VERY SIMPLE MEMBERSHIP INFERENCE AND SYNTHETIC IDENTIFICATION IN DENOISING DIFFUSION MODELS

By

Bowen Qu

Thesis
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE
in

Computer Science

May 10, 2024

Nashville, Tennessee

Approved:

Daniel Moyer, Ph.D.

Jie Ying Wu, Ph.D.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to several individuals who have contributed significantly to the completion of this Master's thesis.

First and foremost, I am profoundly grateful to my mother for her unwavering love, encouragement, and support throughout my academic journey. Her sacrifices, guidance, and belief in my abilities have been the driving force behind my accomplishments.

I owe a debt of gratitude to my advisor, Daniel, whose expertise, patience, and insightful feedback have been invaluable in shaping this thesis. His mentorship has not only enriched my academic experience but also inspired me to strive for excellence in my research pursuits.

I would also like to extend my appreciation to my roommate, Nurshat, for his understanding, encouragement, and occasional study sessions that provided much-needed motivation during challenging times.

Furthermore, I am indebted to all the members of my laboratory, especially Mingxing, for their collaboration, camaraderie, and intellectual exchange, which have enriched my research journey and contributed to the development of this thesis.

Lastly, I would like to thank everyone who has supported me in ways both seen and unseen, contributing to the culmination of this academic endeavor. Your encouragement and belief in my potential have been instrumental in reaching this milestone.

Thank you all for your unwavering support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Introduction

Membership inference is the task of predicting for a given data point and a trained machine learning model whether that data point is part of the training set for the model Shokri et al. (2017). Synthetic identification is the term we are giving to the task of identifying generated data points from real ones, a problem well studied from a variety of angles Goodfellow et al. (2014); Sabir et al. (2019); Dolhansky et al. (2020).

Both of these tasks have a large set of practical applications and implications, Membership inference methods have uses in understanding data privacy, auditing, regulation, data forensics, and transparency in learned models; specifically for generative models, membership inference methods have been proposed as both a tool for auditing copyright complianceChen et al. (2020), as well as an attack vector for obfuscated datasets. Synthetic identification is actively used in systems identifying deep fakes Sabir et al. (2019) and other model generated forgeries, and is performed manually by secondary and tertiary educators for text generation on a regular basis.

Beyond these applications, both tasks are also generally thought to speak to structures and properties of the underlying models. Synthetic identification for natural images often relies upon knowledge of geometry and the physical reality of any real image Ma et al. (2022), as well as the lack thereof in the generated images. Membership inference leverages the overfitting characteristics of a particular model to the training set Hayes et al. (2017), characteristics which are often modulated by training dynamics Yeom et al. (2018).

In this work we describe a very simple method for both membership inference and synthetic identification for the Denoising Diffusion Probabilistic Model (DDPM) class of generative image methods. We show that by measuring the Euclidean norm of the output noise parameter ε we can construct a simple threshold classifier that separates training from test data, as well as identify synthetic data. Using this classifier, we then demonstrate both membership inference and synthetic identification using publicly available weight sets and codebases at a surprisingly high accuracy. We then provide further experiments on dimensionality and number of training steps using our own instances of DDPMs.

We believe that the primary merits of this work are in the simplicity of the method relative to its accuracy, its apparent general applicability, and its connections to the theoretical understanding of diffusion models. While a potential auditing system or detection method for deep-fake images would be of great practical benefit, it is not clear that our particular threat-model and use-case constraints (ability to query and manipulate the generating diffusion model inputs) fit the needs of a potential deployed system for either of those cases.

We present the following:

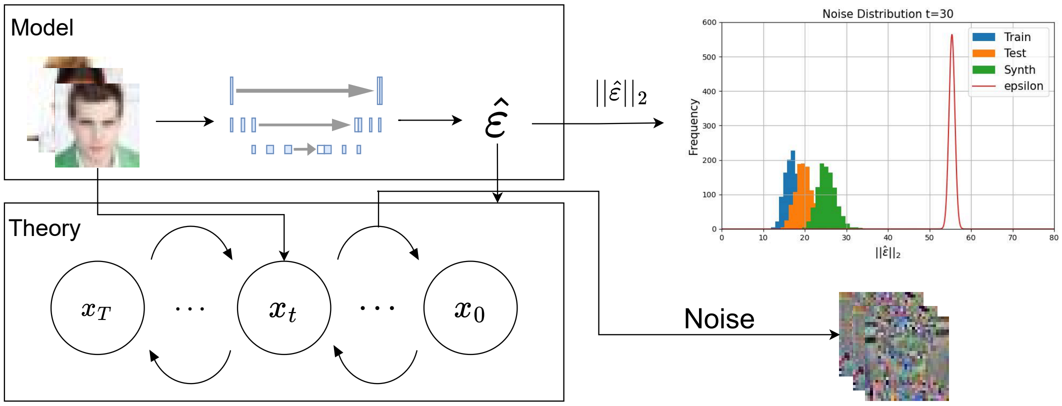

Figure 1.1: A diagram of the DDPM model and our attack process: at left, the DDPM (in theory below, and the practical estimator above), from which we extract the output $\hat{\varepsilon}$. The norm of this output is different across Train, Test, and Synthetic datasets.

