

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

Yan Pang  
University of Virginia  
trv3px@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<sup>1</sup> 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.

## I. INTRODUCTION

The recent developments in image-generative models have been remarkably swift, and many applications based on these models have appeared. Diffusion models [41], [18], [35], [40], [46], [53], [56], [44], [19], [54] 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 [44] was introduced. Leveraging the extensive and high-fidelity LAION-2B [48] 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 [12], [20], [61], [6], [37], [34], [25], [14], [10], [13], [39], [49], [28]. 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 [9], [21], [43], [45], [47], [51], [55], [60], [62], 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 [6], [20], [34] or gradients [37] as attack features. Hu et al. [20] and Matsumoto et al. [34] have utilized losses at different timesteps of querying the model as attack features. Similarly, Carlini et al. [6] 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. [37] 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], [14], [20], [25]. For example, Duan et al. [12], and Kong et al. [25] 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. [20] 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 [7], [17] and VAEs [17]. 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 [34], [61], [63], although more 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](#).

<sup>1</sup>Code accessible at <https://github.com/py85252876/Reconstruction-based-Attack>

In this paper, we present a black-box attack framework fit for state-of-the-art image-generative models. The framework was built on a careful analysis of the objective function of the diffusion model as its theoretical foundation. 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 [23], WIT [57], and MS COCO datasets [30].

Compared with existing black-box methods [34], [63], our attack under four attack scenarios can get 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 AUC scores 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 get at least 83% 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 [1] 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 [6], [20], [12].

**Contributions:** We make the following contributions.

- Many prior black-box attacks [17], [7], [61], [63], [34] 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. [58], 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 scores-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 that our attacks remain effective under 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  $\theta$  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 [24], GANs [15], and diffusion models [18]. Recently, diffusion models have gained significant traction. Building on the classical DDPM (Denoising Diffusion Probabilistic Models), a plethora of models, such as Imagen [46], DALL-E 3 [3], GLIDE [35], Stable Diffusion [44], 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:

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

In Equation 1,  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ , where  $\alpha_i$  is a predefined parameter that decreases incrementally within the interval  $[0, 1]$ . The term  $\epsilon_t$  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 (i.e., U-Net)  $\mathcal{U}_\theta$  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 [54], which has a deterministic reverse process.

2) *Prompt Guided Diffusion Models*: Diffusion models [3], [40], [46], [35], [44] 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 [44]. Distinct from other diffusion generative models [35], [40], [46], Stable Diffusion uniquely conducts both the forward and reverse processes within the latent space. 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  $\mathcal{M}_\theta$  trained with dataset  $\mathcal{D}$ . The attack  $\mathcal{A}$  will access  $\mathcal{M}_\theta$ ,  $\mathcal{D}$  and take data sample  $x$  as input, the output a bit  $b \leftarrow \mathcal{A}^\mathcal{D}(x, \mathcal{M}_\theta) \in \{0, 1\}$  indicating whether  $x$  was used in training (i.e.,  $x \in \mathcal{D}$ ) or not. For simplicity, we use  $\theta$  denoted model  $\mathcal{M}_\theta$  and omit  $\mathcal{D}$ .

Early MIAs predominantly target classification models and use the outputs from classifiers as the data to train their attack models [5], [26], [31], [47], [51], [9], [21], [45], [32], [33]. Shokri et al. [51] 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 predict vector or confidence scores, as well as labels such as members vs non-members, adversaries can subsequently train a binary classifier, acting as an attack model, to predict membership of  $x$  based on the data derived from the querying  $x$  on the target model.

Carlini et al. [5] suggest the use of confidence scores as attack features and the creation of two distributions,  $\mathbb{D}_{\text{in}}$  and  $\mathbb{D}_{\text{out}}$ , based on the confidence scores of samples from the member and non-member sets, respectively. The distributions are then utilized 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

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 [20]	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\times$
	LiRA [6]	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\checkmark$
	LOGAN [34]	$\circ$	$\circ$	$\circ$	$\circ$	$\circ$	$\times$
	GSA [37]	$\circ$	$\bullet$	$\circ$	$\circ$	$\circ$	$\checkmark$
Gray	SecMI [12]	$\circ$	$\bullet$	$\bullet$	$\circ$	$\circ$	$\checkmark$
	PIA [25]	$\circ$	$\circ$	$\bullet$	$\circ$	$\circ$	$\times$
	PFAMI [14]	$\circ$	$\bullet$	$\bullet$	$\circ$	$\circ$	$\checkmark$
Black	GAN-Leaks [34]	$\bullet$	$\circ$	$\bullet$	$\bullet$	$\circ$	$\times$
	Intuition-attack [61]	$\bullet$	$\circ$	$\bullet$	$\bullet$	$\circ$	$\times$
	Distribution-attack [63]	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\circ$	$\times$
	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	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\bullet$	$\checkmark$

for classification models cannot be directly applied to diffusion models. The focus of 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 are deferred to Section II-E1 and Section VII.

### D. Problem Formulation

1) *Threat Model*: Given the target 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



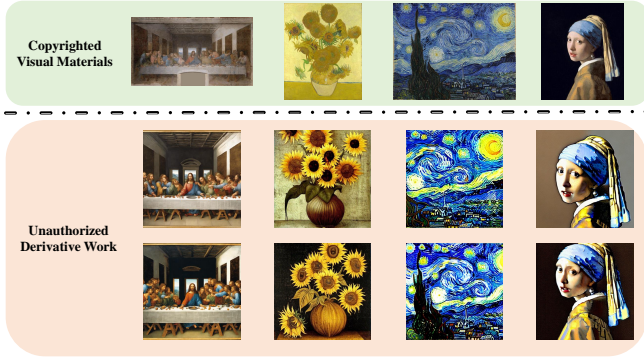


Fig. 1: The capability of current generative models to mimic artistic styles indicates their potential for creating derivatives of any artist’s work with adequate training data, thus leading to concerns about copyright and privacy implications.

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 ( $x = I_q$ ). 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 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  $\mathcal{D}'$ :  $\mathcal{D}^p$  and  $\mathcal{D}^s$ , indicating whether the attacker has access to real samples used to fine-tune  $\mathcal{G}$  or just samples from the same distribution.

2) *Motivation:* Our work aims to reveal privacy leakages of the fine-tuning dataset when fine-tuning a pre-trained image generative model. We focus on the fine-tuning phase because, from the model publishers’ perspective, training a generative model from scratch would entail significant computational costs, and the most efficient strategy for them would be to fine-tune a model that has already been pre-trained on a large-scale dataset. Given the full open-source nature of the Stable Diffusion [44], and the extensive availability of pre-trained models capable of generating photorealistic images from entities like CompVis<sup>2</sup> and Stability AI<sup>3</sup>, 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 Amazon<sup>4</sup>, OctoML<sup>5</sup>, and CoreWeave<sup>6</sup>, 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 leakage issues in this process and raise awareness of them.

