PRELIMINARY VERSION: DO NOT CITE
The AAAI Digital Library will contain the published
version some time after the conference

How Does Data Augmentation Affect Privacy in Machine Learning?

Da Yu[*1], **Huishuai Zhang**[2], **Wei Chen**[2], **Jian Yin**[1], **Tie-Yan Liu**[2]
[1] School of Computer Science and Engineering, Sun Yat-sen University.
Guangdong Key Laboratory of Big Data Analysis and Processing
[2] Microsoft Research Asia
{yuda3@mail2, issjyin@mail}.sysu.edu.cn, {huishuai.zhang, wche, tie-yan.liu}@microsoft.com

Abstract

It is observed in the literature that data augmentation can significantly mitigate membership inference (MI) attack. However, in this work, we challenge this observation by proposing new MI attacks to utilize the information of augmented data. MI attack is widely used to measure the model's information leakage of the training set. We establish the optimal membership inference when the model is trained with augmented data, which inspires us to formulate the MI attack as a set classification problem, i.e., classifying a set of augmented instances instead of a single data point, and design input permutation invariant features. Empirically, we demonstrate that the proposed approach universally outperforms original methods when the model is trained with data augmentation. Even further, we show that the proposed approach can achieve higher MI attack success rates on models trained with some data augmentation than the existing methods on models trained without data augmentation. Notably, we achieve 70.1% MI attack success rate on CIFAR10 against a wide residual network while previous best approach only attains 61.9%. This suggests the privacy risk of models trained with data augmentation could be largely underestimated.

1 Introduction

The training process of machine learning model often needs access to private data, e.g., applications in financial and medical fields. Recent works have shown that the trained model may leak the information of its private training set (Fredrikson, Jha, and Ristenpart 2015; Wu et al. 2016; Shokri et al. 2017; Hitaj, Ateniese, and Pérez-Cruz 2017). As the machine learning models are ubiquitously deployed in real-world applications, it is important to quantitatively analyze the information leakage of their training sets. One fundamental approach reflecting the privacy leakage of a model about its training set is the *membership inference* (Shokri et al. 2017; Yeom et al. 2018; Salem et al. 2019; Nasr, Shokri, and Houmansadr 2018; Long et al. 2018; Jia et al. 2019; Song, Shokri, and Mittal 2019; Chen et al. 2020), i.e., an adversary, who has access to a target model, determines whether a data point is used to train the target model (being a member) or not (not being a member). Membership inference (MI) attack is formulated as a binary classification task. A widely adopted measure for the performance of an MI attack algorithm in literature is the MI success rate over a balanced set that contains half training samples and half test samples. A randomly guessing attack will have success rate of 50% and hence a good MI algorithm should have success rate above 50%.

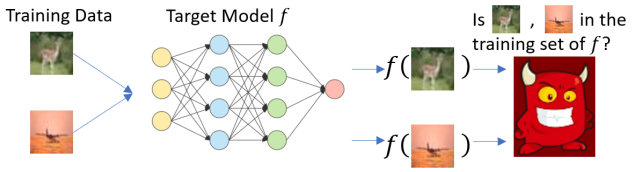

Figure 1: Overview of black-box membership inference in machine learning. The adversary has access to target model's outputs of given samples. The adversary then infers whether the sample is in the target model's training set or not. Higher inference success rate indicates more severe privacy leakage.

It is widely believed that the capability of membership inference is largely attributed to the generalization gap (Shokri et al. 2017; Yeom et al. 2018; Li, Li, and Ribeiro 2020). The larger performance difference of the target model on the training set and on the test set, the easier to determine the membership of a sample with respect to the target model. *Data augmentation* is known to be an effective approach to produce well-generalized models. Indeed, existing MI algorithms obtain significantly lower MI success rate against models trained with data augmentation than those trained without data augmentation (Sablayrolles et al. 2019). It seems that the privacy risk is largely relieved when data augmentation is used.

We challenge this belief by elaborately showing how data augmentation affects the MI attack. We first establish the optimal membership inference when the model is trained with data augmentation from the Bayesian perspective. The optimal membership inference indicates that we should use the set of augmented instances of a given sample rather than a single sample to decide the membership. This matches the intuition because the model is trained to fit the augmented data points instead of a single data point. We also explore the connection between optimal membership inference and group differential privacy, and obtain an upper bound of the

[*]The work was done when this author was an intern at Microsoft Research Asia.

success rate of MI attack.

In this paper, we focus on the *black-box* membership inference (Shokri et al. 2017; Yeom et al. 2018; Salem et al. 2019; Song, Shokri, and Mittal 2019; Sablayrolles et al. 2019). We give an illustration of black-box MI in Figure 1. The black-box setting naturally arises in the *machine learning as a service (MLaaS)* system. In MLaaS, a service provider trains a ML model on private crowd-sourced data and releases the model to users through prediction API. Under the black-box setting, one has access to the model's output of a given sample. Typical outputs are the loss value (Yeom et al. 2018; Sablayrolles et al. 2019) and the predicted logits (Shokri et al. 2017; Salem et al. 2019). We use the loss value of a given sample as it is shown to be better than the logits (Sablayrolles et al. 2019).

