

Black-box Membership Inference Attacks against Fine-tuned Diffusion Models

Yan Pang
University of Virginia
yanpang@virginia.edu

Tianhao Wang
University of Virginia
tianhao@virginia.edu

Abstract—With the rapid advancement of diffusion-based image-generative models, the quality of generated images has become increasingly photorealistic. Moreover, with the release of high-quality pre-trained image-generative models, a growing number of users are downloading these pre-trained models to fine-tune them with downstream datasets for various image-generation tasks. However, employing such powerful pre-trained models in downstream tasks presents significant privacy leakage risks. In this paper, we propose the first scores-based membership inference attack framework tailored for recent diffusion models, and in the more stringent black-box access setting. Considering four distinct attack scenarios and three types of attacks, this framework is capable of targeting any popular conditional generator model, achieving high precision, evidenced by an impressive AUC of 0.95. Our code is accessible at <https://github.com/py85252876/Reconstruction-based-Attack>.

I. INTRODUCTION

The recent developments in image-generative models have been remarkably swift, and many applications based on these models have appeared. Diffusion models [1]–[10] have come to the forefront of image generation. These models generate target images by progressive denoising a noisy sample from an isotropic Gaussian distribution. In an effort to expedite the training of diffusion models and reduce training expenses, Stable Diffusion [8] was introduced. Leveraging the extensive and high-fidelity LAION-2B [11] dataset for training, the Stable Diffusion pre-trained checkpoint, available on HuggingFace, can be fine-tuned efficiently with just a few steps for effective deployment in downstream tasks. This model’s efficiency has spurred an increasing number of usages of Stable Diffusion.

At the same time, there has been a significant amount of research focused on the privacy concerns associated with these models, specifically those related to training data [12]–[16] and those related to model outputs [17]–[19]. Among them, membership inference attacks (MIAs) primarily investigate whether a given sample x is included in the training set of a specific target model. While this line of research was traditionally directed toward classifier models [20]–[28], the

popularity of diffusion models has led to the application of MIAs to examine potential abuses of privacy in their training datasets. Depending on the level of access to the target model, these attacks can be categorized into white-box attacks, gray-box attacks, and black-box attacks.

In a white-box attack scenario, attackers have access to all parameters of a model. Similar to membership inference attack targeting classifiers, attacks against diffusion models also utilize internal model information such as loss [13], [29], [30] or gradients [15] as attack features. Hu et al. [13] and Matsumoto et al. [30] have utilized losses at different timesteps of querying the model as attack features. Similarly, Carlini et al. [29] employed losses across various timesteps but incorporated the LiRA framework to construct two distributions for inferring the membership of a sample x . Pang et al. [15] took a different approach by using the model’s gradients at different timesteps as the attack features, positing that gradient information better reflects the model’s response to x .

Although white-box attacks can achieve high success rates, their limitation lies in the requirement for complete access to the target model’s information, which is often impractical in real-world scenarios. Compared with white-box attack, gray-box approaches do not require full access to the model’s parameters; instead, they only necessitate the intermediate outputs from the diffusion model during the denoising process to serve as features for inference [12], [13], [31]–[35]. For example, Duan et al. [12], and Kong et al. [32] have leveraged the deterministic nature of DDIMs, using the approximated posterior estimation error of intermediate outputs at different timesteps as attack features. Hu et al. [13] have proposed using intermediate outputs to estimate the log-likelihood of samples as attack features. However, these attacks inevitably rely on the intermediate images generated during the model’s operation. In real-world scenarios, if a malicious model is trained using private or unsafe images, typically only the final output image is provided, with efforts made to conceal as many model details as possible. Therefore, the more practical scenarios would be black-box.

There are also black-box attacks for GANs [36], [37] and VAEs [37]. These are based on *unconditional* generative models and involve a highly stochastic generation process that requires *extensive sampling* for inference, which becomes inefficient when directly applied to diffusion models. The other black-box attacks [14], [30], [38], [39], although more

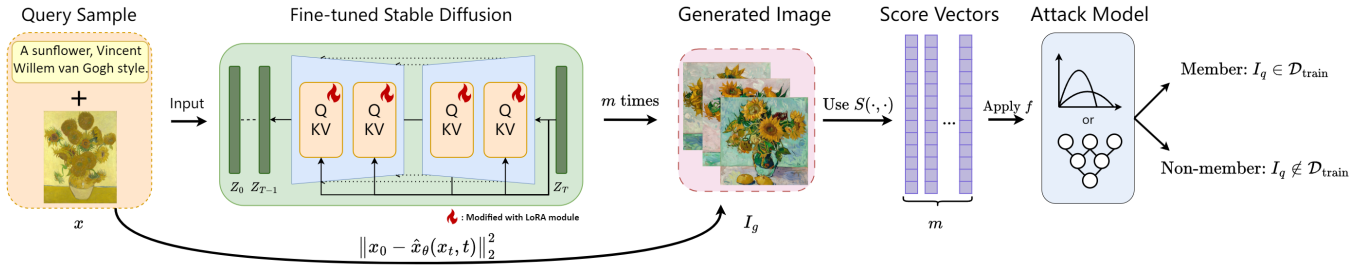


Fig. 1: Our attack takes the query sample x , which consists of an image I_q and a text component T_q , and applies T_q to query the model to get generated image I_g for m times. Then, we compute the similarity score between I_q and each I_g with $S(\cdot, \cdot)$. The m scores are then aggregated using f , and used to train the attack model to determine the membership.

tailored for diffusion models, focus on simulations and lack the necessary conditions to be used in realistic scenarios. We will discuss them in Section II-E.

In this paper, we present a black-box attack framework suitable for state-of-the-art image generative models, as shown in Figure 1. The framework was built on a careful analysis of the objective function of the diffusion model as its theoretical foundation, and compares the generated image and the query image. It also incorporates four potential attack scenarios tailored for different settings of diffusion models. We demonstrate the efficacy of our attack using the pre-trained Stable Diffusion v1-5 and further validate it fine-tuned with CelebA-Dialog [40], WIT [41], and MS COCO datasets [42].

Compared with existing black-box methods [30], [38], our attack under four attack scenarios can achieve 87% accuracy and outperform other methods by nearly 35%. We systematically evaluate all components, including the image encoder, distance metrics, inference steps, and training set sizes. Our method is able to achieve high ROC-AUC across three datasets: 0.95, 0.85, and 0.93. Even using different types of generative models as shadow models to employ the attack, our attack still can obtain at least 83% success rate for four attack scenarios on three datasets. The results show that our attack is robust and fit for real-world requirements. To further comprehensively evaluate our attack, we employed DP-SGD [43] as a defensive strategy to assess the attack’s effectiveness. By reducing the model’s ability to memorize training samples, DP-SGD defends against our attack. This finding is consistent with the outcomes observed in other attacks [12], [13], [29].

Contributions: We make the following contributions.

- Many prior black-box attacks [14], [30], [36]–[39] on image-generative models are no longer practical for the current generation of models and attack scenarios. We propose a black-box membership inference attack framework that is deployable against any generative model by leveraging the model’s memorization of the training data.
- Consistent with the definition in Suya et al. [44], four attack scenarios are considered in which an attacker can perform an attack based on the *query access* as well as the *quality of the initial auxiliary data*, and three different attack models are used to determine the success rate of the attack, respectively.

- The efficacy of the attack is evaluated on the CelebA, WIT, and MS COCO datasets using fine-tuned Stable Diffusion v1-5 as the representative target model. The attack’s impact is analyzed by considering various factors: image encoder selection, distance metrics, fine-tuning steps, inference step count, member set size, shadow model selection, and the elimination of fine-tuning in the captioning model.

Roadmap. Section II reviews key works on denoising generative models and membership inference attacks, including their application against diffusion models. Section III introduces our score-based black-box attack on diffusion models, tailored to four levels of attacker knowledge. Section IV describes our experimental setup, and Section V compares our attack’s effectiveness with existing methods and examines various influencing factors. Section VI shows the effectiveness of our attacks against common defenses. Section VII discusses some other research related to our work. Section VIII concludes the paper, summarizing our main findings and contributions.

II. BACKGROUND

A. Machine Learning

In general, we can classify a machine learning model into discriminative (classification) models and generative models.

1) *Classification Models:* In the context of classification model training, the objective is to map an input x to a category y . The functional representation of the model can be expressed as $y = \mathcal{M}(x)$, where x denotes the input (e.g., an image), \mathcal{M} represents the classification model, and y denotes the corresponding label. The loss in the classification model, which quantifies the discrepancy between the predicted and true labels, can be articulated as follows:

$$L(\theta) = \mathbb{E}_{x,y} [-\log(\mathcal{M}(x)_y)]$$

where θ denotes the parameters of \mathcal{M} , $\mathcal{M}(x)$ denotes the model’s output probability distribution over the possible categories, and $\mathcal{M}(x)_y$ specifically denotes the probability assigned to the correct label y .

2) *Generative Models:* Generative models are designed to generate $\hat{x} = \mathcal{G}(z)$, where z is the randomness not provided by users but inherent to the server hosting the generator \mathcal{G} .

Popular generative models include VAEs [45], GANs [46], and diffusion models [2]. Recently, diffusion models have

gained significant traction. Building on the classical DDPM (Denoising Diffusion Probabilistic Models), a plethora of models, such as Imagen [5], DALL-E 3 [47], GLIDE [3], Stable Diffusion [8], have emerged and can generate high-quality images based on prompt information. In this paper, we mainly focus on diffusion models.

B. Diffusion Models

1) *Foundation of Diffusion Models*: The diffusion model can be conceptualized as a process where a noisy image is incrementally denoised to eventually yield a high-resolution image. Given an image x_0 , the model initially imparts noise via T forward (noisy-adding) processes. At timestep t , the noisy image x_t can be represented as:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t \quad (1)$$

where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, and α_i is a predefined parameter that decreases incrementally within the interval $[0, 1]$. The term ϵ_t is a random Gaussian noise derived using the reparameterization trick from multiple previous forward steps (more details in Appendix A).

The reverse process serves an objective opposite to that of the forward process. Starting from $\hat{x}_T = x_T$, upon obtaining the image \hat{x}_t at timestep t , the reverse process aims to denoise it to retrieve the image \hat{x}_{t-1} . A neural network (e.g., U-Net) \mathcal{U}_θ is trained to predict the noise to be removed at each timestep. The loss function in the training process is defined as:

$$L_t(\theta) = \mathbb{E}_{x_0, \epsilon_t} [\|\epsilon_t - \mathcal{U}_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon_t, t)\|_2^2] \quad (2)$$

Alternatively, this loss function can also be employed to train DDIM [10], which has a deterministic reverse process.

2) *Prompt Guided Diffusion Models*: Diffusion models [3]–[5], [8], [47] mentioned above are also capable of generating high-quality images conditional on prompt information p , denoted as $\hat{x} = \mathcal{G}(z, p)$ (further details can be found in Appendix B). Our experiments primarily utilize the current publicly available state-of-the-art model, Stable Diffusion [8]. Distinct from other diffusion generative models [3]–[5], Stable Diffusion uniquely conducts both the forward and reverse processes within the latent space (images simplified into lower-dimensional data). This approach offers advantages: the noise addition and removal processes operate over a smaller dimensionality, allowing for faster model training at lower computing costs. Additionally, within the latent space, the model can accommodate diverse prompt information to guide image generation. Importantly, Stable Diffusion is open-sourced and provides multiple high-quality pre-trained checkpoints online. This aligns well with the focus of our study on potential privacy concerns when fine-tuning pre-trained models for downstream tasks.

C. Membership Inference Attacks

Membership inference attacks (MIAs) primarily aim to determine whether a target data point x is within the training dataset, often referred to as the *member set*, of a given target model. The motivation behind these attacks is twofold: to

ensure that models are not trained in a manner that misappropriates data and to safeguard against potential privacy breaches. MIA’s underlying principle hinges on exploiting machine learning models’ overfitting and memorization properties. Discerning the model’s different reactions to member and non-member samples makes it feasible to infer the membership of the target point x .

To formalize membership inference attacks, assume there is a data sample x , a model \mathcal{M}_θ trained with dataset \mathcal{D}_m . The attack \mathcal{A} will access \mathcal{M}_θ , \mathcal{D}_m and take data sample x as input. It will then output a bit $b \leftarrow \mathcal{A}^{\mathcal{D}_m}(x, \mathcal{M}_\theta) \in \{0, 1\}$ indicating whether x was used in training (i.e., $x \in \mathcal{D}_m$) or not. For simplicity, we use θ denoted model \mathcal{M}_θ and omit \mathcal{D}_m .

Early MIAs predominantly target classification models and use the outputs from classifiers as the data to train their attack models [21], [23]–[25], [48]–[51]. Shokri et al. [25] introduced a technique for training shadow models designed to use shadow models to approximate the target model’s behavior. By collecting information from these shadow models, such as prediction vectors or training loss, as well as membership labels (e.g., members vs. non-members), adversaries can subsequently train a binary classifier. This classifier acts as an attack model to predict the membership of x based on the data derived from querying x on the target model.

Carlini et al. [52] argued that using loss as an attack feature is inadequate and constitutes a non-membership inference attack. Instead, the likelihood-ratio attack can serve as a better method. They first created two distributions, \mathbb{D}_{in} and \mathbb{D}_{out} , based on the confidence scores of samples from the member and non-member sets, respectively. Then, the distributions are used to calculate the probability density function of query data x in the member set and non-member set.

