Benjamin Hilprecht*, Martin Härterich, and Daniel Bernau

Monte Carlo and Reconstruction Membership Inference Attacks against Generative Models

Abstract: We present two information leakage attacks that outperform previous work on membership inference against generative models. The first attack allows membership inference without assumptions on the type of the generative model. Contrary to previous evaluation metrics for generative models, like Kernel Density Estimation, it only considers samples of the model which are close to training data records. The second attack specifically targets Variational Autoencoders, achieving high membership inference accuracy. Furthermore, previous work mostly considers membership inference adversaries who perform single record membership inference. We argue for considering regulatory actors who perform set membership inference to identify the use of specific datasets for training. The attacks are evaluated on two generative model architectures, Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), trained on standard image datasets. Our results show that the two attacks yield success rates superior to previous work on most data sets while at the same time having only very mild assumptions. We envision the two attacks in combination with the membership inference attack type formalization as especially useful. For example, to enforce data privacy standards and automatically assessing model quality in machine learning as a service setups. In practice, our work motivates the use of GANs since they prove less vulnerable against information leakage attacks while producing detailed samples.

Keywords: Machine Learning, Privacy, Information Security

DOI 10.2478/popets-2019-0067
Received 2019-02-28; revised 2019-06-15; accepted 2019-06-16.

***Corresponding Author: Benjamin Hilprecht:** TU Darmstadt, Darmstadt, Germany, E-mail: benjamin.hilprecht@cs.tu-darmstadt.de (This work was done during an internship at SAP.)
Martin Härterich: SAP SE, Karlsruhe, Germany, E-mail: martin.haerterich@sap.com
Daniel Bernau: SAP SE, Karlsruhe, Germany, E-mail: daniel.bernau@sap.com

1 Introduction

Machine learning is ubiquitous in software applications nowadays. However, the success of machine learning (ML) depends as much on sophisticated algorithms as it does on the availability of large sets of training data. Gathering sufficient amounts of training data for satisfying model generalization has proven cumbersome especially for sensitive data and, in some cases, resulted in privacy violations due to data misuse (e.g., the inappropriate legal basis for the use of National Health Service (NHS) data in the DeepMind project [10, 22, 24]). The desire to identify on which data a model was trained, and thus detect privacy violations gave rise to model inversion, which aims for reconstructing a training dataset with missing parts [7], and membership inference (MI) [21]. Within this work we address the latter, striving to identify whether an individual or a set of individuals, *belong to* a certain training dataset.

Motivated by the recent NHS misuse case we consider two membership inference actors: an adversary performing *single* record MI and a regulator performing *set* MI. Single MI is used in previous work to model an adversary who is mainly interested in identifying individuals within a dataset. However, set MI is relevant for regulatory audits since it can be used to prove that a specific set of records was used to train a model. If the practitioner who trained the model was not authorized to use a specific dataset for this purpose regulators can apply set MI to prove data privacy violations.

We propose and evaluate two novel membership inference attacks against recent generative models, *Generative Adversarial Networks* (GAN) [8] and *Variational Autoencoders* (VAE) [12]. These generative models have become effective tools for (unsupervised) learning with the goal to produce samples of a given distribution after training. Generative models thus have many applications like the synthesis of photo-realistic images, image-to-image translation, and even text [3] or sound [5] synthesis. However, the MI attack of Shokri et al. [21] against discriminative models is not directly applicable to generative models and thus alternative means are required. Moreover, previous attacks on generative models were specialized on GANs [10]. In contrast, our first attack is applicable to every generative model from which one can draw samples. The attack only considers samples which are very close to train or test records giving it an edge over existing methods like the Euclidean attack [9]. The second proposed attack is solely applicable to Variational Autoencoders. Hence, our attacks allow membership inference attacks against a broader class of generative models. In some cases, the attacks formulated in this work yield accuracies close to 100%, clearly outperforming previous work. Furthermore, the regulatory actor performing set MI helps to unveil even slight information leakage. Hence, set MI is of high practical relevance for enforcing data privacy standards.

The close connection of information leakage to overfitting provides another motivation for this work. Severe overfitting will result in the replication of given data in generative models and therefore higher accuracies of membership inference attacks. Given that in extreme cases a linear relationship between the success of membership inference attacks and overfitting has been observed for discriminative models [7] we also want to avoid overfitting in the case of generative models. However, overfitting is neither straightforward to define nor identify for generative models.

As proposed by Hayes et al. [10], the accuracy of attacks in single MI can be used as an indicator for overfitting. We thoroughly compare our attacks against state of the art attacks on generative models introduced by Hayes et al. [10] to further investigate this claim. The proposed type of set membership inference results in higher accuracy values and is potentially a means for identifying even slight overfitting in generative models. For machine learning as a service (MLaaS) our attacks are therefore potentially a means for automatically assessing the quality of the learned generative model more accurately than previous approaches. The main contributions of this work are:

- a membership inference attack based on Monte Carlo integration that exclusively considers small distance samples from the model,
- a membership inference attack designed for Variational Autoencoders: the *Reconstruction attack*,
- and a membership inference variation performing *set membership inference*, which is systematically evaluated and which we envision to be used by regulators to enforce data privacy standards.

We evaluated the attacks on the image datasets MNIST, Fashion-MNIST, and CIFAR-10 for both Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs) which are widely used generative models. For VAEs, the Reconstruction attack yielded accuracies close to 100% for set MI and between 57% and 99% for single MI. The MC attack reached between 72% and 100% set MI accuracy and up to 60% accuracy for single MI. The attacks were less effective on GANs in our experiments. However, the MC attack accuracies against GANs range from 65% to 75% for set MI. In general, the MC attack performs better if the samples drawn from the model are of high quality.

This paper is structured as follows. Section 2 explains the threat model and membership inference attacks considered in this work. In particular, we introduce and formalize two actors who perform single and set membership inference. Furthermore, we argue for their relevance in real-world use cases. In Section 3 we introduce and formalize our two attacks which are applicable to both single and set membership inference. To this end, details regarding GANs and VAEs are provided. The subsequent Section 4 contains an evaluation of our attacks on reference datasets. Related work is discussed in Section 5. A summary and outlook (Section 6) concludes the paper.

2 Membership Inference Attacks

In this section, we introduce the threat model and the two kinds of attacks considered in this paper: single MI and set MI. We start the section by exposing some background on MI.

2.1 Background of Membership Inference

The goal of membership inference (MI) is to gather evidence that a specific record or a set of records belongs to the training dataset of a given machine learning model. MI thus represents an approach for measuring how much a model leaks about individual records of a population. The success rates of MI attacks against a model are tightly linked to overfitting (i.e., the generalization error [30]). The poorer a model generalizes the more specificities it contains about individual training data records.

In this work, two kinds of MI are considered: single MI and set MI. The single MI is comparable to common experiment setups for MI[10, 21]. In the set MI setting a regulator has to recognize which of the two provided sets contains training data records.

2.2 Threat Model

This work considers two actors corresponding to single and set MI, respectively. The first actor is an honest-but-curious adversary $\mathcal{A}$ and the second actor is a regulatory body $\mathcal{R}$. Each actor focuses on a specific task: the adversary $\mathcal{A}$ is common in MI literature and infers whether a single record was present in the training dataset using *single* membership inference. The regulatory body $\mathcal{R}$ performs *set* membership inference to identify whether a set of records was present in the training dataset. This attack can provide evidence that a certain set of training data was illegally used to train a generative model.

Both actors are assumed to have no access to the underlying training dataset of the generative model, and they refrain from activities that maliciously modify this target model. The actors $\mathcal{A}$ and $\mathcal{R}$ can both launch the Monte Carlo (MC) attack as well as the Reconstruction attack. (See Section 3 for details.) The choice of the attack determines the requirements on the information that is available to the actor. The MC attack requires samples drawn from the generative model while the Reconstruction attack has to be able to evaluate the generative model.

2.3 Adversarial Actor: Single MI

Single MI has been used by previous work to evaluate attacks against GANs [10]. In this setting, the honest-but-curious adversary $\mathcal{A}$ has to identify individual records which were used to train the model. To this end M records from the training data and M records from the test dataset $\{x_1, \dots, x_{2M}\}$ are given. Both the MC attack and the Reconstruction attack rely on a function $\hat{f}(x)$ that can be computed for each of the records. The intuition is that this function attains higher values for training data records. Details on how this function is realized are given in the next section. In the following description of the attack types we use the general notation $\hat{f}(x)$.

For every record x_i, $\mathcal{A}$ has to decide whether it was part of the training data. In general, $\mathcal{A}$ picks the M records with the M greatest values of the function $\hat{f}(x)$.

