Membership Inference Attacks on Diffusion Models via Quantile Regression

Shuai Tang [*1] **Zhiwei Steven Wu** [*1 2] **Sergul Aydore** [1] **Michael Kearns** [1 3] **Aaron Roth** [1 3]

Abstract

Recently, diffusion models have become popular tools for image synthesis due to their high-quality outputs. However, like other large models, they may leak private information about their training data. Here, we demonstrate a privacy vulnerability of diffusion models through a *membership inference (MI) attack*, which aims to identify whether a target example belongs to the training set when given the trained diffusion model. Our proposed MI attack learns quantile regression models that predict (a quantile of) the distribution of reconstruction loss on examples not used in training. This allows us to define a granular hypothesis test for determining the membership of a point in the training set, based on thresholding the reconstruction loss of that point using a custom threshold tailored to the example. We also provide a simple bootstrap technique that takes a majority membership prediction over "a bag of weak attackers" which improves the accuracy over individual quantile regression models. We show that our attack outperforms the prior state-of-the-art attack while being substantially less computationally expensive — prior attacks required training multiple "shadow models" with the same architecture as the model under attack, whereas our attack requires training only much smaller models.

1. Introduction

Diffusion models, based on generative neural networks, have gained attention in the field of image generation (Ho et al., 2020; Song & Ermon, 2019). It has been shown that diffusion models are remarkably capable of generating images that are higher-quality than previous approaches such as GANs and VAEs, while also being more scalable. However, as the size of these models has grown drastically over the last decade, so has the privacy concern that these large-scale diffusion models may reveal sensitive information about the dataset they are trained on.

One of the most popular classes of methods to evaluate the privacy risks of machine learning (ML) models is *membership inference (MI) attacks* (e.g., (Homer et al., 2008; Shokri et al., 2017; Jayaraman & Evans, 2019; Jagielski et al., 2020; Nasr et al., 2021; Carlini et al., 2022)), in which an attacker aims to determine if a target example belongs to the training dataset given the trained model. The success of an MI attack falsifies privacy protections on the existence of any individual, i.e. differential privacy guarantees (Dwork & Roth, 2014). MI can also be disclosive if e.g. membership in the training data is determined based on a sensitive attribute (as it would if e.g. the dataset consisted of medical records for patients with a particular disease). In addition, MI attacks can be a building block for other more sophisticated attacks such as *extraction attacks* on generative models (Carlini et al., 2023). In general, a successful MI attack with reasonable side information is a strong indicator of privacy vulnerability. Finally, when applied to differentially private algorithms (Dwork et al., 2006), MI attacks can serve as privacy auditing tools by providing lower bounds on the privacy parameters, which in turn assess the tightness of the privacy analyses (Jagielski et al., 2020; Nasr et al., 2023) and help identify potential errors in the privacy proof or implementation (Tramèr et al., 2022; Stadler et al., 2022).

A majority of the existing MI attacks focus on supervised learning (Yeom et al., 2018; Wang et al., 2019; Jayaraman et al., 2020; Nasr et al., 2021; Shokri et al., 2017; Carlini et al., 2022), and there has been significantly less development on MI attacks against generative models (e.g., (Hayes et al., 2019; van Breugel et al., 2023; Carlini et al., 2023)). The goal of our work is to develop strong MI attacks against state-of-the-art diffusion models.

Our work extends the quantile-regression-based attacks in (Bertran et al., 2023) for supervised learning to attacks for diffusion models. For a given trained diffusion model parameterized by θ, our attack first learns a quantile regression model on public auxiliary data that predicts the α-quantile $q_\alpha(z)$ of θ's reconstruction loss on each example z (formally defined in Definition 2.1). Then we indicate an example is a member of the training set if its reconstruction loss is

[*]Equal contribution [1]Amazon AWS AI/ML [2]Carnegie Mellon University [3]University of Pennsylvania. Correspondence to: Shuai Tang <shuat@amazon.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

lower than its predicted α-quantile. By design, the attack has a false positive rate of α: that is the probability that it incorrectly declares a randomly selected point z that was not used in training to have been used in training is α. We further boost the attack performance of our approach through bagging aggregation over small quantile regression models. We evaluate our attack on diffusion models trained on image datasets, and demonstrate four major advantages:

I. Our quantile-regression-based attack obtains state-of-the-art accuracy on several popular vision datasets. Even though our attacks leverage the same reconstruction loss function considered in (Duan et al., 2023), their attack leverages the same *marginal approach* in (Yeom et al., 2018) that applies a uniform threshold (that is, the α-quantile on the marginal distribution over the reconstruction loss) across all examples. In comparison, our attack is *conditional* since it applies a finer-grained per-example threshold when performing membership inference.

II. Compared to the prior state-of-the-art MI attacks against diffusion models (Pang et al., 2023) also in the white-box setting, we achieve higher accuracy without suffering their computational cost. Similar to the Likelihood Ratio Attack (LiRA) attack proposed by (Carlini et al., 2022), the Gradient-Subsample-Aggregate (GSA) attack in (Pang et al., 2023) requires training multiple *shadow models*, each of which is a diffusion model trained on a randomly drawn dataset. While the accuracy of the MI attack improves as the number of shadow models grows, their approach also becomes computationally prohibitive. However, our approach only requires learning *tiny* quantile regression models.

III. Since our attack does not rely on shadow models, it also requires significantly fewer details about the training algorithm of the diffusion model under attack, such as hyperparameters and network architecture used in training. Our attack is effective even though the quantile regression model has significantly fewer parameters than the attacked diffusion models.