Motivated by the optimal membership inference, we formulate the membership inference as a set classification problem where the set consists of loss values of the augmented instances of a sample evaluated on the target model. We design two new algorithms for the set classification problem. The first algorithm uses threshold on the average of the loss values of augmented instances, which is inspired by the expression of the optimal membership inference. The second algorithm uses neural network as a membership classifier. For the second algorithm, we show it is important to design features that are invariant to the permutation on loss values. Extensive experiments demonstrate that our algorithms significantly improve the success rate over existing membership inference algorithms. We even find that the proposed approaches on models trained with some data augmentation achieve higher MI attack success rate than the existing methods on the model trained without data augmentation. Notably, our approaches achieve $> 70\%$ MI attack success rate against a wide residual network, whose test accuracy on CIFAR10 is more than 95%.

Our contributions can be summarized as follows. First, we establish the optimal membership inference when the model is trained with data augmentation. Second, we formulate the membership inference as a set classification problem and propose two new approaches to conduct membership inference, which achieve significant improvement over existing methods. This suggests that *the privacy risk of models trained with data augmentation could be largely underestimated*. To the best of our knowledge, this is the first work to systematically study the effect of data augmentation on membership inference and reveal non-trivial theoretical and empirical findings.

1.1 Related Work

Recent works have explored the relation between generalization gap and the success rate of membership inference. Shokri et al. (2017); Sablayrolles et al. (2019) empirically observe that better generalization leads to worse inference success rate. Yeom et al. (2018) show the success rates of some simple attacks are directly related to the model's generalization gap. For a given model, Li, Li, and Ribeiro (2020) empirically verify the success rate of MI attack is upper bounded by generalization gap. However, whether the target model is trained with data augmentation, the analysis and algorithms of previous work only use single instance to decide membership. Our work fills this gap by formulating and analyzing membership inference when data augmentation is applied.

Song, Shokri, and Mittal (2019) show *adversarially robust* (Madry et al. 2018) models are more vulnerable to MI attack. They identify one major reason of this phenomenon is the increased generalization gap caused by adversarial training. They also design empirical attack algorithm which leverages the adversarially perturbed image (this process needs white-box access to the target model). In this paper, we choose perturbations following the common practice of data augmentation, which can reduce the generalization gap and do not need white-box access to the target model.

Differential privacy (Dwork et al. 2006b; Dwork, Roth et al. 2014) controls how a single sample could change the parameter distribution in the worst case. How data augmentation affects the DP guarantee helps us to understand how the data augmentation affects membership inference. In Section 7, we give a discussion on the relation between data augmentation, differential privacy, and membership inference.

2 Preliminary

We assume that a dataset D consists of samples of the form $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$, where $\boldsymbol{x}$ is the feature and y is the label. A model f is a mapping from feature space to label, i.e., $f : \mathcal{X} \to \mathcal{Y}$. We assume that the model is parameterized by $\theta \in \mathbb{R}^p$. We further define a loss function $\ell(f(\boldsymbol{x}), y)$ which measures the performance of the model on a data point, e.g., the cross-entropy loss for classification task. We may also written the loss function as $\ell(\theta, d)$ for data point $d = (\boldsymbol{x}, y)$ in this paper. The learning process is conducted by minimizing the *empirical loss*: $\sum_{d \in D} \ell(\theta, d)$.

The data are often divided into training set D_{train} and test set D_{test} to properly evaluate the model performance on unseen samples. The generalization gap G represents the difference of the model performance between the training set and the test set,

$$G = \mathbb{E}_{d \sim D_{test}}[\ell(\theta, d)] - \mathbb{E}_{d \sim D_{train}}[\ell(\theta, d)]. \quad (1)$$

2.1 Data Augmentation

Data augmentation is well known as a good way to improve generalization. It transforms each sample into similar variants and uses the transformed variants as the training samples. We use $\mathcal{T}$ to denote the set of all possible transformations. For a given data point d, each transformation $t \in \mathcal{T}$ generates one augmented instance $t(d) = (\tilde{\boldsymbol{x}}, y)$. For example, if $\boldsymbol{x}$ is a natural image, the transformation could be rotation by a specific degree or flip over the horizontal direction. The set $\mathcal{T}$ then contains the transformations with all possible rotation degrees and all directional flips. The size of $\mathcal{T}$ may be infinite and we usually only use a subset in practice. Let $T \subset \mathcal{T}$ be a subset of transformations. The cardinality of $|T|$ controls the strength of the data augmentation. We use $T(d) = \{t(d); t \in T\}$ and $\ell_T(\theta, d) = \{\ell(\theta, \tilde{d}); \tilde{d} \in T(d)\}$ to denote the set of augmented instances and corresponding loss values. With data augmentation, the learning objective is to fit the augmented instances

$$\theta = \arg\min_{\theta} \sum_{d \in D} \sum_{\tilde{d} \in T(d)} \ell(\theta, \tilde{d}). \quad (2)$$

2.2 Membership Inference

Membership inference is a widely used tool to quantitatively analyze the information leakage of a trained model. Suppose the whole dataset consists of n i.i.d. samples $d_i, \ldots, d_n$ from a data distribution, from which we choose a subset as the training set. We decide membership using n i.i.d. Bernoulli samples $\{m_1, \ldots, m_n\}$ with a positive probability $\mathbb{P}(m_i = 1) = q$. Sample d_i is used to train the model if $m_i = 1$ and is not used if $m_i = 0$. Given the learned parameters θ and d_i, membership inference is to infer m_i, which amounts to computing $\mathbb{P}(m_i = 1|\theta, d_i)$.

