Membership Inference Attack in Face of Data Transformations

Jiyu Chen*, Yiwen Guo†, Hao Chen* and Neil Gong‡
* Department of Computer Science, University of California, Davis
{jiych, chen}@ucdavis.edu
† Independent Researcher
guoyiwen89@gmail.com
‡ Department of Electrical and Computer Engineering, Duke University
neil.gong@duke.edu

***Abstract*—Membership inference attacks (MIAs) on machine learning models, which try to infer whether a sample is in the training dataset of a target model, have been widely studied over recent years as data privacy attracts increasing attention. One unignorable problem in the current MIA threat model is that it assumes the attacker always obtains exactly the same samples as in the training set. In reality, however, the attacker is more likely to gather only a transformed version of the training samples. For instance, portraits downloadable from a social networking website usually are re-scaled and compressed, while the website owner can train models with RAW images. We believe a transformed training sample still causes privacy leakage if the transformation is semantic-preserving. Therefore, we broaden the concept of membership inference into more realistic scenarios by considering data transformations. We introduce two strategies for designing MIAs in face of data transformations: one adapts current MIAs to transformations, and the other tries to reverse the transformations approximately. We demonstrated the effectiveness of our strategies and the significance of considering data transformations by extensive evaluations of multiple datasets with several common data transformations and by comparisons with six state-of-the-art attacks. Moreover, we conduct evaluations on data-augmented and privacy-preserving models protected by three state-of-the-art defenses.**

***Index Terms*—Data Privacy; Membership Inference; Data Transformation**

I. Introduction

Privacy issues in machine learning have been attracting more concerns in recent years. Membership inference attack (MIA) [1], designed to infer whether a sample held by an attacker was utilized for training a target model, is one of the most studied problems. Existing MIAs can cause severe privacy leakage, for example, by analyzing facial recognition systems trained on a particular set of identities or medical models trained on patients' medical records.

However, we notice an unignorable problem in the threat model of all current MIA research, which comes from the assumption that the attacker can always obtain the original training data. The assumption is pretty ideal since what can be collected by the attacker may have been processed and transformed in reality. For example, images downloaded from the Internet may have been compressed; stolen medical records might have missing features. In face of such real-world data transformations, the performance of current MIA methods is yet unknown and rarely explored.

From the perspective of data privacy, one should consider not only specific numeric values but also the semantics of data, especially for human-perceivable ones. Data with semantic-preserving transformations still leak private information, even as much as the original (training) data. For instance, compressed photos differ from their RAW counterparts only in image quality; photos processed by different filters only change in their brightness, contrast, or saturation. Therefore, it is natural to ask the following questions: can we better define "membership" considering the presence of possible data transformations, and can adversaries infer the membership, even if they only have a transformed version of a training member? We show that the answer to both questions is yes. The attacker can achieve a better success rate with strategies adapted to data transformations than directly applying current MIA methods, indicating that model assessment without considering data transformation may underestimate privacy risks.

In Section III-A and III-B, we broaden the concept of membership, define the attack space, and enhance the threat model by considering semantic-preserving transformations. We propose two strategies in Section III-C and III-D for designing MIAs under the presence of data transformations. The first strategy enhances traditional attacks by incorporating possible data transformations in shadow models to select better transferable thresholds or train better attack models. The second strategy is developed based on an observation that the transformed training members are closer to their ϵ-robust areas (defined in Definition 3.2) than non-members, thus we can infer membership via evaluating the distance required to reverse collected samples back to ϵ-robust areas.

In our evaluations, we studied datasets in both image and non-image domains, including CIFAR-10 [2], and Purchase-100 [3]. For CIFAR-10, we evaluated common pixel transformations (Gaussian noise, adversarial noise [4], JPEG compression [5], scaling, photo filtering) and spatial transformations (rotation, translation). For Purchase-100, which has binary one-dimensional feature vectors, we evaluated the scenario where some of the vector entries are missing. Our experimental results show that current MIAs generally fail with a moderate level of transformations or corruptions while the attacker can still achieve decent attack performance with our proposed strategies. Moreover, we also evaluated more realistic scenarios where the target model was trained with data augmentation and state-of-the-art defenses, including differential privacy [6], adversarial regularization [7], and MemGuard [8].

We hope our work can broaden the scope of current MIAs and help ML service providers better assess the actual privacy risks under a more realistic threat model.

II. Related work

In recent years, researchers thoroughly explored the domain of MIA, such as strategies for black-box attacks [1, 9], label-only attacks [10, 11, 12, 13], white-box attacks [14, 15], attacks on generative models [16, 17, 18], and attacks on federated learning [19, 20]. Defenses were also developed from the perspective of privacy-preserving algorithms [6, 21, 22], regularization [1, 7, 23], output obfuscation [1, 8, 13], etc.

Yu et al. [24] discussed the MIAs on data augmented models trained by only transformed data. The attacker tries to infer the membership of the original training samples before augmentation. Our problem settings share a similar condition: what the attacker obtains is not in model training. However, the core difference between our attack scenarios comes from the transformation knowledge of the attacker. On the other hand, we will show that our attack is applicable to both regular and transformed samples on data augmented models. Further discussions on this topic are in Section V-A.

The label-only attacks [10] share a similar intuition with our second strategy in Section III-D. The core assumption of label-only attacks is that training members should lie farther from the decision boundary than non-members. Instead of the decision boundary, we consider a more fine-grained boundary–ϵ-robust area that can be adjusted for different transformations. Similarly, we assume it is easier to move training members back towards ϵ-robust areas.