IV. While both our work and the prior work of (Bertran et al., 2023) rely on quantile regression for MI attacks, an important distinction in ours is the use of bootstrap aggregation — *bagging* — that takes an ensemble of tiny quantile regression models, namely a "bag of weak attackers." Bagging generally improves the attack performance by reducing the variance of the individual models, each of which can be viewed as a weak hypothesis test. The use of bagging immediately enriches the space of MI attacks. Specifically, we use bagging techniques in conjunction with small quantile regression models, which leads to a substantial improvement in accuracy by introducing little computational overhead (due to the small model size). In comparison, the MI attack in (Bertran et al., 2023) primarily leverages a parametric approach to fit a single quantile regression model, and it falls short in performance compared to the shadow models approach (Carlini et al., 2022) on datasets such as CIFAR10.

2. Background and Preliminaries

We here present the objective of a Membership Inference (MI) attack along with required side information, and briefly introduce diffusion models for context.

2.1. Membership inference attacks

Membership inference (MI) is a privacy attack that attempts to predict whether a given example was used to train a machine learning model (e.g., (Homer et al., 2008; Yeom et al., 2018; Shokri et al., 2017; Jagielski et al., 2020; Jayaraman et al., 2020; Nasr et al., 2021; Carlini et al., 2022)). Our work focuses on performing MI attacks on diffusion models.

Problem statement. Given a training dataset $\mathbf{Z}$ drawn from an underlying distribution P, a diffusion model θ is trained on $\mathbf{Z}$. The goal of a membership inference attack is to infer whether a target example z^* is in the training set $\mathbf{Z}$ or not.

Adversary's side information. Similar to almost all prior work on membership inference (Carlini et al., 2022; Duan et al., 2023; Bertran et al., 2023; Shokri et al., 2017), we assume the adversary has access to some public data drawn from P. In the standard terminology of MI, there are two types of access the attacker might have. In a *black-box* attack, the adversary only has access to the generated synthetic data. In a *white-box attack*, the adversary has access to the generative model G and possibly information about its training. We give a white-box attack that requires knowledge of the model parameters, but *not* any information about the training procedure.

2.2. Diffusion Models

We briefly introduce diffusion models at a high level, following the notation of (Ho et al., 2020). For any image, a diffusion model provides a stochastic path from the image to noise. A diffusion model consists of two processes: 1) a T-step diffusion process (denoted as q below) that iteratively adds Gaussian noise to an image, and 2) a denoising process (denoted as p_θ below) that gradually reconstructs the image from noise. Let z_0 be the real image without noise and z_T be the noisy image with the largest amount of noise. The transitions of diffusion and denoising are described as:

$$q(z_t|z_{t-1}) = \mathcal{N}(z_t; \sqrt{1-\beta_t}z_{t-1}, \beta_t I) \quad (1)$$

$$p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \mu_\theta(z_t, t), \Sigma_\theta(z_t, t)), \quad (2)$$

where $q_{z_t|z_{t-1}}$ is the probability distribution of the diffused image z_t given the previous image z_{t-1}, $p_\theta(z_{t-1}|z_t)$ is the probability distribution of the denoised image z_{t-1} given the noisy image z_t, $\mu_\theta(\cdot)$ and $\Sigma_\theta(\cdot)$ are the mean and covariance

of the denoised image, respectively, as parameterized by the model parameters θ , and β_t is a noise schedule that controls the amount of noise added at each step. Moreover, the marginal distribution at any time step t given the example z_0 can be written as

$$q(z_t \mid z_0) = \mathcal{N}(z_t; \sqrt{\overline{\alpha}_t} z_0, (1 - \overline{\alpha}_t) I), \tag{3}$$

where $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. We will work with the following re-parameterization of μ_θ with

$$\mu_\theta(z_t, t) = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{\beta_t}{1 - \overline{\alpha}_t} \epsilon_\theta(z_t, t) \right) \tag{4}$$

where ϵ_θ is a predictor (given by θ) that predicts the noise component given z_t.

Loss function. Many MI attacks proceed by identifying a loss function and making membership inferences by comparing the loss on the target example with a threshold. Intuitively, if the loss is unusually low, then there is evidence that the example was part of the training set. For supervised learning models, MI attacks typically leverage the classification loss (e.g., the cross-entropy loss). For diffusion models, existing work has proposed candidates of loss functions that measure the reconstruction error at different time steps of the diffusion process (Carlini et al., 2020; Duan et al., 2023). We leveraged the t-error function defined in (Duan et al., 2023), which has the compelling advantage that it is deterministic and avoids repeated sampling from the diffusion process. Consider the following deterministic approximation of the diffusion and denoising processes:

$$\begin{aligned} z_{t+1} &= \phi_\theta(z_t, t) \\ &= \sqrt{\overline{\alpha}_{t+1}} f_\theta(z_t, t) + \sqrt{1 - \overline{\alpha}_{t+1}} \epsilon_\theta(z_t, t) \quad (5) \\ z_{t-1} &= \psi_\theta(z_t, t) \\ &= \sqrt{\overline{\alpha}_{t-1}} f_\theta(z_t, t) + \sqrt{1 - \overline{\alpha}_{t-1}} \epsilon_\theta(z_t, t) \quad (6) \end{aligned}$$

where $f_\theta(z_t, t) = \frac{z_t - \sqrt{1-\overline{\alpha}_t}\epsilon_\theta(z_t, t)}{\sqrt{\overline{\alpha}_t}}$ is the estimate of z_0 given the z_t and the prediction $\epsilon_\theta(z_t, t)$. Then we could also define the deterministic reverse result as

$$\Phi_\theta(z_0, t) = \phi_\theta(\cdots \phi_\theta(\phi_\theta(z_0, 0), 1) \ldots, t-1) \tag{7}$$

Now we can define the reconstruction loss function, termed as t-error, that is used in our MI attack.