- a simple membership inference and synthetic identification estimator for Diffusion models (the Euclidean norm of the estimated noise vector),
- results on publicly available datasets as well as publicly available weight sets across several common architectures and image resolutions,
- and a discussion of possible theory and reasoning behind DDPM susceptibility to these methods.

Our code and experiments can be found at https://github.com/lmyhaha/VerySimpleMI.

1.1 Related Work

1.2 DDPM and related generative image models

Denoising Diffusion Probabilistic Models have become the main class of generative image models in current literature. Introduced in their current form in Ho et al. Ho et al. (2020) and as score matching in Song and Ermon Song et al. (2020b), these models and their variants Kingma et al. (2021); Song et al. (2020a); Kong et al. (2023); Nichol and Dhariwal (2021) are the standard image backbone in numerous high-profile generative systems Ramesh et al. (2022); Saharia et al. (2022); Rombach et al. (2022). Diffusion model use also extends beyond 2D images, into 3D point clouds graphics primitives Poole et al. (2022); Nichol et al. (2022); Jun and Nichol (2023), audio Kong et al. (2020), and text Yang et al. (2024).

Extensive theory results can be produced for both the general DDPM class of models and their sampling techniques. These often rely on a stochastic differential equation interpretation of the noising process Song et al. (2020b); De Bortoli et al. (2021); McAllester (2023), but may also be viewed from variational Kingma et al. (2021), information theoretic Kong et al. (2023), or Kernel Density Estimation Alain and Bengio (2014) viewpoints.

1.3 Membership Inference and Synthetic Identification

Membership Inference Shokri et al. (2017) is a recent topic of interest for models with very large parameter sets (e.g. Deep Networks) due to their potential to memorize and regurgiate training examples. Methods for Membership Inference (known as "attacks") may broadly be separated into white-box/high-knowledge and black-box/low-knowledge attacks, where the highest knowledge attack may require access to model weights, and the lowest knowledge attack may not even know the general class of models. Our proposed method is tailored to Diffusion models, and requires manipulation of the time-step variable; however, it does not need access to weights, and it is general across different architectures and implementations of the general DDPM conceptual model. Thus, the proposed model is somewhere between black-box and white-box.

Membership inference is often also referred to as model inference, with some authors making a distinction between the two tasks Yang et al. (2020). In generative image models in particular, a sub-task of membership inference is the inference/recovery of some or all of the training data points; this is sometimes called model inversion.

Loss-based attacks Yeom et al. (2018) are a subset of membership inference attacks leveraging overfit of model to the training set. As the overfitting process proceeds, the model ingests information specific to the dataset, but not about the general task; this manifests as a differential between the loss measured on a training datapoint and the loss measured on a test datapoint drawn from the same (actual) generating data distribution.

Most relevant to our own work, Duan et al. Duan et al. (2023) specify a diffusion model specific loss-based. For each trial point (drawn from test or training sets) they perform a noising process, then measure the loss as estimated by the denoiser at the given noise level (artificial time-step on the noise schedule). This is similar to our own work in that they require the same level of model access, and their attack is tailored to Diffusion methods; we provide a comparison with our own model empirically in Section 3.3, and a discussion of the similarities and difference in Section 4.2.

Synthetic identification is the term that we are giving to identifying generated images from real images. In generative adversarial networks Goodfellow et al. (2014) it is the main training loss, providing gradient information for generating realistic images. A very closely related task, identifying ML-assisted photo manipulations also known as Deep Fake detection Verdoliva (2020); Dolhansky et al. (2020), is very well studied in recent literature Zhao et al. (2021); Haliassos et al. (2021); Nguyen et al. (2022). In the specific case of Diffusion model generation, we find that detection of synthetic data can be done efficiently using the same estimator we use for membership inference, and moreover at higher fidelity than separating training versus test data. We provide hypothese about these phenomena in Section 3.5.

CHAPTER 2

Method

2.1 DDPM

Denoising Diffusion Probabilistic models form the core of a wide class generative image methods. While varying in exact parameterization, architecture, and loss formultion, these methods share an overall structure. Diffusion models consists of two primary processes: a forward noising process and a reverse process. In the forward process, also known as the diffusion process, a data distribution p_{data} is transitioned into a standard Gaussian distribution $\mathcal{N}(0,I)$ by gradually introducing increasing levels of noise ε to data x. Standard notation defines an artificial time variable t which indexes steps from 0 to T, where x_0 is drawn from p_{data} and x_T is assumed to have reached its asymptotic standard Normal distribution. Between those two extremes, the model defines $x_t = \sqrt{\bar{\alpha}_t}x + \sqrt{1-\bar{\alpha}_t}\varepsilon_t$ for $\varepsilon_t \sim \mathcal{N}(0, I\sigma(t))$; the speed of the transition from p_{data} to $\mathcal{N}(0,I)$ is dictated by noise schedule $\sigma(t)$ and $\alpha(t)$, which by convention has $\sigma(T) = 1$.