III. Attacks on Transformed Data

A. Membership reconsidered

Before introducing our considered threat model, we first propose to broaden the scope of membership formally:

Definition 3.1 (Membership (Generalized)): An sample x' is considered a (generalized) training member of a training dataset $\mathcal{D}_0$ if $x' = g(x), x \in \mathcal{D}_0, g \in \mathcal{G}^*$, where $\mathcal{G}^*$ is the set of *semantic-preserving transformations*.

In terms of data privacy, we advocate focusing on the semantics of the target data. There can be special cases, for example, if $x_1 \in \mathcal{D}_0$ and $x_2 \notin \mathcal{D}_0$ are both transformed into the same $x' = g_1(x_1) = g_2(x_2)$ via some $g_1, g_2 \in \mathcal{G}^*$, we will still recognize x' as a training member, since x' can always leak (semantic) information of $x_1 \in \mathcal{D}_0$ as long as g_1 is semantic-preserving. In other words, with Definition 3.1, we only care about whether the post-transformation data x' leaks information in the training dataset. Compared with the traditional definition, which only focuses on fixed training data points, our generalized definition is closer to the attackers' real goal — the semantic information within data. The attackers can further utilize the semantic information to speculate other attributes of the training data.

A question to ask is: what transformation is considered semantic-preserving? The answer can be very subjective and vary from the attacker's downstream goals after membership inference. For instance, adding too much noise can make some small objects inside an image unrecognizable; severe filtering can change the appearance of objects. This paper will limit the discussion of semantic-preserving transformations to common transformations that are frequently seen in the real world, with constrained transformation levels. Examples of common transformations are provided in Section IV-A2.

B. Our threat model

Given Definition 3.1, we then introduce the threat model for attackers, which, to the best of our knowledge, for the first time considers transformations between collected samples and training samples. The other assumptions follow those of the traditional threat model [1]. See as follows.

- **Data distribution**: Assume that the data distribution of the training dataset is $\mathcal{D}$. The attacker can sample from $\mathcal{D}$ without knowledge of any specific members of the training set. On the other hand, the possible training member collected by the attacker can be transformed by some transformations g. The attacker may or may not know the detailed form and parameters of the transformation; however, they can access the transformation as a black box (i.e., obtain the transformed version of any given sample in $\mathcal{D}$). Additionally, the attacker is able to correctly label samples from $\mathcal{D}$.
- **Target model**: The attacker has full knowledge of the model architectures, pre-processing methods, training methods, and hyper-parameters. The attacker does not know the detailed parameters of the target model but can query it as a black box and obtain confidence scores for each class.
- **The attacker's goal**: The attacker wants to obtain an inference function $\mathcal{M}$ for the target model f, so that $\mathcal{M}(x', f) = \mathbf{1}(x \in \mathcal{D}_0)$, where $x' = g(x)$ is the target sample transformed by g, x is the original version of x', and $\mathcal{D}_0 \subset \mathcal{D}$ is the training set for f.

One thing to clarify here is the difference between the attackers' ability to sample from $\mathcal{D}$ and to obtain training members. For example, to attack a face classification model from a social media website, the potential training members are (possibly compressed) face images downloadable from the website. Yet, attackers can always obtain high-resolution face images from other data sources in $\mathcal{D}$.

C. Strategy 1: Adapt current attacks to scenarios with transformations

We start by evaluating current MIAs in face of data transformations. We found that most current attacks would deteriorate without considering data transformations. Detailed results of SOTA attacks can be found in Section IV. Here we take the LT attack [9] as an example, which classifies all samples with losses lower than a threshold as training members. With data transformations, the loss distributions of training and test data change significantly, and the threshold set for regular data in $\mathcal{D}$ can be suboptimal, as can be observed in Figure 1.

Generally, most current attacks encounter a similar problem since they all rely on the differences in output statistics between training and test samples. To obtain the statistics, current attacks use some evaluators $\mathcal{L}$, such as the loss function in [9], the entropy metric in [25] or the binary classifier in [1]. Transformations can change the output distribution of $\mathcal{L}$ significantly, resulting in failed attacks.

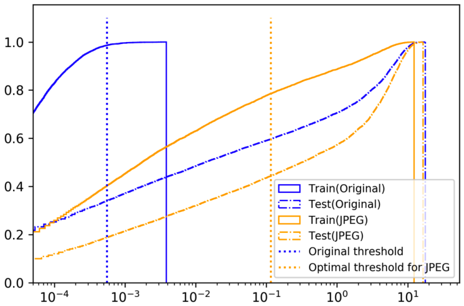

Fig. 1: Illustration of the loss distribution change. The blue curves show the loss distributions of the original training and test samples, while the orange curves show the corresponding loss distributions of JPEG compressed ($q = 10$) versions.

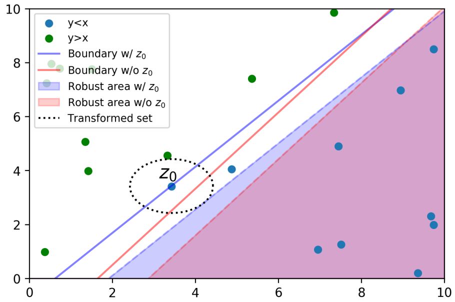

Fig. 2: Illustration of the intuition of Strategy 2 by a simple linear binary classifier. The figure shows that when a sample is used for training, its neighborhood is more likely to be closer to its ϵ-robust areas. Detailed explanations are in Section III-D.