That is to say, membership inference aims to find the posterior distribution of m_i for given θ and d_i. Specifically, Sablayrolles et al. (2019) shows that it is sufficient to use the loss of the target model to determine the membership m_i under some assumption on the the posterior distribution of θ. They predict $m_i = 1$ if $\ell(\theta, d_i)$ is smaller than a threshold τ, i.e.

$$M_{loss}(\theta, d_i) = 1 \quad \text{if} \quad \ell(\theta, d_i) < \tau. \tag{3}$$

This membership inference is well formulated for the model trained with original samples. However, it is not clear how to conduct membership inference and what is the optimal algorithm when data augmentation is used in the training process[1]. We analyze these questions in next sections.

3 Optimal Membership Inference with Augmented Data

When data augmentation is applied, the process

$$\{d_i\} \to \{T(d_i)\} \to \{\theta, m_i\}$$

forms a *Markov chain*, which is due to the described learning process. That is to say, given $T(d_i)$, d_i is independent from $\{\theta, m_i\}$. Hence we have

$$H(m_i|\theta, T(d_i)) = H(m_i|\theta, T(d_i), d_i) \geq H(m_i|\theta, d_i),$$

where $H(\cdot|\cdot)$ is the conditional entropy (Ghahramani 2006), the first equality is due to the Markov chain and the second inequality is due to the property of conditional entropy.

This indicates that we could get less uncertainty of m_i based on $\{\theta, T(d_i)\}$ than based on $\{\theta, d_i\}$. Based on this observation, we give the following definition.

Definition 1. *(Membership inference with augmented data) For given parameters θ, data point d_i and transformation set T, membership inference computes*

$$\mathbb{P}(m_i = 1|\theta, T(d_i)). \tag{4}$$

For the membership inference with augmented data given by Definition 1, we establish an equivalent formula in the Bayesian sense, which sets up the optimal limit that our algorithm can achieve. Without loss of generality, suppose we want to infer m_1. Let $\mathcal{K} = \{m_2, \ldots, m_n, T(d_2), \ldots, T(d_n)\}$ be the status of remaining data points. Theorem 1 provides the Bayesian optimal membership inference rate.

[1] Sablayrolles et al. (2019) directly applies the algorithm (Equation 3) for the case with data augmentation.

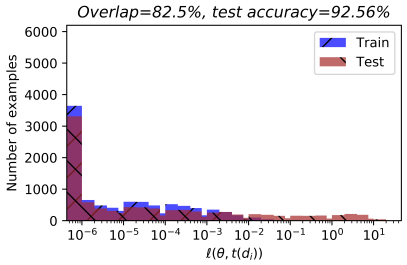

Figure 2: Distribution of single loss values on CIFAR10 dataset. The model is ResNet110 trained with $|T| = 10$. The plot uses 10000 examples from training set and 10000 examples from test set. The dark region is the overlap area between training and test distributions. The membership of a value inside overlap region is hard to decide.

Theorem 1. *The optimal membership inference for given θ and $T(d_1)$ is $\mathbb{P}(m_1 = 1|\theta, T(d_1)) =$*

$$\mathbb{E}_{\mathcal{K}}\left[\sigma\left(\log\left(\frac{\mathbb{P}(\theta|m_1 = 1, T(d_1), \mathcal{K})}{\mathbb{P}(\theta|m_1 = 0, T(d_1), \mathcal{K})}\right) + \log\left(\frac{q}{1-q}\right)\right)\right],$$

where $\sigma(x) := (1 + e^{-x})^{-1}$ is the sigmoid function and $q := \mathbb{P}(m_1 = 1)$ is a constant.

Proof. Apply the law of total expectation and Bayes' theorem, we have

$$\begin{aligned}\mathbb{P}(m_1 = 1|\theta, T(d_1)) &= \mathbb{E}_{\mathcal{K}}[\mathbb{P}(m_1 = 1|\theta, T(d_1), \mathcal{K})] \\ &= \mathbb{E}_{\mathcal{K}}\left[\frac{\mathbb{P}(\theta|m_1 = 1, T(d_1), \mathcal{K})\mathbb{P}(m_1 = 1)}{\mathbb{P}(\theta|T(d_1), \mathcal{K})}\right].\end{aligned} \tag{5}$$

Substitute $q := \mathbb{P}(m_i = 1)$ and let

$$\alpha := \mathbb{P}(\theta|m_1 = 1, T(d_1), \mathcal{K}),\ \beta := \mathbb{P}(\theta|m_1 = 0, T(d_1), \mathcal{K}). \tag{6}$$

Notice that $\mathbb{P}(\theta|T(d_1), \mathcal{K}) = q\alpha + (1-q)\beta$. Then rearranging Eq (5) gives

$$\mathbb{P}(m_1 = 1|\theta, T(d_1)) = \mathbb{E}_{\mathcal{K}}\left[\left(1 + (\frac{1-q}{q})\frac{\beta}{\alpha}\right)^{-1}\right], \tag{7}$$

which concludes the proof. □

We note that the expression in Theorem 1 measures how a single data point affects the parameter posterior in expectation. This is connected with the *differential privacy* (Dwork et al. 2006b,a), which measures how a single data point affects the parameter posterior in the worst case. We give a discussion on the relation between data augmentation, differential privacy, and membership inference in Section 7.

4 Membership Inference with Augmented Data under a Posterior Assumption

In this section we first show the optimal membership inference explicitly depends on the loss values of augmented examples when θ follows a *posterior* distribution. Then we give a membership inference algorithm based on our theory.