The reverse process (denoising) prescribes fitting an estimator $\hat{\varepsilon}$ conditioned on x_{t+1} and t to the aggregate noise (i.e., to the induced noise from the forward process up until t). This estimator $\hat{\varepsilon}(x_t,t)$ is usually a U-Net, and would optimally output $x_t - x_0$ for inputs (x_t,t). It is this network that we will manipulate to generate our estimators. When generating synthetic data points, $\hat{\varepsilon}(x_t,t)$ is used as the gradient in a Langevin sampling method, where temperature corresponds with time-step t.

2.2 Membership Inference and Synthetic Identification Attack

Choose a t “false” time-step (this variable is a hyper-parameter of the attack). We assign a score $s(x^*)$ to each trial datapoint x^*, equal to the ℓ_2 norm of the output of the $\hat{\varepsilon}$ network:

$$s(x^*) = \|\hat{\varepsilon}(x^*,t)\|_2 \tag{2.1}$$

We conduct membership inference directly by thresholding the score at varying values and comparing the proportion of either test or training datapoints above or below the threshold; this is the entire membership inference/synthetic identification procedure. As shown in Section 3, for synthetic identification whether the train dataset has higher scores or lower scores than the synthetic dataset appears to depend on the dimension of the data.

CHAPTER 3

Experiments

3.1 Datasets, Baselines, and Victim Models

We test our method on two sets of two combined datasets: for the first set, we use CelebAHQ Karras et al. (2017) as a training dataset, and the similar but non-overlapping FFHQ Karras et al. (2019) dataset as the test set. For the second dataset, we use ImageNet Deng et al. (2009) as the training dataset, and we use ImageNetV2 Recht et al. (2019) as the test set. For each of these, we also run SecMI, a recently proposed loss based attack Duan et al. (2023). However, evaluations of SecMI exceeded our computational capacity, so we only report partial results for ImageNet.

3.1.1 CelebA-HQ, FFHQ, and Pre-trained Model

CelebA-HQ is a widely used face dataset in computer vision research Karras et al. (2017). It is a subset of the original CelebA dataset Liu et al. (2015), featuring high-resolution facial images of public figures obtained from the internet. The CelebA-HQ dataset offers significantly higher image quality on average than the original larger CelebA dataset.

Due to computational limitations, we downsample the CelebA-HQ dataset. For the initial experiment, we use the DDPM weights published by Ho et al. and rehosted by Hugging Face[1] at a resolution of 256×256 pixels. For experiments with our own trained networks, we use 32×32 and 64×64 resolution.

Because we cannot be sure of exactly which data from CelebA-HQ were used in training the outside model, we use FFHQ as the test-set. These data may be subject to slight variations in collection procedure and domain (CelebA specifically uses public figures, while FFHQ is drawn from a public repository of images which includes regular individuals). However, we believe that it is suitable as a test dataset as they are both found natural images of human faces, aligned and downsampled by us to the appropriate resolution.

3.1.2 ImageNet, ImageNetV2, and Pre-trained Model

ImageNet is a standard dataset comprised of 14 million labeled images covering thousands of object categories. The ImageNetV2 dataset introduces fresh testing data for the ImageNet benchmark, intended to allow replication tests of method performance for methods originally trained on ImageNet. It is comprised of three test sets, each containing 10,000 new images.

These sets, defined by the original authors Recht et al. (2019), include Threshold 0.7 (Test 1), constructed

[1] https://huggingface.co/google/ddpm-celebahq-256

by sampling ten images per class with a selection frequency of at least 0.7, Matched Frequency (Test 2), sampled to align with the MTurk selection frequency distribution of the original ImageNet validation set for each class, and Top Images (Test 3), comprising the ten images with the highest selection frequency in the candidate pool for each class. These three test datasets serve as the test set for an ImageNet trained diffusion model. To reduce computational costs, from each of these we sample 1000 images at random without replacement.

We downsample these images to 256×256 pixels. As the original DDPM authors did not train a model on ImageNet, we instead use an implementation of the similar Guided Diffusion model[2]. While the loss function and architecture have slight difference from the original DDPM, the overall structure remains the same, and importantly $\hat{\varepsilon}(x^*, t)$ has the same functional signature and outputs. As we show, our attacks perform similarly against this variant diffusion model as well.

3.2 Configuration for Training Dynamics Experiments

In order to measure changes in the membership inference and synthetic identification performance with respect to training dynamics, we train our own DDPM models. We use the standard U-Net architecture with an initial channel size of 128, channel multiplier values of [1, 2, 3, 4], attention applied every 2 steps, with 2 residual sub-blocks within every U-Net block, a dropout rate of 0.15, a learning rate of 1e-4, a beta value of 1e-4 for the first time step and 0.02 for the last time step. We allow a maximum of 1000 diffusion time steps (T),

For each model, we randomly select 1000 images from the training dataset to serve as training trial points for evaluation of the membership inference method. We generate 1000 samples from the trained DDPM at differing numbers of training epochs, which become the synthetic datasets.