Our work can also serve as an auditing tool to address related to copyright and privacy infringements. For instance,

if a malicious entity were to use a pre-trained Stable Diffusion model, and subsequently fine-tune it with artworks downloaded from the internet, it would be feasible to quickly train a generative model adept at replicating an artist’s unique style, leading to evident copyright violations. As depicted in Figure 1, for some renowned artworks that are already included in the model’s training set, high-quality replicated versions can now be generated. If malicious users intend to mimic an artist’s style and extensively fine-tune the model using the artist’s works, it is evident that such actions would pose significant infringements on the artist’s copyright.

## 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 [17] 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 [7] employs a similar intuition and, instead of density in the Monte-Carlo attack [17]), roughly speaking, use the shortest distance of the generated samples from the target sample as the criteria. 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 [17] and GAN-Leaks [7] 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 [7]. 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. [34] directly adopted the concept of GAN-Leaks [7] 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. [61] 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.

<sup>2</sup><https://huggingface.co/CompVis/stable-diffusion>

<sup>3</sup><https://github.com/Stability-AI/generative-models>

<sup>4</sup><https://aws.amazon.com/sagemaker/jumpstart/>

<sup>5</sup><https://octoml.ai/blog/the-beginners-guide-to-fine-tuning-stable-diffusion/>

<sup>6</sup><https://docs.coreweave.com/cloud-tools/argo>

Additionally, Zhang et al. [63] 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 (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

Drawing inspiration from GAN-Leaks, our approach aims to determine the membership of query samples through similarity score analysis. However, GAN-Leaks relies on Parzen window density estimation to estimate the probability of query samples [38]. 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 represent the use of similarity scores as an attack feature for determining the membership of query samples.

#### A. Theoretical Foundation

In this section, we aim to establish a detailed theory demonstrating the distance between the query image  $I_q$  and generated image  $I_g$  can be used as a metric to infer the membership of  $x$ . 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 [18], 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 [44], 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 [7] 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  $\hat{x}_\theta$ <sup>7</sup> 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  $x$  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  $\theta$  denotes the parameters of the model.

<sup>7</sup>We previously use  $\mathcal{U}_\theta$  to denote U-Net, now by slightly abusing notations we use  $\hat{x}_\theta$  for easier presentations.

The proof of Theorem 1 can be found at Appendix C-A. In Theorem 1, we show that the training objective for the diffusion model can be seen as reconstructing the training image at each timestep. During the training phase, the diffusion model is trained to predict the  $\epsilon_t$  using Equation 2. Because the existing of Equation 1 and  $\bar{\alpha}_t$  is a pre-defined hyperparameter,  $\hat{x}$  can be easily calculated using  $\epsilon_t$  at each step. Thus, a data sample from the diffusion models' member set should have higher similarity with its replication  $\hat{x}_\theta(x_t, t)$  at each timestep. Naturally, the denoised output from the diffusion model should have higher similarity scores with member set samples. Our work is focused on the black-box attack; the intermediate-generated images will not be used in our attack.

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 VAE and GAN. Hence, the existing black box attacks are to sample a large number of images from the models [34]. And then do the Monte Carlo [17] or GAN-Leaks [7] 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 Stable Diffusion and the unconditional diffusion model is that the former one can do 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. Theorem 2 is to prove this property persists in the Stable Diffusion.

**Theorem 2.** *In a well-trained Stable Diffusion model<sup>8</sup>,  $\hat{z}_\theta$ <sup>9</sup>, the query sample is  $x$ , and the membership of  $x$  is denoted as  $b$  (1/0 for member/non-member). The similarity scores remain a viable metric for assessing the membership of query data  $x$ . A pre-trained text encoder  $\phi_\theta$  is used to convert input conditional prompt information  $p$  into embeddings, which then guide image generation. 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,  $z_0 = \mathcal{E}(x)$ .  $\mathcal{E}$  is the encoder in Stable Diffusion.

The detailed proof of Theorem 2 is presented at Appendix C-B. Compared with the unconditional diffusion model in Theorem 1, Stable Diffusion's training objective function is similar and just adds a pre-trained encoder  $\phi_\theta$  and transfers the diffusion process into latent space. Therefore, we can prove the probability of  $x$  being a member set sample is inversely proportional to

$$\|D(z_0) - D(\hat{z}_\theta(z_t, t, \phi_\theta(p)))\|_2^2.$$

<sup>8</sup>In our work, we used pre-trained Stable Diffusion-v1.5 from CompVis, which is trained for 150,000 A100 hours.

<sup>9</sup> $\hat{z}_\theta$  represents only the U-Net in Stable Diffusion, not including the VAE and text encoder.

This representation quantifies the distance between the synthesized original image and the actual ground truth during training. Considering the realistic situation and settings, we designed four attacks (as shown in Section II-D1) to use this property in different scenarios. However, for general representation, we simplify 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 (i.e., Cosine similarity,  $\ell_1$  or  $\ell_2$  distance, or Hamming distance). Given that a higher similarity score 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 basic inference model relies on computing the similarity 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  $x$ , to which  $I_q$  belongs, originates 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 in Attack-II and Attack-IV, we incorporate the captioning model in our work.

**Image Feature Extractor.** As we follow the high-level intuition of GAN-Leaks and use image similarities to determine membership, we need a metric to formally quantify this similarity. It has been observed that the semantic-level similarities are substantially more effective than pixel-level similarities [61]. So we utilize a pre-trained image encoder (i.e., DETR, BEiT, EfficientFormer, ViT, DeiT) to extract semantic information from the images.

**Captioning Model.** As previously mentioned in Section II-D1, in some scenarios, the data record  $x$  lacks the text component. Consequently, we resort to a captioning model to generate the corresponding text. For our experiments, we utilize BLIP2 [27] as the captioning model. To ensure that 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 target image and compute a similarity score used for MIAs (specific instantiations of  $\mathcal{A}$  to be presented in Section III-C). According to 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 victim model  $m$  times to generate  $m$  images. For each generated image, we calculate a similarity score relative to the query image. Finally, we return these as an  $m$ -dimensional similarity score to determine the target/query data’s membership.

---

### Algorithm 1 High-level Overview of Our Attack.

---

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

- 1: **if**  $T_q \notin x$  **then**
- 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:**  $\mathcal{A} [\langle S(E(I_q), E(I_g^i)) \rangle_{i=1}^m]$  ▷ MIA results.