4.1 Optimal Membership Inference under a Posterior Assumption

In order to further explicate the optimal membership inference (Theorem 1), we need knowledge on the probability density function of θ. Following the wisdom of energy based model (LeCun et al. 2006; Du and Mordatch 2019), we assume the posterior distribution has the form,

$$p(\theta|m_1, T(d_1), \mathcal{K}) \propto \exp\left(-\frac{1}{\gamma}L(\theta)\right), \tag{8}$$

where $L(\theta) = \sum_{i=1}^{n} m_i \sum \ell_T(\theta, d_i) \geq 0$ is the objective to be optimized and γ is the temperature parameter. We note that Eq (8) meets the intuition that the parameters with lower loss on training set have larger chance to appear after training. Let $p_{\mathcal{K}}(\theta) = \frac{\exp(-\frac{1}{\gamma}\sum_{i=2}^{n} m_i \sum \ell_T(\theta,d_i))}{\int_z \exp(-\frac{1}{\gamma}\sum_{i=2}^{n} m_i \sum \ell_T(z,d_i))dz}$ be the PDF of θ given $\mathcal{K}$. The denominator is a constant keeping $\int_z p_{\mathcal{K}}(z)dz = 1$. Theorem 2 present the optimal algorithm under this assumption.

Theorem 2. *Given parameters θ and $T(d_1)$, the optimal membership inference is*

$$\mathbb{P}\left(m_1 = 1|\theta, T(d_1)\right) = \mathbb{E}_{\mathcal{K}}\left[\sigma\left(\tau - \frac{1}{\gamma}\sum \ell_T(\theta, d_1) + c_q\right)\right],$$

where $\tau := -\log\left(\int_z \exp(-\frac{1}{\gamma}\sum \ell_T(z, d_1))p_{\mathcal{K}}(z)dz\right)$, $c_q := \log(q/(1-q))$ *and $\sigma(\cdot)$ is the sigmoid function.*

Proof. For the α and β defined in Eq (6), we have

$$\begin{aligned}\alpha &= \frac{e^{-(1/\gamma)\sum \ell_T(\theta,d_1)} e^{-(1/\gamma)\sum_{i=2}^{n} m_i \sum \ell_T(\theta,d_i)}}{\int_z e^{-(1/\gamma)\sum \ell_T(z,d_1)} e^{-(1/\gamma)\sum_{i=2}^{n} m_i \sum \ell_T(z,d_i)} dz} \\ &= \frac{e^{-(1/\gamma)\sum \ell_T(\theta,d_1)} p_{\mathcal{K}}(\theta)}{\int_z e^{-(1/\gamma)\sum \ell_T(z,d_1)} p_{\mathcal{K}}(z)dz}\end{aligned} \tag{9}$$

and $\beta = p_{\mathcal{K}}(\theta)$. Therefore, we have $\log(\frac{\alpha}{\beta}) =$

$$-\frac{1}{\gamma}\sum \ell_T(\theta, d_1) - \log\left(\int_z e^{-(1/\gamma)\sum \ell_T(z,d_1)} p_{\mathcal{K}}(z)dz\right). \tag{10}$$

Then plugging Eq (10) into Theorem 1 yields Theorem 2. □

The τ in Theorem 2 represents the magnitude of $\ell_T(d_1)$ on parameters trained without $T(d_1)$. Smaller $\sum \ell_T(\theta, d_1)$ indicates higher $\mathbb{P}(m_1 = 1)$. This motivates us to design a membership inference algorithm based on a threshold on loss values (see Algorithm 1). Data points with loss values smaller than such a threshold are more likely to be training data.

A second observation is that the optimal membership inference *explicitly* depends on the set of loss values. Therefore, membership inference attacks against the model trained with data augmentation are ought to leverage the loss values of all augmented instances for a given sample. We give more empirical evidence in Section 4.2.

Algorithm 1: Membership inference with average loss values (M_{mean}).

Input : Set of loss values $\ell_T(\theta, d)$, threshold τ.
Output : Boolean value, $true$ denotes d is a member.
1 Compute $v = mean(\ell_T)$.
2 Return $v < \tau$.

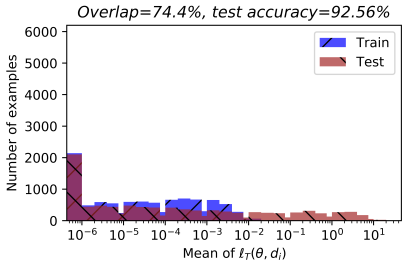

Figure 3: Distribution of the mean of $\ell_T(\theta, d_i)$. The experiment setting is the same as Figure 2. When using mean as metric, the overlap area between training and test distributions is smaller than using single loss, which indicates that $\frac{1}{k}\sum \ell_T(\theta, d_i)$ is a better feature.

4.2 Inference Algorithm in Practice

Inspired by Theorem 2, we predict the membership by comparing $\frac{1}{k}\sum \ell_T(\theta, d_i)$ with a given threshold. The pseudocode is presented in Algorithm 1.

We can set threshold τ in Algorithm 1 based on the outputs of shadow models or tune it based on validation data as done in previous work (Sablayrolles et al. 2019; Song, Shokri, and Mittal 2019). Though simple, Algorithm 1 significantly outperforms M_{loss} by a large margin. The experiment results can be found in Section 6.

We now give some empirical evidence on why M_{mean} is better than M_{loss}. We plot the bar chart of single loss values in Figure 2 (we random sample one loss value for each example). We train the ResNet110 model (He et al. 2016) to fit CIFAR10 dataset[2]. We use the same transformation pool $\mathcal{T}$ as He et al. (2016) which contains horizontal flipping and random clipping. As shown in Figure 2, the overlap area of the loss values between the training samples and the test samples is large when data augmentation is used. For the value inside the overlap area, it is impossible for M_{loss} to classify its membership confidently. Therefore, the overlap area sets up a limit on the success rate of M_{loss}.