Definition 2.1. (t-error) For a given sample z_0 and the deterministic reverse result $\tilde{z}_t = \Phi_\theta(z_0, t)$ at time step t, the *t-error* is the approximated posterior estimation error at step t:

$$\hat{\ell}_t(\theta, z_0) = ||\psi_\theta(\phi_\theta(\tilde{z}_t, t), t) - \tilde{z}_t||^2. \tag{8}$$

Intuitively, the t-error function measures how much we change $\tilde{z}_t$ if we take one step in the deterministic diffusion process ϕ_θ and then rewind back with one step of deterministic denoising ψ_θ. While this loss function is not what the training algorithm optimizes, it provides a deterministic approximation to the loss function during training (Duan et al., 2023; Ho et al., 2020). Thus, smaller t-error values provide evidence that z_0 was used to train the diffusion model θ.

Algorithm 1 Quantile Regression MI attacks for Diffusion Model

Inputs: A set of auxiliary examples D drawn from P, target example z^*, trained diffusion model θ, a choice of t for t-error function. Target false-positive rate α.
for each $z \in D$ **do**
  evaluate the score $\hat{\ell}_t(\theta, z)$
end for
Learn a quantile regression model q_α such that $q_\alpha(z)$ predicts the α-quantile of $\hat{\ell}_t(\theta, z)$ conditioned on z.
Return "IN" if $\hat{\ell}_t(\theta, z^*) \leq q_\alpha(z^*)$, otherwise "NO"

3. MI Attacks with Quantile Regression

Under the setting in Section 2.1, we assume that the attacker has access to a set of public examples D drawn from the underlying distribution P. Given the public dataset D, a choice of t for the t-error function, and the trained diffusion model θ, the attacker learns a quantile regressor q_α such that $q_\alpha(z)$ predicts the α-quantile of the t-error $\hat{\ell}_t(\theta, z)$ for each example z in D, where α is a parameter that controls the false-positive rate. Then on any target example z^*, the attacker declares the example is a member of the training set if and only if the t-error $\hat{\ell}_t(\theta, z^*) \leq q_\alpha(z^*)$. The formal description of the algorithm is in Algorithm 1.

By design, our attack has a false-positive rate of α, which is the probability that an attacker incorrectly declares a randomly selected point z that was not used in training to have been used in training is α. By varying the parameter α, we can then trace the trade-off curves of *true-positive rates* at different false-positive rates.

3.1. Quantile Regression Learner

A generic way to train a quantile regression model is to optimize pinball loss over some function class $\mathcal{Q}$ (e.g., neural networks). Formally, for any observed t-error $\hat{\ell}$ and quantile prediction from q_α at a target level α, the pinball loss is defined as

$$L_\alpha(\hat{\ell}, q_\alpha) = (q_\alpha - \hat{\ell})(\mathbf{1}[\hat{\ell} \leq q_\alpha] - \alpha) \tag{9}$$

where $\mathbf{1}(\cdot)$ is an indicator function. Then we find a quantile regression model $q_\alpha(\cdot)$ that minimizes the *pinball loss*:

$$\min_{q_\alpha \in \mathcal{Q}} \Sigma_{z \in D} L_\alpha(\hat{\ell}_t(\theta, z), q_\alpha(z)), \tag{10}$$

Algorithm 2 Bag of Weak Attackers

Inputs: A set of auxiliary examples D drawn from P, target example z^*, trained diffusion model θ, a choice of t for t-error function, target false-positive rate α, and the number of weak attackers m.
Initialize vote=0
for each $z \in D$ **do**
  evaluate the score $\hat{\ell}_t(\theta, z)$
end for
for $i \in [m]$ **do**
  Bootstrap sampling a dataset D_i from D (Sample with replacement, and $|D_i| = |D|$)
  Learn a α-quantile regression model $q_{\alpha,i}$ on D_i (such that $q_{\alpha,i}(z)$ predicts the α-quantile of the score $\hat{\ell}_t(\theta, z)$ conditioned on z.)
  vote = vote + 1 if $\hat{\ell}_t(\theta, z^*) \leq q_{\alpha,i}(z^*)$
end for
Return "IN" if vote $\geq m/2$, otherwise "NO"

where $\mathcal{Q}$ is the class of quantile regression models. The pinball loss is minimized by the function that predicts for each z the target α-quantile of the t-error conditioned on z.

3.2. Hypothesis Testing Interpretation and Comparison with Shadow Models

A prevalent approach for membership inference attacks is to train *shadow models*—models that are trained by the same algorithm on a randomly drawn dataset that purposefully includes or does not include the target example z^*. In the case of attacking diffusion models, training a shadow model may require days of computation. We will now provide a hypothesis interpretation of membership inference, which shows how our proposed quantile regression attack avoids computing shadow models.