Attack Type 1 (Single Membership Inference). *Let $\mathcal{A}$ be an adversary who is able to compute the function $\hat{f}(x)$ for every record x.*

1. *Choose records $\{x_1, \dots, x_M\}$ from the training data.*
2. *Choose records $\{x_{M+1}, \dots, x_{2M}\}$ from the test data.*
3. *$\mathcal{A}$ is presented the set $\{x_1, \dots, x_{2M}\}$.*
4. *$\mathcal{A}$ labels the M records with highest values $\hat{f}(x_i)$ as training data.*

We denote the M records chosen by $\mathcal{A}$ as $\{x_1^{\mathcal{A}}, \dots, x_M^{\mathcal{A}}\}$. We call the proportion of actual training data in this set

$$\frac{1}{M} \cdot \left|\{i \mid x_i^{\mathcal{A}} \in \{x_1, \dots, x_M\}\}\right|$$

the accuracy of the attack for single MI.

2.4 Regulatory Actor: Set MI

Set MI corresponds to the needs of regulators and auditors aiming to prove data privacy violations in machine learning. One set consisting of M records from the training data $\{x_1, \dots, x_M\}$ and another set consisting of M records from the test data $\{x_{M+1}, \dots, x_{2M}\}$ are shown to a regulator $\mathcal{R}$ in either order. The task of $\mathcal{R}$ is to decide which of the two sets is a subset of the original training data. Contrary to single MI, $\mathcal{R}$ knows which records belong to the same data source (training data or test data). However, $\mathcal{R}$ does not know which set is a subset of the original training data.

Similar to single MI $\mathcal{R}$ computes the function $\hat{f}(x)$ for every record and selects the M records with the M highest values $\hat{f}(x)$. For each of the selected records, $\mathcal{R}$ checks to which set it belongs and eventually selects the set from which most of these records stem as subset of the original training data.[1] Note that this is equivalent to taking the set with the higher median. Since we do not have any prior knowledge on the type of distribution of the $\hat{f}$-values this is more robust than considering e.g. the mean.

Attack Type 2 (Set Membership Inference). *Let $\mathcal{R}$ be an adversary able to calculate the function $\hat{f}(x)$ for every record x.*

1. *Choose records $\{x_1, \dots, x_M\}$ from the training data.*
2. *Choose records $\{x_{M+1}, \dots, x_{2M}\}$ from the test data.*
3. *$\mathcal{R}$ is presented the sets $\{x_1, \dots, x_M\}$ and $\{x_{M+1}, \dots, x_{2M}\}$.*
4. *$\mathcal{R}$ identifies the M records with highest values $\hat{f}(x_i)$.*
5. *$\mathcal{R}$ chooses the set from which most of these records stem.*
6. *If both have the same number of representatives $\mathcal{R}$ picks one set randomly.*

[1] If an equal number of records belong to the first and the second set, $\mathcal{R}$ picks one of the sets with probability 50%.

Table 1. Comparison of Attacks

Attack	Required Access	Applicable	Idea
White-box	Discriminator	GANs	Evaluate Discriminator
Black-box	Samples from Generative Model	Generative Models	Train auxiliary GAN on samples and evaluate Discriminator
Monte Carlo	Samples from Generative Model	Generative Models	Monte Carlo approximation on close samples
Reconstruction Attack	VAE model	VAEs	VAE reconstructs training data more precisely

The accuracy of an attack of this type is defined as the average success rate of $\mathcal{R}$, i.e., the probability that $\mathcal{R}$ identifies the true subset of the training data.

2.5 Relevance for Real-World Use Cases

The formalized MI attack types are an alternative to assessing a single record x by computing $\hat{f}(x)$ and considering the record part of the training data if the value exceeds a threshold. While the single record approach is conceptually similar, the formalized types contributed in this work are closer to real-world use cases. For example, in machine learning as a service (MLaaS) applications access to both test and training data is implicitly given. Hence, the single MI and set MI attack types can be automatically conducted. Increased MI attack accuracies suggest that the model quality is insufficient w.r.t. privacy.

Figure 1 visualizes the regulatory use case. The regulator $\mathcal{R}$ suspects that a certain dataset was illegally used to train a model (b). Actually, even more data was used illegally (c). Moreover, some legally obtained data might have been used. Together with the illegal data, it represents the complete training data (d). $\mathcal{R}$'s set of suspected data is used as train set in the set MI attack (a). $\mathcal{R}$ also needs test data (f) from which a subset (e) is used as test set for the attack. If the attack is successful the illegal use can be proven. Otherwise, the attack does not perform better than random guessing. By repeating the attack for multiple choices of subsets (a) and (f) $\mathcal{R}$ ensures statistical significance. Note that $\mathcal{R}$ does not need to know the entire training data since the MI attacks also work for subsets of the entire training data. The accuracy does not depend on the concrete subset choice as we will show in our experiments in Section 4.

Note that in both single and set MI we assume that there are exactly as many test as train records. In the regulatory use case of set MI this is realistic since a sample of the larger of the two sets can be used if they are not of equal size. To make the results of single and set MI comparable, and to be in line with the balanced setting in previous work [21], we also decided to use this setup in single MI. Note that this is potentially an advantage for $\mathcal{A}$.

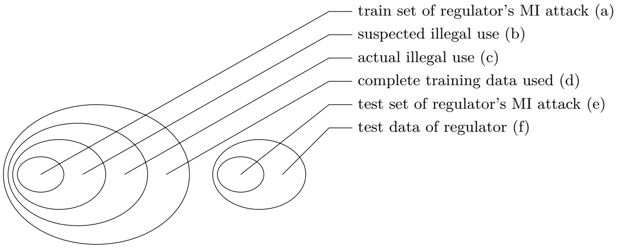

train set of regulator's MI attack (a)
suspected illegal use (b)
actual illegal use (c)
complete training data used (d)
test set of regulator's MI attack (e)
test data of regulator (f)

Fig. 1. Venn diagram of training and test data in the regulatory use case for $\mathcal{R}$.

3 Attack Details

In this section we introduce two novel MI attacks. They can be used for both single and set MI. The first attack, namely the *Monte Carlo* attack (Section 3.2) compares samples drawn from the model to either test or train records. Opposing to existing approaches, only very close samples are considered. Indeed, this distinguishes the attacks from previous approaches like the Euclidean attack [9] and made the attacks effective. Furthermore, the *Reconstruction* attack (Section 3.3) which is optimized for VAEs is presented. A comparison of our attacks and state-of-the-art attacks is given in Table 1. Again, an attack is fully specified by the function $\hat{f}(x)$ which will be introduced in the following. Since in the description of the attacks details about generative models are required, we briefly describe VAEs and GANs in the next section.

3.1 Generative Models

Generative models are ML models that are trained to learn the joint probability distribution $p(X, Y)$ of features X and labels Y of training data. In this paper we apply two decoder based models re-

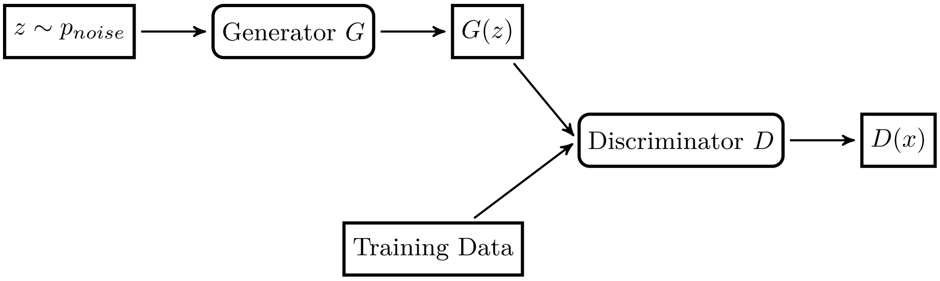


Fig. 2. Architecture of a Generative Adversarial Network (GAN).

lying on neural networks, namely *Generative Adversarial Networks* (GANs) [8] and *Variational Autoencoders* (VAEs) [12]. Note, however, that our Monte Carlo attack is applicable to all generative models from which one can draw samples. The reconstruction attack specifically targets VAEs.

3.1.1 Generative Adversarial Networks

A GAN consists of two competing models, a *generator* G and a *discriminator* D, which are trained in an adversarial manner (i.e., compete against each other). We describe the approach in detail referring to Figure 2.

To generate artificial data a prior z is sampled from a prior distribution p_{noise} (e.g., Gaussian) and fed as input into the generator G. The task of the discriminator D is to output the probability that generated samples stem either from the training data or G. However, G tries to fool D by generating samples that D misclassifies. Hence, the outputs $G(z)$ should look similar to the training data x (i.e. records sampled from p_{data}). This is expressed as a two-player zero-sum game via the following objective function:

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_{noise}}[\log(1-D(G(z)))].$$

Gradients are computed for G and D during training, and usually, after already a few steps of training G produces realistic outputs. A conditional generative model is obtained by providing a condition c (e.g., a class label) as an input both to the generator and the discriminator [8].