3.3 Performance

We present our findings on both the CelebA-HQ and ImageNet datasets, utilizing the Area Under the ROC Curve (AUC) as our metric, as detailed in Table 3.1. Figure 3.2 depicts the comprehensive AUC values across the range of t values from 0 to 200. The proposed attack method is effective when selecting certain values of t smaller than 100, , with AUC exceeding 0.7. Remarkably, when t is set to 10, the AUC can even reach approximately 0.95. Figure 3.1 illustrates the membership set, test set, and synthesis set of the mean of the noise within both datasets. It is apparent that the mean of the noise in the synthesis set is considerably lower than that of the membership set, facilitating the detection of the synthesis set from the membership test. Moreover, despite the high AUC, the mean between the membership set and test set remains relatively

[2] https://github.com/openai/guided-diffusion

Table 3.1: AUC Comparison on CelebA-HQ and ImageNet Datasets at 256×256 Resolution with FFHQ and ImageNetv2 Test Sets. Due to lack computational resources, not all SecMI baseline comparsions could be completed.

Dataset	Set	AUC			
		t=0	t=10	t=50	t=100
CelebA-HQ	SecMI Train vs. Test	0.85	0.81	0.65	0.35
	Train vs. Test	0.69	0.80	0.85	0.90
	SecMI Train vs. Synth	0.99	1.00	0.99	0.99
	Train vs. Synth inv	1.00	1.00	0.94	0.63
ImageNet	Train vs. Test Set 1	0.62	0.87	0.47	0.45
	SecMI Train vs. Test Set 1	0.59	0.83	-	-
	Train vs. Test Set 2	0.61	0.87	0.49	0.47
	Train vs. Test Set 3	0.61	0.88	0.49	0.47
	Train vs. Synth inv	0.53	0.90	0.70	0.69
	SecMI Train vs. Synth	0.42	0.45	-	-

Table 3.2: AUC Comparison of Our Method on CelebA-HQ Dataset at Various Data Resolutions.

Resolution	Set	Diffusion Step t			
		0	10	50	100
32×32	Test	0.42	0.57	0.95	**0.99**
	Synth	0.76	**1.00**	**1.00**	0.98
64×64	Test	0.46	0.49	0.79	**0.93**
	Synth	0.43	**1.00**	**1.00**	0.96
256×256	Test	0.69	0.80	0.85	**0.90**
	Synth inv	**1.00**	**1.00**	0.94	0.63

close. However, we observe an abnormal increase in the synthesis set compared to the membership test, a phenomenon we will delve into further in the subsequent discussion.

3.4 Resolution Effect

We also explore the impact of different resolutions on our analysis. We experiment with resolutions of 32×32, 64×64, and 256×256 using the CelebA-HQ dataset. For lower resolutions, specifically 32×32 and 64×64, we partition the dataset to train our own model, reserving unused images for the test set. Conversely, for the higher resolution of 256×256, we utilize the aforementioned pretrained model and FFHQ dataset images for testing purposes.

In Table 3.2, we compare the AUC values across different resolutions. As t increases, the accuracy improves when comparing the train set with the test set. However, the disparity between the train set and synthesis set distributions is most pronounced when t is approximately 10. Further details can be observed in Figure 3.3. Notably, for low resolutions, the magnitude of noise in both the synthesis set and test set

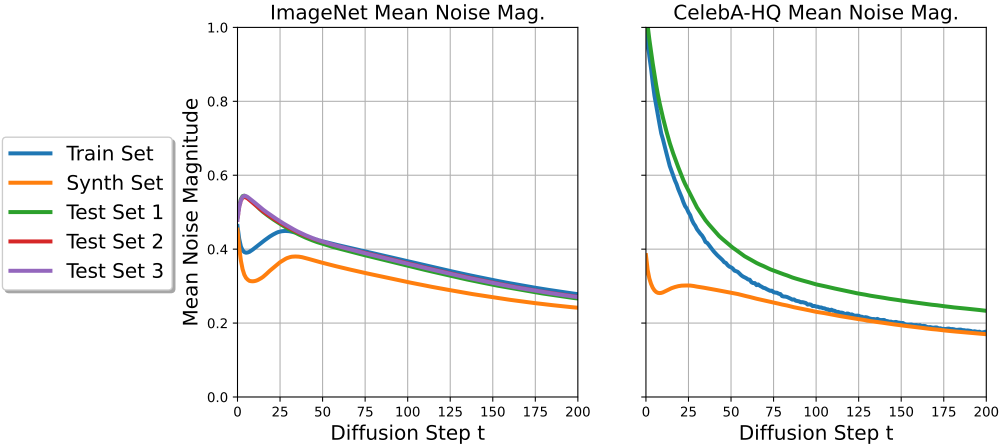

Figure 3.1: Mean Noise Magnitude Comparison Across First 200 Diffusion Steps for Different Datasets at 256 × 256 Resolution. **On the left:** Performance on ImageNet dataset, with three Test Sets representing the Test Matched Frequency set , Test Threshold0.7 set, and Test Top Images set in ImageNetv2. **On the right:** Performance on CelebA-HQ dataset, with Test Set1 representing the FFHQ set.

consistently exceeds that of the train set. Conversely, for higher resolutions, the noise magnitude in the synthesis set is lower than that of the train set, a point we will elaborate on in the subsequent sections. Nevertheless, when we reverse the magnitude, comparing the train magnitude with the synth magnitude also yields a high AUC.