MIAs against Diffusion Models. In the context of MIA against diffusion models, due to the structural differences between diffusion models and classification models, as well as the dissimilarities in their inputs and outputs, MIAs designed for classification models cannot be directly applied to diffusion models. The focus of the research lies in how to construct features for MIA. We classify existing attacks against diffusion models as white-box, gray-box, and black-box, and introduce them separately. In white-box attacks, methods in this setting exploit the loss (derived from each timestep using Equation 2) and gradients (via backpropagation through the model). Gray-box attacks typically necessitate access to a model’s intermediate outputs but do not require any internal model information. For gray-box attacks targeting diffusion models, the model’s denoising trajectory, particularly the noisy images, is utilized as attack data. In contrast, black-box attacks operate without knowledge of the model’s internal mechanics or process outputs, relying solely on the final generated images for analysis. In Table I, we compare all existing attacks. Each type of attack’s details is deferred to Section II-E1 and Section VII.

TABLE I: The symbols \circ \bullet and \bullet represent an attacker’s fully authorized, partially authorized, and unauthorized data access, respectively. Symbols \checkmark and \times denote the use and non-use of a technique, respectively. ‘HP’: stands for the model’s parameter settings. ‘TD’: training data used to train the target model. ‘IV’: model’s internal values, including loss and gradient. ‘IO’: internal outputs (noisy images). ‘TSC’: components (text and image) of the target sample. ‘SMs’: whether the attack employs shadow models.

	Method	HP	TD	MIV	IOs	TSC	SMs
White	Loss-based [13]	\circ	\circ	\circ	\circ	\circ	\times
	LiRA [29]	\circ	\circ	\circ	\circ	\circ	\checkmark
	LOGAN [30]	\circ	\circ	\circ	\circ	\circ	\times
	GSA [15]	\circ	\bullet	\circ	\circ	\circ	\checkmark
Gray	SecMI [12]	\bullet	\bullet	\bullet	\circ	\circ	\checkmark
	PIA [32]	\circ	\circ	\bullet	\circ	\circ	\times
	PFAMI [31]	\circ	\bullet	\bullet	\circ	\circ	\checkmark
	DRC [34]	\circ	\circ	\bullet	\circ	\bullet	\times
	CLiD [35]	\bullet	\bullet	\bullet	\circ	\bullet	\checkmark
	Structure-Based [33]	\circ	\circ	\bullet	\circ	\circ	\times
Black	GAN-Leaks [30]	\bullet	\circ	\bullet	\bullet	\circ	\times
	Intuition-attack [14]	\bullet	\circ	\bullet	\bullet	\circ	\times
	Pixel-attack [39]	\bullet	\circ	\bullet	\bullet	\circ	\times
	Distribution-attack [38]	\bullet	\bullet	\bullet	\bullet	\circ	\times
	Our Attack-I	\bullet	\bullet	\bullet	\bullet	\circ	\checkmark
	Our Attack-II	\bullet	\bullet	\bullet	\bullet	\bullet	\checkmark
	Our Attack-III	\bullet	\bullet	\bullet	\bullet	\circ	\checkmark
	Our Attack-IV	\bullet	\bullet	\bullet	\bullet	\bullet	\checkmark

D. Problem Formulation

1) *Threat Model*: Given the query sample x and black-box access to the target image-generative model \mathcal{G} , the goal of the attacker is to determine whether x was used to train \mathcal{G} . More specifically, we focus on the *fine-tuning* process, namely, we care about the privacy of the fine-tuning dataset of \mathcal{G} , and do not care about the pre-training dataset. We focus on fine-tuning because (1) the attacks will be similar for direct training, while the computational cost for experiments on fine-tuning MIA will be much smaller, and (2) the pretraining and fine-tuning paradigm is more popular with modern large models, and we will provide motivations for focusing on the fine-tuning process later.

We categorize the threat model into four scenarios (as shown in Table I) with two dimensions, namely:

- **Target Sample Component.** One distinct property of the current image generator models is that there exists the flexibility to input text prompt to guide model generation. Two configurations for the attacker can be considered: First, the query data x aligns with the training data as a text-image pair ($x = \langle T_q, I_q \rangle$, where T_q denotes the text component and I_q denotes the corresponding image component). Second, the attacker only obtains a suspect image potentially revealing private information without a corresponding caption ($x = \langle I_q \rangle$). As our focus here is MIAs on text-to-image generative models, the scenario where x solely consists of text is not deemed practical and hence, is not discussed.
- **Auxiliary Dataset.** Similar to all other MIAs, we assume

TABLE II: Results of utilizing our attack as an auditing tool. ‘Member’ refers to the similarity between generated images mimicking a specific artist or style and the actual works of that artist or style. The ‘Non-member’ sample similarity score is calculated by querying the model with samples not seen during fine-tuning and comparing them to their ground-truth images. ‘Diff.’ refers to the difference in similarity scores between member samples and non-member samples. Additionally, the attack results demonstrate the effectiveness of the attack using a shadow model trained on an auxiliary dataset. The similarity scores are calculated as an average of 3 generated images. The art style with the highest attack accuracy is highlighted in bold.

Art Style (Artist)	Member	Non-member	Diff.	ROC-AUC
Vincent van Gogh	0.92	0.44	0.48	0.91
Baishi Qi	0.77	0.40	0.37	0.88
Ukiyo-e	0.70	0.45	0.25	0.87
Uemura Shoen	0.88	0.41	0.47	0.89
Wanostyle	0.80	0.49	0.31	0.87
Ken Kelly	0.80	0.47	0.33	0.89
Shanshui Painting	0.88	0.47	0.41	0.86

an auxiliary dataset \mathcal{D}' is available. It is used to train the shadow models \mathcal{G}^s to mimic the behavior of the target model. We consider two scenarios for \mathcal{D}' , indicating whether the auxiliary dataset overlaps with the training set of the target model. In the case of the first scenario, \mathcal{D}' contains 50% real samples that were used to fine-tune the target model. In contrast, the second scenario represents \mathcal{D}' is sampled from the same distribution as the target model’s training set but without any overlap.

2) *Motivation*: Our work aims to reveal privacy infringements in the datasets used for fine-tuning image generative models. Given the full open-source nature of the Stable Diffusion [8], and the extensive availability of pre-trained models capable of generating photorealistic images from entities like CompVis¹ and Stability AI², there has been an increasing trend of leveraging these pre-trained models for fine-tuning to specific downstream tasks. Furthermore, an increasing number of companies, such as Amazon³, OctoML⁴, and CoreWeave⁵, are offering services in this domain. The data privacy issue during the training of these downstream tasks has not been explicitly studied. Our work seeks to uncover privacy violations in this process and raise awareness of them.

3) *Case Study*: To better demonstrate the risks of data misuse during the fine-tuning process and the ability of models to steal artistic styles, we designed a simple case study in this section. In the study, we collected seven models from Civitai⁶, a website that shares fine-tuned models. These models were utilized to generate images that mimic the styles of specific artists. We define synthesized samples that mimic the

¹<https://huggingface.co/CompVis/stable-diffusion>

²<https://github.com/Stability-AI/generative-models>

³<https://aws.amazon.com/sagemaker/jumpstart/>

⁴<https://octoml.ai/blog/the-beginners-guide-to-fine-tuning-stable-diffusion/>

⁵<https://docs.coreweave.com/cloud-tools/argo>

⁶<https://civitai.com/>

same art style as the fine-tuning data as ‘Member’ and those unrelated to the fine-tuning set as ‘Non-member’. The ‘Member’ similarity score is calculated by comparing the generated images with artworks by the same artists. In contrast, the ‘Non-member’ similarity score is computed by querying the model with data not used during fine-tuning and measuring the similarity between the generated and the ground-truth images. We present the ‘Member’ and ‘Non-member’ similarity scores for each of the seven models in Table II. In the case study, Cosine similarity served as the distance metric, and DeiT was employed to extract the features. For each model, we collected 60 samples for both member and non-member categories. Each sample was queried three times, and the average of these three similarity scores was used as the final similarity score for that sample.

According to Table II, the similarity score of ‘Member’ is at least 0.25 higher than ‘Non-member’ samples. In some cases, such as when imitating Vincent van Gogh’s works, the difference in similarity between ‘Member’ and ‘Non-member’ samples can reach up to 0.48. The distinct difference in similarity scores demonstrates that the model can copy the relevant artistic style after being trained on an artist’s work.

4) *Auditing Tool*: From the perspective of artists, it is concerning that models can steal their artistic styles after being trained on their works. To protect their copyright, artists need an auditing tool to detect suspicious models that may have used their work without authorization. Such an auditing tool is crucial today, as websites like Civitai already contain many fine-tuned models capable of imitating artistic styles and anime characters. Our work is based on the model’s memorization of samples during training. By incorporating the objective function of diffusion models (more details can be found in Section III), we designed a score-based membership inference attack. As observed in Table II, the significant difference between the ‘Member’ and the ‘Non-member’ similarity score suggests that our attack can serve as an auditing tool to detect potential misuse of training data.

To evaluate the feasibility of our attack as an auditing tool and ensure consistency with subsequent experiments, we used the shadow model technique to attack each target model (model from Civitai). We set the auxiliary dataset does not overlap with the member and non-member sets of the target model. Additionally, the sizes of all member and non-member sets are identical. Then, we trained an MLP as the attack model using the shadow model’s member and non-member data, and we present the attack ROC-AUC in Table II. The preliminary results of these attacks undoubtedly demonstrate the feasibility of our approach.

E. Existing Solutions

1) *Black-box MIA against Traditional Image-generative Models*: There are existing black-box MIAs targeting VAEs and GANs. They share a similar underlying idea, which is that if the target sample x was used during training, the generated samples would be close to x . Monte-Carlo attack [37] invokes the target model many times to generate many samples first.

Given x , it measures the number of generated samples within a specific radius. The more samples there are, the higher the likelihood that x is part of the member set.

GAN-Leaks [36] employs a similar intuition, using the shortest distance of the generated samples from the target sample as the criterion. It also proposes another attack assuming an extra ability to optimize the noise input z to the generator (which is not strictly the black-box setting; we will describe it and compare with it in the evaluation) so it can reduce the number of generated samples. More formal details about these attacks are deferred to Appendix D.

The reason we cannot apply Monte-Carlo attack [37] and GAN-Leaks [36] to diffusion models is that both attack methods require the model to sample a large number of images. Diffusion models progressively denoise during the inference process, involving dozens of steps, unlike VAEs and GANs, which require only a single step. Both Monte-Carlo attack and GAN-Leaks need to construct 100K samples to achieve optimal attack performance [36]. For the diffusion model, this will take even hundreds of times longer in terms of computing time. Furthermore, these attacks are unsuitable for conditional generative models. Although GAN-Leaks proposed a partial-black attack, conditional embedding (e.g., text embedding) in diffusion models is significantly more complex than the initial noise z in GANs and VAEs. Therefore, traditional black-box MIAs are not feasible for current diffusion models.

2) *Black-box MIA against Recent Diffusion Models*: Matsumoto et al. [30] directly adopted the concept of GAN-Leaks [36] to diffusion models. However, as diffusion models are more complex, the attack is bottlenecked by the time required to sample a large number of samples.

Wu et al. [14] leveraged the intuition that the generated samples exhibit a higher degree of fidelity in replicating the training samples, and demonstrate greater alignment with their accompanying textual description. However, the authors did not use the shadow model technique and only tested their attack on off-the-shelf models with explicitly known training sets. In the realistic setting where the training set is unknown (which is the purpose of MIAs), their attack cannot work.

Dubinski et al. [39] designed their attack against API-based generative machine learning services (e.g., Midjourney⁷) by directly comparing the pixel-level error between generated samples and known training samples. However, similar to Wu et al. [14], they did not use the shadow model technique in the black-box scenario (they trained shadow models in white-box settings) and assumed the attacker already knew the training dataset (LAION Aesthetics v2.5+), using it as the member set. This attack assumes the attacker can access excessive information, making it impractical in real-world scenarios.

Additionally, Zhang et al. [38] trained a classifier based on samples generated by the target model (labeled 1) and samples not used in training (labeled 0). The classifier can then determine whether the target sample was used in training. However, it needs to (1) know the non-training samples, and

⁷<https://www.midjourney.com/home>

(2) ensure the two distributions (of generated samples and non-training samples) are different enough. Both conditions are not necessarily true in a realistic setting.

III. METHODOLOGY

In this section, we introduce our attack, which is based on the model’s memorization of training samples. Current black-box membership inference attacks on generative models, such as GAN-Leaks [36] and Monte-Carlo attacks [37], also exploit this characteristic. However, GAN-Leaks relies on Parzen window density estimation to estimate the probability of query samples [53] that belong to the training set. This method often results in unstable probability estimates due to the large sampling size, as we mentioned in Section II-E1. We propose utilizing the *intrinsic characteristics of diffusion models with formal proofs* to design a more efficient and suitable attack for diffusion models. Specifically, we leverage the training objective of diffusion models to more directly and intuitively quantify the model’s memorization of query samples using similarity scores. Based on the results of the similarity score analysis, we determine the membership of query samples.