3.1.2 Variational Autoencoders

VAEs [12] consist of two networks - an encoder E and a decoder D. During training each record x is given to the encoder which outputs the mean $E_\mu(x)$ and variance $E_\Sigma(x)$ of a Gaussian distribution. A latent variable z is sampled from this distribution $N(E_\mu(x), E_\Sigma(x))$ and fed into the decoder D. The reconstruction $D(z)$ should be close to the training data record x.

During training two terms need to be minimized. First, the reconstruction error $\|D(z) - x\|$. Second $KL(N(E_\mu(x), E_\Sigma(x))||N(0,1))$, the *Kullback-Leibler divergence* between the distribution of the latent variables z and the unit Gaussian. The second term prevents the network from only memorizing certain latent variables because the distribution should be similar to the unit Gaussian. In practice, both the encoder E and the decoder D are neural networks. Kingma et al. [12] provide details on how to train those networks given the training objective with the reparametrization trick. Moreover, they motivate the training objective as a lower bound on the log-likelihood. Sampling from the VAE is achieved by sampling a latent variable $z \sim N(0,1)$ and passing z through the decoder network D. The outputs of the decoder $D(z)$ then serve as samples. Like for GANs, a conditional variant is obtained by providing a condition c as input to the decoder and the encoder.

3.2 Monte Carlo Attack

In the following section we introduce the first attack which is applicable to all generative models. The intuition behind the Monte Carlo attack is that the generator G overfits if it tends to output datasets close to the provided training data. Formally, let $U_\varepsilon(x)$ denote the ε-neighborhood of x defined as $U_\varepsilon(x) = \{x' \mid d(x,x') \leq \varepsilon\}$ with respect to some distance d. If a sample g of the generative model G is likely to be close to a record x the probability $P(g \in U_\varepsilon(x))$ is increased. It can be rewritten as

$$P(g \in U_\varepsilon(x)) = \mathbb{E}_{g\sim p_{generator}}\left(\mathbf{1}_{g\in U_\varepsilon(x)}\right)$$

and approximated via Monte Carlo integration [17]

$$\hat{f}_{MC-\varepsilon}(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}_{g_i \in U_\varepsilon(x)}, \tag{1}$$

where $g_1, \ldots, g_n$ are samples from $p_{generator}$. Note that samples g_i of the generator G are ignored if their distance to the training data record x is higher than ε. In this attack, the estimation $\hat{f}_{MC-\varepsilon}(x)$ plays the role of the function $\hat{f}(x)$ attaining higher values for training data records.

An alternative is provided by incorporating the exact distances $d(z_i, x)$ between samples $g_1, \ldots, g_n$ and training data x, and computing

$$\mathbb{E}_{g\sim p_{generator}}\left(-\mathbf{1}_{g\in U_\varepsilon(x)}\log\left(d(g,x)+\delta\right)\right)$$

where a small δ is chosen to clip off large values ("avoid log(0)") if the distance is zero. The logarithm is to ensure that outliers do not affect the results too much. The Monte Carlo approximation is then given by

$$\hat{f}_{MC-d}(x) = -\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}_{g_i\in U_\varepsilon(x)}\log d(g_i, x) \quad . \qquad (2)$$

Here, the estimation $\hat{f}_{MC-d}(x)$ plays the role of the function $\hat{f}(x)$ used to conduct the attack types presented above.

In the case of GANs and VAEs one obtains $g_i \sim p_{generator}$ by sampling from $z_i \sim p_{noise}$ and computing $g_i = G(z_i)$ and $g_i = D(z_i)$, respectively. Note that only a sufficiently large amount of samples has to be provided and no additional information is required. Of course, both attack variants depend on the specification of the distance $d(\cdot,\cdot)$. See below for details.

A further alternative to the attacks discussed could be realized using a Kernel Density Estimator (KDE) [18]. In the following we briefly compare the Monte Carlo attack with this metric. An estimation of the likelihood $\hat{f}(x)$ of a data point x using KDE is given by

$$\hat{f}_{KDE}(x) = \frac{1}{nh^d}\sum_{i=1}^{n}K\left(\frac{x-g_i}{h^d}\right), \qquad (3)$$

where K is typically the Gaussian kernel and h denotes the bandwidth. If this likelihood $\hat{f}_{KDE}(x)$ is significantly higher for training data than for test data the model fails to generalize. Likewise the approximate likelihood values $\hat{f}_{KDE}(x)$ can be used as the function $\hat{f}(x)$ to conduct the single and set MI attack types. However, this attack variation did not perform better than random guessing and is therefore not considered in our evaluation section.

Note that KDE (3) can indeed be interpreted as a special case of the proposed distance based method (2), where

$$d(x, g_i) = 1/\exp(h^d \cdot K((x-g_i)/h^d)), \text{ and}$$
$$\varepsilon = \max_{i=1,\ldots,n} d(x, g_i) \quad .$$

As KDE does not perform well for MI against generative models this stresses that choosing the right distance function seems to be key. In contrast to KDE, our attacks exclusively consider samples significantly close to training data x. To fully specify the Monte Carlo attacks concrete distance measures and heuristics for choosing ε are required. We describe our approach for this in the next two subsections.

3.2.1 Distance Measures

Both Monte Carlo (MC) attack variants require a distance function $d(\cdot,\cdot)$ and the distance plays an important role for the success of the MI attack. Therefore, a distance metric suited for the specific data under consideration has to be chosen. For neural networks, image recognition has become a key task and consequently, we formulate distance metrics for image data in the following paragraphs.

Principal Components Analysis. Images are initially represented as a vector of their pixel intensities. A principal component analysis (PCA) is then applied to all vectors in the test dataset. The top 40 components are kept while all other components are discarded. When computing the distance between two new images the PCA transformation is first applied to their vectors of pixel intensities. The Euclidean distance of the two resulting vectors with 40 components each is then defined as the distance of the images.

Histogram of Oriented Gradients. Histogram of Oriented Gradients (HOG) [4] is a computer vision algorithm enabling the computation of feature vectors for images. First, the image is separated into cells. Second, the occurrences of gradient orientations in the cells are counted and a histogram is computed. The histograms are normalized block-wise and concatenated to obtain a feature vector. Again the Euclidean distance of these vectors is used as image distance. This approach was successfully used by Ebrahimzadeh et al. [6] for an MNIST data classifier.

Color Histogram. According to the intensities in the three color channels, the pixels are sorted into bins. For the pixels of one image, this results in a color histogram (CHIST) which can be represented as a feature vector. The Euclidean distance of these vectors is defined as the image distance.

3.2.2 Heuristics for ε

For the attack all pairwise distances $d(x_i, g_j)$ of the records x_i and samples g_j need to be computed. Samples with distances greater than ε to the training data records are ignored. Hence, an appropriate choice of ε is crucial for the success of the attack. We thus formulate two heuristics in the following.

Percentile Heuristic. The first heuristic is to use a fixed percentile of all pairwise distances $d(x_i, g_j)$ as ε. By choosing the 0.1% percentile of the distances as ε we can ensure that the corresponding samples in an ε-neighborhood are sufficiently close. Note that the MC-ε and MC-d approaches are not necessarily equivalent if this heuristic is employed.

Median Heuristic. The second heuristic avoids the need to choose an additional parameter such as the percentile value. Again, the idea is to exploit the measured distances in the Monte Carlo computation. In this approach, the median of the minimum distance to each record x_i for all the generated samples g_j is chosen:

$$\varepsilon = \underset{1 \le i \le 2M}{\text{median}} \left(\min_{1 \le j \le n} d(x_i, g_j) \right) \quad . \tag{4}$$

If ε is chosen according to the median heuristic (4) the results of MC-ε and MC-d are equivalent in both the single and set MI types as there are always exactly M records with $\hat{f}_{MC-\varepsilon}(x_i) > 0$ and $\hat{f}_{MC-d}(x_i) > 0$. A comparison of the MC attack variants is provided in the evaluation in Section 4.

3.3 Reconstruction Attack

The reconstruction attack is solely applicable to VAEs. During training, reconstructions $D(z)$ close to the current training data record x are rewarded. Hence, for training data more precise reconstructions of the VAE can be expected. However, the outputs $D(z)$ are not deterministic. They depend on the latent variable z which is sampled from the distribution $N(E_\mu(x), E_\Sigma(x))$ whose parameters are the output of the encoder network E. Hence, we repeat this process n times and set

$$\hat{f}_{\text{rec}}(x) = -\frac{1}{n} \sum_{i=1}^{n} \| D(z_i) - x \| \tag{5}$$

where z_i $(i = 1, \dots, n)$ are samples from the distribution $N(E_\mu(x), E_\Sigma(x))$. This term is frequently used in practice as part of the loss function of VAEs. One of the contributions of this work is to apply this loss to the problem of membership inference. Specifically, the function $\hat{f}_{\text{rec}}(x)$ is applied in the attack types as the discriminating function $\hat{f}(x)$. This induces the Reconstruction attack. Note that this attack considers a strong adversary $\mathcal{A}$ with access to the VAE model.