Figure 3.4 illustrates the AUC values, indicating that in lower resolutions, the model tends to overfit, making it easier for our method to detect the training set from the test or synthesis set. However, as the resolution increases, although our method still performs effectively, the range of t values that yield optimal results becomes narrower.

3.5 Inversion on synthesis on high resolution data

In our preceding experiments, we observed consistent disparities in the performance of the synthesis set between high and low resolutions. Several hypotheses may explain this phenomenon.

Firstly, it could be attributed to overfitting of the diffusion model. As evident from the progression of model training, the performance trends of the train set, test set, and synth set tend to converge at higher resolutions. However, while the training loss stabilizes, the mean noise magnitude, as depicted in Figure 3.3, exhibits greater divergence at higher pretrained resolutions. This suggests the possibility of overfitting occurring in the sampling algorithm as training epochs increase.

Another potential factor is inherent to the resolution and sampling methodology itself. Notably, we observe that the mean noise magnitude in 64 × 64 higher resolutions exhibits greater similarity compared to

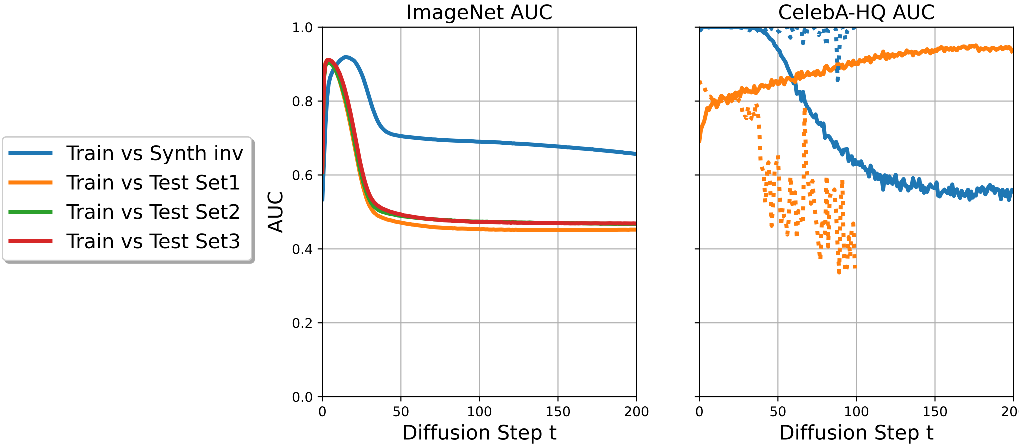

Figure 3.2: AUC Comparison Across First 200 Diffusion Steps for Different Datasets at 256 × 256 resolution. **Left image:** Performance on ImageNet dataset, with three Test Sets representing the Test Matched Frequency set , Test Threshold0.7 set, and Test Top Images set in ImageNetv2; **Right image:** Performance on CelebA-HQ dataset, with Test Set1 representing the FFHQ set. SecMI baselines as **dashed** (- -) curves for their respective experimental groups, which are denoted by color.

resolutions like 256 × 256 than that in 32 × 32 resolution, also shown in Figure 3.3. At higher resolutions, real-world images may contain continuous color blocks where colors transition smoothly or remain constant. However, the DDPM sampling method generates random noise from pure Gaussian noise, potentially resulting in less smooth transitions within these blocks.

3.6 Overfitting Problems

We conducted evaluations on different training epochs, training both the 32 × 32 and 64 × 64 models for 2000 epochs. Both models showed convergence of training loss and validation loss around the 800th epoch. In the table corresponding to the 800th epoch, setting t between 50 to 100 resulted in our method reaching an AUC of approximately 1. Additionally, for the higher resolution 64x64 model, perfect prediction was achieved around the 1200th epoch.

However, as illustrated in Figure 3.5, the mean noise magnitude exhibits greater variance with increasing training epochs. A decrease in magnitude within the train set alongside an increase within the test set suggests potential overfitting of the model. Furthermore, Figure 3.6, depicting the AUC across different training epochs, demonstrates an increase in AUC as epochs progress, indicating a widening difference between train and test or synthesis distributions, indicative of overfitting.