---

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.** The threshold-based MIA takes the similarity scores, apply a statistical function (e.g., mean, median)  $f$  to it, and then decide membership if the result is above a pre-defined threshold  $\tau$ , i.e.,

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

**Distribution-based Membership Inference Attack.** Following the work by Carlini et al. [5], 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

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

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

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

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

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

and

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

**Classifier-based Membership Inference Attack.** Given



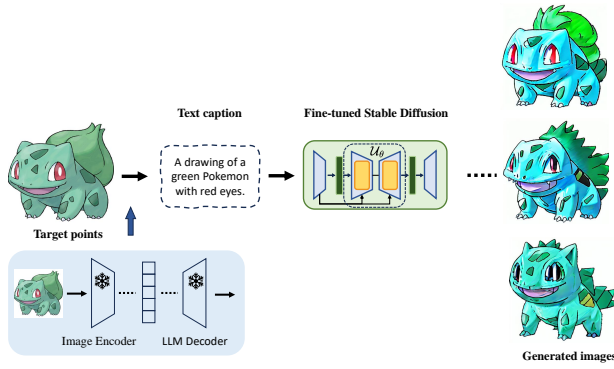


Fig. 2: Composition of target points using solely  $I_q$  with text caption generation by BLIP2 [27] captioning model, followed by image generation denoted as  $I_g$  using the fine-tuned Stable Diffusion model.

that the obtained similarity score is represented as a high dimensional vector, the classifier-based MIA feeds  $f \left[ \langle S(E(I_q), E(I_g^i)) \rangle_{i=1}^m \right]$  directly into a classifier (we use a multilayer perceptron in our evaluation). This approach aligns with the methods of Shokri et al. [51], 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 [48] and LAION-Aesthetics. To guarantee the integrity and effectiveness of our work, we utilize the MS COCO [30], CelebA-Dialog [23], and WIT datasets [57] for evaluation, ensuring that there is no overlap with the pretraining dataset. MS COCO is widely used for training and testing various text-to-image models. The CelebA-Dialog dataset, with its extensive facial data and descriptions, aligns well with our interest in sensitive data descriptions. Meanwhile, the WIT dataset, curated from web scraping Wikipedia for images and their associated text descriptions, offers a diverse range of images and distinct textual styles, serving as an excellent benchmark for assessing model robustness.

**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 [30] has been extensively utilized in various image generation models, including experiments on DALLE 2 [40], Imagen [46], GLIDE [35], and VQ-Diffusion [16]. 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

TABLE II: The default parameters used in Section V.

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

text-based facial generation, manipulation, and face image captioning. Facial information has consistently been regarded as private; hence, utilizing CelebA-Dialog [23] in this study aligns with our objective of detecting malicious users fine-tuning the Stable Diffusion model [44] 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], [20], [34], [25], [61], [6], [5], we will employ Attack Success Rate (ASR), Area Under the 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 v1-5<sup>10</sup> checkpoints as our pre-trained models. 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 II.

##### C. Competitors

For our evaluation, we first compare our work with existing black-box attacks on diffusion models [34], [14]. 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. [34]** employed the full-black attack framework from GAN-Leaks.

**Zhang et al. [63]** 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

<sup>10</sup><https://huggingface.co/runwayml/stable-diffusion-v1-5>

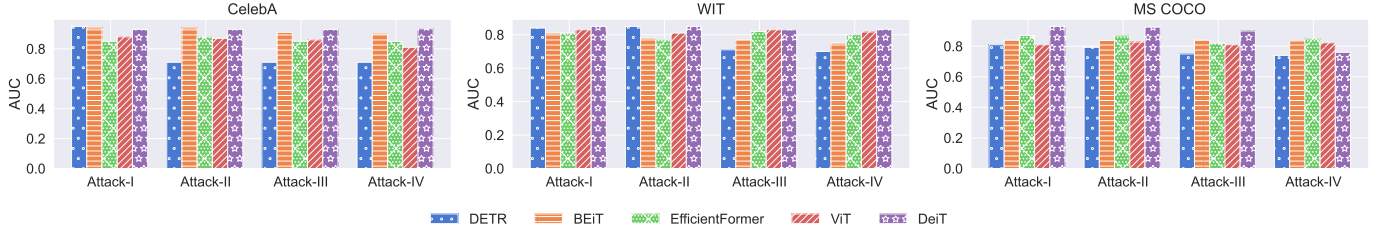


Fig. 3: AUC results on three datasets and four attack scenarios comparing five different image feature extractors.

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  $I_q$ , then employ the similarity between  $I_q$  and  $I_q$  to ascertain the membership of  $x$ .

**Attack-II** ( $x = I_q, \mathcal{D}' \cap \mathcal{D}_m \neq \emptyset$ ) In this scenario, the attacker does not possess a conditional prompt that can be directly fed into the target model. As illustrated in Figure 2, the attack scenario necessitates the use of 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** ( $x = I_q, \mathcal{D}' \cap \mathcal{D}_m = \emptyset$ ) is similar to the second 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 III. 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. [63]’s approach 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, 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

TABLE III: Accuracy comparison between the attacks by Zhang et al. [63] and Matsumoto et al. [34] (Applying GAN-Leaks against the diffusion model) versus our methods: Attack-I, Attack-II, Attack-III, and Attack-IV, illustrating the distinct success rates.

Attack type	CelebA-Dialog		
	ASR	AUC	TPR@FPR=1%
Matsumoto et al. [34]	0.52	0.50	0.01
Zhang et al. [63]	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

the attack. Under such constraints, the effectiveness of attacks by Zhang et al. [63] 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. [34] 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. 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 DETR [4], BEiT [2], EfficientFormer [29], ViT [11], and DeiT [59] five distinct image feature extractors to generate image embeddings, with the aim of observing the impact of various image features on the success rate of attacks. Furthermore, 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