4 Evaluation

The two MI attacks formulated in this paper are evaluated in comparison to the white and black-box MI attacks of Hayes et al. [10] against generative models trained on MNIST, Fashion MNIST, and CIFAR-10 throughout Sections 4.3 to 4.7.

The *white box attack* is solely applicable to GANs and requires access to the discriminator D. Specifically, the discriminator D plays the role of the function $\hat{f}(x)$ in this attack.

The *black box attack* overcomes the limitation of the white box attack in that it requires no access to D. It is therefore not solely applicable to GANs. For the black box attack, an auxiliary GAN is trained with samples $g_1, \dots, g_n$ from the target model and the discriminator D' of this newly trained model is used in a white box manner. In experiments, the white box attack performed significantly better than the black box attack [10].

In general, our MC attacks outperformed state of the art, i.e. the white box attack of Hayes [9], for both MNIST and Fashion MNIST which are considered very hard datasets due to their simplicity. Since it is an upper bound for the accuracy, also the black box attack is outperformed. However, the MC attacks are dominated by the white box attacks on CIFAR-10. This is due to the bad sample quality which is essential if only very close samples are considered. As a consequence of the low accuracies, we decided not to compare it with the black-box attacks. In contrast, the Reconstruction attack specialized for VAEs constantly provides the highest accuracies with up to 100% single and set accuracies even for CIFAR-10.

Since several parameters have to be chosen before the attacks are applied a study of the effect of these parameters is presented in Section 4.2. Moreover, additional experiments on VAEs trained on the MNIST dataset are provided in Sections 4.4 and 4.5. These experiments are not performed for the other datasets or GANs to avoid redundancy and are solely for the purpose of evaluating the effect of regularization and training data sizes.

4.1 Setup

We evaluated the attacks of Hayes et al. [9], the Monte Carlo and the Reconstruction attacks for differing 10% subsets of the MNIST, Fashion MNIST and the CIFAR-10 dataset. The simple nature of MNIST has proven to result in low MI precision in previous work, the more complex Fashion-MNIST and CIFAR-10 datasets result in higher MI precision. Thus, the three chosen datasets represent three varying difficulties w.r.t. MI. To ensure a fair comparison we executed all experiments repeatedly and report standard deviations. Neural networks are implemented with tensorflow [1], and for the HOG and PCA computations, the python libraries scikit-image and scikit-learn [19] are used. Experiments were run on Amazon Web Services p2.xlarge (GAN) and c5.2xlarge (VAE) instances.

We first describe the datasets and models used before analyzing the parameters of the attacks.

4.1.1 MNIST

MNIST is a standard dataset in machine learning and computer vision consisting of $70,000$ labeled handwritten digits which are separated into $60,000$ training and $10,000$ test records.[2] Each digit is a 28×28 grayscale image. In all subsequent datasets only a 10% subset of the training images is used for training to provoke overfitting. The remaining 90% of the training data is used as test data to compute the accuracies of the attacks. The actual MNIST test data is only used to define the PCA transformation for the PCA based distance. This ensures that the distance is not influenced by the specific choice of the training data or the remaining 90%. Attacks are performed against two state of the art generative models, namely GANs (cf. Section 3.1.1) and VAEs (cf. Section 3.1.2). For the GAN we employ the widely used deep convolutional generative adversarial network (DCGAN) [20] architecture which aims to improve both stability and quality of GANs for image generation. This network relies on convolutional neural networks (CNN) which are state of the art for many computer vision tasks. We trained the DCGAN for 500 epochs (i.e., until convergence) with a mini batch size of 128.[3] For the VAE we apply a standard architecture[4] with 90% Dropout and a mini batch size of 128. Due to the different convergence behavior, the VAE is only trained for 300 epochs. For both models, GAN and VAE, we utilize the conditional variant s.t. we can control which digit is generated.

4.1.2 Fashion MNIST

This dataset is intended to serve as a direct drop-in replacement for MNIST [28]. Like MNIST it consists of $60,000$ training and $10,000$ test 28×28 grayscale images representing 10 fashion classes such as trousers, pullovers etc. The goal of using this dataset is to overcome the limitation of MNIST being too simple for various computer vision tasks. The same model architectures as that for MNIST are used for the conditional GAN and VAE on this dataset.

4.1.3 CIFAR-10

The CIFAR-10 dataset [13] consists of $60,000$ 32×32 color images representing 10 classes such as airplane, automobile etc. There are $50,000$ train and $10,000$ test records. Within the evaluation a GAN[5] and a VAE[6] are trained on a random 10% subset of the original dataset.

4.2 Attack Parameters

The effects of the attack parameters are analyzed in the following. Specifically, for the MC attacks the effect of the heuristic for setting ε and the number of samples n for the Monte Carlo integration are studied. We expect these to be similar for both GANs and VAEs. Hence, the analysis is restricted to the case of VAEs. For the Reconstruction attack, we study how the number of samples n for the reconstruction error estimation affects the accuracy.

[2] http://yann.lecun.com/exdb/mnist/

[3] We used https://github.com/yihui-he/GAN-MNIST as a starting point.

[4] We used https://github.com/hwalsuklee/tensorflow-mnist-VAE as a starting point.

[5] We used https://github.com/4thgen/DCGAN-CIFAR10 as a starting point.

[6] We used https://github.com/chaitanya100100/VAE-for-Image-Generation as a starting point.

4.2.1 Monte Carlo Attack

The single and set MI accuracies against VAEs trained on MNIST for different choices of ε are reported in Table 2 for $\mathcal{A}$ and $\mathcal{R}$, respectively. Note that the results of the MC-ε and MC-d attacks do not differ significantly. This suggests that the main contribution is the introduction of ε effectively ignoring samples which are further than ε away from the training records. In the case of the median heuristic, the two MC attack variants yield equivalent performances as expected. However, the median heuristic outperforms the percentile heuristic.

Besides the heuristic for ε, a sample size for the Monte Carlo approximation has to be chosen. Hence, we also analyze the performance of the MC-ε attack depending on the sample size. Again, the MC-ε attack is equivalent to the MC-d attack in the case of the median heuristic. The single and set accuracies are stated in Figure 3 for $\mathcal{A}$ and $\mathcal{R}$, respectively. In general, higher percentile values ignore fewer samples since ε is increased. A smaller sample size is required to achieve optimal accuracy for these percentiles. However, the accuracy of higher percentile values is inferior to the ones of lower percentile values.

For example, the 10% percentile attack already reaches its optimum in the minimal case of $3,000$ samples and the 1% percentile saturates at 10^4 samples. The 0.1% percentile approach is gaining higher accuracies and does not level off at 10^6 samples. It is noticeable that the median heuristic always outperforms the other heuristics. We conjecture this heuristic to level off at a higher sample size. However, in practice there is a trade-off between computational effort and accuracy of the attack. To study the effect 20 experiments for the median heuristic with 10^7 samples each are conducted, achieving a single record MI accuracy of $59.80 \pm 3.50\%$ for $\mathcal{A}$ and a set MI accuracy of $100.00 \pm 0.00\%$ for $\mathcal{R}$. In the subsequent experiments, we always use 10^6 samples for the Monte Carlo simulations.

The median heuristic is superior to the percentile heuristic for all sample sizes. Moreover, no parameter like the percentile is required. Thus, in all subsequent experiments we apply the median heuristic for which the MC-ε and MC-d attacks are equivalent. We refer to these equivalent approaches simply as *MC attack*.

4.2.2 Reconstruction Attack

We also study the effect of the sample size n to approximate the reconstruction error

$$\hat{f}_{\text{rec}}(x) = -\frac{1}{n}\sum_{i=1}^{n}\|D(z_i) - x\|. \tag{6}$$

In preliminary experiments even small sample sizes of $n = 300$ yielded good accuracies. This suggests that the estimator $\hat{f}_{\text{rec}}(x)$ is accurate enough for small n values. To ensure optimal results we conduct the subsequent experiments with $n = 10^6$ for the Reconstruction attacks against VAEs trained on MNIST and Fashion MNIST. For CIFAR we just use $n = 10^5$ samples as we already achieve accuracies of $\approx 100\%$ both in single and set MI.

4.3 Results on MNIST

Having analyzed the parameters of our proposed attacks, we now compare their accuracies with the recent white-box and black-box attacks of [10]. To stabilize the results 10 different 10% subsets of the MNIST data are chosen as training data for the GAN and VAE models. For every subset 10 single and set MI attacks are conducted with $M = 100$. While we apply the white-box attack against the GAN, we are limited to the black-box attack in case of the VAE as the latter model does not feature a discriminator. In order to test the black-box attack, a new GAN is trained with 10^6 samples from the target VAE.