A. Theoretical Foundation

We aim to establish a detailed theory demonstrating the similarity score between the query image I_q and generated image I_g can be used as a metric to infer the membership of x . It is important to note that a high similarity score between I_q and I_g indicates a low distance between the two images. We leverage the internal property of the diffusion model, which is inherently structured to optimize the log-likelihood: If x is in the training set, its likelihood of being generated should be higher. However, due to the intractability of calculating log-likelihood in diffusion models, these models are designed to use the Evidence Lower Bound (ELBO) as an approximation of log-likelihood [2], as shown later in Equation 7. In Theorem 1, we first argue that ELBO of the diffusion model can be interpreted as a chain of generating images at any given timestep that approximates samples in the training set. Then, in Theorem 2, based on the loss function of the Stable Diffusion [8], we extend the result and demonstrate that this argument remains valid. Therefore, we can reasonably employ the similarity between the generated images and the query image as our attack. Note that GAN-Leaks [36] also shares this intuition of using similarity. However, it relies more on intuition and lacks a solid foundation, as the training of GANs is different (not a streamlined process as in diffusion).

From the perspective of the training process, we proposed these two theorems that facilitate our attack.

Theorem 1. *Assuming we have a pre-trained diffusion model \hat{x}_θ ⁸ with its training set \mathcal{D}_m , and use a bit b to represent the membership of query sample x (1 for member and 0 for non-member). The higher similarity scores between the query data*

⁸We previously use \mathcal{U}_θ to denote U-Net, now by slightly abusing notations we use \hat{x}_θ for easier presentations.

and its generated image $\hat{x}_\theta(x_t, t)$, the higher the probability of $\Pr[b = 1|x, \theta]$.

$$\Pr[b = 1|x, \theta] \propto -\|x_0 - \hat{x}_\theta(x_t, t)\|_2^2$$

where θ denotes the parameters of the model.

Proof: [Proof Sketch] We first demonstrate that diffusion models use ELBO to approximate the log-likelihood of the training dataset. By restructuring the optimization function, we find that the diffusion model primarily focuses on predicting the noise ϵ_t at t -th step. Using Equation 1, we show that the objective function of the diffusion model can also be expressed in terms of predicting \hat{x} at each step. Therefore, a data sample from the diffusion model’s member set is expected to have higher similarity with its replication $\hat{x}_\theta(x_t, t)$ at each step. As a result, the denoised sample from the diffusion model should naturally exhibit higher similarity scores with member set samples. The full proof can be found at Appendix C-A.

In the above, we have linked the probability of query sample x belonging to the member set to its similarity score with generated images in the unconditional diffusion model. For this type of diffusion model, although we can prove the training image has this property with its replica. We still cannot design the black-box attack on it because the inference process is random. We cannot control the unconditional diffusion model to reconstruct the specific data sample. This generation process is the same with VAEs and GANs. Hence, the existing black box attacks are to sample a large number of images from the models [30]. And then do the Monte Carlo [37] or GAN-Leaks [36] attack.

However, we can employ this property to execute the membership inference attack with conditional diffusion models (e.g., Stable Diffusion). The main difference between conditional and unconditional diffusion models is that the former can perform conditional generation. According to the prompt input, Stable Diffusion can generate an image that aligns with it. Therefore, we can use prompts to guide the model and synthesize images for a specific data sample. In Theorem 2, we prove this property valid in the Stable Diffusion.

Theorem 2. *For a well-trained Stable Diffusion model⁹, \hat{z}_θ ¹⁰, the query sample is x , and the membership of x is denoted as b (1/0 for member/non-member). D/E refers to the decoder/encoder module of the VAE in Stable Diffusion. A pre-trained text encoder, ϕ_θ , converts the input conditional prompt p into the text embedding that guides image generation. The similarity scores remain a viable metric for assessing the membership of query data x . This relationship can be expressed in the following mathematical formulation:*

$$\Pr[b = 1|x, \theta] \propto -\|D(z_0) - D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))\|_2^2$$

Where z_t represents the latent representation, $z_0 = E(x)$.

⁹In our work, we used the pre-trained Stable Diffusion-v1-5 from CompVis, which was trained for 150,000 A100 hours.

¹⁰ \hat{z}_θ represents only the U-Net in Stable Diffusion, excluding the VAE and text encoder.

Proof: [Proof Sketch] To establish [Theorem 2](#), we begin by examining the loss function of Stable Diffusion. We find that the optimization objective and the diffusion process in Stable Diffusion remain consistent with the unconditional diffusion model. However, the diffusion/denoising process is moving from the pixel level to the latent space. Through reinterpreting the noise prediction ϵ_t at each step, the optimization objective of Stable Diffusion can also be viewed as predicting the initial latent variable z_0 at each step. By incorporating the Decoder D, we prove that in Stable Diffusion, the member sample $D(z_0)$ should have a higher similarity score with its replicate $D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))$. The detailed proof of [Theorem 2](#) is presented at [Appendix C-B](#).

Considering the realistic situation and settings, we designed four attacks (as shown in [Section IV-C](#)) to use this property in different scenarios. However, for general representation, we simply denote the image generated by the model as I_g (which also corresponds to \hat{x} in [Section II-A2](#), $\hat{x}_\theta(x_t, t)$ in [Theorem 1](#), and $D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))$ in [Theorem 2](#)). The similarity score between I_g and I_q from the query data x can be represented as $S(I_q, I_g)$. Here, S is a distance metric (e.g., Cosine similarity, ℓ_1 or ℓ_2 distance, or Hamming distance). Given that a higher similarity score (low distance) indicates a higher probability of the data being a training sample, the inference model can be formulated accordingly.

$$\mathcal{A}_{base}(x, \theta) = \mathbb{1}\{S(I_q, I_g) \geq \tau\} \quad (3)$$

The base inference model relies on computing the similarity scores between I_g and I_q . If the similarity score $S(I_q, I_g)$ exceeds a certain threshold, the inference model will determine that the data record x associated with I_q comes from the member set.

B. Attack Pipeline

According to [Section III-A](#), our attack needs to calculate the similarity between query image I_q and generated image I_g . We choose to compute the image embedding similarity scores by using image feature extractors. Also, to execute our attack on query data that lacks text components, we incorporate the captioning model in our work. Our work seeks to uncover privacy violations in this process and raise awareness of them.

Image Feature Extractor. As we follow the high-level intuition of GAN-Leaks and use image similarities to determine membership, we employ distance metrics (e.g., Cosine similarity, ℓ_1 or ℓ_2 distance, or Hamming distance) to formally quantify this similarity. It has been observed that the semantic-level similarities are substantially more effective than pixel-level similarities [14]. Therefore, we utilize a pre-trained image encoder (i.e., DETR, BEiT, EfficientFormer, ViT, DeiT) to extract semantic representations from the images.

Captioning Model. In our work, under certain scenarios, the query data x may lack the text component T_q and only include I_q . Consequently, we resort to a captioning model to generate the corresponding text. For our experiments, we utilize BLIP2 [54] as the captioning model. To ensure that

Algorithm 1 High-level Overview of Our Attack.

Input: Query sample x , target model \mathcal{G} , distance metrics $S(\cdot, \cdot)$, the image captioning model \mathcal{C} , the instantiation of attack \mathcal{A} , the statistical function f , and the image feature extractor E .

- 1: **if** $T_q \notin x$ **then** ▷ Check for text components in x .
- 2: $T_q = \mathcal{C}(I_q)$ ▷ Synthesize the text for \mathcal{G} .
- 3: **end if**
- 4: **for** $i = 1$ **to** m **do** ▷ Perform m repetitive queries.
- 5: $I_g^i = \mathcal{G}(T_q)$
- 6: **end for**

Output: $\mathcal{A}(f[\langle S(E(I_q), E(I_g^i)) \rangle_{i=1}^m])$ ▷ MIA results.

the generated textual descriptions closely match the style of the model’s training dataset, we also consider further use of the auxiliary dataset to fine-tune the captioning model.

Attack Overview. [Algorithm 1](#) gives the high-level overview of our attack. The intuition is to compare the generated images with the query image and compute a similarity score used for MIAs (specific instantiations of \mathcal{A} to be presented in [Section III-C](#)). Depending on whether the text is available or not, we might need the captioning model to synthesize the text. Once the captioning is complete, we repeatedly query the target model m times for each query image, then apply a statistical function f (e.g., mean, median) to aggregate the m similarity score vectors for each query image. Finally, we return the aggregated similarity scores to determine the target/query data’s membership.

Note that while the attack pipeline is perhaps straightforward, its intuition relies on the formal analysis of the diffusion models. We first describe its theoretical foundation and then instantiate it with different MIA paradigms based on the output score in the following.

C. Instantiations

Utilizing the scores obtained from [Algorithm 1](#), we instantiate three different types of MIAs according to [Section II-C](#). In our evaluation, we try all three of them, and observe the last one is usually the most effective one.

Threshold-based Membership Inference Attack. Since the threshold-based MIA uses a scalar for comparison, the similarity scores obtained after applying f are calculated for each image patch (e.g., ViT generate 196 patches, more details in [Appendix E](#)). Therefore, to compute the overall image similarity, these similarity scores need to be averaged, i.e.,

$$\frac{1}{k} \sum_{j=1}^k f \left[\langle S(E(I_q), E(I_g^i)) \rangle_{i=1}^m \right]_j \geq \tau \quad (4)$$

Where k refers to the patch size used by the image feature extractors, S represents the distance metrics, and E denotes the image feature extractor. It is important to note that τ is determined in advance using member and non-member samples from the shadow model. Specifically, when the statistical

function f is mean, we calculate each query sample’s average feature similarity score, then average these scores across all patches and scale them (using Min-Max scaling [55]) to the range $[0, 1]$. After scaling, we use Youden’s index [56] to determine the best threshold τ that yields the highest AUC. This τ is then used to attack the target model.

Distribution-based Membership Inference Attack. Following the work by Carlini et al. [52], we know we can also use the likelihood ratio attack against diffusion models. In our analysis, we leverage similarity scores derived from shadow models to delineate two distinct distributions: \mathbb{Q}_{in} and \mathbb{Q}_{out} . Specifically: For \mathbb{Q}_{in} , consider image I that belong to the member set \mathcal{D}_m . We then define \mathbb{Q}_{in} as

$$\mathbb{Q}_{in} = \left\{ f \left[\left\langle S(E(I), E(I_g^i)) \right\rangle_{i=1}^m \right] \mid I \in \mathcal{D}_m \right\}.$$

Similarly, for \mathbb{Q}_{out} , when image I are part of the non-member set \mathcal{D}_{nm} , we have

$$\mathbb{Q}_{out} = \left\{ f \left[\left\langle S(E(I), E(I_g^i)) \right\rangle_{i=1}^m \right] \mid I \in \mathcal{D}_{nm} \right\}.$$

For target query point I_q , membership inference can be deduced by assessing:

$$\Pr \left[f \left[\left\langle S(E(I_q), E(I_g^i)) \right\rangle_{i=1}^m \right] \mid \mathbb{Q}_{in} \right]$$

and

$$\Pr \left[f \left[\left\langle S(E(I_q), E(I_g^i)) \right\rangle_{i=1}^m \right] \mid \mathbb{Q}_{out} \right]$$

Classifier-based Membership Inference Attack. Given that the obtained similarity score is represented as a high dimensional vector, the classifier-based MIA feeds $f \left[\left\langle S(E(I_q), E(I_g^i)) \right\rangle_{i=1}^m \right]$ directly into a classifier (we use a multilayer perceptron in our evaluation). This approach aligns with the methods of Shokri et al. [25], leveraging the machine learning model as the inference model to execute the attack.

In evaluation, although we can use different functions of f , we observe a simple f that takes the mean of all m similarity scores performs pretty stable, so we just use the mean function for all three MIAs throughout the evaluation.

IV. EXPERIMENT SETUP

A. Datasets

Stable Diffusion v1-5 is pre-trained on LAION-2B [11] and LAION-Aesthetics. To guarantee the integrity and effectiveness of our work, we utilize the MS COCO [42], CelebA-Dialog [40], and WIT datasets [41] for evaluation, ensuring that there is no overlap with the pre-training dataset. We label the samples in the member set as the positive class and the non-member samples as the negative class.

MS COCO is a large-scale dataset featuring a diverse array of images, each accompanied by five similar captions, amounting to a total of over 330k images. The MS COCO dataset [42] has been extensively utilized in various image generation

TABLE III: The default parameters used in Section V.

Parameters	Experiment setting for our work
Training data size	100
Epoch number	500
Resolution	512×512
Batch size	4
Learning rate	5×10^{-5}
Gradient accumulation steps	4
Inference step	30
Image feature extractor	DeiT
Captioning model	BLIP2
Distance metrics	Cosine similarity
Attack type	Classifier-based

models, including experiments on DALL-E 2 [4], Imagen [5], GLIDE [3], and VQ-Diffusion [57]. In this work, we randomly selected 50k images along with their corresponding captions to do the experiments. Each image is paired with a single caption to fine-tune the model.

CelebA-Dialog is an extensive visual-language collection of facial data. Each facial image is meticulously annotated and encompasses over 10,000 distinct entities. Given that each face image is associated with multiple labels and a detailed caption, the dataset is suitable for a range of tasks, including text-based facial generation, manipulation, and face image captioning. Facial information has consistently been regarded as private; hence, utilizing CelebA-Dialog [40] in this study aligns with our objective of detecting malicious users fine-tuning the Stable Diffusion model [8] for simulating genuine face generation.

WIT is a vast image-text dataset encompassing a diverse range of languages and styles of images and textual descriptions. It boasts 37.6 million image-text pairs and 11.5 million images, showcasing remarkable diversity. We leverage this dataset specifically to evaluate the robustness of our attack in handling such heterogeneous data.