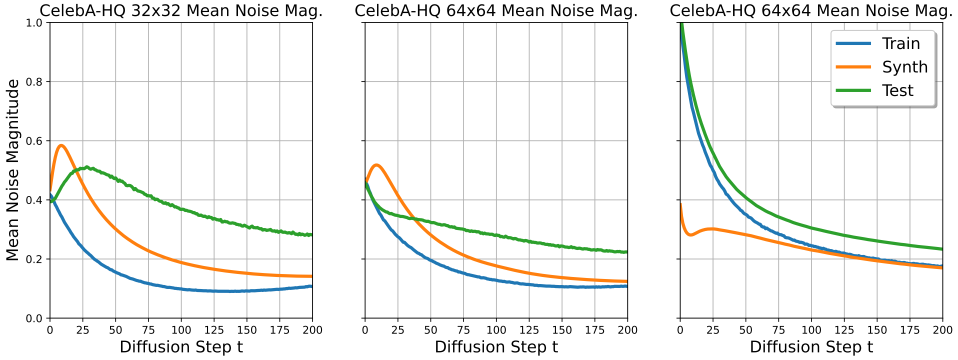

Figure 3.3: Comparison of Mean Noise Magnitude Across First 200 Diffusion Steps of CelebA-HQ at Various Data Resolutions. **Left image:** 32×32 resolution; **Middle image:** 64×64 resolution; **Right image:** 256×256 resolution. For 32×32 and 64×64 resolutions, the test set is part of the CelebA-HQ dataset. For 256×256 resolution, the test set is the FFHQ set.

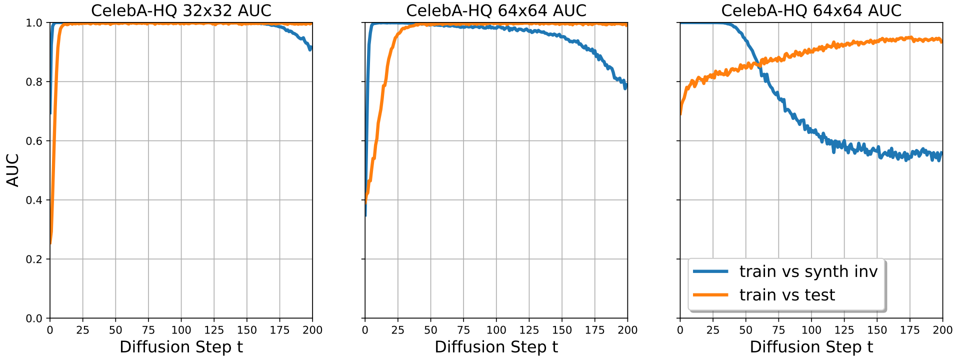

Figure 3.4: Comparison of AUC Across First 200 Diffusion Steps of CelebA-HQ at Various Data Resolutions. **Left image:** 32×32 resolution; **Middle image:** 64×64 resolution; **Right image:** 256×256 resolution. For 32×32 and 64×64 resolutions, the test set is part of the CelebA-HQ dataset. For 256×256 resolution, the test set is the FFHQ set.

Table 3.3: AUC Variation of Our Method on CelebA-HQ Across Different Training Epochs under 2000.

Epoch	Set	32×32				64×64			
		$t = 0$	$t = 10$	$t = 50$	$t = 100$	$t = 0$	$t = 10$	$t = 50$	$t = 100$
400	Test	0.51	0.54	0.64	0.69	0.50	0.49	0.56	0.63
	Synth	0.67	1.00	0.99	0.96	0.73	1.00	1.00	0.98
800	Test	0.42	0.57	0.95	0.99	0.46	0.49	0.79	0.93
	Synth	0.76	1.00	1.00	0.98	0.43	1.00	1.00	0.96
1200	Test	0.35	0.80	0.99	0.99	0.44	0.52	0.96	0.99
	Synth	0.69	1.00	1.00	0.99	0.48	1.00	1.00	0.99
1600	Test	0.29	0.97	1.00	1.00	0.41	0.62	0.99	1.00
	Synth	0.70	1.00	1.00	1.00	0.35	1.00	1.00	0.99
2000	Test	0.26	0.99	1.00	1.00	0.39	0.66	0.99	1.00
	Synth	0.69	1.00	1.00	1.00	0.35	1.00	0.99	0.98

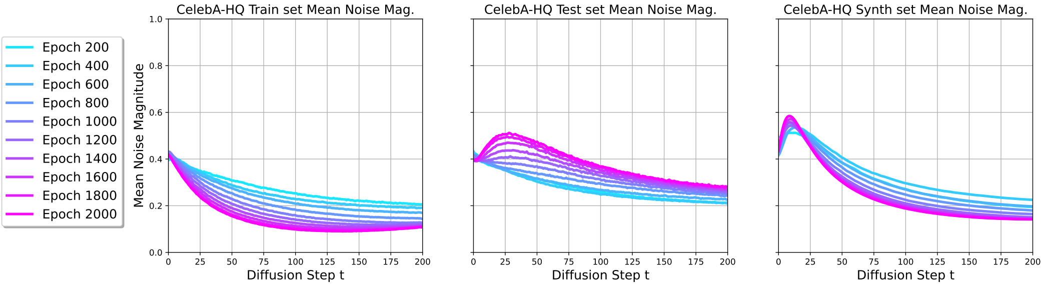

Figure 3.5: Comparison of Mean Noise Magnitude Across Diffusion Steps of CelebA-HQ in 32 × 32 Data Resolution Across Different Training Epochs. **Left image:** Evaluation on Train Set; **Middle image:** Evaluation on Test Set; **Right image:** Evaluation on Synth Set.