The starting point for the shadow-models approach is to consider the following two competing hypotheses about a trained generative model θ:

$$\begin{aligned} & H_0 : \theta \sim A(\mathbf{Z}) \mid z^* \notin \mathbf{Z} \\ \text{and} \quad & H_1 : \theta \sim A(\mathbf{Z}) \mid z^* \in \mathbf{Z}. \end{aligned} \tag{11}$$

which correspond to whether or not the training algorithm A uses a dataset $\mathbf{Z}$ that includes the target example z^*. To determine whether the observed trained model θ was drawn from the distribution H_0 or H_1, algorithms such as Likelihood Ratio Attack (LiRA) (Carlini et al., 2022) learn how to distinguish the two distributions by generating observations from them. Note that generating an observation from a distribution requires training an entirely new shadow model using a training dataset that includes z^* (in the case of H_0) or does not include z^* (in the case of H_1).

Similar to the approach in Bertran et al. (2023), we consider distributions over examples with and without conditioning on the fixed observed output θ. Formally, we consider the following hypotheses:

$$\begin{aligned} & H_0 \colon z^* \sim P \mid A(\mathbf{Z}) = \theta \\ \text{and} \quad & H_1 \colon z^* \sim \mathbf{Z} \mid A(\mathbf{Z}) = \theta \end{aligned} \tag{12}$$

That is, H_0 asserts that z^* is a random example drawn from the underlying distribution P, but H_1 asserts z^* is a random example drawn from a dataset $\mathbf{Z}$ that produces the trained model θ. In this view, MI is about distinguishing two distributions over examples instead of trained models. This alternative hypothesis testing view motivates our approach that learns quantile regression models over examples z.

Compared to the competing attack using shadow models (GSA (Pang et al., 2023) in Sec.4), our proposed attack achieves a higher accuracy without paying for their prohibitive computational cost. Another benefit of our attack is that it does not require knowledge about the training hyperparameters and network architecture, whereas training a shadow model requires the precise details of training.

3.3. Bag of Weak Attackers

One of the challenges, as is also mentioned in (Bertran et al., 2023), is that sometimes directly optimizing pinball loss (in (10)) is difficult in practice since the gradient is either $-\alpha$ or $1-\alpha$ depending on whether the quantile prediction is smaller or larger than the target. Rather than tackling the challenge directly, we take advantage of the fact that our quantile regression models here can be a much smaller neural network than the targeted diffusion models, which enables us to train multiple models with minimal computational overhead. In particular, we adopt the bagging approach (Breiman, 1996) to improve the generalization of our attack by taking an ensemble of tiny models trained on bootstrapped datasets. Since individual weak models may have relatively poor attack performance, and bagging improves the performance, we call our approach "bag of weak attackers". After learning, each weak attacker makes a binary decision on an input sample to decide whether this sample is used in training the diffusion model or not, and we simply take the majority vote over all weak attackers to obtain the final decision. The formal description of the algorithm is in Algorithm 2. It is worth noting that quantile regression models only need to be trained once before attacking any sample, and can be used to attack many samples simultaneously.

Another benefit of our bagging approach is that our attack does not require any hyperparameter optimization (HPO), as we are training quantile regression models using tiny neural networks. In comparison, the attack in Bertran et al. (2023) (on supervised learning models) chooses to optimize a surrogate Gaussian likelihood objective (instead of pinball loss) and requires substantial HPO.

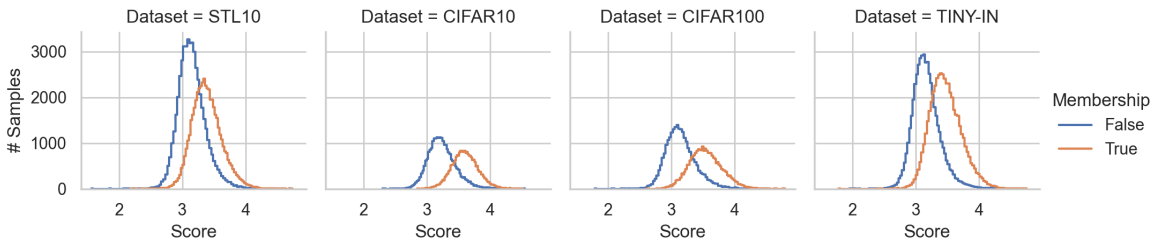


Figure 1: Distributions of the (negative) log transformation of the t-error on members and nonmembers of a dataset. It is clear that on each dataset, members and nonmembers have slightly different marginal score distributions, however, they are not drastically different from each other, which explains why the marginal baselines are not optimal, and also motivates our approach that conditions the score prediction on the input sample.

4. Experimental Details

We demonstrate the effectiveness of our membership inference attack via quantile regression on four denoising diffusion probabilistic models (Ho et al., 2020) (DDPMs) trained on CIFAR-10, CIFAR-100 (Krizhevsky, 2009), STL-10 (Coates et al., 2011) and Tiny-ImageNet, respectively. On each dataset, data samples are split into two halves, and one half is regarded as the private samples $\mathbf{Z}$ for training a DDPM. The other half is then split into two sets, including one as the public samples D that are auxiliary information, and the other as the holdout set for testing. On public samples, we train quantile regression models.

The base for our quantile regression model is a ResNet model, and it is attached with multiple prediction heads, each of which predicts target α-quantile for a specific value of α. Compared to the standard ResNet-18 model for classifying CIFAR-10, we design each attacker to be much smaller than the standard ResNet model. We present results with a varying number of weak attackers, each with a varying number of parameters.

In our experiments, we use the same diffusion models trained on CIFAR10 and CIFAR100 released by (Duan et al., 2023), and trained our own diffusion models — specifically DDPM — on STL10 and Tiny-ImageNet using a lower resolution, 32×32, due to the limit of computing resources. Each diffusion model was trained with 80k steps, and it took around 2 days to finish training on a single V100 GPU card. After obtaining these trained diffusion models, for membership inference attacks, we use a fixed $t = 50$ in the t-error function. Duan et al. (2023) suggested that the choice of t does not influence the results drastically. More training details can be found in Appendix B.