For the Monte Carlo estimator $\hat{f}_{MC}$ we use the PCA and HOG based distances introduced in Section 3.2.1. The CHIST distance is not applicable since MNIST solely consists of grayscale images. As described in the previous section we use $n = 10^6$ samples and the median heuristic. The resulting accuracies are depicted in Figure 5. The dotted horizontal baseline at 50% is the average success rate of random guessing. In general, the accuracies of single MI for $\mathcal{A}$ are significantly lower than those of set MI for $\mathcal{R}$. Furthermore, all attacks are much more successful if applied against VAEs instead of GANs. This suggests that in general there is less overfitting in GANs. This observation is consistent with the Annealed Importance Sampling measurements by Wu et al. [27].

The black-box and white-box attack do not perform significantly better than the baseline in both experiments. The MC attack clearly outperforms these attacks in the experiments. When used with PCA distance our MC attack can even infer set membership with nearly

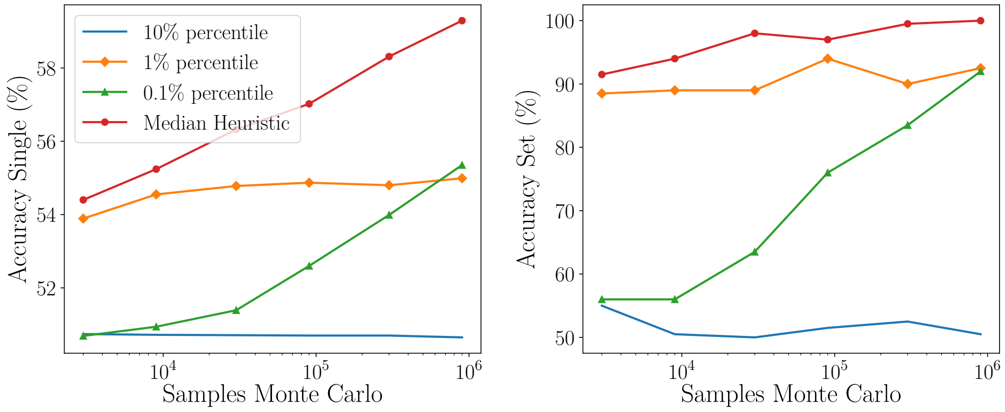
Table 2. Set accuracies for $\mathcal{R}$ depending on ε values

(a) HOG-based distance

Heuristic/Percentile	HOG-based distance			
	GAN Monte Carlo-d	GAN Monte Carlo-ε	VAE Monte Carlo-d	VAE Monte Carlo-ε
Median	63.76±3.83	63.76±3.83	83.50±2.43	83.50±2.43
0.01%	63.76±3.68	66.11±3.70	81.00±2.59	82.25±2.50
0.10%	63.76±3.71	62.08±3.65	74.50±2.90	71.75±2.98
1.00%	60.07±3.84	59.73±3.86	59.50±3.24	54.00±3.29

(b) PCA-based distance

Heuristic/Percentile	PCA-based distance			
	GAN Monte Carlo-d	GAN Monte Carlo-ε	VAE Monte Carlo-d	VAE Monte Carlo-ε
Median	74.84±3.25	74.84±3.25	99.75±0.25	99.75±0.25
0.01%	74.84±3.31	71.94±3.40	95.50±1.34	91.75±1.80
0.10%	64.84±3.69	59.68±3.78	94.75±1.52	95.50±1.43
1.00%	47.42±3.77	51.61±3.76	60.75±3.21	58.50±3.29



(a) Adversarial actor: Single MI

(b) Regulartory actor: Set MI

Fig. 3. MC attack accuracy (differing scales) on MNIST with PCA based distance against VAEs depending on sample size.

100% accuracy against a VAE. For the GAN the accuracy is still about 75%. In general, accuracies are inferior if the HOG distance is used. As a side fact, the Monte Carlo based attacks with PCA distance take ≈ 7 minutes each on a p2.xlarge instance on AWS. Currently, at the cost of 0.90 US $ per hour, the attacks only cause minor costs. The specialized Reconstruction attack is superior to the MC attack in the case of the VAE yielding ≈ 70% and 100% in the single and set MI attack, respectively. The high accuracies of the attacks we proposed make them especially attractive for the regulatory use case depicted in Section 2.2.

4.4 Effect of Subset Choice

It is unclear how the specific choice of the MNIST 10% subset influences the accuracy of the MC attack. In Figure 4 the average MC attack performance with PCA distance against VAEs trained on different subsets are plotted. Attack performances seem independent of the specific subset. We also conduct an F-test to evaluate whether the single accuracy means of the four VAEs are different at 10^6 samples resulting in a p-value ≈ 0.64. Hence, the hypothesis that the means are equal can be accepted with high probability, i.e. the choice of the subset does not significantly influence the attack results. We conclude that the accuracy depends on the size of the training data rather than its specific members.

Table 3. Accuracies depending on MNIST training data size

Size	Monte Carlo (PCA dist.)		Reconstruction attack	
	Single	Set	Single	Set
40%	50.79±0.27	57.50±3.24	57.35±0.37	98.50±1.11
20%	57.05±0.32	94.75±1.39	62.23±0.38	100.00±0.00
10%	59.93±0.26	99.75±0.25	70.09±0.37	100.00±0.00

Table 4. Accuracies depending on MNIST Dropout Keep Rates

Rate	Monte Carlo (PCA dist.)		Reconstruction attack	
	Single	Set	Single	Set
50%	51.45±0.26	64.75±3.19	53.77±0.34	86.00±3.18
70%	53.17±0.29	78.50±2.71	58.31±0.40	97.00±1.56
90%	59.93±0.26	99.75±0.25	70.09±0.37	100.00±0.00

We remark that in the experiment setups $M = 100$ samples of the 10% subset of the training data and 100 samples of the remaining 90% training data are chosen. The set MI experiments yield high accuracies. Therefore, if a regulator suspects that some dataset was used for training a model this can be recognized with the novel attacks even though other data might have been part of the training data as well. This is an analogous case to the experiment described. Though of course more training data was used, we focus on 100 samples. It is very likely that the inappropriately used data is not the only data used to train the model. Hence, the practicability of the MC attack is increased since the regulator does not need to know all the training data to prove that a certain subset was used.

4.5 Effect of Training Data Size and Regularization — Mitigations

We also investigate how the size of the training dataset influences the success of the attacks for the MNIST dataset. For this, five VAEs are trained with 20 experiments each since the effect should be similar for GANs. The results for the MC attack and Reconstruction attack are depicted in Table 3. When using 40% of the training data instead of the usual 10% the accuracy shrinks from 60% to 51% for single MI and from nearly 100% to only about 58% for set MI in the case of the MC attack. As expected, for 20% the effects are less significant. Clearly, more training data would further reduce the effectiveness of the attacks. However, in the case of the Reconstruction attack, the effects are less significant. Even if 40% are used the set accuracy is still about 100% meaning that the Reconstruction attack is more robust.

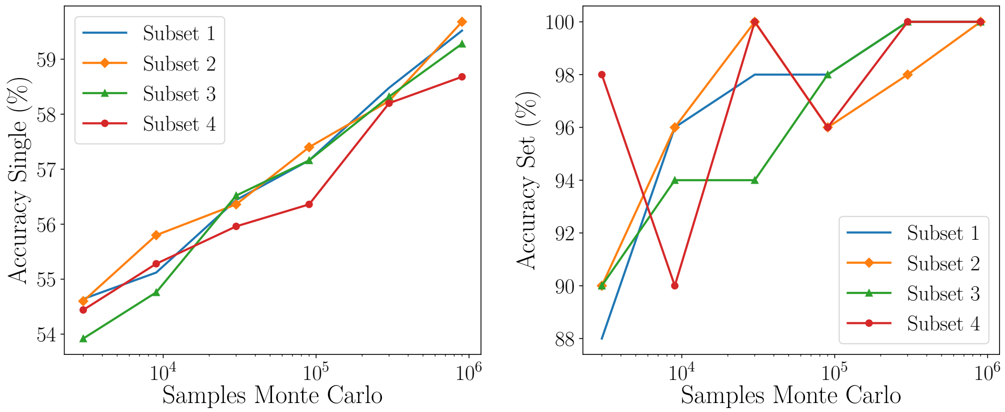
In general, the performance declines suggest that generative models make use of the additional information provided by additional training data. Similar effects were observed before in the case of the white-box attack [10]. However, often in practice the amount of training data is a bottleneck for training generative models. In consequence, one could use regularization methods to improve the generalization such as *dropout* [23]. In the case of dropout, certain neurons are switched off during training with given probability to increase the resistance of the network. In the standard case we already use dropout with a keep probability of 90% both in the encoder and decoder of the VAE. We also conduct experiments for the MC and Reconstruction attack at lower keep rates of 70% and 50%. The accuracy in the set MI type decreases to 79% at a keep probability of 70% and to 65% at an even reduced keep probability of 50% for the MC attack. Again, the effects are less significant for the Reconstruction attack still yielding $\approx 86\%$ set MI accuracy for a 50% keep rate. Detailed results are reported in Table 4. The results indicate that dropout can indeed be used in practice to mitigate the proposed MI attacks. This can also be observed in the case of the white-box attack [10]. However, a lower keep probability also causes the generated images to get increasingly blurry (cf. Appendix, Figure 6). Hence, there is an inherent trade-off between high image quality and low MI attack accuracies.