Next, we plot the distribution of $\frac{1}{k}\sum \ell_T(\theta, d_i)$ in Figure 3. The overlap area in Figure 3 is significantly smaller compared to Figure 2. This indicates classifying the mean of $\ell_T(\theta, d_i)$ is easier than classifying a single loss value.

5 Membership Inference with Augmented Data Using Neural Network

We have shown that the mean of loss values is the optimal membership inference when θ follows a posterior assumption

[2]https://www.cs.toronto.edu/~kriz/cifar.html.

and demonstrate its good empirical performance. However, if in practice θ does not exactly follow the posterior assumption, it is possible to design features to incorporate more information than the average of loss values to boost the membership inference success rate. In this section, we use more features in $\ell_T(\theta, d)$ as input and train a neural network $\mathcal{N}$ to do the membership inference. The general algorithm is presented in Algorithm 2.

Algorithm 2: Membership inference with neural network.

Input : Set of loss values of a target sample $\ell_T(\theta, d)$; MI network $\mathcal{N}$ and hyperparameters $\mathcal{H}$; some raw data $\mathcal{S} := \{(\ell_T(\theta, \hat{d}), \mathbb{1}_{\hat{d} \in D_{train}}\}$.

Output : boolean value, $true$ denotes d is a member.

1 Build input feature vectors $\boldsymbol{v}$ from $\ell_T(\theta, \hat{d})$ and construct a training set $\mathcal{S}' := \{(\boldsymbol{v}, \mathbb{1}_{\hat{d} \in D_{train}})\}$;

2 Use the training set $\mathcal{S}'$ and hyperparameters $\mathcal{H}$ to train MI network $\mathcal{N}$;

3 Return $\mathcal{N}(\ell_T(\theta, d))$.

In Algorithm 2, each record in raw data $\mathcal{S}$ consists of the loss values of a given example and corresponding membership. The training data of MI network is built from $\mathcal{S}$. Specifically, the loss values of each record are transformed into the input feature vector of MI network $\mathcal{N}$.

Then the key point is to design input feature of the network $\mathcal{N}$. We first use the raw values in $\ell_T(\theta, d)$ as features. We show this solution has poor performance because it is not robust to the permutation on loss values. Then we design permutation invariant features through the *raw moments* of $\ell_T(\theta, d)$ and demonstrate its superior performance.

5.1 A Bad Solution

A straightforward implementation is to train a neural network as a classifier whose inputs are the loss values of all the augmented instances for a target sample. The pseudocode of this implementation is presented in Algorithm 3. We refer to this approach as M_{NN_loss}. Surprisingly, the success rate of M_{NN_loss} is much worse than M_{mean} though M_{NN_loss} has access to more information.

Algorithm 3: Generating input features from raw losses.

Input : Set of loss values $\ell_T(\theta, \hat{d})$.

Output : Feature vector $\boldsymbol{v}$.

1 Concatenate the elements in ℓ_T into vector $\boldsymbol{v}$.

2 Return $\boldsymbol{v}$.

We note that different from standard classification task, the order of elements in set $\ell_T(\theta, d) = \{\ell(\theta, \tilde{d}); \tilde{d} \in T(d)\}$ should not affect the decision of the MI classifier because of the nature of the problem. However, the usual neural network is not invariant to the permutation on input features. For a neuron with non-trivial weights, changing the positions of input features would change its output. We illustrate this phenomenon in Figure 4: the order of elements in ℓ_T, which is not relevant to the target sample's membership, however has large influence on the output of network.

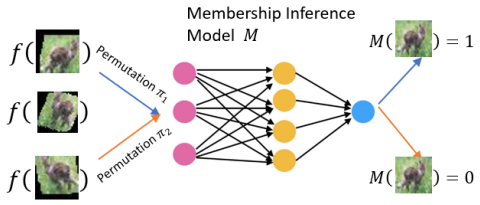

Figure 4: Neural network is not robust to permutation of input features. Changing the order of features will change the prediction. However, the order of augmented instances is not relevant to the membership.

5.2 Building Permutation Invariant Features

Inspired by the failure of M_{NN_loss}, we design features that are invariant to the permutation on $\ell_T(\theta, d)$. We first define functions whose outputs are permutation invariant with respect to their inputs. Then we use permutation invariant functions to encode the loss values into permutation invariant features.

Recall $k = |T|$ is the number of augmented instances for each sample. Let $a \in \mathbb{R}^k$ be a vector version of $\ell_T(\theta, d)$. Let $\pi \in \Pi$ be a permutation of a and $P_\pi \in \mathbb{R}^{k \times k}$ be its corresponding permutation matrix. The following definition states a transformation function satisfying the permutation invariant property.

Definition 2. *A function $f : \mathbb{R}^k \to \mathbb{R}^p$ is permutation invariant if for arbitrary $\pi_i, \pi_j \in \Pi$ and $a \in \mathbb{R}^k$:*

$$f(P_{\pi_i} a) = f(P_{\pi_j} a).$$

Clearly, the *mean* function in Algorithm 1 satisfies Definition 2. However, using the $mean$ to encode $\ell_T(\theta, d)$ may introduce too much information loss.