We adopt the same evaluation metric as prior work (Bertran et al., 2023; Carlini et al., 2022). Specifically, we are interested in the True Positive Rates (TPRs) at very low False Positive Rates (FPRs). Intuitively, a successful membership inference attack should identify true members with high accuracy, and in the meantime, make few mistakes in declaring nonmembers as members.

4.1. Competing attacks

Marginal baseline. The first one is a simple **marginal** baseline, which only looks at the error distributions of members and nonmembers, respectively. For a target FPR value α, it computes the quantile on t-errors of the public samples, and then the performance of this marginal baseline is evaluated on the private samples and the holdout set. It is clear that the marginal baseline only produces a single threshold for a target FPR, and it does not condition on the input images, whereas ours learns to predict the threshold for a given image, thus each image has a different threshold for a target FPR. Figure 1 illustrates the error distributions on individual datasets, and it is clear that the error distribution of members is different from that of nonmembers, but there is not a single threshold that can clearly distinguish the two distributions. Our attack produces, for a fixed α-value, a sample-conditioned threshold by learning quantile regression models, thus, for each sample under attack, the threshold is unique, which empowers the MI attack.

Shadow models attack (GSA) The other competing attack is also a white-box attack that adapts the approach of Likelihood Ratio Attack (LiRA) (Carlini et al., 2022) with gradient information (Pang et al., 2023), namely **GSA**. Similar to LiRA, GSA requires training many shadow models on random subsets of the public data, and thus suffers a significantly higher computational cost compared to our quantile-regression-based attack and the marginal baseline.

4.2. Main Results

Numerical results are presented in Table 1, and they are averaged across 10 random seeds. The weak attacker in our experiment is a small neural network with only 5,666 parameters, and the number of weak attackers is 7. We can see that our attack on CIFAR10, besides being much more efficient, outperforms GSA attacks when we focus on lower TPR (0.1%). We also have demonstrated the effectiveness of

Table 1: Performance of MI Attacks. Each weak attack is a neural network with only 5,666 parameters, and the number of weak attackers is 7.

Dataset	CIFAR-10		CIFAR-100		TINY-IN		STL-10	
MI Attack (TPR @ fixed FPR)	@ 1%	@ 0.1%	@ 1%	@ 0.1%	@ 1%	@ 0.1%	@ 1%	@ 0.1%
Bag of Weak Attackers	99.94%	99.86%	99.89%	99.75%	99.99%	99.98%	99.98%	99.98%
GSA_1 (Shadow Models)	99.70%	82.90%	-	-	-	-		
GSA_2 (Shadow Models)	97.88%	58.57%	-	-	-	-		
Marginal Baseline	9.6%	0.7%	11.06%	5.76%	8%	0.32%	5.78%	0.55%

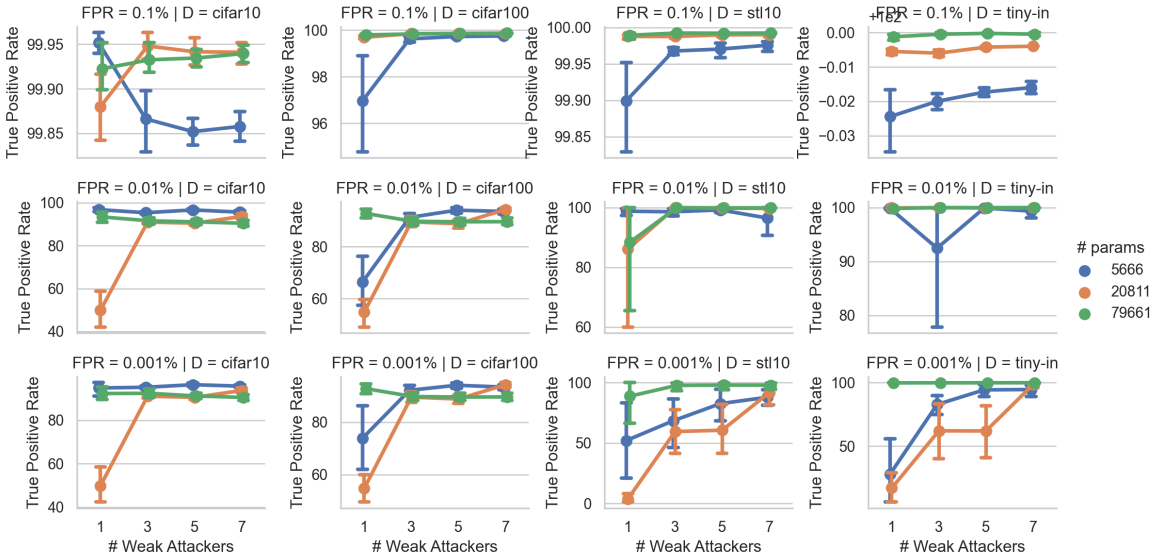


Figure 2: The Effect of Bagging. The bagging predictor takes the majority vote over the decisions made by weak attackers - quantile regressors in our case. Due to the variance reduction aspect of bagging, we expect the attack performance to increase as the number of weak attackers increases. We present the effect of bagging at three fixed FPR values, including 0.1%, 0.01%, and 0.001%, on four diffusion models, and we also present the impact on three neural networks of different sizes. Overall, the attack performance demonstrates a non-decreasing trend as the number of weak attackers increases.

our attack on diffusion models trained on CIFAR100, STL10 and Tiny-ImageNet in Table 1. Besides the performance improvement, our algorithm requires no knowledge about the training algorithm of the diffusion models.