In order to accommodate the distribution change, it is straightforward to select better thresholds. Thresholds in MIAs are commonly obtained from shadow models trained with data sampled from $\mathcal{D}$ to mimic the target model. Hence, our first strategy shown in Algorithm 1 is to adapt current attack methods to data transformations by explicitly considering transformations in the shadow models when selecting thresholds. In detail, the shadow models are still trained with data sampled from $\mathcal{D}$, while the inputs to $\mathcal{L}$ for generating statistics are transformed data. Note that the transformation is applied as a black-box function via our assumption in Section III-B.

We take the adapted LT attack as an example (namely, Loss-thresholding with transformation, or LTT) for evaluation in Section IV and V. Other attacks can be adapted similarly (e.g., adapted label only attack (LOT) in Table VI).

Algorithm 1: Adapt current attacks to transformations

```
Input: shadow model $f_s$, black box transformation
       function $g$, shadow model's training set $X$ and
       labels $Y$, shadow model's test set $X_t$ and labels
       $Y_t$, metric function $\mathcal{L}$
Output: Best threshold $\tau^*$ on the shadow model
$L \leftarrow \mathcal{L}(f_s(g(X)), Y)$;
$L_t \leftarrow \mathcal{L}(f_s(g(X_t)), Y_t)$;
$\tau^* \leftarrow \mathrm{argmax}_\tau(\mathrm{sum}(\mathbf{1}(L < \tau)) + \mathrm{sum}(\mathbf{1}(L_t > \tau)))$;
    ▷ Select $\tau$ that can optimally
    distinguish $L$ and $L_t$
return $\tau^*$;
```

D. Strategy 2: Reverse transformation

While the first strategy adapts current attacks, we introduce another strategy from the following perspective: training members tend to lie around its local minima. Thus, moving a transformed training member towards its original version may decrease its loss more drastically than a non-member.

To get into more details, we first provide the definition for an ϵ-*robust area*, where ϵ serves as an important hyper-parameter to control the area's range.

Definition 3.2 (ϵ-robust area): An ϵ*-robust area* $R_\epsilon(x, f, \mathcal{G})$ of a given sample x, a target model f, and a family of transformation $\mathcal{G}$, is defined as:

$$R_\epsilon(x, f, \mathcal{G}) = \{g(x, \delta) | \mathcal{L}(f(g(x, \delta)), y) < \epsilon, g \in \mathcal{G}\},$$

where $g(\cdot, \delta)$ is some transformation function from a family of functions $\mathcal{G}$ parametrized by δ which controls its transformation level, $\mathcal{L}$ evaluates the prediction loss with label y, and $\epsilon \in \mathcal{R}^+$.

An ϵ-robust area represents the neighborhood of a trained instance via some transformations with a loss lower than ϵ. Given Definition 3.2, our intuition is that a transformed training member is likely to show a shorter distance to its ϵ-robust area. We illustrate with a toy linear model $f(x, y) = \mathbf{1}(x > y)$ in Figure 2. We trained two models with and without z_0, respectively. The blue and red regions are ϵ-robust areas of z_0, corresponding to the two models. Obviously, z_0 and perturbations within the black circle are closer to the blue region, for which z_0 is a training member.

In order to estimate distances to ϵ-robust areas, we define the *reverse transformation function* $g^-(\cdot, \theta)$ in Definition 3.3, and we limit the optimization of the parameter θ of $\mathcal{G}'$ in a metric space (Θ, d) with a distance metric d.

Definition 3.3 (Reverse transformation function): A reverse transformation function $g_\theta^- = g^-(\cdot, \theta)$ of $g_\delta = g(\cdot, \delta)$ is a function from a family of transformations $\mathcal{G}'$ parameterized by θ such that:

$$\forall \delta \in \Delta, \exists \theta \in \Theta, g_\theta^- \circ g_\delta = \mathcal{I},$$

where Δ and Θ are parameter spaces for δ and θ, respectively, and $\mathcal{I}(x) = x$ is the identity function.

Given all these, our second strategy (named reverse transformation, RT) is summarized in Algorithm 2, which searches the shortest path from a transformed sample towards a ϵ-robust area. The update function u is designed differently for different transformations. Examples of u are in Section IV.

Once the shortest path is found, we will decide the membership via a preset threshold distance τ. The threshold is also from shadow models: the attacker evaluates reverse transformation distances for training and non-training data with transformations and then selects a threshold to maximize the inference accuracy on shadow models.

Algorithm 2: Reverse transformation (RT)

```
Input: transformed sample $x'$, true label $y$, target model
       $f$, loss function $\mathcal{L}$, robust area radius $\epsilon$, reverse
       transformation function $g^{-}$, update function $u$,
       attack threshold $\tau$, maximum iteration $n$, metric
       $d$
Output: membership $m \in \{\text{True}, \text{False}\}$
$\theta \leftarrow \mathbf{0}$;
$i \leftarrow 0$;
while $\mathcal{L}(f(g^{-}(x', \theta)), y) >= \epsilon$ and $i < n$ do
    $\theta \leftarrow u(\theta)$;
    $i \leftarrow i + 1$;
end
return $d(\theta) < \tau \;\; \& \;\; i < n$;
```

We would like to mention that the reverse transformation function g^{-} is not necessarily the mathematical inverse of g, considering that not all transformations are invertible (i.e., $\exists g^{-}$,such that $g^{-} \neq g^{-1}$). For a specific sample, functions that output the same result as g^{-1} may not be unique, and g^{-} can be any one of them. Generally, the reverse transformation can be any operation that undoes the transformation. Still, the closer g^{-} to g^{-1} (if it exists), the better the results. In practice, an approximation of the reverse transformation already suffice, and we will show it with experiments.