To better preserve the information, we turn to the raw moments of $\ell_T(\theta, d)$. The i_{th} raw moment v_i of a probability density (mass) function $p(z)$ can be computed as $v_i = \int_{-\infty}^{+\infty} z^i p(z) dz$. The moments of $\ell_T(\theta, d)$ can be computed easily because $\ell_T(\theta, d)$ is a valid empirical distribution with uniform probability mass. Shuffling the loss values would not change the moments. More importantly, for probability distributions in bounded intervals, the moments of all orders uniquely determines the distribution (known as *Hausdorff moment problem (Shohat and Tamarkin 1943)*). The pseudocode of generating permutation invariant features through raw moments is in Algorithm 4.

We note that any classifier using the features generated by Algorithm 4 is permutation invariant with respect to $\ell_T(\theta, d)$. We then use Algorithm 4 to construct $\mathcal{S}'$ in Algorithm 2. This approach is referred to as $M_{moments}$. In our experiments, $M_{moments}$ achieves the highest inference success rate. Experiments details and results can be found in Section 6.

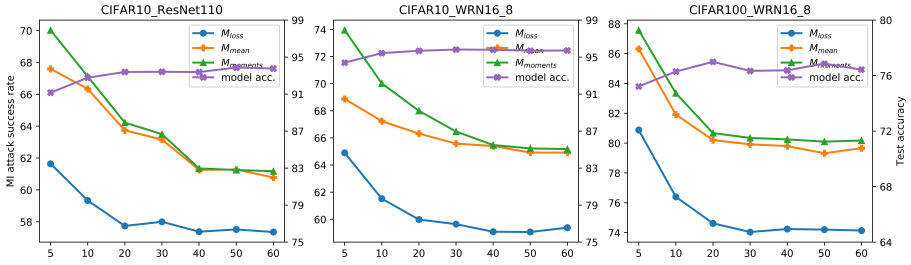

Figure 5: Membership inference success rates with varying k on CIFAR10 and CIFAR100. The left y-axis denotes the membership inference attack success rate. The right y-axis denotes the test accuracy of target models. Our algorithms achieve universally better performance on different datasets and models with varying choices of k.

Algorithm 4: Generating permutation invariant features through raw moments.

Input : Set of loss values $\ell_T(\theta, d)$; the highest order of moments m.

Output : Permutation invariant features $\boldsymbol{v}$

for $i \in [m]$ **do**

  Compute the normalized i_{th} raw moment:

  $v_i := \left(\frac{1}{|T|}\sum_{l\in\ell_T(\theta,d)} l^i\right)^{1/i}$,

end

Concatenate $\{v_i; i \in [m]\}$ into a vector $\boldsymbol{v}$.

Return $\boldsymbol{v}$.

6 Experiments

In this section, we empirically compare the proposed inference algorithms with state-of-the-art membership inference attack algorithm, and demonstrate that the proposed algorithms achieve superior performance over different datasets, models and choices of data augmentation.

We first introduce the datasets and target models with the details of experiment setup.

Datasets We use benchmark datasets for image classification: CIFAR10, CIFAR100, and ImageNet1000. CIFAR10 and CIFAR100 both have 60000 examples including 50000 training samples and 10000 test samples. CIFAR10 and CIFAR100 have 10 and 100 classes, respectively. ImageNet1000 contains more than one million high-resolution images with 1000 classes. We use the training and validation sets provided by ILSVRC2012[3].

Details of used data augmentation We consider 6 standard transformations in image processing literature to build the data augmentation pool $\mathcal{T}$. The details are listed as below.

1. Flip: Flip the image horizontally with probability $p = 0.5$.
2. Crop: Take a random corp of the image.
3. Rotation: Rotate the image by $d \in [-15, 15]$ degrees.
4. Translation: Translate the image by $d \in [-6, 6]$ pixels.
5. Shear: Shear the image by $d \in [-15, 15]$ degrees.
6. Cutout (DeVries and Taylor 2017): Erase a 4×4 box at random location.

For each $t \in \mathcal{T}$, the operations are applied with a random order and each operation is conducted with a randomly chosen parameter (e.g. random rotation degrees). Following the common practice, we sample different transformations for different training samples.

Target models We choose target models with varying capacity, including a small convolution model used in previous work (Shokri et al. 2017; Sablayrolles et al. 2019), deep ResNet (He et al. 2016) and wide ResNet (Zagoruyko and Komodakis 2016). The small convolution model contains 2 convolution layers with 64 kernels, a global pooling layer and a fully connected layer of size 128. The small model is trained for 200 epochs with initial learning rate 0.01. We decay the learning rate by 10 at the 100-th epoch. Following Shokri et al. (2017); Sablayrolles et al. (2019), we randomly choose 15000 samples as training set for the small model. The ResNet models for CIFAR is a deep ResNet model with 110 layers and a wide ResNet model WRN16-8. The detailed configurations and training recipes for deep/wide ResNets can be found in the original papers. For ImageNet1000, we use the ResNet101 model and follow the training recipe in Sablayrolles et al. (2019).

Implementation details of membership inference algorithms All the augmented instances are randomly generated. We use k to denote the number of augmented instances for one image. The number of augmented images is the same for training target models and conducting membership inference attacks. The benchmark algorithm is M_{loss}, which achieves the state-of-the-art black-box membership inference success rate (Sablayrolles et al. 2019). For M_{loss}, *we report the best result among using every element in* $\ell_T(\theta, d)$ *and the loss of original image*. We tune the threshold of M_{loss} and M_{mean} on valid data following previous work (Sablayrolles et al. 2019; Song, Shokri, and Mittal 2019). For M_{NN_loss} and $M_{moments}$, we use 200 samples from the training set of target model and 200 samples from the test set to build the training data of inference network. The inference network has two hidden layers with 20 neurons and Tanh non-linearity as activation function. We randomly choose 2500 samples from the training set of target model and 2500 samples from the test set to evaluate the inference success rate. The samples

[3] http://image-net.org/challenges/LSVRC/2012/.