4.6 Results on Fashion MNIST

Samples of the trained VAE and GAN models are provided in Figure 7 (Appendix). They show that the GAN produces more detailed samples compared to the VAE.

To stabilize our results we train five GANs and VAEs on different 10% subsets of the dataset. For each model 20 single record MI and set MI experiments are conducted. We do not evaluate the black-box attack for the VAE as it performed significantly worse than the MC attack and Reconstruction attack in the previous MNIST experiments. The white-box attack is not applicable since VAEs do not provide a discriminator D. Figure 5 provides an overview of the results.

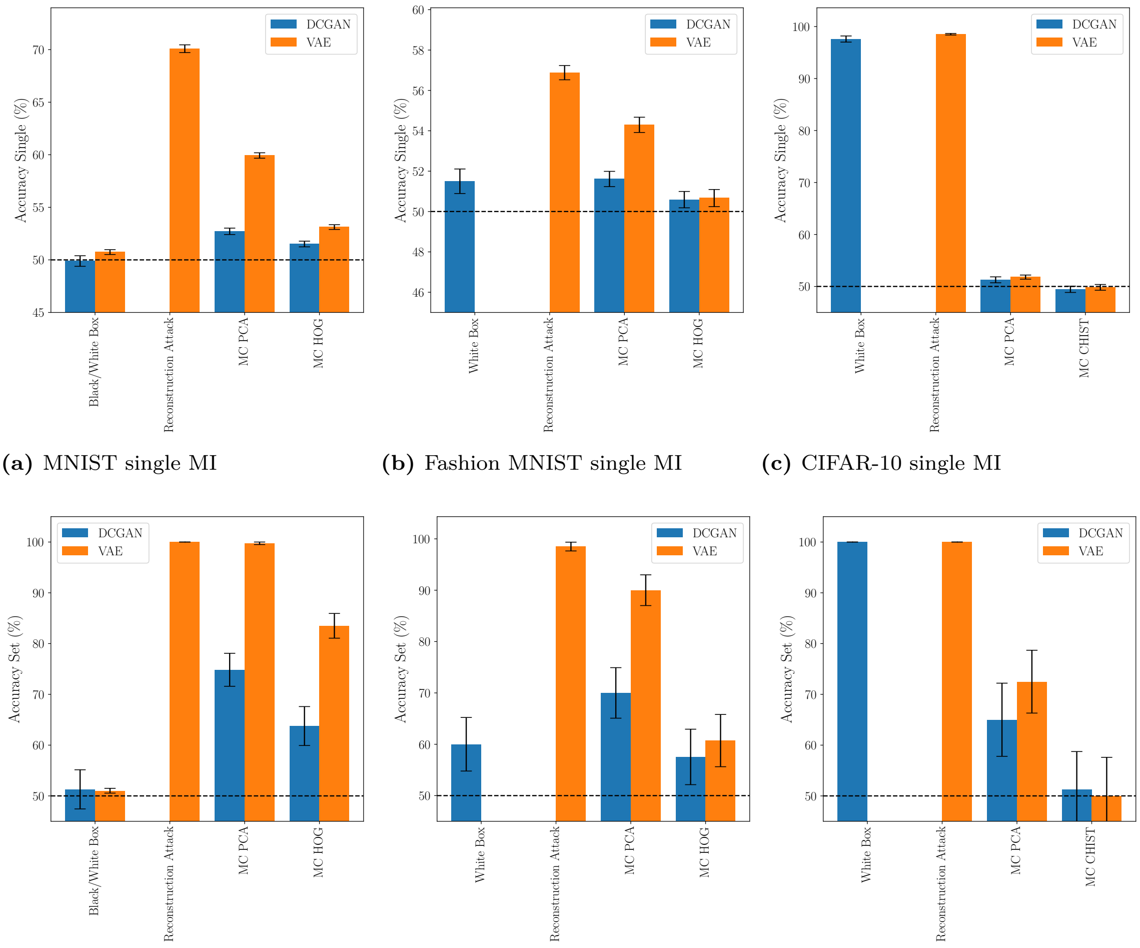
Compared to MNIST, the MC attack performs slightly worse on this dataset. As before, the attacks are more successful in against the VAE providing additional evidence that GANs generalize better. This surprises because the samples created by the GAN are more detailed. The white-box attack performs better with this



(a) Adversarial actor: Single Membership Inference

(b) Regulatory actor: Set Membership Inference

Fig. 4. MC attack accuracy (differing scales) on MNIST with PCA distance depending on sample size for four different training subsets.



(a) MNIST single MI

(b) Fashion MNIST single MI

(c) CIFAR-10 single MI

(d) MNIST set MI

(e) Fashion MNIST set MI

(f) CIFAR-10 set MI

Fig. 5. Average attack accuracy (differing scales) for single and set MI on the datasets.

dataset achieving about 60% accuracy for set MI against GANs. However, it is still inferior to the proposed MC attacks with PCA distance (70% accuracy). Again, our reconstruction attack significantly outperforms all other attacks in the case of the VAE yielding $\approx$ 57% and $\approx$ 99% in the single and set case.

4.7 Results on CIFAR-10

Samples of the models after training are provided in Figure 8. Though state of the art models are applied, they do not succeed in learning the data effectively as the samples are very blurry and real objects cannot be identified. This is similar to Hayes et al. [10]. Hence we expect the MC attacks to perform worse on these datasets due to their reliance on samples which are very close to the training data. However, when the overall quality is bad we do not expect individual samples to replicate the training data.

MC distances are calculated by the known PCA based distance with 120 components. Moreover, we examine the CHIST distance (cf. Section 3.2.1) instead of the HOG distance for two reasons. First, the images are very blurry so it is very unlikely that oriented gradients yield a good distance. Second, it is now possible to employ the CHIST distance as it relies on colors and could potentially be less affected by blurry images.

Contrary to the 100 experiments for MNIST and Fashion MNIST, 40 experiments were sufficient for significant results for CIFAR-10. The results of the white-box attack and the novel MC and Reconstruction attacks are depicted in Table 6. Figure 5 provides an overview of the results. The MC attack with CHIST distance is not significantly better than random guessing. If the PCA based distance is employed the accuracy increases to roughly 51% and 52% for single MI and 65% and 73% set MI against the GAN and VAE, respectively. Again, the choice of the distance metric d is crucial. Surprisingly the attack exhibits an accuracy better than random guessing despite the bad sample quality. However, unlike the MNIST and Fashion MNIST datasets, the white-box attack outperforms the MC attack for the GAN trained on CIFAR-10. This is most likely due to the bad sample quality of the generator.

The white-box attack achieves an accuracy of nearly 100% in single record MI as well as set MI. This implicates that despite the bad sample quality the discriminator effectively remembers the training data. A similar accuracy can be observed for the reconstruction attack in the case of the VAE. This suggests that the reconstruction attack we propose is an effective means of assessing VAEs as it constantly outperformed all other attacks. Note that for GANs the white-box attack cannot play this role as it performs worse than the novel MC attacks on MNIST and Fashion MNIST.

5 Related Work

The range of attacks against neural networks and their applications is wide and various approaches have been contributed. We now review the prior work and relate it to our findings.

In the case of adversarial examples, input data is systematically manipulated to disturb inference as formulated by Huang et al. [11]. In the case of adversarial training, sample data is poisoned, e.g., to introduce stealthy features which may be exploited later on [16, 29]. Common to these examples is an attacker who actively influences the result of either learning or inference of a model.

In contrast, this work considers an honest-but-curious adversary having access to an already trained model, or at least to samples from a generative model. This adversary infers knowledge about the training data records. Previous work in this setup follows two main directions: Model inversion attacks as formulated by Fredrikson et al. [7] and Tramer et al. [26] try to directly reconstruct training data based on the output of a model to which the attacker has black-box access. Instances of this approach can make use of a confidence score for the output in a discriminative model [7].

Our approach follows the other main direction of data leakage attacks: membership inference. The goal of this attack is to identify the data used to train the model. Shokri et al. [21] apply such attacks against discriminative networks. We focus on generative models similar to Hayes et al. [10], and also evaluate our attacks in comparison to their white- and black-box attacks. The white-box attack, where the discriminator of the trained model must be accessible, is restricted to GANs. The black box attack solely requires access to samples from the model. We further structure the class of membership inference attacks by assuming two different types of actors: an honest-but-curious adversary $\mathcal{A}$ performing single MI, and a regulatory actor $\mathcal{R}$ performing set MI. The first attack type has already been used in previous work to evaluate attacks against generative models [10]. In parallel to our work, Liu et al. [14] came up with an approach for the application of MI to a set

Table 5. Accuracy of the white-box, Reconstruction and MC Attacks on Fashion MNIST for single record MI and set MI.