IV. Evaluation

A. Experiment settings

1) Datasets: In the main paper, we report the results of two datasets: CIFAR-10 [2] and Purchase-100 [3] due to the page limit. Sources and results of other datasets will be released upon paper publication.

We train the target model and shadow models by randomly selecting a training set of size 10000 and also a test set of size 10000. Note that the training set for the target model and the shadow models are independent.

2) Transformations: Image datasets lie in continuous spaces with spatial information, so we consider the following seven semantic-preserving transformations. The first five are pixel value transformations, and the other two are spatial transformations, all commonly seen in real-world websites, social media applications, CAPTCHA services, etc.

- **Gaussian noise**: Gaussian noise is one of the most common noises that would appear in images. In this case, the transformed sample is $x' = x + \delta$, where $\delta \sim N(\mu, \sigma^2)$. In our experiments, we set $\mu = 0$ and consider a random standard deviation $\sigma \in (\sigma_{min}, \sigma_{max})$.
- **Adversarial noise**: Adversarial noise [4] is defined as noises crafted to change the model output significantly. We apply adversarial noises generated by PGD attacks [26] with a fixed L^{∞}-norm of 0.05. We consider three different adversarial settings: 1-step PGD, 10-step PGD, and targeted-loss PGD (which runs until the loss exceeds the target loss).
- **JPEG compression**: JPEG [5] is a popular lossy compression method for digital images. Our experiments evaluate samples with different compression quality parameters q ($q \in [0, 100]$, higher q means better image quality).
- **Scaling**: Scaling is a common image processing operation when placing an image into a larger or smaller placeholder. In our experiment, we consider scaling the target image by bilinear interpolation. We will evaluate both down-scaling ($r < 1$) and up-scaling ($r > 1$) scenarios, where r is the scaling factor.
- **Filtering**: Photo filtering appears everywhere in social media applications when people share their photos. Our experiments evaluate attacks on three popular filters embedded in Instagram — Clarendon, Gingham, and Moon, all implemented in the *pilgram* library [27].
- **Rotation**: Rotation is a common spatial transformation that keeps the image semantics. Since rotation is a fixed transformation once the degree of the rotation δ is given, and the attacker can easily identify it. So we consider the scenario where the degrees of rotation are randomly selected in a range: $\delta \in (\delta_{min}, \delta_{max})$.
- **Translation**: Translation is another common spatial transformation that preserves semantics. Without loss of generality, we consider translating the image to the left-top direction by (d, d), where d is a small translation distance in terms of image pixels towards both horizontal and vertical directions.

Records in Purchase-100 lie in discrete space and have no explicit spatial relationship. So we consider the following realistic scenario:

- **Missing features**: Some features in the data records are missing. For Purchase-100, which has binary features, we consider pre-processing the data by randomly filling the entries of missing features, making the scenario the same as when some features are randomly flipped. We assume that the attacker knows the ratio γ of flipped features in the original records, but the attacker does not know their exact positions. Note that the flipped positions may not be the same for all records.

For the cases where there exists randomness in the transformations (e.g., Gaussian noise with a range of σ), we transform each sample by a randomly selected parameter in the parameter space multiple times and compute the average loss or reverse transformation distance to select the threshold.

3) Target models: For CIFAR-10, we trained VGG-19 [28] models by an SGD optimizer with a 1e-3 learning rate and a 5e-4 weight decay rate for 100 epochs. The final model has a training accuracy of 99% and testing accuracy of 69%.

TABLE I: Accuracies of MIAs on CIFAR-10 regular models

(a) Smaller transformations

Att.	Trans.	Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.1]	1-step	50	10	Clarendon	$[0°, 10°]$	3
Current	CC	65.64%	66.21%	78.89%	66.68%	67.97%	66.72%	66.10%	54.53%
	LT	80.47%	77.32%	62.63%	74.13%	70.32%	72.78%	69.51%	51.61%
	ST	78.32%	75.84%	64.86%	73.40%	70.38%	72.04%	68.99%	51.92%
	ET	77.58%	76.44%	68.49%	75.59%	73.11%	73.81%	71.32%	52.56%
	NN	79.46%	76.33%	54.23%	73.10%	68.33%	72.18%	67.94%	50.42%
	LO	76.27%	72.19%	68.59%	71.58%	71.14%	72.11%	68.47%	54.81%
Ours	LTT	-	**77.62**%	79.39%	**75.89**%	73.81%	**74.51**%	71.69%	54.54%
	RT	-	75.71%	**79.63**%	74.63%	**74.34**%	72.94%	**78.73**%	**67.04**%