Model	Dataset	$\|T\|$	Test accuracy	M_{loss}	M_{NN_loss}	M_{mean}	$M_{moments}$
2-layer ConvNet	CIFAR10	$k=0$	59.7	83.7	83.6	83.7	83.7
	CIFAR10	$k=3$	64.6	82.2	85.7	90.3	**91.3**
ResNet110	CIFAR10	$k=0$	84.9	65.4	65.4	65.4	65.6
	CIFAR10	$k=10$	92.7	58.8	61.8	66.3	**67.1**
WRN16-8	CIFAR10	$k=0$	89.7	62.9	62.8	62.8	62.9
	CIFAR10	$k=10$	95.2	61.9	63.1	68.9	**70.1**
ResNet101	ImageNet	$k=10$	93.9	68.3	68.9	73.9	**75.2**

Table 1: Membership inference success rates (in $\%$). We report top-1 test accuracy for CIFAR10 and top-5 accuracy for ImageNet. The numbers under algorithm name are the attack success rates. When $k=0$, we run the proposed methods with 10 randomly augmented instances as input anyway. The baseline attack M_{loss} is introduced in Section 2. The row with $k=0$ denotes the model is trained without data augmentation. Test accuracy denotes the target model's classification accuracy on test set.

used to evaluate inference success rate have no overlap with inference model's training data. Other details of implementation can be found in our submitted code.

Experiment Results We first present the inference success rate with a single k. We use $k=10$ as default. For 2-layer ConvNet, we choose $k=3$ because its small capacity. The results are presented in Table 1.

When data augmentation is used, algorithms using $\ell_T(\theta, d)$ universally outperform M_{loss}. Algorithm 3 has inferior inference success rate compared to M_{mean} and $M_{moments}$ because it is not robust to permutation on input features. The best inference success rate is achieved by $M_{moments}$, which utilizes the most information while being invariant to the permutation on $\ell_T(\theta, d)$.

Remarkably, when $k=10$, $M_{moments}$ has inference success rate higher than 70% against WRN16-8, whose top-1 test accuracy on CIFAR10 is more than 95%! Moreover, in Table 1, our algorithm on models trained with data augmentation obtains higher inference success rate than previous algorithm (M_{loss}) on models trained without data augmentation. We note that the generalization gap of models with data augmentation is much smaller than that of models without data augmentation. *This observation challenges the common belief that models with better generalization provides better privacy.*

We further plot the inference success rates of M_{loss}, M_{mean} and $M_{moments}$ with varying k in Figure 5. For all algorithms, the inference success rate gradually degenerates as k becomes large. Nonetheless, our algorithms consistently outperform M_{loss} by a large margin for all k.

7 Connection with Differential Privacy

Differential privacy (DP) measures how a single data point affects the parameter posterior in the worst case. In this section, we show an algorithm with DP guarantee can provide an upper bound on the membership inference. DP is defined for a random algorithm $\mathcal{A}$ applying on two datasets D and D' that differ from each other in one sample, denoted as $D \sim^1 D'$. Differential privacy ensures the change of arbitrary instance does not significantly change the algorithm's output.

Definition 3. *(ϵ - differential privacy (Dwork et al. 2006b)) A randomized learning algorithm $\mathcal{A}$ is ϵ-differentially private with respect to D if for any subset of possible outcome S*

$$\max_{D \sim^1 D'} \frac{\mathbb{P}(\mathcal{A}=S|D)}{\mathbb{P}(\mathcal{A}=S|D')} \leq e^{\epsilon}.$$

However, in the formula of Theorem 1, the change/removal of one sample d_1 indicates change/removal of a set of training instances $T(d_1)$. We need *group differential privacy* to give upper bound on the quantity of Theorem 1.

Let D be a training set with n samples and $D \sim^k D'$ denote that two datasets differ in k instances. Group different privacy and differential privacy are connected via the following property.

Remark 1. *(Group differential privacy (Dwork, Roth et al. 2014)) If $\mathcal{A}$ is ϵ-differentially private with respect to D, then it is also $k\epsilon$-group differentially private with respect to D for the group size k.*

Let $D_{aug} = \{T(d_i); m_i = 1, i \in [n]\}$ be the augmented training set with k transformations, i.e., $|T(d_i)| = k$. For mean query based algorithms (e.g. gradient descent algorithm), the sensitivity of any instance is reduced to $\frac{1}{k}$. Therefore, a learning algorithm $\mathcal{A}$ that is ϵ-differentially private with respect to dataset D is $\frac{\epsilon}{k}$-differentially private with respect to D_{aug}[4]. With this observation, we have an upper bound on the optimal membership inference in Theorem 1.

Proposition 1. *If the learning algorithm is $\frac{\epsilon}{k}$-differentially private with respect to D_{aug}, we have*

$$\mathbb{P}(m_1 = 1|\theta, T(d_1)) \leq \sigma\left(\epsilon + \log(q/(1-q))\right).$$

Proof. For any given $\mathcal{K}$, we have

$$\frac{\mathbb{P}(\theta|m_1=1, T(d_1), \mathcal{K})}{\mathbb{P}(\theta|m_1=0, T(d_1), \mathcal{K})} \leq \max_{D_{aug} \sim^k D'_{aug}} \frac{\mathbb{P}(\mathcal{A}=S|D_{aug})}{\mathbb{P}(\mathcal{A}=S|D'_{aug})} \leq e^{\epsilon}. \quad (11)$$

The first inequality is due to the definitions of $T(d_1)$ and group differential privacy, and the second inequality is due to the property of group differential privacy (Remark 1). Substituting Eq (11) into Theorem 1 yields the desired bound. □

[4]The $\frac{\epsilon}{k}$-DP is at instance level, i.e. $D_{aug} \sim^1 D'_{aug}$.