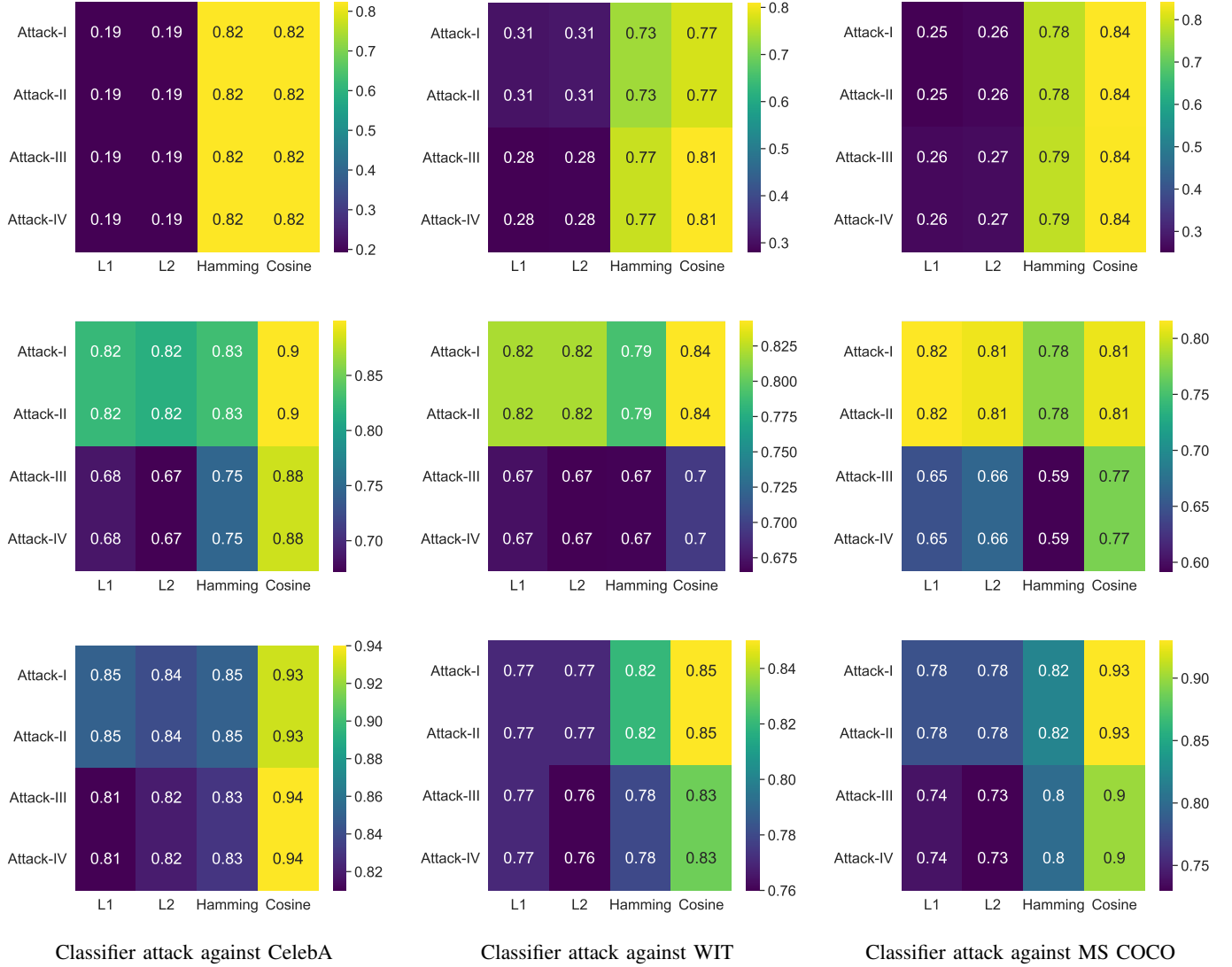


Fig. 4: Experimental evaluation of three attack types (*threshold-based*, *distribution-based*, *classifier-based*) across three datasets in four scenarios (**Attack-I**, **Attack-II**, **Attack-III**, **Attack-IV**) highlighting Cosine similarity’s superior and stable performance across all metrics and attack types.

**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` [59] 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 experimental investigations.

A more comprehensive comparison including **threshold-based** and **distribution-based** of these five image encoders is presented in [Appendix E](#).

### C. Different Distance Metrics

In the previous section, we picked `DeiT` [59] as the most stable and efficient image feature extractor. However, our attack framework also necessitates a reliable and consistent similarity metric to compute the closeness between embed-

dings. We conducted systematic and extensive experiments, and as demonstrated in [Figure 4](#), we thoroughly assessed various attack scenarios and types across all datasets to test their effects on Cosine similarity,  $\ell_1$  distance,  $\ell_2$  distance, and Hamming distance.

From [Figure 4](#), 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,  $\ell_1$  and  $\ell_2$  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.

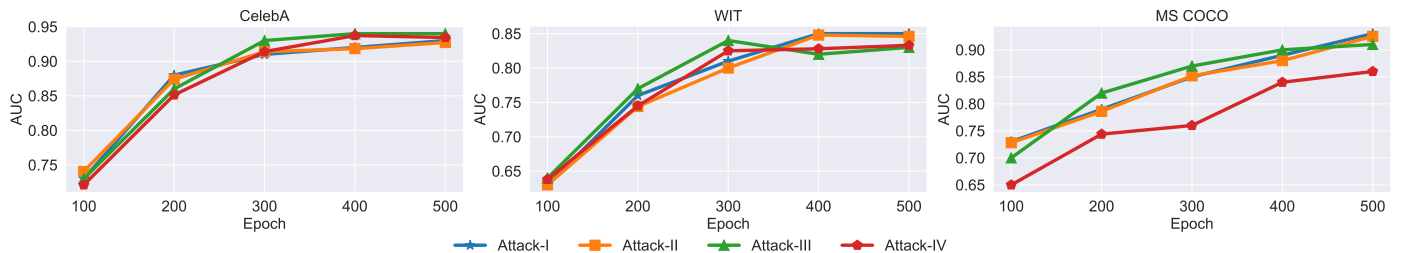


Fig. 5: Relationship between epoch progression and AUC score in **Attack-I**, **Attack-II**, **Attack-III**, and **Attack-IV**, indicating increasing memorization within image generation models over fine-tuning epochs.

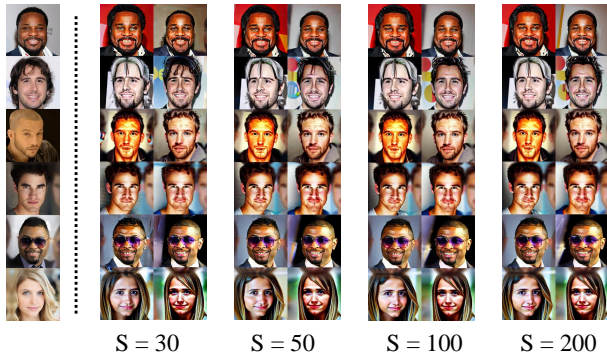


Fig. 6: Model fine-tuning with CelebA-Diolog and image synthesis at inference steps of 30, 50, 100, and 200.

#### D. 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 II](#). 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.

In [Figure 5](#), 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 extra biases that might 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 F](#) for reference.

TABLE IV: Alignment with DDIM [54] denoting ‘S’ as inference steps; experimentation under **Attack-III** scenario measuring FID at varying inference step counts.