B. Evaluation Metrics

To systematically evaluate the efficacy of our proposed attack, we opted for multiple evaluation metrics as performance indicators. Similar to other comparable attacks [12]–[14], [29], [30], [32], [52], we employ ASR (Accuracy of Membership Inference), Area Under the ROC Curve (AUC), and True Positive Rate (TPR) at low False Positive Rate (FPR) as our evaluation metrics. In Section V, all experiments are evaluated under the condition that the member set and non-member set have the same size.

We opted to use Stable Diffusion v1-5¹¹ checkpoints as our pre-trained models. The fine-tuning code script was modified from the Huggingface Diffusers package¹². All experiments were carried out using two Nvidia A100 GPUs, and each fine-tuning of the model required an average of three days. We presented the default fine-tuning and attack settings in Table III.

¹¹<https://huggingface.co/runwayml/stable-diffusion-v1-5>

¹²<https://huggingface.co/docs/diffusers/v0.9.0/en/training/text2image>

C. Baseline Attacks

For our evaluation, we first compare our work with existing black-box attacks on diffusion models [30], [31]. For our attack, based on the categorization provided in Section II-D1, the attacker will obtain information of two distinct dimensions, leading to four different scenarios. We call them Attack-I to Attack-IV. Below we introduce them in more detail.

Matsumoto et al. [30] employed the full-black attack framework from GAN-Leaks.

Zhang et al. [38] utilized a novel attack strategy involving a pre-trained ResNet18 as a feature extractor. This approach focuses on discriminating between the target model’s generated image distribution and a hold-out dataset, thereby fine-tuning ResNet18 to become a binary classification model.

Attack-I ($x = \langle T_q, I_q \rangle, \mathcal{D}' \cap \mathcal{D}_m \neq \emptyset$) In this attack scenario, we assume the attacker has access to partial samples from the actual training (fine-tuning) set of the target model (attacker’s auxiliary data \mathcal{D}' overlaps with the fine-tuning data \mathcal{D}_m). Furthermore, x includes both the image and the corresponding text (caption information). An attacker can directly utilize T_q to obtain I_g , then employ the similarity between I_g and I_q to ascertain the membership of x .

Attack-II ($x = \langle I_q \rangle, \mathcal{D}' \cap \mathcal{D}_m \neq \emptyset$) In this scenario, the attacker does not possess a conditional prompt that can be directly fed into the target model. The attacker needs to use an image captioning model to produce a caption for I_q . This caption is subsequently used as the input for \mathcal{G} . The process culminates in the computation of similarity between the query image I_q and the image generated by \mathcal{G} .

Attack-III ($x = \langle T_q, I_q \rangle, \mathcal{D}' \cap \mathcal{D}_m = \emptyset$) is similar to the first scenario (the difference is the attacker’s auxiliary dataset does not intersect with the target training dataset). The attack (as shown in Algorithm 1) is the same, but we expect a lower effectiveness.

Attack-IV ($x = \langle I_q \rangle, \mathcal{D}' \cap \mathcal{D}_m = \emptyset$) is similar to the third scenario (there is no overlap between the attacker’s auxiliary dataset and the target member set). This attack represents the hardest situation, and we think it will get the lowest accuracy.

V. EXPERIMENTS EVALUATION

A. Comparison with Baselines

Results are shown in Table IV. We ensure consistency in simulating real-world scenarios, wherein the number of images that a malicious publisher can sample from the target generator is limited. Under the constraint of limited sample size, we observe that the accuracy of both baseline attacks nearly equates to random guessing. We conjecture that this is due to their reliance on a large number of synthesis images for decision-making. Specifically, Zhang et al. [38] requires learning the distributional differences between generated image samples and non-member samples using ResNet18, based on a substantial volume of images sampled from the target model,

TABLE IV: Comparison between the attacks by Zhang et al. [38], Matsumoto et al. [30] (applying GAN-Leaks against the diffusion model) versus our methods. The best attack result is highlighted in bold.

Attack type	CelebA-Dialog		
	ASR	AUC	TPR@FPR=1%
Matsumoto et al. [30]	0.52	0.50	0.01
Zhang et al. [38]	0.51	0.49	0.01
Attack-I	0.85	0.93	0.53
Attack-II	0.88	0.93	0.60
Attack-III	0.87	0.94	0.54
Attack-IV	0.87	0.93	0.57

and subsequently applying this knowledge to assess the input query data. However, such an attack premise falters in realistic scenarios where a malicious model publisher restricts the number of images a user can obtain from the model, preventing attackers from sampling a large volume of images to conduct the attack. Under such constraints, the effectiveness of attacks by Zhang et al. [38] and others is inevitably compromised, as the insufficient sample size hampers the ability to accurately discern the differences between the two data distributions. Similarly, the approach by Matsumoto et al. [30] encounters a hurdle; in scenarios of limited generative sample availability, it becomes challenging to find a suitable reconstruction counterpart and calculate its distance from the original data record. Consequently, these methods fail to achieve high attack success rates under sample-restricted conditions. In contrast, the four attacks we propose still attain a high success rate despite the limited number of generative samples. This is attributed to our attacks being based on the similarity scores as proposed in Section III-A, which, while influenced by the quality of the model’s generated images, is not hindered by the quantity of these images.

B. Impact of Different Thresholds

In our work, we introduced three different types of attacks: *threshold-based*, *distribution-based*, and *classifier-based* attacks. The *threshold-based* attack is the most straightforward one. It does not require calculating means and variances to form distributions or training a classification model. We can directly use the τ obtained from the shadow model to determine the membership of samples in the target model. Since the *threshold-based* attack uses a one-dimensional threshold for judgment, it may reduce accuracy and become less stable to some extent. Therefore, before exploring other influence factors, we aim to validate whether the threshold obtained from the shadow model can be effectively used to attack the target model and to test the impact of different τ on attack accuracy.

From the results shown in Figure 2, it can be observed that using the shadow model to determine the best threshold τ allows for a successful attack on the target model. The threshold τ , calculated using sample data from the shadow model, is more effective at distinguishing between the target model’s member and non-member samples compared to other nearby values. This consistent result is evident across the

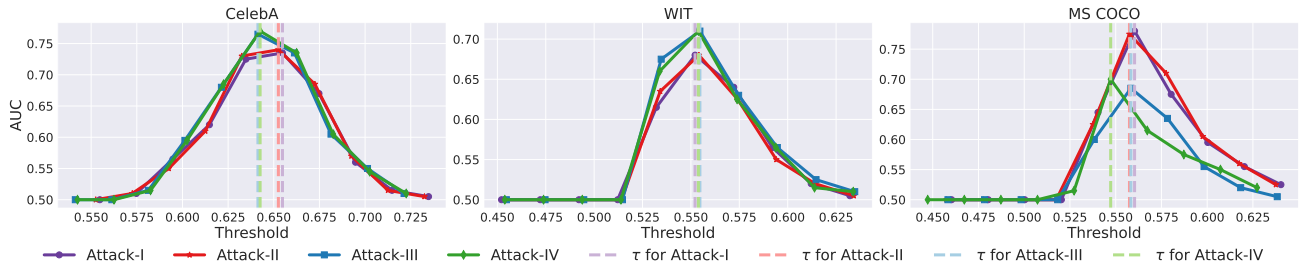


Fig. 2: Impact of different threshold values on attack results using the ROC-AUC metric across three datasets. Here, τ represents the best threshold selected for each attack from the shadow model based on the AUC scores.

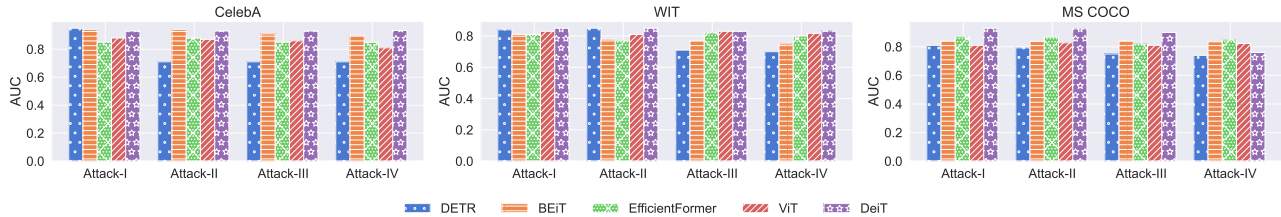


Fig. 3: AUC results on three datasets and four attack scenarios comparing five different image feature extractors.

TABLE V: The AUC scores of three attack types (*threshold-based*, *distribution-based*, *classifier-based*) across three datasets in four scenarios (**Attack-I**, **Attack-II**, **Attack-III**, **Attack-IV**) highlighting Cosine similarity’s superior and stable performance across all metrics and attack types. The best performance in each scenario is highlighted in bold.

Method	CelebA				WIT				WIT				
	ℓ_1	ℓ_2	Hamming	Cosine	ℓ_1	ℓ_2	Hamming	Cosine	ℓ_1	ℓ_2	Hamming	Cosine	
Threshold	Attack-I	0.30	0.33	0.71	0.88	0.39	0.38	0.73	0.69	0.40	0.37	0.78	0.84
	Attack-II	0.27	0.30	0.69	0.83	0.39	0.39	0.71	0.67	0.44	0.43	0.80	0.82
	Attack-III	0.30	0.34	0.74	0.82	0.40	0.40	0.67	0.69	0.40	0.39	0.79	0.84
	Attack-IV	0.41	0.43	0.48	0.82	0.43	0.47	0.77	0.66	0.37	0.38	0.77	0.74
Distribution	Attack-I	0.79	0.83	0.86	0.93	0.82	0.82	0.79	0.84	0.81	0.81	0.79	0.82
	Attack-II	0.83	0.82	0.83	0.93	0.80	0.83	0.77	0.82	0.82	0.81	0.78	0.81
	Attack-III	0.68	0.67	0.75	0.94	0.67	0.67	0.56	0.70	0.65	0.66	0.59	0.72
	Attack-IV	0.65	0.66	0.73	0.88	0.66	0.65	0.68	0.70	0.66	0.66	0.58	0.67
Classifier	Attack-I	0.74	0.84	0.85	0.93	0.73	0.75	0.78	0.82	0.73	0.77	0.76	0.86
	Attack-II	0.79	0.76	0.86	0.93	0.73	0.73	0.79	0.82	0.77	0.78	0.74	0.91
	Attack-III	0.81	0.75	0.83	0.94	0.70	0.71	0.78	0.79	0.53	0.73	0.80	0.83
	Attack-IV	0.77	0.73	0.82	0.93	0.75	0.74	0.70	0.79	0.52	0.62	0.78	0.82

three datasets included in the experiment. The AUC of our four attacks all exceeds 0.7, demonstrating the feasibility of threshold-based attacks. Moreover, the impact of different thresholds on attack performance is shown in Figure 2. We found that even with a deviation (e.g., 0.02), the τ obtained from the shadow model still achieves good attack accuracy on the target model.

C. Impact of Different Image Encoder

As our attack is a similarity scores-based attack, and we measure the distance between the query image I_q and the image I_g generated by the target model using the embeddings $E(I_g)$ and $E(I_q)$. However, due to the multitude of high-performance image encoder models, each with its unique pre-trained dataset and model architecture, we employed five distinct image feature extractors: DETR [58], BEiT [59], EfficientFormer [60], ViT [61], and DeiT [62]. Our goal was to observe the impact of various image features on the success rate of attacks by generating image embeddings from these models. The extractor yielding the highest success

rate will be selected as the default image feature extractor for subsequent experiments.

As depicted in Figure 3, our five image feature extractors excel across four different attack scenarios within the *classifier-based* attack domain. Each maintains an AUC score exceeding 0.7, underscoring the robustness of our attack framework across different feature extractors. Notably, the implementation of DeiT [62] as the feature extraction model yielded a marginally higher and more consistent success rate compared to the other image encoders. Therefore, we selected DeiT as the default image encoder for future experiments.

A more comprehensive comparison including *threshold-based* and *distribution-based* of these five image encoders is presented in Appendix G.

D. Impact of Different Distance Metrics

In the previous section, we picked DeiT [62] as the most stable and efficient image feature extractor. However, our attack framework also necessitates a reliable and consistent distance metric to compute the similarity score between embeddings. We conducted systematic and extensive experiments,

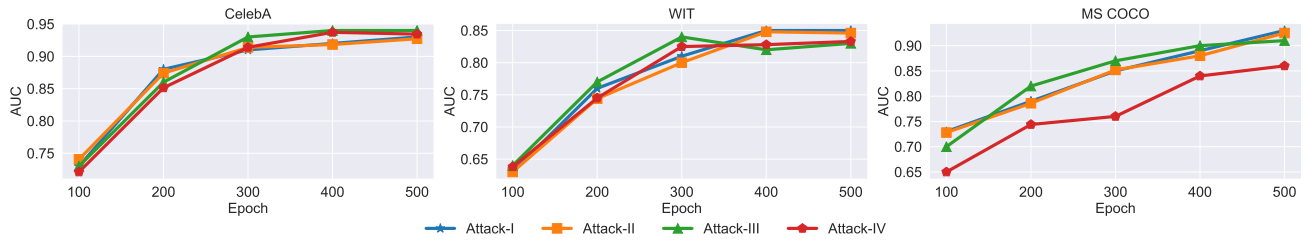


Fig. 4: Relationship between epoch progression and AUC score in **Attack-I**, **Attack-II**, **Attack-III**, and **Attack-IV**, indicating increasing memorization within image generation models over fine-tuning epochs.