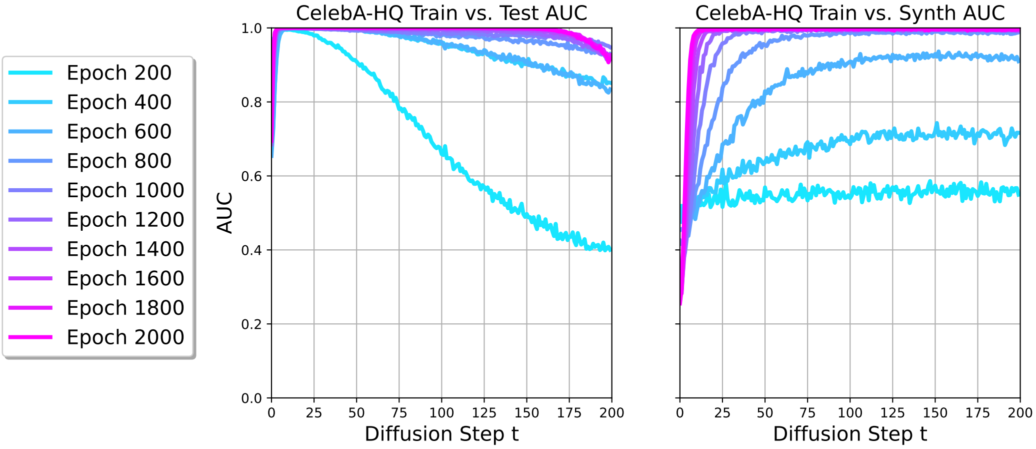

Figure 3.6: Comparison of AUC Across Diffusion Steps of CelebA-HQ in 32 × 32 Data Resolution Across Different Training Epochs. **Left image:** Train Set Compared with Test Set; **Right image:** Train Set Compared with Synth Set.

CHAPTER 4

Discussion

4.1 Hypotheses on driving mechanism: why does this work?

The intuition of loss based attacks is fairly clear: “The model may overfit to the training data, learning more about the specific dataset than the generating distribution. This disparity can be leveraged to identify the training set out of sets of real data points.” Because the proposed score is not the loss function of the diffusion model, it is technically not a loss-based attack, though it is obviously both related and similar in some aspects. Importantly, the score seems to be lower on training data than on test in many cases, and this difference is modulated by training. However, if it were purely an overfitting issue of the Diffusion model, our score function should approach zero across all diffusion times for increasingly overfit models. As shown in Figure 3.5, for $t \in [75, 150]$ the score drops progressively closer to zero as training progresses. However, for t approaching $t = 0$, the mean estimated noise value appears stable.

An alternative viewpoint of the same phenomenon can be understood from Alain and Bengio 2014 Alain and Bengio (2014), wherein they characterize the behavior of Denoising Auto-encoders (a result which remains applicable to the iterated diffusion steps of DDPM, and its reverse denoising process). They characterize the optimal denoising auto-encoder as essentially a convolution of the generating distribution with a mollifier/smoothing function of bandwidth equal to the noise distribution. They then extend this to cover the empirical data distribution case (a set of δ-distributions), where a denoising auto-encoder acts as a Parzen window/Kernel density estimator Parzen (1962) using Gaussian kernel functions. In this interpretation, our proposed score function is purposefully misspecifying the bandwidth parameter of the (DDPM provided) KDE.

For the CelebA-HQ/FFHQ experiments, this misspecification has noticeable effects between Train and Test even as we progress along the noise curve to $t = 200$, but seemingly vanishing effect when comparing the Train vs Synthetic datasets. It is tempting to hypothesize that the vanishing effect is due to inaccuracies of the sampling process being erased (smoothed away) at wider bandwidth parameters, but this is *not* reflected in behavior in ImageNet. While direct comparison of the noise schedule is incorrect here (the CelebA-HQ and ImageNet models are trained separately, with slightly different loss functions), it is clear that the ImageNet train and test scores converge rapidly, while the synthetic scores remain separate.

4.2 Comparison with Duan et al.Duan et al. (2023) 2023 (SecMI)

Duan et al. Duan et al. (2023) define a loss-based attack for Diffusion models. For a given output, they add noise up to some step t, then denoise a step, *re*-noise a step (by computing the estimated noise vector at the $t-1$ point and moving backward along that vector), and then compute error from their noised sample. This attack should in theory have the same characteristics of the loss function on training data.

Similar to our attack, this method uses an artificial timestep and the Diffusion model's own dynamics to produce a scoring function. It has very high fidelity with respect to synthetic identification, and performs well (0.85 AUC to our proposed method's 0.90 AUC) for CelebA-HQ train-test identification.

Unlike our attack, it recreates the denoising process (as expected of a loss-based attack); we believe that this is essentially querying the fitted density estimation at the trial points. While asymptotically this is equivalent to the loss, directly querying that density estimation function or its gradient (as per our method) should be more efficient. SecMI requires multiple more evaluations of the network, and uses steps along the estimated noise vector as a proxy for denoising and re-noising by a single time-step.