(a) Single MI

Model	Accuracy Single (%)			
	White-box attack	Reconstruction attack	Monte Carlo PCA distance	Monte Carlo HOG distance
GAN	**51.50±0.61**	**not applicable**	**51.61±0.38**	**50.59±0.41**
VAE	**not applicable**	**56.88±0.35**	**54.29±0.38**	**50.67±0.43**

(b) Set MI

Model	Accuracy Set (%)			
	White-box attack	Reconstruction attack	Monte Carlo PCA distance	Monte Carlo HOG distance
GAN	**60.00±5.22**	**not applicable**	**70.00±4.92**	**57.53±5.40**
VAE	**not applicable**	**98.50±0.86**	**90.00±2.99**	**60.71±5.09**

Table 6. CIFAR-10 accuracy for MC and White-box attack

(a) Accuracy of single MI

Model	Accuracy Single (%)			
	White-box attack	Reconstruction attack	Monte Carlo PCA distance	Monte Carlo CHIST distance
GAN	**97.60 ± 0.59**	**not applicable**	**51.28 ± 0.57**	**49.45 ± 0.60**
VAE	**not applicable**	**98.52 ± 0.15**	**51.80 ± 0.40**	**49.83 ± 0.55**

(b) Accuracy of set MI

Model	Accuracy Set (%)			
	White-box attack	Reconstruction attack	Monte Carlo PCA distance	Monte Carlo CHIST distance
GAN	**100.00 ± 0.00**	**not applicable**	**65.00 ± 7.21**	**51.25 ± 7.49**
VAE	**not applicable**	**100.00 ± 0.00**	**72.50 ± 6.19**	**50.00 ± 7.60**

of samples simultaneously. Their approach is to train a network that acts as an inverse for the generator. They then measure the (L2-)distance of the generator applied to the thus calculated preimage of a sample to the sample itself. The decision to classify a sample as training data is based on a threshold applied to this distance. In their co-membership inference attack they simultaneously train and evaluate the network A on multiple samples (either all training data or all test data). Hence their decision function implicitly changes for different input data. However, our set membership inference provides a framework where a discriminating function f (which is fixed per attack) is evaluated by $\mathcal{R}$ on a the members of two sets of samples (from training resp. test data) in order to amplify subtle differences in the values of f and to compensate for outliers.

Part of our work can be seen as a generalization of previous approaches to evaluate generative models. According to Theis et al. [25] the choice of metrics may have a strong influence on the result of such model evaluations. Specifically, the use of KDE is problematic since the error may be large. Hence, Theis et al. [25] suggest not to use KDE for the evaluation of generative models. A key difference of our MC attack in comparison to KDE is that it only considers samples very close to the training data. Arora et al. [2] recently evaluated GANs by analyzing near duplicate samples of GANs with the Birthday paradox. Their results lead to the similar conclusion that close samples are of high interest to assess the model quality.

Model quality is related to overfitting. Yeom et al. [30] study the relationship between overfitting and the success of both membership inference and model inversion attacks and quantify the advantage of them. Opposed to our work, their analysis considers discriminative models. We could empirically show a similar effect for generative models. Overfitting increased the accuracy of all examined attacks. This aligns with the results of Hayes et al. [10] for their white-box attack.

We use histograms of oriented gradients (HOG) [4], color histograms and PCA to quantify distances between images. A different approach would be an algorithm built upon local key point descriptors such as the scale-invariant feature transform (SIFT) algorithm [15]. In preliminary experiments, SIFT yielded lower accuracies while being less efficient to compute. Hence, it is not considered in our evaluation section.

6 Conclusion

We suggest two membership inference attacks for generative models: the *Monte Carlo (MC) attack* and the *Reconstruction attack*. While the first is applicable to all generative models the latter is specialized for VAEs. Both attacks significantly outperform state of the art attacks against generative models often yielding accuracies close to 100%. In particular, the Reconstruction attack against VAEs outperformed all other attacks on all datasets. For CIFAR-10 the single and set MI even reached $\approx$ 100%. Even with dropout or more training data, the accuracies have proven robust.

On datasets with very good sample quality the MC attack outperformed state of the art. This supports the use of our formulated attacks to evaluate both overfitting and information leakage of generative models. On a dataset with very poor sample quality, however, the white box-attack [9] outperformed our approaches. This is not very surprising as the MC attacks rely on a replication of training data characteristics which cannot be observed if the sample quality is insufficient.

In general, we observed in this work that VAEs are more vulnerable to the MI attacks. This suggests that VAEs are more prone to overfitting than GANs if the same amount of training data is available. Hence, the novel MI attacks formulated within this work give insights into the performance of different generative models and regularization techniques. In particular, the use of GANs being less vulnerable while producing detailed samples is motivated.

7 Acknowledgements

We thank the anonymous reviewers and our shepherd, Shruti Tople, for critically reading this paper and suggesting numerous improvements. This work has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 825333 (MOSAICROWN).

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In *Proc. of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI)*, pages 265–283, Berkeley, CA, USA, 2016. USENIX Assoc.

[2] S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (gans). In *International Conference on Machine Learning*, pages 224–232, 2017.

[3] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Jozefowicz, and S. Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893, Piscataway, NJ, USA, 2005. IEEE.

[5] C. Donahue, J. McAuley, and M. Puckette. Synthesizing audio with generative adversarial networks. *arXiv preprint arXiv:1802.04208*, 2018.

[6] R. Ebrahimzadeh and M. Jampour. Efficient handwritten digit recognition based on histogram of oriented gradients and svm. *International Journal of Computer Applications*, 104(9), 2014.

[7] M. Fredrikson, S. Jha, and T. Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1322–1333, New York, NY, USA, 2015. ACM.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Proc. of Advances in Neural Information Processing Systems 27 (NIPS)*, pages 2672–2680. NIPS Foundation, 2014.

[9] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. Logan: Evaluating privacy leakage of generative models using generative adversarial networks. *arXiv preprint arXiv:1705.07663*, 2017.

[10] J. Hayes, L. Melis, G. Danezis, and E. De Cristofaro. LOGAN: Membership Inference Attacks Against Generative Models. *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2019(1), 2019.

[11] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, and J. D. Tygar. Adversarial machine learning. In *AISec*, 2011.

[12] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.

[14] K. S. Liu, B. Li, and J. Gao. Generative model: Membership attack, generalization and diversity. *CoRR*, abs/1805.09898, 2018.

[15] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages

1150–1157. Ieee, 1999.

[16] M. Mozaffari-Kermani, S. Sur-Kolay, A. Raghunathan, and N. K. Jha. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE journal of biomedical and health informatics*, 19(6):1893–1905, 2015.

[17] A. B. Owen. *Monte Carlo theory, methods and examples*. 2013.

[18] E. Parzen. On estimation of a probability density function and mode. *The annals of mathematical statistics*, 33(3):1065–1076, 1962.

[19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[20] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[21] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *Proc. of the 2017 IEEE Symposium on Security and Privacy (S&P)*, pages 3–18, Piscataway, NJ, USA, 2017. IEEE.

[22] Sky News. The guardian view on google's nhs grab: legally inappropriate, 2017.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[24] The Guardian Online. The guardian view on google's nhs grab: legally inappropriate, 2017.

[25] L. Theis, A. van den Oord, and M. Bethge. A note on the evaluation of generative models. In *Proc. of the 4th International Conference on Learning Representations (ICLR)*, 2016.

[26] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart. Stealing machine learning models via prediction apis. In *Proc. of the 2016 USENIX Security Symposium*, pages 601–618, Berkeley, CA, USA, 2016. USENIX Assoc.