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

#### E. Impact of Number of Inference Step

The quality of images generated by current diffusion models, including the Stable Diffusion [44] presented in our work, is influenced not only by the number of fine-tuning steps but also significantly by the number of inference steps. These models predominantly utilize Denoising Diffusion Implicit Models [54] as their sampling method. The Frchet Inception Distance is able to shift significantly from 13.36 to 4.04 when varying the sampling steps from 10 to 1000. This dramatic 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 IV](#), the variations in attack accuracy are not immediately pronounced. However, upon a broader examination, it becomes evident that as the number of S (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 doesn’t lead to a substantial boost in attack success rate, we present samples generated at different inference steps in [Figure 6](#). 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 doesn’t drastically impact the attack success rate.

The experimental results obtained from the additional two datasets are presented in [Appendix G](#).

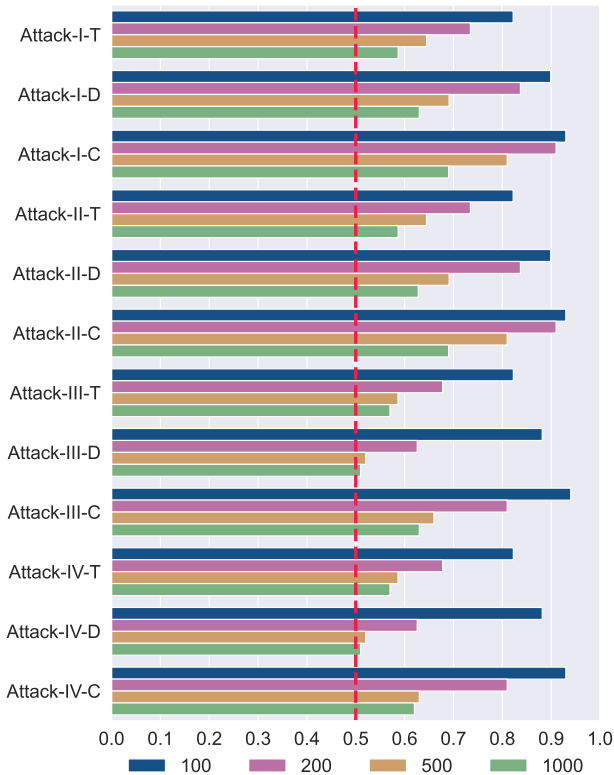


Fig. 7: Attack nomenclature and performance trends: ‘T’ for threshold-based, ‘D’ for distribution-based, and ‘C’ for classifier-based attacks, with accuracy inversely related to training set size.

#### F. Impact of Different Size of Auxiliary Dataset

From our observations across white-box [6], [34], [20], gray-box [12], [25], [20], and black-box attacks [34], the accuracy of these attacks is significantly influenced by the size of training set. As the training set of the target model,  $\mathcal{G}^t$ , 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 only 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 V: Use of Kandinsky [42] as shadow model and Stable Diffusion [44] as target model in conducting attacks, demonstrating the maintained efficacy of all four attack methodologies.

Dataset	A-I-S	A-I-A	A-II-S	A-II-A	A-III-S	A-III-A	A-IV-S	A-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

#### G. Impact of the Selection of Shadow Models

To enhance the universality 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 will employ a conditional image generator, Kandinsky [42], which has a different architectural design from the Stable Diffusion [44], as our shadow model. This model will be fine-tuned using the same auxiliary dataset mentioned in the Section IV-A, and the results will be displayed in Table V.

In Table V, 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.

#### H. 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 VI, 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



TABLE VI: Impact of not fine-tuning the captioning model on the success rates of [Attack-II](#) and [Attack-IV](#) across various datasets.

Dataset	Attack-II		Attack-IV	
	With tuning	Without tuning	With tuning	Without 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

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 reason is the image captioning model may have introduced biases in the generated  $T_q$ , 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. 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 caption 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

Our approach aligns with other membership inference attacks against diffusion models [\[6\]](#), [\[7\]](#), [\[12\]](#), [\[61\]](#). We aim to employ Differential Privacy Stochastic Gradient Descent (DP-SGD) [\[1\]](#) to evaluate the robustness of our attack. The primary purpose of this defense method is to prevent the inference of training samples from the model by diminishing the model’s memorization of individual samples, achieved by introducing noise into the gradient updates during model training. Implementing DP-SGD in other membership inference attacks targeting diffusion models [\[6\]](#), [\[12\]](#), [\[61\]](#) has consistently hindered normal model convergence, resulting in the generation of low-quality images.

In this part, we test our four attacks and employ 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  $\epsilon \in \{1, 4, 10\}$  (different  $\epsilon$  gives different noise multiplier  $\sigma$ ).

We show the experimental results in [Table VII](#). It illustrates that the attack success rate of four attacks significantly decreases after implementing DP-SGD [\[1\]](#) as the defensive method. When we use 100 samples to fine-tune the model

and set  $\epsilon = 1$ , both [Attack-III](#) and [Attack-IV](#) have been greatly impacting, dropping to around 50% (random guess). For [Attack-I](#) and [Attack-II](#), both attacks allow attackers to be able to access the target model’s data sample partially. Therefore, it can still get over 55% attack accuracy and seems robust to the defense. We also measure attack success rate at low FPRs, according to Carlini et al. [\[5\]](#). From [Table VII](#), we noticed the TPR value when FPR=1% has dropped from 0.58 and 0.51 to 0.01 and 0.01. This phoneme shows that [Attack-I](#) and [Attack-II](#) also lose their functionality in these defense settings.

When we change the  $\epsilon$  value from 1 to 4 and 10, the attack accuracy increases but still cannot show their effectiveness. We hypothesize that our attack relies on the model’s memorization of training data, which enables the use of similarity scores as an indicator for assessment. However, the application of DP-SGD reduces the model’s memorization capacity for training data, leading to decreased similarity scores in member set samples and, resulting in a lower attack success rate. The experiment outcome has aligned with other works [\[6\]](#), [\[20\]](#), [\[12\]](#).