(b) Larger transformations

Att.	Trans.	Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.3]	loss=10	1	0.5	Moon	$[0°, 30°]$	7
Current	CC	65.64%	63.48%	50.22%	56.15%	58.65%	64.41%	60.09%	49.93%
	LT	80.47%	64.78%	50.02%	51.98%	53.46%	60.56%	59.03%	50.01%
	ST	78.32%	64.57%	50.10%	52.06%	53.64%	60.69%	58.97%	49.93%
	ET	77.58%	65.50%	50.03%	52.93%	54.94%	63.20%	60.51%	49.67%
	NN	79.46%	63.14%	53.25%	51.37%	51.90%	59.75%	58.14%	50.87%
	LO	76.27%	63.33%	50.07%	52.96%	55.17%	61.33%	59.25%	49.70%
Ours	LTT	-	65.67%	56.17%	56.70%	58.91%	**65.56**%	61.33%	49.99%
	RT	-	**65.79**%	**65.01**%	**57.10**%	**61.35**%	65.44%	**77.11**%	**56.81**%

For Purchase-100, we trained the same architecture used by [1] by an SGD optimizer with a 1e-2 learning rate and a 1e-7 weight decay rate for 200 epochs. The final model has a 98% training accuracy and 72% testing accuracy.

Note that all models in this section are regular models without data augmentation and privacy defenses.

4) Evaluation metrics: The primary metric to compare the effectiveness of each MIA is attack accuracy (on 10000 training and 10000 non-training data). The baseline accuracy is $\sim 50\%$ since the membership inference functions have binary outputs (member/non-member). A success rate of around 50% means the attack is no better than random guessing.

5) Selection of ϵ-robust areas: In RT attacks, ϵ serves as an important hyper-parameter that controls the range of the robust area, which decides the strength of the reverse transformation. An improperly selected ϵ can weaken the attack.

Empirically, we found that the optimal ϵ is always with a close magnitude of the median loss of the input data distribution. In our experiments, we select best results from ϵs that are set as $\lambda_\epsilon \in \{10^0, 10^{-1}, 10^{-2}\}$ times of the median loss of the (transformed) training members.

B. Evaluation on CIFAR-10

Together with attacks designed by our proposed strategies, we also evaluate how current MIAs performs against transformed samples. In this section, we select six SOTA attacks: (1) Classification correctness (CC) [9], (2) Loss thresholding (LT) [9], (3) Confidence score thresholding (ST) [29], (4) Entropy thresholding (ET) [25], (5) NN-based attack (NN) [1], and (6) Label-only attack (LO) [10].

As mentioned in Section III-D, we need to select two functions—the reverse transformation function and the update function—for the RT attack. Specifically, we consider the following two families of reverse transformation functions for the image transformations listed in Section IV-A:

- **Noises on pixel values** For transformations that modifies pixel values, such as Gaussian noises, adversarial noises, JPEG compression, scaling, and filtering, we consider a simple reverse transformation function $g^{-}(x', \theta) = x' + \theta$.
- **Affine transformations** For both rotation and translation, they can be represented by affine transformations parameterized by some affine transformation matrices θ which applies the following transformation to every pixel:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \theta_{00} & \theta_{01} \\ \theta_{10} & \theta_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \theta_{02} \\ \theta_{12} \end{bmatrix}.$$

We also apply affine transformations as the reverse transformation function.

On top of the reverse transformation function, the update function u is designed in Algorithm 3. It decreases the loss $\mathcal{L}$ w.r.t. the reverse transformation parameter θ via zeroth-order gradient descent controlled by the sign direction and a step size λ. u is called at most $n = 100$ times to avoid nonconvergence. After reaching the ϵ-robust area, we apply L^2-norm as the distance metric for membership inference.

The attack accuracy of current and our attacks on CIFAR-10 models are shown in Table I, with the best accuracy of

Algorithm 3: Update function u for transformations on pixel values

```
Input: transformed sample $x'$, true label $y$, target model
       $f$, loss function $\mathcal{L}$, reverse transformation
       function $g^-$ parameterized by the noise $\theta$, step
       size $\lambda$, small constant $\sigma_0$
Output: updated noise $\theta$
$\eta \leftarrow$ sample($N(0, \sigma_0^2)$);
$g \leftarrow \frac{\mathcal{L}(f(g^-(x',\theta+\eta)),y)-\mathcal{L}(f(g^-(x',\theta)),y)}{\eta}$;  ▷ Gradient
  estimation
$\theta \leftarrow \theta - \lambda \cdot$ sign($g$);
return $\theta$;
```

each scenario highlighted. When the transformations are small, regular attacks can still perform well due to the model's robustness to small transformations. However, their accuracy would drop significantly when the transformation becomes larger or adversarially designed. CC attack's accuracy sometimes gets better than attacking original samples for small transformations because slight noises cause fewer misclassifications in the train set, which aligns with our intuition.

Table I shows that both LTT and RT attacks outperform the rest, especially for larger transformations. The results indicate that attacks that focus on original samples can underestimate the real privacy risks. In addition, we can see that RT outperforms LTT in several cases, such as adversarial noises, rotations, and translations. The former is due to the design of RT, which exploits the local minima property of training samples, while LTT relies only on logits. The latter two support that RT can perform better when the reverse transformation function is close to the mathematical inverse of the transformation.

C. Evaluation on Purchase-100

Different from images, gradient estimation via binary features would be inaccurate. Instead, we propose the following reverse transformation function $g^-(x', \mathcal{S})$: flipping x' by all the binary features whose indices are in the set $\mathcal{S}$. The corresponding update function u is defined in Algorithm 4. We flip the single feature that decreases the loss value most whenever we call u on a given input. We repeat this process until the sample is moved back to the ϵ-robust area. Finally, we define $d(\mathcal{S}) = |\mathcal{S}|$, which is the total number of features flipped, as the distance metric for membership inference.