4.3. Time Consumed in Preparing An Attack

Apart from simply using a uniform threshold on the marginal distribution, other white-box attack approaches including ours, involve training machine learning models. Efficient attack algorithms enable attackers to launch attacks more frequently on a machine learning system, and, on the bright side, it also enables frequent privacy auditing for people who are maintaining these systems.

LiRA attack on a target diffusion model trains shadow models, of which each is a diffusion model of the same size. (Pang et al., 2023) proposed to extract gradient information, and it introduces additional latency overhead on top of learning shadow models. On the contrary, our approach learns simple quantile regression models. To demonstrate the efficiency of our algorithm in preparing the attack, we estimated the clock time on a single V100 GPU card, and the rough estimates are presented in Table 2.

4.4. Impact of Bagging

Since our algorithm doesn't require knowledge of the training algorithm of the target diffusion model, the quantile regression model technically can be an arbitrary machine learning model. Therefore, with an inappropriately chosen model architecture for quantile regression, the attack performance at low FPR may not be desirable. Our proposal to alleviate this issue to through bagging, hence namely "bag of weak attackers". We now present an empirical ablation study of the impact of bagging on various model sizes.

We select the base architecture for our quantile regression as a ResNet model (He et al., 2016), however, we significantly reduce the number of channels in each layer to make it much smaller in terms of the number of parameters and make it much more efficient to train and conduct inference. In our

Table 2: Time Consumed in Preparing An Attack on Tiny-ImageNet.

Preparation Steps	Computing Scores on Public Samples using the Target Model	Learning Models
GSA (6 Shadow models)	0 mins (not required)	2 days (x 6 shadow models = 12 days)
Quantile (7 regression models)	8 mins	4 mins (x 7 regression models = 28 mins)

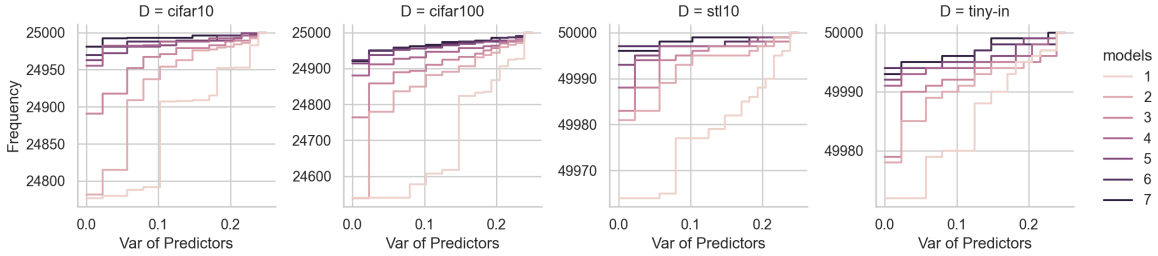


Figure 3: Variance Reduction of Bagging. We present CDF plots of variances of bagging predictors with an increasing number of quantile regression models on four datasets in our experiments. The x-axis indicates the variance estimated from 10 random runs, and the y-axis indicates the cumulative count of samples used in training the diffusion model. Ideally, with an increasing number of predictors in bagging, prediction variances of samples decrease, thus, more samples enjoy low prediction variances. We can see that, as the number of models increases in the bagging predictor, the CDF plot gradually becomes flatter, which means that the predictions on more samples now have lower variance. Thus, it shows that, by taking the majority vote of weak attackers, the prediction variance decreases, which results in better performance.

experiments, the smallest ResNet for quantile regression has only 5.6×10^3 parameters.

We consider three configurations of the number of channels in ResNet, which then results in three base models with different numbers of parameters, including 5.6×10^3, 2.0×10^4, and 8.0×10^4. Details are in Appendix A. It is worth noting that the number of parameters in a single diffusion model in our experiments is 7.1×10^8, and our attack models are significantly smaller than the target model. We then vary the number of weak attackers from 1 to 7. Each configuration is run with 10 random seeds.

Improving Attack Performance. Figure 2 presents results that indicate the effectiveness of our "bag of weak attackers". We can see that, for the smallest model, essentially the weakest attacker, increasing the number of attackers improves the performance drastically at low FPR, and for the other two models, bagging improves in some cases, but in general, is not detrimental to the attack performance.

This is particularly interesting that the attacker first doesn't need to know the training algorithm of the diffusion model, which is required for LiRA attacks, and even if the attacker chooses a small quantile regression model, which might be a weak attacker, they can still obtain nearly perfect attack performance through bootstrap ensemble, namely, training a small number of small quantile regression models.

Reducing Variance of Individual Predictors. The performance improvement of bagging is mainly derived from variance reduction by taking an ensemble over several predictors. In our case, we directly take a majority vote over the decisions of individual weak attackers, and Figure 2 already shows that overall the attack performance improves as the number of weak attackers increases. It would be interesting to show that the performance improvement indeed comes from variance reduction.