## VII. RELATED WORK

We review further related works on membership inference attacks and extraction 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 (i.e., predict vector [\[8\]](#), [\[21\]](#), [\[22\]](#), [\[32\]](#), [\[50\]](#), [\[51\]](#), top-k confidence score [\[47\]](#), [\[51\]](#)); 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. [\[20\]](#) and Matsumoto et al. [\[34\]](#) adopt approaches similar to that of Yeom et al. [\[62\]](#), determining membership by comparing the loss at various timesteps to a specific threshold. Carlini et al. [\[6\]](#) argue that mere threshold-based determinations are insufficient and proposed training multiple shadow models and utilizing the distribution of loss across each timestep established by these shadow models to execute an online LiRA attack [\[5\]](#). Pang et al. [\[37\]](#) 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, Duan et al. [\[12\]](#) and Kong et al. [\[25\]](#) leveraged the deterministic properties of generative process in DDIM [\[54\]](#) for their attack designs. Duan et al. [\[12\]](#) employed the approximated posterior estimation error as attack features, while Kong et al. [\[25\]](#) 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

TABLE VII: Explore the attack accuracy under DP-SGD defense our four attack methods’ accuracy declines. Experiments include two different sizes of fine-tuning datasets and three  $\epsilon$  values. ‘Vanilla’ means without the DP-SGD version.

		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	0.665	0.713	0.12
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	0.673	0.700	0.05

predicted value. Fu et al. [14] use the intermediate output to calculate the probabilistic fluctuations between target points and neighboring points.

### C. Extracting Attack

In recent few years, not only do the owners of models potentially infringe upon privacy through the misuse of data in training, but users of the models may also violate privacy by extracting sensitive training data from the models. This is particularly concerning for image generation models trained on high-value data, such as medical imaging models [52]. The training data for these models constitute private information. However, there currently exist extracting attacks targeting generative models [6], capable of extracting training samples from the models. Undoubtedly, this poses a significant infringement on the intellectual property rights of the model owners.

Carlini et al. [6] posited that models are more prone to memorizing duplicated samples, a hypothesis central to their strategy for extracting data attack. Rather than a bit-by-bit comparison, they opted for comparing CLIP embeddings due to the dataset’s vast size. This approach facilitated the identification of potential duplicates based on the  $\ell_2$  distance between samples. The crux of their attack involved using the captions of these identified duplicates. Specifically, for each duplicated image, its corresponding caption was input 500 times, generating 500 candidate images. These images were then analyzed within a constructed graph, where connections were drawn between pairs of candidate images separated by less than a threshold distance  $d$ . The key indicator of a successful attack was the identification of the largest clique in the graph, comprising at least 10 nodes, which was considered evidence of a memorized image.

## 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-H, 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.

## REFERENCES

- [1] 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.
- [2] H. Bao, L. Dong, S. Piao, and F. Wei, “Beit: Bert pre-training of image transformers,” 2022.
- [3] 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>
- [4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” 2020.
- [5] N. Carlini, S. Chien, M. Nasr, S. Song, A. Terzis, and F. Tramer, “Membership inference attacks from first principles,” 2022.
- [6] N. Carlini, J. Hayes, M. Nasr, M. Jagielski, V. Sehwag, F. Tramer, 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.
- [7] 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.
- [8] 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.
- [9] C. A. Choquette-Choo, F. Tramer, N. Carlini, and N. Papernot, “Label-only membership inference attacks,” 2021.
- [10] T. Dockhorn, T. Cao, A. Vahdat, and K. Kreis, “Differentially private diffusion models,” *arXiv preprint arXiv:2210.09929*, 2022.
- [11] 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.
- [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] V. Fernandez, P. Sanchez, W. H. L. Pinaya, G. Jacenków, S. A. Tsaftaris, and J. Cardoso, “Privacy distillation: reducing re-identification risk of multimodal diffusion models,” *arXiv preprint arXiv:2306.01322*, 2023.
- [14] 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.

- [15] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.
- [16] 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.
- [17] B. Hilprecht, M. Härterich, and D. Bernau, "Monte carlo and reconstruction membership inference attacks against generative models," *Proceedings on Privacy Enhancing Technologies*, 2019.
- [18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," 2020.
- [19] J. Ho and T. Salimans, "Classifier-free diffusion guidance," 2022.
- [20] H. Hu and J. Pang, "Membership inference of diffusion models," *arXiv preprint arXiv:2301.09956*, 2023.
- [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] B. Jayaraman, L. Wang, K. Knipmeyer, Q. Gu, and D. Evans, "Revisiting membership inference under realistic assumptions," 2021.
- [23] 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.
- [24] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," 2022.
- [25] 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.
- [26] 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.
- [27] 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. 19 730–19 742.
- [28] 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.
- [29] Y. Li, G. Yuan, Y. Wen, J. Hu, G. Evangelidis, S. Tulyakov, Y. Wang, and J. Ren, "Efficientformer: Vision transformers at mobilenet speed," 2022.
- [30] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollr, "Microsoft coco: Common objects in context," 2015.
- [31] Y. Liu, Z. Zhao, M. Backes, and Y. Zhang, "Membership inference attacks by exploiting loss trajectory," 2022.
- [32] 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.
- [33] 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.
- [34] T. Matsumoto, T. Miura, and N. Yanai, "Membership inference attacks against diffusion models," 2023.
- [35] 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.
- [36] A. B. Owen, "Monte carlo theory, methods and examples," 2013.
- [37] 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.
- [38] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [39] 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.
- [40] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical text-conditional image generation with clip latents," 2022.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] A. Sablayrolles, M. Douze, Y. Ollivier, C. Schmid, and H. Jgou, "White-box vs black-box: Bayes optimal strategies for membership inference," 2019.
- [46] 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.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] R. Shokri, M. Strobil, and Y. Zick, "On the privacy risks of model explanations," 2021.
- [51] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," 2017.
- [52] N. K. Singh and K. Raza, "Medical image generation using generative adversarial networks," 2020.
- [53] 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.
- [54] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," 2022.
- [55] 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.
- [56] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," 2021.
- [57] 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.
- [58] F. Suya, A. Suri, T. Zhang, J. Hong, Y. Tian, and D. Evans, "Sok: Pitfalls in evaluating black-box attacks," *arXiv:2310.17534*, 2023.
- [59] 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. 10 347–10 357.
- [60] S. Truex, L. Liu, M. E. Gursay, 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.
- [61] Y. Wu, N. Yu, Z. Li, M. Backes, and Y. Zhang, "Membership infer-



ence attacks against text-to-image generation models,” *arXiv preprint arXiv:2210.00968*, 2022.

- [62] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha, “Privacy risk in machine learning: Analyzing the connection to overfitting,” 2018.
- [63] 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.