In the experiment, we select a fixed $\epsilon = 0.5$ for the reverse transformation attack, but we vary the missing ratio γ from 0.05 to 0.40 to see how it would affect our attacks.

Table II shows the performance of SOTA and our attacks on Purchase-100, with the best highlighted. Similar to the results of CIFAR-10, SOTA attacks still work at a small γ. However, the success rate decreases quickly as γ increases. We can see that both LTT and RT outperform other attacks. Specifically, LTT performs better than RT when the transformation is small (with missing ratios $\gamma \leq 0.3$), while RT can still successfully infer training members when the transformation is large (with a missing ratio $\gamma = 0.3$) when all current attacks drop to random guesses when the missing ratio $\gamma = 0.25$.

Algorithm 4: Update function u for missing features

```
Input: transformed sample $x'$, true label $y$, target model
       $f$, loss function $\mathcal{L}$, reverse transformation
       function $g^-$, set of flipped feature indices $\mathcal{S}$,
       number of features $n$
Output: updated set of flipped features $\mathcal{S}$
$l \leftarrow \mathcal{L}(f(g^-(x', \mathcal{S})), y)$;
$k \leftarrow -1$;
for $i \leftarrow 0$ to $n$ do
    if $i \notin \mathcal{S}$ and $\mathcal{L}(f(g^-(x', \mathcal{S} + \{i\})), y) < l$ then
        $l \leftarrow \mathcal{L}(f(g^-(x', \mathcal{S} + \{i\})), y)$;
        $k \leftarrow i$;
    end
end
$\mathcal{S} \leftarrow \mathcal{S} + \{k\}$;
return $\mathcal{S}$;
```

V. Evaluation on more general scenarios

A. Evaluation on data augmented models

In Section III and IV, we only consider training models with original samples to focus our attention to data transformations. However, data augmentation via transformations is also common in reality, where samples used in model training and collected by the attacker are variations of the original samples via two different transformations.

This section will evaluate our attacks on data augmented models trained on CIFAR-10. Specifically, we consider the following data augmentation when training the target model and assume the attacker knows the augmentation: (1) Random affine transformations with a random rotation of up to 10 degrees and a random translation of up to 3 pixels in each direction, (2) Color jittering with a random adjustment of brightness, contrast, and saturation with a maximum ratio of 0.2, and (3) Gaussian noise from $N(0, 0.1)$. The target model trained by data augmentation has a 99% training accuracy and a 73% testing accuracy.

In addition to the SOTA attacks we evaluated in Section IV, we also consider the moment attack (MM) proposed by [24] specifically designed for data-augmented models. We set the number of the augmented instances to be 10 and concatenate moments of orders from 1 to 20, which is the same as in [24].

We also evaluated our attack strategies on original samples since they can be regarded as an (inverse) transformed version of the augmented samples in model training. Specifically, suppose x is the original data, $x' = g(x)$ is the augmented data with transformation g, and $x = g^{-1}(x')$ where g^{-1} is the mathematical inverse of g. We can then switch the role of x and x', that is, x' is now regarded as the original sample and x is the transformed version of x' via transformation g^{-1}. As a result, attacking x is the same as having access to transformation g^{-1}, so we can apply our attack strategies

TABLE II: Accuracies of MIAs on Purchase-100

Att. \ γ		Original	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40
Current	CC	63.33%	61.72%	57.54%	53.74%	51.61%	50.54%	50.18%	50.17%	50.13%
	LT	70.68%	61.15%	55.47%	52.42%	51.11%	50.37%	50.20%	50.12%	50.10%
	ST	59.36%	56.51%	53.78%	52.26%	51.30%	50.64%	50.32%	50.39%	50.21%
	ET	70.99%	61.46%	55.62%	52.47%	51.09%	50.38%	50.20%	50.11%	50.09%
	NN	65.65%	60.58%	56.01%	52.93%	51.37%	50.63%	50.28%	50.16%	50.09%
	LO	70.65%	59.96%	54.88%	52.22%	50.81%	50.29%	50.08%	49.98%	50.01%
Ours	LTT	-	**64.09**%	**58.03**%	55.40%	52.79%	51.96%	50.70%	50.11%	49.83%
	RT	-	61.94%	57.27%	**55.49**%	**53.70**%	**52.90**%	**52.02**%	**50.57**%	**50.41**%

TABLE III: Accuracies of MIAs on CIFAR-10 augmented models

Att. \ Trans.		Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.3]	loss=10	1	0.5	Moon	$[0°, 30°]$	7
Current	CC	63.32%	62.65%	51.58%	54.59%	56.43%	60.08%	61.02%	51.62%
	LT	72.49%	67.36%	50.16%	51.50%	53.47%	58.66%	60.71%	50.59%
	ST	71.55%	66.43%	50.41%	51.76%	53.71%	58.03%	60.61%	50.77%
	ET	71.56%	66.95%	50.17%	51.58%	53.58%	58.59%	60.45%	50.58%
	NN	71.05%	67.95%	52.57%	50.17%	53.70%	56.10%	58.38%	50.80%
	LO	71.81%	65.41%	50.37%	51.14%	54.08%	58.00%	58.75%	49.88%
	MM	**73.76**%	58.61%	52.25%	52.32%	56.65%	57.88%	61.94%	50.43%
Ours	LTT	72.49%	**68.10**%	61.42%	55.06%	56.98%	60.57%	62.42%	52.10%
	RT	73.20%	67.30%	**67.12**%	**56.19**%	**60.50**%	**61.08**%	**71.28**%	**57.32**%

the same as attacking transformed samples on regular models. Note that our first strategy is essentially the same as regular attacks when attacking original samples on augmented models.