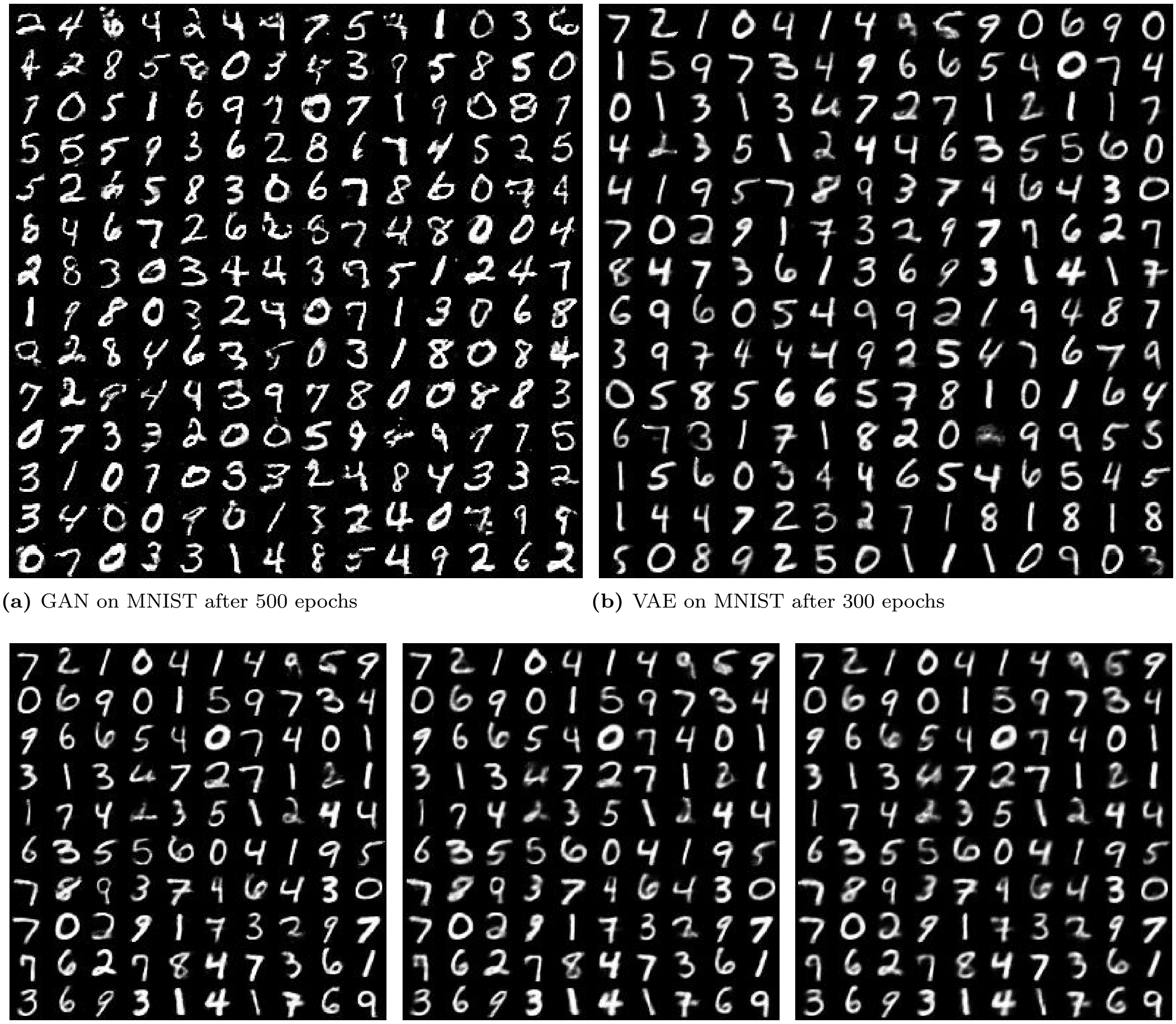
[27] Y. Wu, Y. Burda, R. Salakhutdinov, and R. Grosse. On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*, 2016.

[28] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[29] C. Yang, Q. Wu, H. Li, and Y. Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.

[30] S. Yeom, M. Fredrikson, and S. Jha. The unintended consequences of overfitting: Training data inference attacks. *arXiv preprint arXiv:1709.01604*, 2017.

Appendix: Additional Figures



(a) GAN on MNIST after 500 epochs

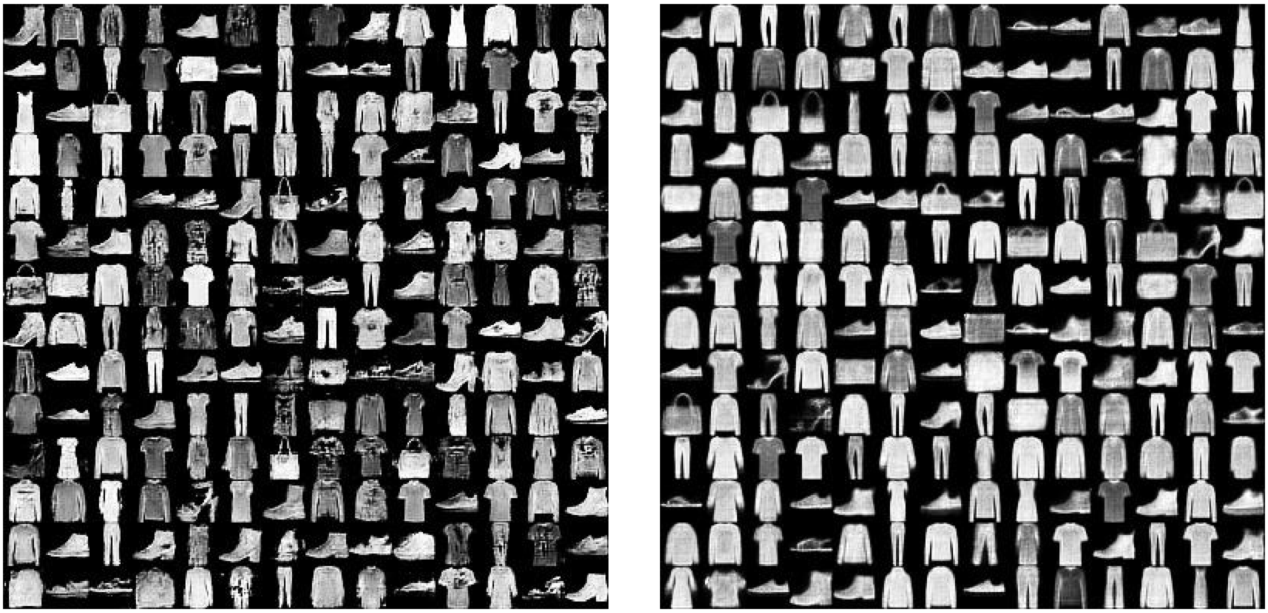
(b) VAE on MNIST after 300 epochs

(c) VAE with 90% Keep Probability

(d) VAE with 70% Keep Probability

(e) VAE with 50% Keep Probability

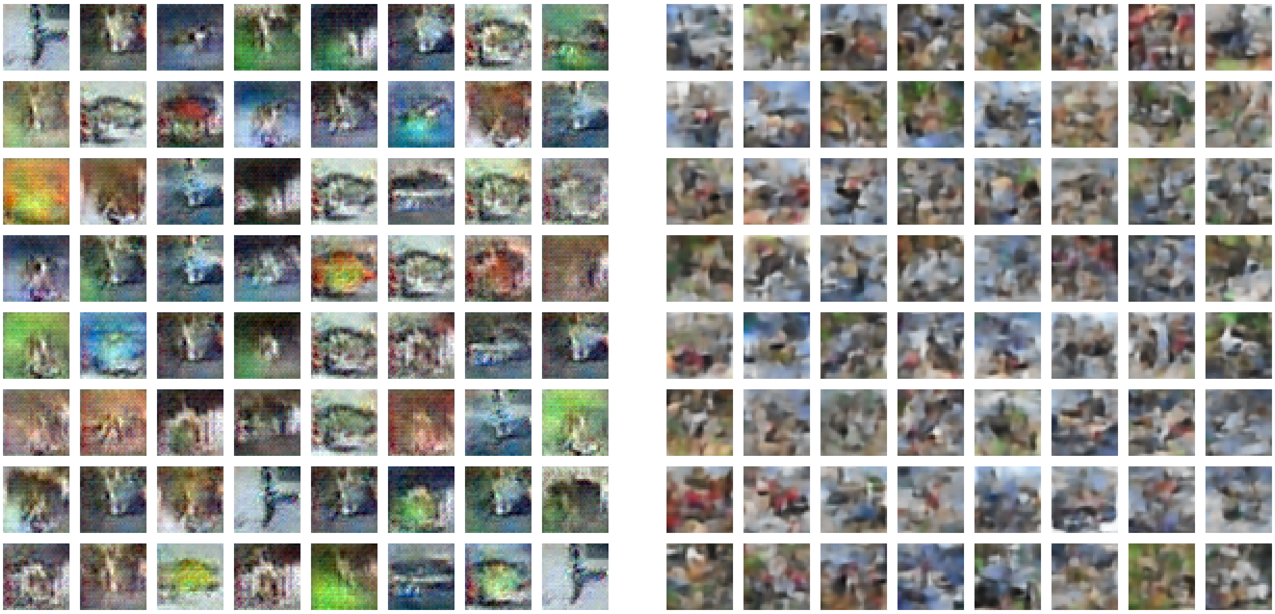
Fig. 6. Generated samples of the trained models.



(a) GAN on Fashion MNIST after 500 epochs

(b) VAE on Fashion MNIST after 300 epochs

Fig. 7. Generated samples of the trained models.



(a) GAN after 1000 epochs

(b) VAE after 1200 epochs

Fig. 8. Generated images of a GAN and a VAE after training on CIFAR-10 dataset.