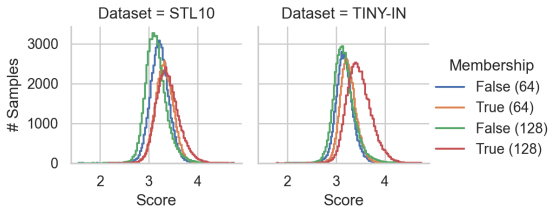
We select the smallest model architecture in our experiments, which is the one with only 5,666 parameters. For a fixed number of models in bagging, we estimate the per-sample variance on the same holdout set by running our algorithm several times, and we plot the Cumulative Distribution Function (CDF) of the per-sample variance over the samples in the holdout set in Figure 3. In general, as the number of models in bagging increases, the CDF shows a steeper increase in the area where the variance of predictors is small. Therefore, more models in bagging lead to smaller variance.

4.5. Impact of the Size of the Model Under Attack

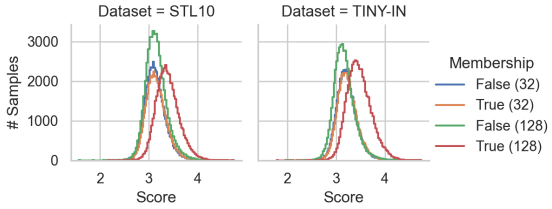
In our previous experiments, the core architecture of the DDPM — UNet — has 128 channels in its first layer, and as shown in Figure 1, the learned diffusion model assigns on average lower reconstruction errors to private training samples than public samples, but with significant overlaps. Here, we reduce the number of channels to 64, and then to 32, in the first layer of DDPM , and train diffusion models on Tiny-ImageNet and STL10, respectively, to demonstrate the impact of the target model size.

Figure 4 shows the reconstruction errors produced by the smaller models on the private training samples and the public samples. These plots show transformed errors using

$-\log(\ell)$, so large scores indicate small reconstruction errors. In comparison to reconstruction errors obtained from models with 128 base channels in our previous experiments, smaller models have a smaller generalization gap, but also produce worse reconstruction errors on the unseen public set. Specifically, both smaller models trained on Tiny-ImageNet have almost no difference in the error distributions. It means that a single threshold wouldn't perform well in these scenarios, which motivates us to learn per-sample thresholds.



(a) Target models with 64 base channels

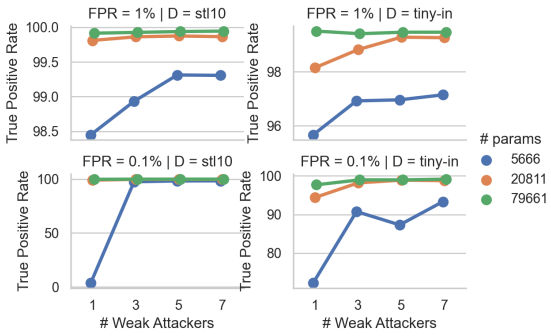


(b) Target models with 32 base channels

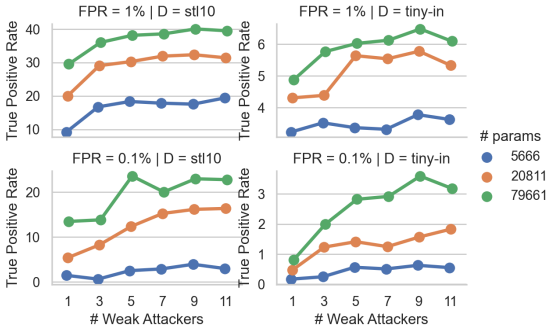
Figure 4: Reconstruction errors from target models of two different sizes in comparison to those from target models in our main experiments. As the model size decreases, the generalization gap between the reconstruction errors on the private training samples and that on the public samples decreases as well, which renders attacks that only use a single threshold for all samples under attack impractical, Especially on Tiny-ImageNet, the gap is almost zero. It is worth noting that, with a decreasing number of channels, the reconstruction errors on the public samples gradually become worse, which renders the target model less generalizable.

The performance of our attack is presented in Figure 5. Even though the error distributions, including one on the private training data, and one on the public data, are almost identical in some cases, our attack still achieves good performance because our attack produces a threshold conditioned on the sample under attack.

Specifically, on target models with 64 base channels, our attack again provides almost perfect performance on both STL10 and Tiny-ImageNet; on target models with 32 base channels, the error distribution on the private training samples is identical to that on the public samples visually, which potentially gives the false impression that it would be impossible to determine whether a data sample is in the training set or not, our attack provides strong performance on STL10,



(a) Attacking Target Models with 64 Base Channels.



(b) Attacking Target Models with 32 Base Channels.

Figure 5: Attack performance of our bag-of-weak-attackers algorithm. Our attack previously provided nearly perfect performance on attacking target models with 128 base channels, and here, the observation is similar on attacking target models with 64 base channels. Our attack provides strong performance on target models with 32 base channels even when the generalization gaps are almost zero.

and non-trivial performance on Tiny-ImageNet. In addition, our attack only requires tiny weak attackers that are easy and fast to train. The positive effect of bagging is prominent, and the computational overhead of launching an attack is negligible in contrast with training a diffusion model.

5. Conclusion

We demonstrate the success of our attack via quantile regression on diffusion models. With side information, including auxiliary samples from the data distribution, the results show the effectiveness of our membership inference attack on four diffusion models trained on different datasets respectively. Besides the performance, our attack is also computationally efficient compared to prior approaches based on shadow models. Moreover, the effectiveness and the efficiency of our algorithm indicate that diffusion models are indeed extremely vulnerable to MI attacks, and extra care should be taken when releasing a trained diffusion model. An exciting future direction is to investigate whether we can extend our current approach to a black-box attack setting, where we do not have direct access to the trained model θ.

Impact Statement

We demonstrate the privacy risks of using large-scale diffusion models with a simple attack algorithm, and it raises privacy concerns on deploying diffusion models, especially large-scale ones. Techniques with theoretical guarantees of privacy protections, including Differential Privacy, should be considered.