## APPENDIX A MORE DETAILS FOR DIFFUSION MODELS

Diffusion models have two phases: the forward diffusion process and the reverse denoising process. In the forward process, an image  $x$  is sampled from the true data distribution. The image  $x$  undergoes a series of noise addition steps for  $T$  iterations, resulting in a sequence  $x_1, x_2, \dots, x_{T-1}, x_T$ , until  $x_T$  becomes an image equation to an isotropic Gaussian noise distribution. The magnitude of noise introduced at each step is controlled by a parameter  $\alpha_t$ , where  $\alpha_t \in [0, 1]$ , which gradually decreases over time. At step  $t$ , the noise image  $x_t$  can be represent as:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1 - \alpha_t}\epsilon; \quad \epsilon \sim \mathcal{N}(0, 1)$$

Given that the original work [18] considers the forward process as a Markov chain and employs the reparameterization trick, it is possible to directly derive the noisy image  $x_t$  at step  $t$  from the original image  $x_0$ . Based on the definition  $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ , the training objective for diffusion models is to obtain a denoising network capable of sampling  $x_{t-1}$  from  $x_t$  according to the distribution  $\mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \sigma_t^2 \mathbf{I})$ . When conditioned on  $x_0$  and  $x_t$  from  $q(x_{t-1}|x_t, x_0)$ , the ground-truth distribution of  $x_{t-1}$  is given by  $\mathcal{N}(x_{t-1}; \tilde{\mu}_t(x_t, x_0), \sigma_t^2 \mathbf{I})$ . Given that variance is fixed as a hyperparameter, the focus is on calculating the difference between  $\mu_\theta(x_t, t)$  and  $\tilde{\mu}_t(x_t, x_0)$ . 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

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

As a result, by deriving  $x_t$  from  $x_0$  using Equation 1 and omitting the weight term as suggested by Ho et al. [18], the loss function can be articulated as presented in Equation 2.

## APPENDIX B MORE DETAILS FOR CLASSIFIER-FREE GUIDANCE

As the field advances, diffusion models can create content from user-given prompts, primarily using classifier-free guidance [19]. Many diffusion models, including Imagen [46], DALL-E 2 [40], and Stable Diffusion [44], which utilize the classifier-free guidance mechanism, are trained on dual objectives; however, they can be represented with a single model during training by probabilistically setting the conditional prompt to null. 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  $w$  represents the guidance scale factor, where a higher value of  $w$  results in improved alignment between image and text at the potential expense of 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.

$$\begin{aligned} \log p(x) &\geq \mathbb{E}_{q(x_{1:T}|x_0)} \left[ \log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)} \right] \\ &\dots \\ &= \underbrace{\mathbb{E}_{q(x_1|x_0)} [\log p_\theta(x_0|x_1)]}_{L_0} - \underbrace{\mathcal{D}_{KL}(q(x_T|x_0) \| p(x_T))}_{L_T} \\ &\quad - \underbrace{\sum_{t=2}^T \mathbb{E}_{q(x_t|x_0)} [\mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))]}_{L_{t-1}} \end{aligned} \quad (7)$$

The primary focus of optimization is on  $L_{t-1}$ , as explicated in the original work [18]. 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 [18], 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  $\theta$ . 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  $\epsilon_t$  as the targeted prediction of the diffusion model, aligning with the optimization objectives stated in both DDPM [18] and DDIM [54]. 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 [44], 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  $\mathcal{U}_\theta$  predicts the noise at timestep  $t$  based on  $z_t$  and the embedding generated by  $\phi_\theta$ , which takes  $p$  as its input. Given that the forward process of the Stable Diffusion [44] 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.** Hilprecht et al. [17] argue that generative models can overfit and memorize data due to their ability to capture specific details. Given a query sample  $x$ , attackers can utilize the generative model to sample  $k$  images. Define an  $\epsilon$ -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 [36], the Monte Carlo attack can be expressed as:

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

Furthermore, they try to employ the Kernel Density Estimator (KDE) [38] as a substitute for  $\hat{f}_{MC-\epsilon}(x)$ . The estimation

of the likelihood of  $x$  using KDE can be expressed as:

$$\hat{f}_{KDE}(x) = \frac{1}{nh^d} \sum_{i=1}^k K\left(\frac{x - x'_i}{h^d}\right)$$

where  $h_d$  is the bandwidth and  $K$  is the Gaussian kernel function. If  $x \in \mathcal{D}_m$ , the likelihood  $\hat{f}_{KDE}(x)$  should be substantially higher than the likelihood when  $x \in \mathcal{D}_{nm}$ . However, upon experimentation, it was observed that using  $\hat{f}_{KDE}(x)$  as the criterion for the attack did not yield attack accuracy better than random guessing.

**GAN-Leaks Attack.** In a further examination of the memorization with generative models, Chen et al. [7] 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 further articulated this observation as:

$$\Pr[y_q = 1|x, \theta] \propto \Pr_{\mathcal{G}}[x|\theta_v]$$

However, due to the inability to represent generated data distribution with an explicit density function, computing the precise probability becomes intractable. Therefore, Chen et al. [7] also employed the KDE method [38] 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  $P_z$ . Besides, Chen et al. [7] propose the approximation of Equation 14:

$$\Pr_{\mathcal{G}}(x|\theta_v) \approx \frac{1}{k} \sum_{i=1}^k \exp(-d(x, \mathcal{G}(z_i))); \quad z_i \sim P_z \quad (15)$$

For the  $k$  samples from  $\mathcal{G}$ , we use the distance metric  $d(\cdot, \cdot)$  to measure and sum the distances between each sample  $g_i$  and the query point  $x$ . Equation 13 and Equation 15 indicate that the only practical way to improve attack success rates in models with such stochastic sampling is by significantly increasing the number of samples  $g_i$ . For a full-black attack, around  $100k$  samples are required for a query point  $x$  to achieve an attack AUC close to 0.60. This undoubtedly results in significant overhead. Then, Chen et al. [7] introduced the concept of a partial-black attack. Specifically, the attacker first employs

$$z^* = \arg \min_z L(x, \mathcal{G}(z))$$

to identify the optimal latent code  $z^*$ . Subsequently,  $g_i$  is generated using  $\mathcal{G}(z^*)$  to compare with  $x$ . The partial-black attack method boosts the success rate of attacks and uses fewer data samples, but requires finding the optimal  $z^*$ . In GANs, the input latent code  $z$  is typically a 100-dimensional random noise. However, in newer conditional diffusion models, the complexity and size of input embeddings have greatly increased. Additionally, the latest diffusion models use explicit prompt information instead of random latent codes, making the partial-black attack strategy used for GANs unsuitable for these new-generation generative models.

APPENDIX E  
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.