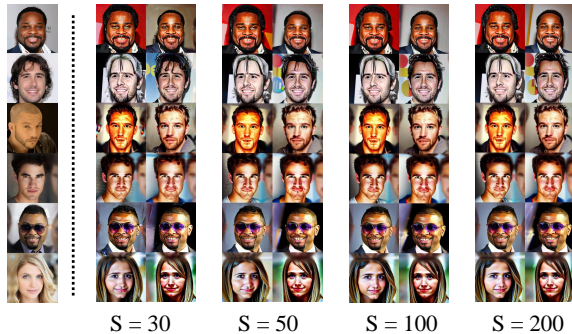


Fig. 5: The inference steps of 30, 50, 100, and 200 showed no noticeable differences in the overall structure of the generated images. Only subtle details, such as hair, exhibited variations.

and as demonstrated in **Table V**, we thoroughly assessed various attack scenarios and types across all datasets to test their effects on Cosine similarity, ℓ_1 distance, ℓ_2 distance, and Hamming distance.

From **Table V**, it is evident that using Cosine similarity as the distance metric yields optimal results for the computed distance vector, regardless of the attack scenario and type employed. We hypothesize that this phenomenon can be attributed to the focal point of our computation: the image embedding vectors generated by the encoder for both I_q and I_g . Cosine similarity is inherently adept at measuring the similarity between two vectors. In contrast, ℓ_1 and ℓ_2 norms are more suitable for quantifying pixel-level discrepancies between I_q and I_g , making them less efficient for evaluating the distance between two vectors.

E. Impact of Fine-tuning Steps

We then investigated the influence of the number of fine-tuning steps on the success rate of attacks. Evaluations were conducted at intervals of 100 epochs, ranging from 100 to 500 epochs, to measure the attack success rate. The default image encoder and distance metrics are Deit and Cosine similarity; all fine-tuning settings are aligned with **Table III**. As the model’s memorization of the training data can be equated to overfitting effects, it is anticipated that with an increased number of fine-tuning steps, the model increasingly exhibits a tendency towards overfitting and enhanced memorization of the training samples. Consequently, when querying the model with member set samples compared to non-member samples, a more distinct similarity discrepancy should be observed.

TABLE VI: Alignment with DDIM [10] denoting ‘S’ as inference steps; experimentation under **Attack-III** scenario measuring FID at varying inference step counts. The results show the inference steps did not affect attack performance. For each type of attack, we highlight the optimal attack result corresponding to each evaluation metric.

S	Threshold-based			Distribution-based			Classifier-based			FID
	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	
30	0.75	0.8225	0.30	0.76	0.8816	0.50	0.865	0.93	0.54	8.77
50	0.765	0.8146	0.25	0.77	0.8920	0.37	0.85	0.93	0.58	7.835
100	0.74	0.8172	0.26	0.745	0.8818	0.40	0.855	0.94	0.61	7.527
200	0.745	0.8125	0.39	0.74	0.8869	0.49	0.87	0.94	0.58	7.472

In **Figure 4**, we present the results of the *classifier-based* attacks under four attack scenarios: **Attack-I**, **Attack-II**, **Attack-III**, and **Attack-IV**. The outcomes indicate that **Attack-I** and **Attack-III** achieve higher success rates compared to the other two scenarios. This can be attributed to the fact that when utilizing the query data sample x , it inherently comprises the text caption T_q . As a result, neither **Attack-I** nor **Attack-III** require the employment of a caption model to generate corresponding text descriptions based on I_q . This circumvents the introduction of additional biases that could cause discrepancies between the model-generated images and I_q itself.

We have also included the results for *threshold-based* and *distribution-based* attacks under these four scenarios in the **Appendix H** for reference.

F. Impact of Number of Inference Step

The quality of images generated by current diffusion models, including the Stable Diffusion [8] presented in our work, is influenced not only by the number of fine-tuning steps but also by the number of inference steps. These models predominantly utilize DDIM [10] as their sampling method. The Fréchet Inception Distance is able to shift moderately from 13.36 to 4.04 when varying the sampling steps from 10 to 1000. This change highlights the capability of a higher number of inference steps to produce images of superior quality. Given that the foundation of our attack relies on the distance between generated and original images, we posit that an increased number of inference steps, which results in images closely resembling the original and of better quality, would correspondingly enhance the attack’s success rate.

As illustrated in **Table VI**, the variations in attack accuracy are not immediately pronounced. However, upon a broader examination, it becomes evident that as the number of S

TABLE VII: Use of Kandinsky [63] as shadow model and Stable Diffusion [8] as target model in conducting attacks, demonstrating the maintained efficacy of all four attack scenarios. We use the ‘X’-‘Y’ format to represent different experiment settings in the table, where ‘X’ means four attack scenarios, ‘Y’ being ‘S’ or ‘A’ denotes whether the shadow model is the same as or different from the target model. For each comparison, the optimal result is marked in bold.

Dataset	I-S	I-A	II-S	II-A	III-S	III-A	IV-S	IV-A
CelebA	0.93	0.87	0.93	0.86	0.93	0.86	0.93	0.85
WIT	0.83	0.81	0.83	0.84	0.84	0.84	0.83	0.83
MS COCO	0.92	0.89	0.92	0.91	0.89	0.89	0.76	0.74

(inference steps) increases, there is a gradual uptrend in the success rate of attacks. Notably, attacks based on classifiers yield the highest accuracy. To delve deeper into the reason why an increased number of inference steps does not lead to a substantial boost in attack success rate, we present samples generated at different inference steps in Figure 5. It becomes apparent that as the number of inference steps rises, only certain localized features of the generated images are affected. The overall style remains largely undisturbed, with no significant discrepancies observed. This observation potentially explains why altering the inference steps does not drastically impact the attack success rate.

The experimental results obtained from the additional two datasets are presented in Appendix I.

G. Impact of Different Size of Auxiliary Dataset

From our observations across white-box [13], [29], [30], gray-box [12], [13], [32], and black-box attacks [30], the accuracy of these attacks is significantly influenced by the size of training set. As the training set of the target model, encompasses more samples, its “memorization” capability for individual samples diminishes. This is attributed to the fact that an increase in training data can decelerate the model’s convergence rate, impacting its ability to fit all the training sets accurately. As a result, many attacks do not demonstrate effective performance as the dataset size expands. In this work, we investigate how increasing the size of the dataset used by the target model affects the success rate of our black-box attack. Given that our work is predicated on leveraging pre-trained models for downstream tasks, where the downstream datasets usually do not contain a vast number of samples, we have established our training dataset sizes at 100, 200, 500, and 1000. Using the CelebA dataset, we aim to assess the variations in the performance of the three attack types when the attacker is privy to four distinct values of knowledge.

As illustrated in Figure 7, the attack success rate tends to decrease as the number of images in the training set increases. However, even when the users use 1,000 samples to fine-tune the target models, in the scenarios of Attack-I and Attack-III, a classifier used as the attack model can still achieve a success rate of over 60%.

TABLE VIII: Impact of not fine-tuning the captioning model on the success rates of Attack-II and Attack-IV across various datasets. The best result for each dataset is marked in bold.

Dataset	Attack-II		Attack-IV	
	With tuning	W/o tuning	With tuning	W/o tuning
CelebA	0.93	0.59	0.93	0.60
WIT	0.83	0.70	0.83	0.56
MS COCO	0.93	0.79	0.73	0.65

H. Impact of the Selection of Shadow Models

To examine the generalization and applicability of our attack methodology in real-world scenarios, we propose to further relax the assumptions pretraining to the attack environment. In our prior experiments, all results were predicated on the use of shadow models mirroring the target model’s structural framework to generate training data for the attack inference model. However, in practical settings, malicious model publishers may withhold any specific details about the model, offering only a user interface. Under such circumstances, it is not advisable to confine ourselves to a specific type of shadow model. Instead, a more effective approach would be to leverage the memorization properties of image generators when creating training data for the attack, thus diversifying and strengthening the attack strategy.

Therefore, we employed a conditional image generator, Kandinsky [63], which has a different architectural design from Stable Diffusion [8], as our shadow model. This model was fine-tuned using the same auxiliary dataset mentioned in Table III, and the results are displayed in Table VII.

In Table VII, we evaluate attackers with different knowledge across three datasets, employing a classifier as the attack inference model. The notation ‘*-S’ indicates attacks conducted using a shadow model with the same architecture as the target model. Conversely, ‘*-A’ denotes scenarios where the target model is anonymous to the attacker. Hence, the shadow model and the target model are architecturally dissimilar. The experimental data indicate that altering the shadow model has only a minimal effect on the success rate of the attacks, with all attacks still capable of achieving a relatively high level of success. This further substantiates the robustness and generalizability of our attack framework.

I. Impact of Eliminating Fine-Tuning in Captioning Models

In our work, within the attack environments designed for Attack-II and Attack-IV, the attacker does not have full access to the query point x , but only a query image I_q . In previous sections, for these two attack scenarios, we initially used auxiliary data to fine-tune the image captioning model before generating matching prompt information based on the query image. However, this approach significantly increases the time cost of the attack. Therefore, we use an image captioning model that has not been fine-tuned to generate image descriptions. We then carry out the attack based on these generated descriptions.

From Table VIII, it is evident that without fine-tuning the captioning model, there is a varying degree of reduction in the success rates of attacks across different datasets. Notably,

when using CelebA-Dialog as the test set, the success rate of the attack drops by nearly 30%, leading to a marked inconsistency in the attack outcomes. Unlike changing the types of shadow models, a captioning model without tuning more conspicuously diminishes the effectiveness of the attacks. We posit the image captioning model may have introduced biases in the generated text component T_q , adversely affecting the quality of the resultant images.

Takeaways: We compared the four attack scenarios we proposed with existing black-box attacks and found that our accuracy significantly surpasses the established baselines. Then, we tested the feasibility of implementing the most straightforward threshold-based attack using compressed high-dimensional similarity scores. To thoroughly evaluate the accuracy and stability of our attacks, we conducted tests employing various image encoders, distance metrics, fine-tuning steps, and inference procedures, as well as different sizes of auxiliary datasets. Additionally, we experimented with changing the types of shadow models and testing without fine-tuning the image captioning model to test our attacks’ generalization and robustness. Our findings reveal a strong correlation between the attacks’ success rate and the generated images’ quality. Higher quality images lead to increased attack success rates, which aligns with the theory of similarity scores mentioned in [Section III-A](#).

VI. DEFENSE

In this part, we want to employ Differential Privacy Stochastic Gradient Descent (DP-SGD) [43] to evaluate the robustness of our attack. DP-SGD adds noise into the gradient during the training phase and provides a guarantee that the presence/absence of any single training sample only incurs quantifiably limited differences [64] and thus diminishes the model’s memorization of individual samples.

We tested our four attacks and employed a classifier as the inference model against the MS COCO dataset. Due to the limit of time and computing resources, we only selected two different sizes of datasets, 100 and 200. For the DP-SGD mechanism, we set clipping norm $C = 1$, $\delta = 1 \times 10^{-3}$, sampling rate $q = 4/(\text{dataset size})$, epoch number is 500, and target privacy budget (with a slight abuse of notation) $\epsilon \in \{1, 4, 10\}$ (different ϵ gives different noise multiplier σ).

The experimental results can be seen in [Table IX](#). It illustrates that the attack success rate of four attacks significantly decreases after implementing DP-SGD [43] as the defensive method. When we use 100 samples to fine-tune the model and set $\epsilon = 1$, four attacks have been greatly impacted dropping to around 50% (random guess). When we change the ϵ value from 1 to 4 and 10, the attack accuracy increases but still cannot show their effectiveness. From [Table IX](#), we noticed TPR at FPR=1% has dropped to 0.01. This phoneme further demonstrates that [Attack-I](#), [Attack-II](#), [Attack-III](#), and [Attack-IV](#) all lose their functionality in these defense settings.

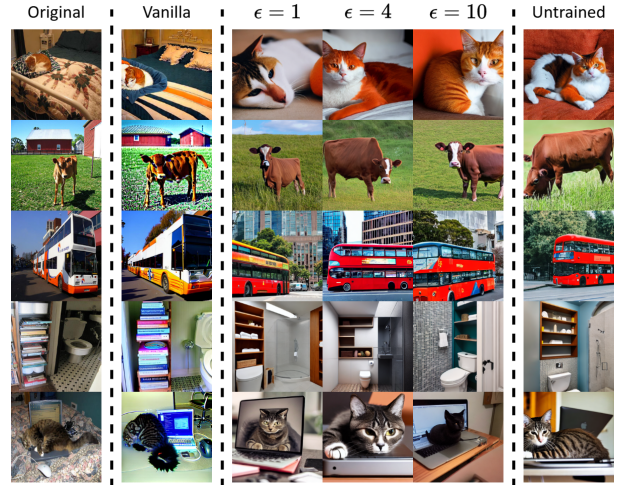


Fig. 6: Effect of adding DP-SGD [43] on model memorization. ‘Original’ represents the training samples, while ‘Vanilla’ denotes samples generated after fine-tuning without using DP-SGD. $\epsilon = 1, 4, 10$ indicate samples generated by the fine-tuned model after applying DP-SGD at varying levels. ‘Untrained’ represents samples generated by the Stable Diffusion v1-5 without fine-tuning. The generated images in the same row are from the same prompt.