Surprisingly, the accuracy actually decreases with increased t, even over small intervals of t. This may be due to the strongest loss signals being closer to $t = 0$ (due to concentration of the Gaussian density about the datapoint as $t \rightarrow 0$).

4.3 Limitations

Due to computational limitations, we did not experiment with many of the popular variations of diffusion models; notably, both Variational Diffusion Models Kingma et al. (2021) and Latent Diffusion Rombach et al. (2022) use a stochastic output layer. We do not expect this to change behaviors significantly, though the Latent Diffusion embedding (initial encoder/decoder phase) may affect dimensionality-dependent behaviors.

It is also difficult to imagine an auditing membership inference system which will work with high fidelity using either our attack or the Duan et al. 2023 Duan et al. (2023) attack (SecMI). Both methods perform very well at synthetic identification, though their threat model becomes more complicated without knowledge of the exact generating method. Still, we believe both the loss based attacks and our own method have value in showing potential idiosyncrasies in current Diffusion models.

CHAPTER 5

Conclusion

In this paper we have presented a very simple method for membership inference and synthetic identification on Diffusion models, showing it to have high fidlity performance on two large datasets and multiple experimental conditions. Benefiting from its simplicity, the attack may be interpreted as exploiting Diffusion model structure in several ways, providing novel insight into the models themselves and their behavior as distribution estimators.

References

Alain, G. and Bengio, Y. (2014). What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593.

Chen, D., Yu, N., Zhang, Y., and Fritz, M. (2020). Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362.

De Bortoli, V., Thornton, J., Heng, J., and Doucet, A. (2021). Diffusion schrödinger bridge with applications to score-based generative modeling. *Advances in Neural Information Processing Systems*, 34:17695–17709.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., and Ferrer, C. C. (2020). The deepfake detection challenge (dfdc) dataset. *arXiv preprint arXiv:2006.07397*.

Duan, J., Kong, F., Wang, S., Shi, X., and Xu, K. (2023). Are diffusion models vulnerable to membership inference attacks? *arXiv preprint arXiv:2302.01316*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Haliassos, A., Vougioukas, K., Petridis, S., and Pantic, M. (2021). Lips don't lie: A generalisable and robust approach to face forgery detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5039–5049.

Hayes, J., Melis, L., Danezis, G., and De Cristofaro, E. (2017). Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851.

Jun, H. and Nichol, A. (2023). Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196.

Karras, T., Laine, S., and Aila, T. (2019). A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410.

Kingma, D., Salimans, T., Poole, B., and Ho, J. (2021). Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707.

Kong, X., Brekelmans, R., and Steeg, G. V. (2023). Information-theoretic diffusion. *arXiv preprint arXiv:2302.03792*.

Kong, Z., Ping, W., Huang, J., Zhao, K., and Catanzaro, B. (2020). Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*.

Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

Ma, J., Chai, L., Huh, M., Wang, T., Lim, S.-N., Isola, P., and Torralba, A. (2022). Totems: Physical objects for verifying visual integrity. In *European Conference on Computer Vision*, pages 164–180. Springer.

McAllester, D. (2023). On the mathematics of diffusion models. *arXiv preprint arXiv:2301.11108*.

Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Nguyen, D. T., Huynh-The, T., Nahavandi, S., Nguyen, T. T., Pham, Q.-V., and Nguyen, C. M. (2022). Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525.

Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., and Chen, M. (2022). Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*.

Nichol, A. Q. and Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR.

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

Poole, B., Jain, A., Barron, J. T., and Mildenhall, B. (2022). Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*.

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. (2022). Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3.

Recht, B., Roelofs, R., Schmidt, L., and Shankar, V. (2019). Do imagenet classifiers generalize to imagenet? *CoRR*, abs/1902.10811.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695.

Sabir, E., Cheng, J., Jaiswal, A., AbdAlmageed, W., Masi, I., and Natarajan, P. (2019). Recurrent convolutional strategies for face manipulation detection in videos. *Interfaces (GUI)*, 3(1):80–87.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al. (2022). Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. (2017). Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE.

Song, J., Meng, C., and Ermon, S. (2020a). Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2020b). Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*.

Verdoliva, L. (2020). Media forensics and deepfakes: an overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932.

Yang, L., Yu, Z., Meng, C., Xu, M., Ermon, S., and Cui, B. (2024). Mastering text-to-image diffusion: Recaptioning, planning, and generating with multimodal llms. *arXiv preprint arXiv:2401.11708*.

Yang, Z., Shao, B., Xuan, B., Chang, E.-C., and Zhang, F. (2020). Defending model inversion and membership inference attacks via prediction purification. *arXiv preprint arXiv:2005.03915*.

Yeom, S., Giacomelli, I., Fredrikson, M., and Jha, S. (2018). Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE.

Zhao, H., Zhou, W., Chen, D., Wei, T., Zhang, W., and Yu, N. (2021). Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2185–2194.