoRR*, abs/2007.1, 2020. 1, 2, 4, 5

[25] Alexander Selvikvåg Lundervold and Arvid Lundervold. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift für Medizinische Physik*, 29(2):102–127, 2019. 1

[26] Milad Nasr, R Shokri, and Amir Houmansadr. Comprehensive Privacy Analysis of Deep Learning: Standalone and Federated Learning under Passive and Active White-box Inference Attacks. *ArXiv*, abs/1812.0, 2018. 2, 8

[27] Milad Nasr, R Shokri, and Amir Houmansadr. Machine Learning with Membership Privacy using Adversarial Regularization. *ACM SIGSAC*, 2018. 2

[28] Maria-Irina Nicolae, Mathieu Sinn, Minh-Ngoc Tran, Beat Buesser, Ambrish Rawat, Martin Wistuba, Valentina Zantedeschi, Nathalie Baracaldo, Bryant Chen, Heiko Ludwig, Ian Molloy, and Ben Edwards. Adversarial Robustness Toolbox v1.0.0. *arXiv: Learning*, 2018. 6

[29] Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating Training Data Influence by Tracing Gradient Descent. In H Larochelle,

M Ranzato, R Hadsell, M F Balcan, and H Lin, editors, *NeurIPS*, volume 33, pages 19920–19930. Curran Associates, Inc., 2020. 3

[30] Md.Atiqur Rahman, Tanzila Rahman, Robert Laganière, and Noman Mohammed. Membership Inference Attack against Differentially Private Deep Learning Model. *Trans. Data Priv.*, 11:61–79, 2018. 2

[31] Shahbaz Rezaei and Xin Liu. On the Difficulty of Membership Inference Attacks. *CVPR*, pages 7888–7896, 2021. 1, 2, 5

[32] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs Black-box: Bayes Optimal Strategies for Membership Inference. In *ICML*, 2019. 1, 2

[33] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models. *ArXiv*, abs/1806.0, 2019. 1, 2, 4

[34] Andrea Schioppa, Polina Zablotskaia, David Vilar, and Artem Sokolov. Scaling Up Influence Functions. *ArXiv*, abs/2112.0, 2021. 3

[35] Xiaoting Shao, Arseny Skryagin, Wolfgang Stammer, Patrick Schramowski, and Kristian Kersting. Right for Better Reasons: Training Differentiable Models by Constraining their Influence Functions. *AAAI*, 35(11):9533–9540, 5 2021. 2

[36] Virat Shejwalkar and Amir Houmansadr. Membership Privacy for Machine Learning Models Through Knowledge Transfer. In *AAAI*, 2021. 2

[37] R Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership Inference Attacks Against Machine Learning Models. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18, 2017. 1, 2, 4

[38] Liwei Song and Prateek Mittal. Systematic Evaluation of Privacy Risks of Machine Learning Models. In *USENIX Security Symposium*, 2020. 2

[39] Liwei Song, R Shokri, and Prateek Mittal. Privacy Risks of Securing Machine Learning Models against Adversarial Examples. *ACM SIGSAC*, 2019. 4

[40] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. *CVPR*, pages 1701–1708, 2014. 1

[41] Stacey Truex, Ling Liu, Mehmet Emre Gursoy, Lei Yu, and Wenqi Wei. Towards Demystifying Membership Inference Attacks. *ArXiv*, abs/1807.0, 2018. 4

[42] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pages 268–282, 2018. 1, 2