Similarly, the MM attack may also be adapted to attacking transformed samples on regular models if we switch the role of original and transformed samples. Unfortunately, there is an inevitable difficulty: an essential assumption for initiating the MM attack is the knowledge of the data augmentation, which would be g^{-1} in this case. In reality, g^{-1} is not available for most g. Moreover, many transformations are strictly lossy, and even invertible transformations can have information loss due to value clipping. As a result, we only evaluate the original version of the MM attack when attacking transformed samples.

Results of data augmented models are in Table III. Like attacking unaugmented models, LTT and RT outperform all other attacks when facing transformations. On the other hand, MM achieves the best attack accuracy among all SOTA attacks when attacking original untransformed samples. We can see that RT can also achieve comparable performance to MM (both $> 73\%$) on original samples.

B. Evaluation on defended models

This section evaluates how effective the SOTA defenses are when facing data transformation. Here we provide evaluation results of CIFAR-10 models trained with three popular defenses: (1) Differentially-private (DP) training [6], (2) Adversarial regularizations [7], and (3) MemGuard [8]. Similarly, we assume that attackers know the defense algorithms so they can adaptively train shadow models with the same defenses.

For the first defense, we applied the implementation from *Opacus* [30] with an RMSprop optimizer for better convergence. Then we set a gradient clipping value of 1.2, a noise multiplier of

TABLE IV: Accuracies of MIAs on CIFAR-10 models trained with DP

Att. \ Trans.		Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.3]	loss=10	1	0.5	Moon	$[0°, 30°]$	7
Current	CC	50.67%	50.48%	50.00%	50.24%	50.34%	50.62%	50.51%	50.07%
	LT	50.79%	50.53%	50.01%	50.30%	50.27%	**50.68**%	50.53%	50.38%
	ST	50.55%	50.32%	50.04%	50.08%	50.25%	50.50%	50.19%	**50.53**%
	ET	50.86%	50.53%	50.00%	50.28%	50.35%	50.69%	50.57%	50.19%
	NN	49.82%	50.14%	50.17%	50.18%	49.97%	50.08%	50.18%	49.75%
	LO	50.56%	50.25%	50.01%	50.31%	50.30%	50.26%	50.21%	49.35%
Ours	LTT	-	**50.91**%	**51.42**%	**50.54**%	**50.78**%	50.33%	**51.10**%	50.49%
	RT	-	50.89%	51.32%	50.52%	50.75%	50.32%	50.92%	50.40%

TABLE V: Accuracies of MIAs on CIFAR-10 models trained with adversarial regularization

Att.	Trans.	Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.3]	loss=10	1	0.5	Moon	$[0°, 30°]$	7
Current	CC	60.85%	55.30%	51.15%	51.93%	53.55%	54.66%	54.91%	50.07%
	LT	60.96%	55.39%	51.35%	51.79%	53.12%	54.51%	54.83%	50.31%
	ST	58.91%	54.97%	52.27%	51.46%	52.34%	54.27%	53.75%	50.11%
	ET	60.03%	54.51%	50.25%	51.68%	53.37%	53.99%	54.48%	50.12%
	NN	60.88%	55.78%	**53.75**%	50.50%	50.95%	54.03%	51.18%	50.20%
	LO	61.45%	52.45%	50.70%	51.96%	52.61%	54.10%	52.69%	49.50%
Ours	LTT	-	**57.08**%	50.39%	**52.52**%	53.62%	55.54%	55.05%	50.52%
	RT	-	56.12%	51.01%	52.37%	**53.72**%	**55.68**%	**55.48**%	**51.00**%

TABLE VI: Accuracies of MIAs on CIFAR-10 models protected by MemGuard

Att.	Trans.	Original	Gaussian(σ)	Adversarial	JPEG(q)	Scaling(r)	Filtering	Rotation(δ)	Translation(d)
		-	[0,0.3]	loss=10	1	0.5	Moon	$[0°, 30°]$	7
Current	CC	65.37%	63.92%	50.10%	**56.11**%	58.03%	**64.27**%	60.35%	49.93%
	LT	65.37%	63.92%	50.10%	56.11%	58.03%	64.27%	60.35%	49.93%
	ST	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%	50.00%
	ET	65.37%	63.92%	50.10%	56.11%	58.03%	64.27%	60.35%	49.93%
	NN	50.58%	49.63%	50.00%	48.83%	49.23%	49.88%	49.00%	49.65%
	LO	76.85%	65.30%	50.05%	53.45%	56.50%	62.40%	60.40%	**50.55**%
Ours	LTT	-	62.27%	50.00%	50.00%	58.03%	64.27%	60.51%	50.00%
	RT	-	62.68%	50.00%	50.00%	50.00%	64.27%	60.55%	50.00%
	LOT	-	**71.50**%	**50.10**%	54.50%	**59.25**%	63.95%	**62.34**%	50.35%

2.3, and trained the target model for 100 epochs, achieving a (5, 1e-5)-differential privacy. For the second defense, we followed the attack model architecture in [7] and set the regularization weight $\lambda = 5$. For the third defense, we applied the same implementation as in [10]. All the other training settings are the same as in Section IV-A in order to control variates. The final performance (training accuracy/testing accuracy) of the models trained with the first two defenses are (1) 40%/39%, and (2) 75%/52%, respectively. MemGuard is directly applied to the undefended model and does not affect model performance.