To understand how DP-SGD mitigates our attack, we presented the generated images from different settings in [Figure 6](#). We observe that compared to the ‘Vanilla’ version (fine-tune without DP-SGD), DP-SGD prevents the model’s memorization of training samples. For instance, in the first row, it is clear that DP-SGD omits detailed features of the training images, such as pillows. This effect is observed regardless of the value of ϵ . The generated images with defense remain very similar to those produced by the untrained Stable Diffusion v1-5 model. DP-SGD weakens the model’s memorization of the training samples, thereby reducing the similarity score and rendering our four attacks ineffective.

VII. RELATED WORK

We further review related work on white-box and gray-box membership inference attacks against diffusion models.

A. White-Box MIA

In the white-box setting, the attacker has access to the parameters of the victim model. Note that in MIA for classification tasks, it is observed that having black-box means can sufficient enough information (e.g., predict vector [21], [25], [50], [65]–[67], top- k confidence score [24], [25]); but in MIA for generative models, because the model is more complicated and directly applying existing MIAs is not successful, white-box attacks are investigated.

Both Hu et al. [13] and Matsumoto et al. [30] adopt approaches similar to that of Yeom et al. [28], determining membership by comparing the loss at various timesteps to a specific threshold. Carlini et al. [29] argue that mere threshold-based determinations are insufficient and proposed training

TABLE IX: Attack accuracy under DP-SGD defense. Our four attack methods’ accuracy declines. Experiments include two different sizes of datasets and three ϵ values. ‘Vanilla’ means without DP-SGD. The highest accuracy is marked in bold.

		Attack-I			Attack-II			Attack-III			Attack-IV		
		ASR \uparrow	AUC \uparrow	T@1%F \uparrow	ASR \uparrow	AUC \uparrow	T@1%F \uparrow	ASR \uparrow	AUC \uparrow	T@1%F \uparrow	ASR \uparrow	AUC \uparrow	T@1%F \uparrow
100	$\epsilon = 1$	0.581	0.646	0.01	0.532	0.654	0.01	0.495	0.498	0.00	0.522	0.524	0.00
	$\epsilon = 4$	0.592	0.651	0.01	0.575	0.647	0.01	0.515	0.514	0.01	0.535	0.534	0.01
	$\epsilon = 10$	0.595	0.641	0.02	0.560	0.644	0.02	0.56	0.522	0.01	0.545	0.522	0.01
	Vanilla	0.843	0.911	0.58	0.845	0.909	0.51	0.831	0.893	0.38	0.765	0.813	0.19
200	$\epsilon = 1$	0.593	0.632	0.01	0.628	0.676	0.01	0.493	0.502	0.00	0.548	0.524	0.01
	$\epsilon = 4$	0.601	0.652	0.01	0.618	0.670	0.01	0.523	0.516	0.01	0.515	0.506	0.01
	$\epsilon = 10$	0.585	0.632	0.03	0.643	0.655	0.02	0.535	0.504	0.01	0.542	0.541	0.02
	Vanilla	0.767	0.863	0.30	0.730	0.812	0.11	0.695	0.728	0.09	0.773	0.800	0.14

multiple shadow models and utilizing the distribution of loss across each timestep established by these shadow models to execute an online LiRA attack [52]. Pang et al. [15] leveraged the norm of gradient information computed from timesteps uniformly sampled across total diffusion steps as attack data to train their attack model.

B. Gray-box MIA

Gray-box access does not acquire any internal information from the model. However, given that diffusion models generate images through a progressive denoising process, attacks in this setting assume the availability of intermediate outputs during this process. In particular, several works leveraged the deterministic properties of generative process in DDIM [10] for their attack designs. Duan et al. [12] employed the approximated posterior estimation error as attack features, while Kong et al. [32] used the magnitude difference $\|x_{t-t'} - x'_{t-t'}\|_p$ from the denoising process as their attack criterion, where $x_{t-t'}$ represents the ground truth and $x'_{t-t'}$ denotes the predicted value. Fu et al. [31] use the intermediate output to calculate the probabilistic fluctuations between target points and neighboring points. Similarly, Zhai et al. [35] sampled multiple times at different denoising steps, with the likelihood discrepancy between the conditional and unconditional generations as the criterion. Fu et al. [34] based their approach on the structural similarity between intermediate outputs and the original images. Li et al. [33] found that the similarity between reconstructed images and the original images after degradation can also serve as a standard for evaluation.

VIII. CONCLUSION

In this work, we introduce a black-box membership inference attack framework specifically designed for contemporary conditional diffusion models. Given the rapid development of diffusion models and the abundance of open-source pre-trained models available online, we focus on the potential privacy issues arising from utilizing these pre-trained models fine-tuned for downstream tasks. Recognizing the absence of effective attacks against the current generation of conditional image generators, we leverage the objective function of diffusion models to propose a black-box similarity scores-based membership inference attack. Our experiments not only demonstrate the flexibility and effectiveness of this attack but also highlight significant privacy vulnerabilities in image

generators, underscoring the need for increased attention to these issues.

However, our attacks still face certain limitations. As discussed in Section V-I, both Attack-II and Attack-IV critically rely on a captioning model that has been fine-tuned using an auxiliary dataset. We hope future work can effectively address this challenge.

ACKNOWLEDGMENT

We thank the reviewers for their valuable comments and suggestions. This work is supported by NSF CCF-2217071 and OAC-2319988.

REFERENCES

- [1] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever, “Zero-shot text-to-image generation,” in *International Conference on Machine Learning*. PMLR, 2021, pp. 8821–8831.
- [2] J. Ho, A. Jain, and P. Abbeel, “Denoising diffusion probabilistic models,” 2020.
- [3] A. Nichol, P. Dhariwal, A. Ramesh, P. Shyam, P. Mishkin, B. McGrew, I. Sutskever, and M. Chen, “Glide: Towards photorealistic image generation and editing with text-guided diffusion models,” 2022.
- [4] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, “Hierarchical text-conditional image generation with clip latents,” 2022.
- [5] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, “Photorealistic text-to-image diffusion models with deep language understanding,” 2022.
- [6] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, “Deep unsupervised learning using nonequilibrium thermodynamics,” in *International Conference on Machine Learning*. PMLR, 2015, pp. 2256–2265.
- [7] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, “Score-based generative modeling through stochastic differential equations,” 2021.
- [8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, “High-resolution image synthesis with latent diffusion models,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.
- [9] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” 2022.
- [10] J. Song, C. Meng, and S. Ermon, “Denoising diffusion implicit models,” 2022.
- [11] C. Schuhmann, R. Beaumont, R. Vencu, C. Gordon, R. Wightman, M. Cherti, T. Coombes, A. Katta, C. Mullis, M. Wortsman, P. Schramowski, S. Kundurthy, K. Crowson, L. Schmidt, R. Kaczmarczyk, and J. Jitsev, “Laion-5b: An open large-scale dataset for training next generation image-text models,” 2022.
- [12] J. Duan, F. Kong, S. Wang, X. Shi, and K. Xu, “Are diffusion models vulnerable to membership inference attacks?” in *International Conference on Machine Learning*. PMLR, 2023, pp. 8717–8730.
- [13] H. Hu and J. Pang, “Membership inference of diffusion models,” *arXiv preprint arXiv:2301.09956*, 2023.
- [14] Y. Wu, N. Yu, Z. Li, M. Backes, and Y. Zhang, “Membership inference attacks against text-to-image generation models,” *arXiv preprint arXiv:2210.00968*, 2022.

- [15] Y. Pang, T. Wang, X. Kang, M. Huai, and Y. Zhang, "White-box membership inference attacks against diffusion models," *arXiv preprint arXiv:2308.06405*, 2023.
- [16] K. Li, C. Gong, Z. Li, Y. Zhao, X. Hou, and T. Wang, "Meticulously selecting 1% of the dataset for pre-training! generating differentially private images data with semantics query," *arXiv preprint arXiv:2311.12850*, 2023.
- [17] S. Shan, J. Cryan, E. Wenger, H. Zheng, R. Hanocka, and B. Y. Zhao, "Glaze: Protecting artists from style mimicry by {Text-to-Image} models," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 2187–2204.
- [18] S.-Y. Chou, P.-Y. Chen, and T.-Y. Ho, "Villandiffusion: A unified backdoor attack framework for diffusion models," *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [19] S. Peng, Y. Chen, C. Wang, and X. Jia, "Protecting the intellectual property of diffusion models by the watermark diffusion process," *arXiv preprint arXiv:2306.03436*, 2023.
- [20] C. A. Choquette-Choo, F. Tramèr, N. Carlini, and N. Papernot, "Label-only membership inference attacks," in *International conference on machine learning*. PMLR, 2021, pp. 1964–1974.
- [21] B. Hui, Y. Yang, H. Yuan, P. Burlina, N. Z. Gong, and Y. Cao, "Practical blind membership inference attack via differential comparisons," *arXiv preprint arXiv:2101.01341*, 2021.
- [22] S. Rezaei and X. Liu, "On the difficulty of membership inference attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7892–7900.
- [23] A. Sablayrolles, M. Douze, Y. Ollivier, C. Schmid, and H. Jégou, "White-box vs black-box: Bayes optimal strategies for membership inference," 2019.
- [24] A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes, "MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models," 2018.
- [25] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2017.
- [26] L. Song and P. Mittal, "Systematic evaluation of privacy risks of machine learning models," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 2615–2632.
- [27] S. Truex, L. Liu, M. E. Gursoy, L. Yu, and W. Wei, "Demystifying membership inference attacks in machine learning as a service," *IEEE transactions on services computing*, vol. 14, no. 6, pp. 2073–2089, 2019.
- [28] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, "Privacy risk in machine learning: Analyzing the connection to overfitting," in *2018 IEEE 31st computer security foundations symposium (CSF)*. IEEE, 2018, pp. 268–282.
- [29] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwal, F. Tramèr, B. Balle, D. Ippolito, and E. Wallace, "Extracting training data from diffusion models," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 5253–5270.
- [30] T. Matsumoto, T. Miura, and N. Yanai, "Membership inference attacks against diffusion models," 2023.
- [31] W. Fu, H. Wang, C. Gao, G. Liu, Y. Li, and T. Jiang, "A probabilistic fluctuation based membership inference attack for diffusion models," *arXiv e-prints*, pp. arXiv–2308, 2023.
- [32] F. Kong, J. Duan, R. Ma, H. Shen, X. Zhu, X. Shi, and K. Xu, "An efficient membership inference attack for the diffusion model by proximal initialization," 2023.
- [33] Q. Li, X. Fu, X. Wang, J. Liu, X. Gao, J. Dai, and J. Han, "Unveiling structural memorization: Structural membership inference attack for text-to-image diffusion models," *arXiv preprint arXiv:2407.13252*, 2024.
- [34] X. Fu, X. Wang, Q. Li, J. Liu, J. Dai, and J. Han, "Model will tell: Training membership inference for diffusion models," *arXiv preprint arXiv:2403.08487*, 2024.
- [35] S. Zhai, H. Chen, Y. Dong, J. Li, Q. Shen, Y. Gao, H. Su, and Y. Liu, "Membership inference on text-to-image diffusion models via conditional likelihood discrepancy," *arXiv preprint arXiv:2405.14800*, 2024.
- [36] D. Chen, N. Yu, Y. Zhang, and M. Fritz, "Gan-leaks: A taxonomy of membership inference attacks against generative models," in *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, 2020, pp. 343–362.
- [37] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, 2019.
- [38] M. Zhang, N. Yu, R. Wen, M. Backes, and Y. Zhang, "Generated distributions are all you need for membership inference attacks against generative models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4839–4849.
- [39] J. Dubiński, A. Kowalczyk, S. Pawlak, P. Rokita, T. Trzcinski, and P. Morawiecki, "Towards more realistic membership inference attacks on large diffusion models," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024, pp. 4860–4869.
- [40] Y. Jiang, Z. Huang, X. Pan, C. C. Loy, and Z. Liu, "Talk-to-edit: Fine-grained facial editing via dialog," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.
- [41] K. Srinivasan, K. Raman, J. Chen, M. Bendersky, and M. Najork, "Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 2443–2449.
- [42] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.
- [43] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 308–318.
- [44] F. Suya, A. Suri, T. Zhang, J. Hong, Y. Tian, and D. Evans, "Sok: Pitfalls in evaluating black-box attacks," *arXiv:2310.17534*, 2023.
- [45] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.
- [46] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
- [47] J. Betker, G. Goh, L. Jing, TimBrooks, J. Wang, L. Li, LongOuyang, JuntangZhuang, JoyceLee, YufeiGuo, WesamManassra, PrafullaDhariwal, CaseyChu, YunxinJiao, and A. Ramesh, "Improving image generation with better captions." [Online]. Available: <https://api.semanticscholar.org/CorpusID:264403242>
- [48] J. Li, N. Li, and B. Ribeiro, "Membership inference attacks and defenses in classification models," in *Proceedings of the Eleventh ACM Conference on Data and Application Security and Privacy*, 2021, pp. 5–16.
- [49] Y. Liu, Z. Zhao, M. Backes, and Y. Zhang, "Membership inference attacks by exploiting loss trajectory," 2022.
- [50] Y. Long, V. Bindschaedler, L. Wang, D. Bu, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "Understanding membership inferences on well-generalized learning models," *arXiv preprint arXiv:1802.04889*, 2018.
- [51] Y. Long, L. Wang, D. Bu, V. Bindschaedler, X. Wang, H. Tang, C. A. Gunter, and K. Chen, "A pragmatic approach to membership inferences on machine learning models," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2020, pp. 521–534.
- [52] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramèr, "Membership inference attacks from first principles," 2022.
- [53] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [54] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19730–19742.
- [55] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.
- [56] W. J. Youden, "Index for rating diagnostic tests," *Cancer*, vol. 3, no. 1, pp. 32–35, 1950.
- [57] S. Gu, D. Chen, J. Bao, F. Wen, B. Zhang, D. Chen, L. Yuan, and B. Guo, "Vector quantized diffusion model for text-to-image synthesis," 2022.
- [58] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.
- [59] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," 2022.
- [60] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," 2022.
- [61] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