Proposition 1 tells that if the learning algorithm is $\frac{\epsilon}{k}$-DP with respect to D_{aug} , which is true for differentially private gradient descent (Bassily, Smith, and Thakurta 2014), the upper bound of the optimal membership inference is not affected by the number of transformations k. This is in contrast with previous membership inference algorithm that only considers single instance (Sablayrolles et al. 2019), i.e., formulated as $\mathbb{P}(m_1 = 1|\theta, \tilde{d}_1)$, where $\tilde{d}_1$ can be any element in $T(d_1)$. Due to the result in Sablayrolles et al. (2019), the upper bound of $\mathbb{P}(m_1 = 1|\theta, \tilde{d}_1)$ scales with $\frac{\epsilon}{k}$ for mean query based algorithms, which monotonically decreases with k. This suggests the algorithm in Sablayrolles et al. (2019) has limited performance especially when k is large.

8 Conclusion

In this paper, we revisit the influence of data augmentation on the privacy risk of machine learning models. We show the optimal membership inference in this case explicitly depends on the augmented dataset (Theorem 1). When the posterior distribution of parameters follows the Bayesian posterior, we give an explicit expression of the optimal membership inference (Theorem 2). Our theoretical analysis inspires us to design practical attack algorithms. Our algorithms achieve state-of-the-art membership inference success rates against well-generalized models, suggesting that the privacy risk of existing deep learning models may be largely underestimated. An important future research direction is to mitigate the privacy risk incurred by data augmentation.

9 Acknowledgments

Da Yu and Jian Yin are supported by the National Natural Science Foundation of China (U1711262, U1611264,U1711261,U1811261,U1811264, U1911203), Guangdong Basic and Applied Basic Research Foundation (2019B1515130001), Key R&D Program of Guangdong Province (2018B010107005). Huishuai Zhang and Jian Yin are corresponding authors.

References

Bassily, R.; Smith, A.; and Thakurta, A. 2014. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *Annual Symposium on Foundations of Computer Science*.

Chen, M.; Zhang, Z.; Wang, T.; Backes, M.; Humbert, M.; and Zhang, Y. 2020. When Machine Unlearning Jeopardizes Privacy. *arXiv preprint arXiv:2005.02205* .

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* .

Du, Y.; and Mordatch, I. 2019. Implicit Generation and Modeling with Energy Based Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Dwork, C.; Kenthapadi, K.; McSherry, F.; Mironov, I.; and Naor, M. 2006a. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006b. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*.

Dwork, C.; Roth, A.; et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* .

Fredrikson, M.; Jha, S.; and Ristenpart, T. 2015. Model inversion attacks that exploit confidence information and basic countermeasures. In *ACM SIGSAC Conference on Computer and Communications Security*.

Ghahramani, Z. 2006. Information theory. *Encyclopedia of Cognitive Science* .

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

Hitaj, B.; Ateniese, G.; and Pérez-Cruz, F. 2017. Deep models under the GAN: information leakage from collaborative deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*.

Jia, J.; Salem, A.; Backes, M.; Zhang, Y.; and Gong, N. Z. 2019. MemGuard: Defending against Black-Box Membership Inference Attacks via Adversarial Examples. In *2019 ACM SIGSAC Conference on Computer and Communications Security*.

LeCun, Y.; Chopra, S.; Hadsell, R.; Ranzato, M.; and Huang, F. 2006. A tutorial on energy-based learning. *Predicting structured data* .

Li, J.; Li, N.; and Ribeiro, B. 2020. Membership Inference Attacks and Defenses in Supervised Learning via Generalization Gap. *arXiv preprint arXiv:2002.12062* .

Long, Y.; Bindschaedler, V.; Wang, L.; Bu, D.; Wang, X.; Tang, H.; Gunter, C. A.; and Chen, K. 2018. Understanding membership inferences on well-generalized learning models. *arXiv preprint arXiv:1802.04889* .

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations*.

Nasr, M.; Shokri, R.; and Houmansadr, A. 2018. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security*.

Sablayrolles, A.; Douze, M.; Ollivier, Y.; Schmid, C.; and Jégou, H. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. *International Conference on Machine Learning* .

Salem, A.; Zhang, Y.; Humbert, M.; Berrang, P.; Fritz, M.; and Backes, M. 2019. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *Network and Distributed Systems Security (NDSS) Symposium* .

Shohat, J. A.; and Tamarkin, J. D. 1943. *The problem of moments*. American Mathematical Soc.

Shokri, R.; Stronati, M.; Song, C.; and Shmatikov, V. 2017. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy (SP)*.

Song, L.; Shokri, R.; and Mittal, P. 2019. Privacy risks of securing machine learning models against adversarial examples. In *ACM SIGSAC Conference on Computer and Communications Security*.

Wu, X.; Fredrikson, M.; Jha, S.; and Naughton, J. F. 2016. A methodology for formalizing model-inversion attacks. In *IEEE Computer Security Foundations Symposium*.

Yeom, S.; Giacomelli, I.; Fredrikson, M.; and Jha, S. 2018. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE 31st Computer Security Foundations Symposium (CSF)*.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* .