TABLE VIII: Comparative analysis of five different image encoders using *threshold-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	0.57	0.64	0.02	0.79	0.86	0.41	0.70	0.76	0.26	0.73	0.78	0.01	0.75	0.82	0.43
	Attack-II	0.60	0.64	0.02	0.77	0.86	0.41	0.70	0.76	0.29	0.73	0.78	0.18	0.76	0.82	0.45
	Attack-III	0.57	0.65	0.01	0.79	0.86	0.40	0.71	0.76	0.23	0.67	0.79	0.12	0.75	0.82	0.20
	Attack-IV	0.59	0.66	0.05	0.78	0.86	0.39	0.68	0.76	0.24	0.69	0.76	0.13	0.77	0.84	0.32
WIT	Attack-I	0.59	0.66	0.07	0.56	0.62	0.06	0.66	0.76	0.12	0.72	0.79	0.02	0.68	0.77	0.17
	Attack-II	0.57	0.66	0.14	0.57	0.62	0.05	0.61	0.70	0.03	0.57	0.61	0.01	0.62	0.67	0.02
	Attack-III	0.60	0.66	0.02	0.60	0.61	0.03	0.66	0.71	0.09	0.63	0.71	0.03	0.64	0.69	0.13
	Attack-IV	0.57	0.62	0.01	0.66	0.73	0.03	0.69	0.80	0.13	0.70	0.82	0.02	0.73	0.80	0.20
MS COCO	Attack-I	0.71	0.73	0.01	0.71	0.80	0.14	0.74	0.82	0.11	0.68	0.77	0.17	0.79	0.84	0.05
	Attack-II	0.63	0.72	0.01	0.75	0.80	0.15	0.75	0.82	0.11	0.70	0.78	0.18	0.79	0.85	0.06
	Attack-III	0.63	0.72	0.01	0.69	0.81	0.23	0.76	0.82	0.15	0.68	0.79	0.01	0.76	0.84	0.13
	Attack-IV	0.61	0.72	0.09	0.71	0.81	0.24	0.78	0.82	0.17	0.68	0.79	0.24	0.70	0.74	0.01

TABLE IX: Comparative analysis of five different image encoders using *distribution-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	0.62	0.66	0.03	0.76	0.90	0.65	0.70	0.83	0.41	0.74	0.84	0.40	0.73	0.90	0.61
	Attack-II	0.61	0.66	0.03	0.79	0.90	0.66	0.71	0.82	0.32	0.73	0.85	0.39	0.74	0.90	0.64
	Attack-III	0.56	0.58	0.01	0.74	0.85	0.51	0.61	0.71	0.20	0.64	0.74	0.13	0.76	0.88	0.50
	Attack-IV	0.59	0.61	0.01	0.72	0.86	0.61	0.61	0.70	0.21	0.67	0.73	0.16	0.77	0.88	0.52
WIT	Attack-I	0.66	0.72	0.12	0.70	0.83	0.27	0.70	0.82	0.22	0.70	0.85	0.30	0.69	0.84	0.41
	Attack-II	0.58	0.68	0.07	0.70	0.81	0.14	0.66	0.78	0.19	0.71	0.84	0.23	0.68	0.84	0.34
	Attack-III	0.57	0.57	0.01	0.62	0.69	0.15	0.62	0.68	0.20	0.60	0.67	0.10	0.61	0.70	0.26
	Attack-IV	0.51	0.55	0.01	0.61	0.71	0.14	0.60	0.66	0.23	0.56	0.64	0.11	0.64	0.69	0.09
MS COCO	Attack-I	0.62	0.71	0.19	0.65	0.80	0.33	0.67	0.79	0.14	0.59	0.73	0.43	0.73	0.81	0.36
	Attack-II	0.60	0.71	0.18	0.64	0.80	0.32	0.68	0.80	0.15	0.61	0.74	0.41	0.72	0.81	0.36
	Attack-III	0.56	0.57	0.03	0.62	0.70	0.09	0.61	0.63	0.10	0.61	0.67	0.06	0.70	0.77	0.13
	Attack-IV	0.55	0.57	0.02	0.63	0.70	0.06	0.59	0.63	0.12	0.62	0.67	0.07	0.62	0.67	0.07

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	0.933	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 F 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.

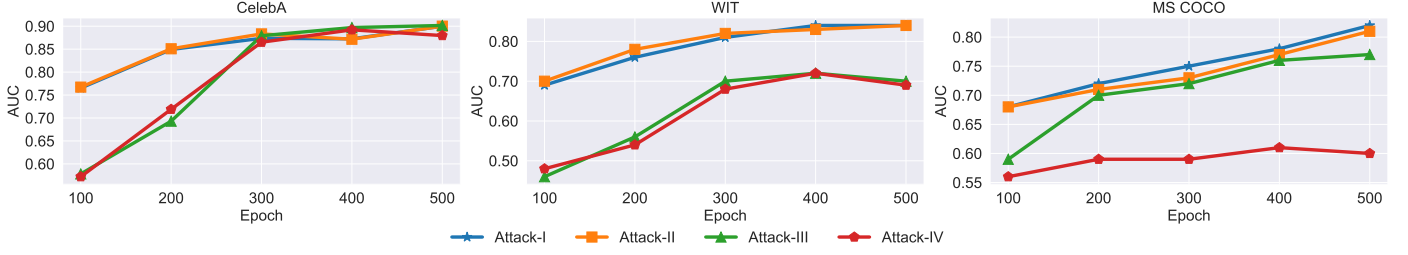


Fig. 8: Correlation between increased fine-tuning steps and enhanced accuracy of *distribution-based* attack.

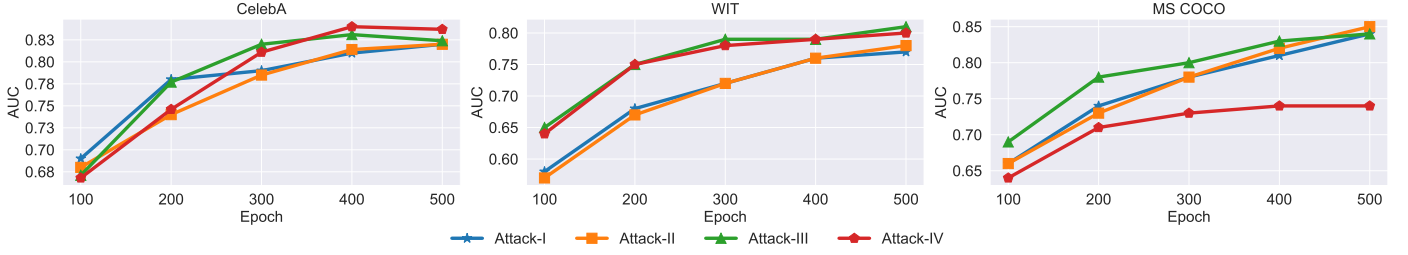


Fig. 9: Increase in success rate of *threshold-based* attack with more fine-tuning steps

## APPENDIX G 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.

TABLE XI: Experiment results for more inference steps on MS COCO and WIT.

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