- [62] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in *International conference on machine learning*, PMLR, 2021, pp. 10347–10357.
- [63] A. Razzhigaev, A. Shakhmatov, A. Maltseva, V. Arkhipkin, I. Pavlov, I. Ryabov, A. Kuts, A. Panchenko, A. Kuznetsov, and D. Dimitrov, “Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion,” *arXiv preprint arXiv:2310.03502*, 2023.
- [64] C. Dwork, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” *Journal of Privacy and Confidentiality*, vol. 7, no. 3, pp. 17–51, 2016.
- [65] M. Chen, Z. Zhang, T. Wang, M. Backes, M. Humbert, and Y. Zhang, “When machine unlearning jeopardizes privacy,” in *Proceedings of the 2021 ACM SIGSAC conference on computer and communications security*, 2021, pp. 896–911.
- [66] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, “Revisiting membership inference under realistic assumptions,” 2021.
- [67] R. Shokri, M. Strobel, and Y. Zick, “On the privacy risks of model explanations,” 2021.
- [68] Y. Pang and T. Wang, “Black-box membership inference attacks against fine-tuned diffusion models,” *arXiv preprint arXiv:2312.08207*, 2023.
- [69] A. B. Owen, “Monte carlo theory, methods and examples,” 2013.

Due to page limitations, the complete version of the appendix can be found at [68].

APPENDIX A MORE DETAILS FOR DIFFUSION MODELS

Given noised sample x_t and timestep t , the diffusion model is trained to make the predicted distribution $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$ approach the ground-truth distribution $\mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \sigma_t^2 \mathbf{I})$. Applying Bayes’ rule to the ground-truth distribution

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t} x_0 \quad (5)$$

The objective of the training process is to closely approximate $\mu_\theta(x_t, t)$ with $\tilde{\mu}_t(x_t, x_0)$. Then, parameterize

$$\mu_\theta(x_t, t) = \frac{\sqrt{\alpha_t}(1 - \bar{\alpha}_{t-1})}{1 - \bar{\alpha}_t} x_t + \frac{\sqrt{\bar{\alpha}_{t-1}}(1 - \alpha_t)}{1 - \bar{\alpha}_t} \hat{x}_\theta(x_t, t) \quad (6)$$

By deriving x_t from x_0 using Equation 1 and omitting the weight term, the loss function is given in Equation 2.

APPENDIX B MORE DETAILS FOR CLASSIFIER-FREE GUIDANCE

A conditional generation without an explicit classifier is achieved using the denoising network $\bar{\mathcal{U}}_\theta(x_t, t, p)$, where

$$\bar{\mathcal{U}}_\theta(x_t, t, p) = (w + 1) \cdot \mathcal{U}_\theta(x_t, t, p) - w \cdot \mathcal{U}_\theta(x_t, t).$$

The variable w is the guidance scale factor; a higher w improves image-text alignment but may reduce image fidelity.

APPENDIX C MORE DETAILS FOR THEORETICAL FOUNDATION

A. Proof for Theorem 1

Proof: Diffusion models employ the ELBO to approximate the log-likelihood $p(x)$ of the entire training dataset.

$$\log p(x) \geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[\log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \right]$$

...

$$\begin{aligned} &= \underbrace{\mathbb{E}_{q(x_1|x_0)} [\log p_\theta(x_0|x_1)]}_{L_0} - \underbrace{\mathcal{D}_{KL}(q(x_T|x_0) \| p(x_T))}_{L_T} \\ &\quad - \underbrace{\sum_{t=2}^T \mathbb{E}_q(x_t|x_0) [\mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))]}_{L_{t-1}} \end{aligned} \quad (7)$$

The primary focus of optimization is on L_{t-1} , as explicated in the original work [2]. The other terms are treated as constants and independent decoders. The objective function can be rewritten as:

$$\min_{\theta} \mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)).$$

Based on the assumption in DDPM [2], to elucidate further:

$$\begin{aligned} &\arg \min_{\theta} \mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t)) \\ &= \arg \min_{\theta} \mathcal{D}_{KL}(\mathcal{N}(\tilde{\mu}_t(x_t, x_0), \sigma_t^2 \mathbf{I}) \| \mathcal{N}(\mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})) \\ &= \arg \min_{\theta} \frac{1}{2\sigma_t^2} \left[\|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|_2^2 \right] \end{aligned} \quad (8)$$

In Equation 8, $q(x_{t-1}|x_t, x_0)$ represents the ground truth distribution of x_{t-1} given x_t and x_0 , while $p_\theta(x_{t-1}|x_t)$ denotes the predicted distribution of x_{t-1} parameterized by θ . The term $\tilde{\mu}_t(x_t, x_0)$ corresponds to the mean of the ground truth distribution $q(x_{t-1}|x_t, x_0)$, and $\mu_\theta(x_t, t)$ corresponds to the mean of the predicted distribution $p_\theta(x_{t-1}|x_t)$.

From Equation 5 and Equation 6 in Appendix A (which gives more details about diffusion models), we can rewrite Equation 8 as:

$$\arg \min_{\theta} \frac{1}{2\sigma_t^2} \frac{\bar{\alpha}_{t-1}(1 - \alpha_t)^2}{(1 - \bar{\alpha}_t)^2} \left[\|x_0 - \hat{x}_\theta(x_t, t)\|_2^2 \right] \quad (9)$$

Equation 9 can also be further developed by substituting and expressing x_0 using x_t according to Equation 1, and by introducing ϵ_t as the targeted prediction of the diffusion model, aligning with the optimization objectives stated in both DDPM [2] and DDIM [10]. However, our aim is to demonstrate that the optimization goal of the diffusion model supports the use of similarity scores as an indicator for determining the membership of query data. Consequently, the objective function is merely reformulated in the form of Equation 9. Given that the likelihood of all training data should be higher than that of data not in the training set, and as inferred from Equation 7 and Equation 9, if a data point x has a higher likelihood, the norm $\|x_0 - \hat{x}_\theta(x_t, t)\|$ at any timestep in the model should be smaller, indicating that the image generated by the model is closer to the original image. This can be expressed as:

$$\Pr[b = 1|x, \theta] \propto -\|x_0 - \hat{x}_\theta(x_t, t)\|_2^2 \quad (10)$$

■

B. Proof for Theorem 2

Proof: In the original paper [8], the loss function of the Stable Diffusion model is described as follows:

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x), \epsilon \sim \mathcal{N}(0,1), t} [\|\epsilon_t - \mathcal{U}_\theta(z_t, t, \phi_\theta(p))\|_2^2]$$

The latent code z_t is of a much smaller dimension than that of the original image. The denoising network \mathcal{U}_θ predicts the noise at timestep t based on z_t and the embedding generated by ϕ_θ , which takes p as its input. Given that the forward process of the Stable Diffusion [8] is fixed, Equation 1 remains applicable. Therefore, by substituting in the expression $\epsilon_t = \frac{z_t - \sqrt{\alpha_t} z_0}{\sqrt{1 - \alpha_t}}$ and discarding other weight terms, we can rederive the loss function of the Stable Diffusion model as:

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x), t} [\|z_0 - \hat{z}_\theta(z_t, t, \phi_\theta(p))\|_2^2] \quad (11)$$

As seen from Equation 11, Stable Diffusion is essentially trained to optimize image predictions at any given timestep to closely approximate the original image $D(z_0)$, where D is the decoder in Stable Diffusion. For the Stable Diffusion model, we can still distinguish between member samples and non-member samples by the similarity scores $\|D(z_0) - D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))\|_2^2$, which is expressed as:

$$\Pr[b = 1 | x, \theta] \propto -\|D(z_0) - D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))\|_2^2 \quad (12)$$

APPENDIX D

MORE DETAILS FOR TRADITIONAL BLACK-BOX ATTACKS

Monte Carlo Attack. Given a query sample x , attackers can utilize the generative model to sample k images. Define an ϵ -neighborhood set $U_\epsilon(x)$ as $U_\epsilon(x) = \{x' \mid d(x, x') \leq \epsilon\}$. Intuitively, if a larger number of g_i are close to x , the probability $\Pr[x' \in U_\epsilon(x)]$ will also be greater. Through the Monte Carlo Integration [69], the Monte Carlo attack can be expressed as:

$$\hat{f}_{MC-\epsilon}(x) = \frac{1}{k} \sum_{i=1}^k \mathbb{1}_{x'_i \in U_\epsilon(x)} \quad (13)$$

GAN-Leaks Attack. Chen et al. [36] posited that the closer the generated data distribution $p_\theta(\hat{x})$ is to the training data distribution $q(x)$, the more likely it is for \mathcal{G} to generate a query datapoint x . They employed the KDE method [53] and sampled k times to estimate the likelihood of x . This can be expressed as:

$$\Pr_{\mathcal{G}}(x|\theta) = \frac{1}{k} \sum_{i=1}^k K(x, \mathcal{G}(z_i)); \quad z_i \sim P_z \quad (14)$$

Here, K denotes the kernel function, and z_i represents the input to \mathcal{G} , which sample from latent code distribution P_z .

APPENDIX E

MORE DETAILS ON THE ATTACK FRAMEWORK

We use an example to show how our attack works. Assume we have a query image I_q and a generated image I_g . After

extracting features using an image feature extractor E , the resulting vector has dimensions $[\text{patch_size}, \text{latent_size}]$. For example, when using ViT as E , the height and width of the image are first resized to 224×224 , and then divided into 196 patches, each of size 16×16 , and a latent representation typically of size 768 is calculated for each patch. Extracting features from I_q and I_g results in two vectors of size $[196, 768]$.

We then calculate the patch-wise similarity score, resulting in 196. Note that we can use patches of other granularities. In the extreme case, we can just use the CLS token of ViT, which gives 768 latent space features of the whole image. But this may overlook some details. So we choose the most fine-grained patches available.

If we query the target model m times, we obtain a similarity score vector of size $[m, 196]$ for all generated images. By applying our defined statistical function f , we aggregate the similarity scores from multiple generated images to produce a final similarity score vector of size 196. This vector is then used as input for threshold-based, distribution-based, and classifier-based attack models.

APPENDIX F

MORE DETAILS FOR DIFFERENT SIZE OF AUXILIARY DATASET

Figure 7 shows that in all attack scenarios, the attack performance decreases as the size of the auxiliary data increases. However, the classifier-based attack still maintains a ROC-AUC above 0.6.

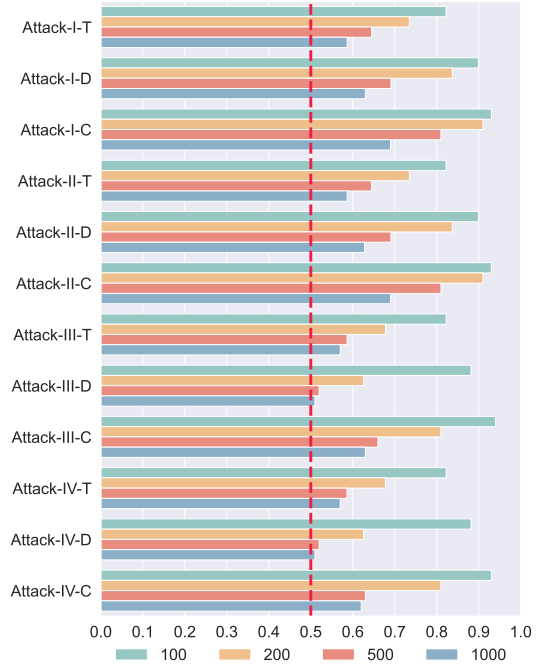


Fig. 7: Attack nomenclature and performance trends: ‘T’ for threshold-based, ‘D’ for distribution-based, and ‘C’ for classifier-based attacks, with accuracy inversely related to training set size.

APPENDIX G
MORE DETAILS FOR COMPARING FIVE DIFFERENT IMAGE ENCODERS

To comprehensively analyze the influence of various image feature extractors on attack success rates, we evaluated the performance of five distinct image feature extractors across three types of attacks, within four attack scenarios obtained by the attacker, on three datasets. For each attack, we highlighted the optimal results of each evaluation metric across different image feature extractors. In Table X (and Tables XI and XII in Appendix G of the full paper [68]), DeiT is the most stable image encoder and achieves the best attack performance.

TABLE X: Comparative analysis of five different image encoders using *classifier-based* attack across three datasets.