References

Bertran, M. A., Tang, S., Roth, A., Kearns, M., Morgenstern, J., and Wu, S. Scalable membership inference attacks via quantile regression. *pre-print*, 2023.

Breiman, L. Bagging predictors. *Machine learning*, 24: 123–140, 1996.

Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., Oprea, A., and Raffel, C. Extracting training data from large language models. *ArXiv*, abs/2012.07805, 2020.

Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramèr, F. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1897–1914, 2022. doi: 10.1109/SP46214.2022.9833649.

Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 5253–5270, 2023.

Coates, A., Ng, A., and Lee, H. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.

Duan, J., Kong, F., Wang, S., Shi, X., and Xu, K. Are diffusion models vulnerable to membership inference attacks? In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8717–8730. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/duan23b.html.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, TCC '06, pp. 265–284, New York, NY, USA, 2006.

Hayes, J., Melis, L., Danezis, G., and De Cristofaro, E. Logan: Membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019:133–152, 01 2019. doi: 10.2478/popets-2019-0008.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J. V., Stephan, D. A., Nelson, S. F., and Craig, D. W. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

Jagielski, M., Ullman, J., and Oprea, A. Auditing differentially private machine learning: How private is private sgd? In *Advances in Neural Information Processing Systems*, NeurIPS '20, 2020. https://arxiv.org/abs/2006.07709.

Jayaraman, B. and Evans, D. Evaluating differentially private machine learning in practice. In *USENIX Security Symposium*, 2019.

Jayaraman, B., Wang, L., Evans, D., and Gu, Q. Revisiting membership inference under realistic assumptions. *arXiv preprint arXiv:2005.10881*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. 2009. URL https://api.semanticscholar.org/CorpusID:18268744.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

Nasr, M., Song, S., Thakurta, A., Papernot, N., and Carlini, N. Adversary instantiation: Lower bounds for differentially private machine learning. In *IEEE Symposium on Security & Privacy*, IEEE S&P '21, 2021. https://arxiv.org/abs/2101.04535.

Nasr, M., Hayes, J., Steinke, T., Balle, B., Tramèr, F., Jagielski, M., Carlini, N., and Terzis, A. Tight auditing of differentially private machine learning. *CoRR*, abs/2302.07956, 2023. doi: 10.48550/arXiv.2302.07956.

Pang, Y., Wang, T., Kang, X., Huai, M., and Zhang, Y. White-box membership inference attacks against diffusion models. *arXiv preprint arXiv:2308.06405*, 2023.

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

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Stadler, T., Oprisanu, B., and Troncoso, C. Synthetic data - anonymisation groundhog day. In Butler, K. R. B. and Thomas, K. (eds.), *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pp. 1451–1468. USENIX Association, 2022. URL https://www.usenix.org/conference/usenixsecurity22/presentation/stadler.

Tramèr, F., Terzis, A., Steinke, T., Song, S., Jagielski, M., and Carlini, N. Debugging differential privacy: A case study for privacy auditing. *CoRR*, abs/2202.12219, 2022. URL https://arxiv.org/abs/2202.12219.

van Breugel, B., Sun, H., Qian, Z., and van der Schaar, M. Membership inference attacks against synthetic data through overfitting detection. In Ruiz, F. J. R., Dy, J. G., and van de Meent, J. (eds.), *International Conference on Artificial Intelligence and Statistics, 25-27 April 2023, Palau de Congressos, Valencia, Spain*, volume 206 of *Proceedings of Machine Learning Research*, pp. 3493–3514. PMLR, 2023. URL https://proceedings.mlr.press/v206/breugel23a.html.

Wang, L., Jayaraman, B., Evans, D., and Gu, Q. Efficient privacy-preserving nonconvex optimization. *arXiv preprint arXiv:1910.13659*, 2019.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, CSF '18, pp. 268–282, 2018. https://arxiv.org/abs/1709.01604.

A. Training details of attack models

All attack models in our experiments are trained with the same following optimization settings:

1. optimizer: Adam (Kingma & Ba, 2015)
2. batch size: 128
3. initial learning rate: 1e-3
4. number of training epochs: 200
5. learning rate scheduler: cosine annealing without warm restarts (Loshchilov & Hutter, 2017)

Each attack model follows the ResNet architecture (He et al., 2016), and we adopt a publicly available GitHub repository [1] for our experiments. Rather than using the original ResNet architecture with 64 output channels in the first convolutional layer, to reduce the number of parameters and to reduce the training and inference latency, we adjust the number of output channels in the first convolutional layer, and adjust the subsequent layers accordingly. Table 3 settings describe the adjustments.

Table 3: Configurations of Attack Models

Output Channels of the First Convolutional Layer	Total Number of Parameters
1	5.6×10^3
2	2.0×10^4
4	8.0×10^4

B. Training details of target models

Two target models — diffusion models on Tiny ImageNet and STL10 — are trained using the same code base as provided in (Duan et al., 2023), and we directly used the diffusion models trained on CIFAR10 and CIFAR100 released in the same code base. Target models are trained using the released codebase[2] by (Duan et al., 2023), and we didn't change the codebase for our experiments.

To obtain the smaller diffusion models, we used the same codebase and only changed the number of the base channels in the UNet to 32, which significantly reduced the total number of parameters, but it still took around 40 hours to finish training.

[1] https://github.com/kuangliu/pytorch-cifar
[2] https://github.com/jinhaoduan/SecMI