Table IV, V, and VI show the attack results on defended models. In Table IV, we can see that DP with a small privacy budget provides the best privacy guarantee, reducing all attacks to ($\sim 50\%$), with a large privacy-utility trade-off. The trade-off also exists in adversarial regularizations. MemGuard doe not suffer from the trade-off while providing excellent robustness against attacks that heavily rely on confidence vectors. From Table VI, we can see that ST, NN, and our RT are deteriorated by MemGuard, while LT and ET are essentially the same as CC. On the other hand, we can see that MemGuard is not effective under LO, which has also been reported in [10]. Thus, an adaptive attacker can apply our strategy 1 to LO and achieve better attack performance when facing transformed data, as shown in row LOT in Table VI.

In summary, we can see that all three defenses are effective against all MIAs. Though the overall success rate is reduced significantly, applying our attack strategies can always help increase the performance when facing data transformations.

VI. Conclusion

In this paper, we study MIA in face of potential transformation, which is somehow overlooked by previous research. We believe transformed data is of similar interest to attackers if it retains semantic and private information of the original training data. On this point, we have made an attempt toward broadening the concept of membership and generalizing the threat model of MIAs, by taking semantic-preserving data transformations into account. We have proposed two strategies for designing attacks to infer data that has probably been transformed. The former strategy directly improves current attacks by setting optimal parameters (e.g., thresholds) based on knowledge of transformations. The latter leverages the difference in transformation sensitivity between training and test samples and tries to distinguish them by comparing their shortest distance to ϵ-robust areas. We have shown by extensive experimental evaluations that, derived from the two strategies, one can outperform current state-of-the-arts on inferring transformed members, with a variety of different transformation types and levels. We hope our paper can inspire future work towards generalizing MIAs to more realistic settings and reducing possible bias in accessing model privacy in practice.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1801751, No. 1956364 and No. 1937786.

References

[1] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[2] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[3] Kaggle, "Kaggle: Acquire valued shoppers challenge," https://www.kaggle.com/c/acquire-valued-shoppers-challenge, 2014.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[5] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[6] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 308–318.

[7] M. Nasr, R. Shokri, and A. Houmansadr, "Machine learning with membership privacy using adversarial regularization," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 634–646.

[8] J. Jia, A. Salem, M. Backes, Y. Zhang, and N. Z. Gong, "Memguard: Defending against black-box membership inference attacks via adversarial examples," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 259–274.

[9] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*. IEEE, 2018, pp. 268–282.

[10] C. A. C. Choo, F. Tramer, N. Carlini, and N. Papernot, "Label-only membership inference attacks," *arXiv preprint arXiv:2007.14321*, 2020.

[11] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *2020 ieee symposium on security and privacy (sp)*. IEEE, 2020, pp. 1277–1294.

[12] Z. Li and Y. Zhang, "Membership leakage in label-only exposures," *arXiv preprint arXiv:2007.15528*, 2020.

[13] S. Rahimian, T. Orekondy, and M. Fritz, "Sampling attacks: Amplification of membership inference attacks by repeated queries," *arXiv preprint arXiv:2009.00395*, 2020.

[14] K. Leino and M. Fredrikson, "Stolen memories: Leveraging model memorization for calibrated white-box membership inference," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 1605–1622.

[15] A. Sablayrolles, M. Douze, C. Schmid, Y. Ollivier, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5558–5567.

[16] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro, "Logan: Membership inference attacks against generative models," in *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2019, no. 1. De Gruyter, 2019, pp. 133–152.

[17] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models." *Proc. Priv. Enhancing Technol.*, vol. 2019, no. 4, pp. 232–249, 2019.

[18] K. S. Liu, C. Xiao, B. Li, and J. Gao, "Performing co-membership attacks against deep generative models," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 459–467.

[19] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov, "Exploiting unintended feature leakage in collaborative learning," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 691–706.

[20] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *2019 IEEE symposium on security and privacy (SP)*. IEEE, 2019, pp. 739–753.

[21] Q. Chen, C. Xiang, M. Xue, B. Li, N. Borisov, D. Kaarfar, and H. Zhu, "Differentially private data generative models," *arXiv preprint arXiv:1812.02274*, 2018.

[22] L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou, "Differentially private generative adversarial network," *arXiv preprint arXiv:1802.06739*, 2018.

[23] J. Chen, Y. Guo, Q. Zheng, and H. Chen, "Protect privacy of deep classification networks by exploiting their generative power," *Machine Learning*, vol. 110, no. 4, pp. 651–674, 2021.

[24] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, "How does data augmentation affect privacy in machine learning?" 2021.

[25] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th {USENIX} Security Symposium ({USENIX} Security 21)*, 2021.

[26] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[27] A. Kamakura, "pilgram: A python library for instagram filters,," https://github.com/akiomik/pilgram, 2019.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models," *arXiv preprint arXiv: 1806.01246*, 2018.

[30] Facebook, "Opacus: Train pytorch models with differential privacy," https://opacus.ai/, 2020.