		DETR			BEiT			EfficientFormer			ViT			DeiT		
		ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%
CelebA	Attack-I	0.66	0.70	0.10	0.87	0.95	0.64	0.80	0.87	0.37	0.81	0.88	0.26	0.87	0.93	0.49
	Attack-II	0.67	0.69	0.09	0.88	0.94	0.57	0.82	0.88	0.38	0.80	0.88	0.29	0.88	0.94	0.61
	Attack-III	0.67	0.71	0.07	0.84	0.91	0.57	0.81	0.87	0.42	0.79	0.83	0.40	0.87	0.94	0.52
	Attack-IV	0.67	0.71	0.10	0.84	0.91	0.58	0.78	0.84	0.44	0.78	0.83	0.38	0.88	0.93	0.60
WIT	Attack-I	0.74	0.79	0.11	0.70	0.80	0.30	0.76	0.81	0.13	0.77	0.83	0.06	0.79	0.84	0.22
	Attack-II	0.73	0.77	0.10	0.69	0.77	0.27	0.71	0.78	0.11	0.74	0.80	0.16	0.78	0.85	0.15
	Attack-III	0.65	0.72	0.07	0.71	0.78	0.17	0.78	0.82	0.22	0.78	0.82	0.21	0.77	0.83	0.29
	Attack-IV	0.64	0.69	0.08	0.72	0.77	0.11	0.76	0.81	0.16	0.77	0.82	0.05	0.75	0.83	0.25
MS COCO	Attack-I	0.72	0.75	0.17	0.77	0.84	0.24	0.78	0.87	0.20	0.73	0.82	0.20	0.85	0.93	0.61
	Attack-II	0.75	0.80	0.06	0.77	0.85	0.16	0.81	0.87	0.35	0.75	0.83	0.20	0.85	0.92	0.56
	Attack-III	0.70	0.78	0.16	0.78	0.84	0.44	0.78	0.82	0.28	0.71	0.80	0.20	0.83	0.89	0.30
	Attack-IV	0.70	0.76	0.20	0.80	0.83	0.40	0.76	0.83	0.27	0.75	0.82	0.31	0.69	0.74	0.16

APPENDIX H
MORE EXPERIMENTAL RESULTS FOR VARYING FINE-TUNING STEPS

In this part, we want to examine the impact of increasing fine-tuned steps on the outcomes of different types of attacks. The distribution-based attack results can be found in Figure 8, and the threshold-based attack is illustrated in Figure 9 of full paper [68]. All these experiment results show that attack accuracy increases with more fine-tuning steps.

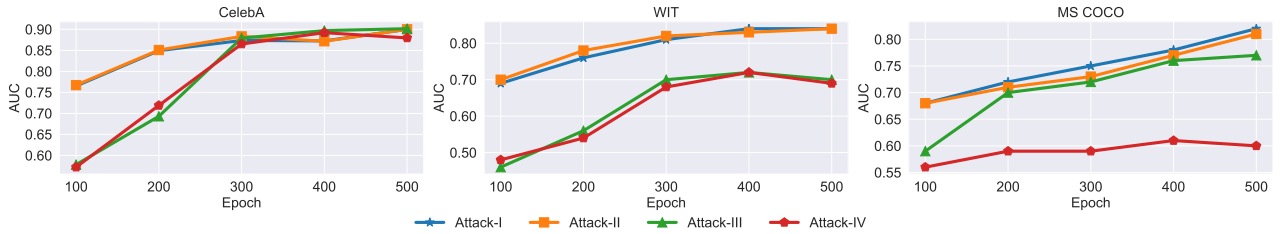


Fig. 8: Correlation between increased fine-tuning steps and enhanced accuracy of *distribution-based* attack.

APPENDIX I
MORE EXPERIMENTAL RESULTS FOR DIFFERENT NUMBER OF INFERENCE STEPS

To evaluate how inference steps affect attack performance, we conducted experiments on the WIT and MS COCO datasets, with results detailed in Table XI. We highlighted the best attack results for each evaluation metric across different inference steps. The results indicate that the inference steps do not affect attack accuracy.

TABLE XI: Experiment results for more inference steps on MS COCO and WIT. The best attack result is marked in bold.

S	MS COCO									FID	WIT									FID
	Threshold-based			Distribution-based			Classifier-based				Threshold-based			Distribution-based			Classifier-based			
	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%		ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	ASR	AUC	T@F=1%	
30	0.76	0.84	0.13	0.70	0.77	0.13	0.84	0.90	0.42	8.49	0.71	0.81	0.23	0.61	0.70	0.26	0.78	0.82	0.29	6.73
50	0.74	0.84	0.13	0.69	0.77	0.11	0.84	0.91	0.20	7.24	0.71	0.80	0.20	0.62	0.72	0.25	0.75	0.82	0.30	5.83
100	0.76	0.84	0.15	0.70	0.76	0.11	0.85	0.90	0.23	6.46	0.71	0.79	0.17	0.65	0.74	0.09	0.76	0.83	0.32	5.58
200	0.77	0.84	0.16	0.71	0.75	0.11	0.83	0.88	0.21	6.46	0.70	0.79	0.20	0.62	0.72	0.14	0.77	0.83	0.33	5.56

A. Description & Requirements

Our work proposed a black-box membership inference attack against fine-tuned diffusion models. The primary components of our work are: 1) A dataset that the attacker wants to use to fine-tune the model, 2) A pre-trained stable diffusion model and the fine-tuned LoRA module, 3) A feature extractor to obtain features from the generated images, and 4) An attack model to examine the membership of the query data.

1) *How to access:* Users can access our code repository for the experiment code at¹³. In this repository, we also provide the bash commands to run our code. We included the fine-tuned LoRA module, BLIP checkpoint, and the exemplary dataset in the artifact package. Users can use this dataset to fine-tune a new LoRA module or directly use the one we have trained. Score vectors for each experiment are also stored in the artifact package. These score vectors correspond to target model member scores, target model non-member scores, shadow model member scores, and shadow model non-member scores. Users can use these four vectors as input to test our attack’s accuracy. We have also uploaded the artifact package with all of the experimental data to the Zenodo repository at¹⁴.

2) *Hardware dependencies:*

- **GPU:** NVIDIA GTX A6000 or higher.
- **RAM:** 252 GB minimum.
- **CPU:** AMD Ryzen Threadripper PRO 5955WX 16-Cores or equivalent.

3) *Software dependencies:*

- **Anaconda:** Anaconda3-2023.03
- **Python:** Python 3.10.14
- **Pytorch:** Pytorch 2.0.1
- **Packages:** The package dependencies are specified in `environment.yml` at the code repository.

4) *Benchmarks:* None.

B. Artifact Installation & Configuration

All required model checkpoints and datasets are included in the artifact package. Due to the settings of **Attack-II** and **Attack-IV** in our work, we provided trained BLIP checkpoints in the artifact package. Users can choose to fine-tune the LoRA module and modify the default training configuration. The scripts for image generation, similarity score calculation, and attack accuracy calculation contain the default runtime configurations.

In our work, we tested several variables that could influence the accuracy of the attack. In each experiment, we set other variables default to **Table XII**.

¹³<https://github.com/py85252876/Reconstruction-based-Attack>

¹⁴<https://zenodo.org/records/13371475>

TABLE XII: The default settings used in our experiments.

Parameters	Experiment setting for our work
Inference step	30
Resolution	512 × 512
Image encoder	DeiT
Fine-tuning epochs	500
Distance metrics	Cosine similarity
Size of dataset	100

C. Experiment Workflow

Our work workflow contains five parts.

- **Collect Dataset:** The first step of our work is to collect the data that we want to fine-tune the model.
- **Fine-tuned Model:** Use the prepared dataset to fine-tune the LoRA module. During this phase, the parameters of U-Net, VAE, and the text encoder are frozen.
- **Synthesized Images:** Generate images based on the dataset from the fine-tuned model.
- **Generated Score Matrix:** Calculate the similarity score between the generated images and the query image, and compute the average similarity score for each query sample.
- **Discriminate Member and Non-member Samples:** Implement attacks using threshold-based, distribution-based, and classifier-based attack models based on the generated score matrix.

D. Major Claims

- (C1): We consider four attack scenarios where an attacker can execute an attack based on the level of *query access* and the *quality of the initial auxiliary data*. Three different types of attack models are used to evaluate the success rate of these attacks.
- (C2): The impact of the attack was analyzed by considering various factors: image encoder selection (E1), distance metrics (E2), fine-tuning steps (E3), inference step count (E4), dataset size (E5).

E. Evaluation

To evaluate our work, several preliminary steps are required, including dataset pre-processing and model fine-tuning. We have included the fine-tuned module and BLIP checkpoints in the artifact package to facilitate quicker reproduction of the results.

1) *Impact of Different Image Encoder:* **E1** [E1] [12 hours training + 40 minutes attack]: This part of the experiment focuses on comparing the impact of different image encoders on attack effectiveness. The image encoders included in the experiment are DETR, BEiT, EfficientFormer, ViT, and DeiT. *[Preparation]* After setting up the environment using the `environment.yml` file in the code repository, the user can employ the provided dataset files and the `train_text_to_image_lora.py` to train the LoRA module. Once the training is finished, use the

`inference.py` to generate images. By default, three images are generated for each query data point.

[Execution] After generating images for each query data point, use `cal_embedding.py` to extract image features and calculate similarity scores. In this process, the user can use the `--image_encoder` configuration to test the accuracy of five different image encoders extracting image features for the attack. Each attack will produce five different sets of similarity score vectors. These five sets of similarity vectors, obtained using different feature extractors, are then used as inputs for the `test_accuracy.py` to execute the attack.

[Results] The experimental results should align with those in Figure 3. This figure shows that among the five different encoders, DeiT achieves the highest attack success rate.

2) *Impact of Different Distance Metrics.*: **E2** [E2] [12 hours training + 40 minutes attack]: In this section of the experiment, we focus on determining the most effective distance metrics for evaluating feature vectors extracted from generated images and query data. The distance metrics tested in this experiment include ℓ_1 , ℓ_2 , the Hamming distance, and the cosine similarity.

[Preparation] Similar to **E1**, we need to fine-tune the LoRA module and use the BLIP checkpoints for **Attack-II** and **Attack-IV**. Subsequently, according to the definitions of different attack scenarios, we will load the appropriate checkpoints and run `inference.py` to generate images.

[Execution] After generating three images for each query data point using the default configuration, we calculate similarity scores with `cal_embedding.py`. It is important to note that while **E1** focused on selecting the image encoder, this section compares different distance metrics. We control the comparison using the `--method` parameter. Finally, we input the similarity vectors obtained from the four distance metrics into `test_accuracy.py` to test the accuracy of the attacks.

[Results] The observed experimental results should align with those in Section V-D. In that section, Cosine similarity as a distance metric demonstrates superior attack performance.

3) *Impact of Fine-tuning Steps.*: **E3** [E3] [12 hours training + 40 minutes attack]: The foundation of our attack is the hypothesis that the model retains the memorization of the training data. However, the effectiveness of this memorization is significantly influenced by the fine-tuning steps. In this section, we perform attacks on the model at the 100, 200, 300, 400, and 500 epochs.

[Preparation] Similar to **E1** and **E2**, we need to prepare the fine-tuned LoRA module and BLIP checkpoints. In this section, we use the DeiT image encoder and Cosine similarity as the distance metric, as determined in **E1** and **E2**, to yield superior attack performance. These will be set as the default settings for both the current and subsequent experiments.

[Execution] When storing the LoRA module checkpoints, we save them at every 100 epoch. We then use these five checkpoints to generate images and calculate similarity score vectors. Finally, we perform the attacks using `test_accuracy.py`.

[Results] The experimental results should align with Figure 4 in Section V-E. As the number of fine-tuning steps increases, the model’s memorization of the training samples strengthens, leading to improved attack accuracy.

4) *Impact of Inference Step.*: **E4** [E4] [12 hours training + 40 minutes attack]: According to DDIM [10], the quality of generated images is influenced by the number of inference steps. Therefore, in this experiment, we investigate the impact of inference steps set to 30, 50, 100, and 200 on the attack success rate.

[Preparation] As with the previous experiments, we need to prepare the LoRA module and BLIP model checkpoints. In this experiment, the image encoder, distance metrics, and fine-tuning steps will all use the default settings specified in Table III.

[Execution] This part of the experiment focuses on the image generation stage. We control the inference steps by modifying the `--inference` parameter in `inference.py`. After generating images with different inference steps, we use `cal_embedding.py` to obtain the similarity score vectors and `test_accuracy.py` to calculate the attack success rate.

[Results] The results of the experiment should demonstrate that varying the inference steps does not affect attack accuracy.

5) *Impact of Different Size of Dataset.*: **E5** [E5] [12 hours training + 40 minutes attack]: In other membership inference attacks, the size of the dataset significantly impacts the attack’s effectiveness. In this section, we set the dataset sizes to 100, 200, 500, and 1000, keeping the number of fine-tuning epochs constant, and then evaluate the attack effectiveness.

[Preparation] As with the previous experiments, we prepare the trained LoRA module and BLIP checkpoints. Other attack parameters are set according to Table III.

[Execution] This part of the experiment primarily distinguishes the fine-tuned LoRA modules using different dataset sizes. Due to time constraints, larger datasets require more time for training. Therefore, we have included the pre-trained LoRA modules in the artifact package for immediate use.

[Results] The experimental results should align with Figure 7 in Section V-G, indicating that smaller datasets achieve better attack accuracy when the number of training epochs is consistent.

F. Customization

For customization, users can specify the fine-tuning epochs and other relevant parameters, such as the learning rate.

G. Notes

Our future work will include the implementation of memorization phenomena and additional literature supplementation. These additions will not affect the final conclusions drawn from the above experiments.