White-box Membership Inference Attacks against Diffusion Models

Yan Pang
University of Virginia
yanpang@virginia.edu

Tianhao Wang
University of Virginia
tianhao@virginia.edu

Xuhui Kang
University of Virginia
qhv6ku@virginia.edu

Mengdi Huai
Iowa State University
mdhuai@iastate.edu

Yang Zhang
CISPA Helmholtz Center for
Information Security
zhang@cispa.de

Abstract

Diffusion models have begun to overshadow GANs and other generative models in industrial applications due to their superior image generation performance. The complex architecture of these models furnishes an extensive array of attack features. In light of this, we aim to design membership inference attacks (MIAs) catered to diffusion models. We first conduct an exhaustive analysis of existing MIAs on diffusion models, taking into account factors such as black-box/white-box models and the selection of attack features. We found that white-box attacks are highly applicable in real-world scenarios, and the most effective attacks presently are white-box. Departing from earlier research, which employs model loss as the attack feature for white-box MIAs, we employ model gradients in our attack, leveraging the fact that these gradients provide a more profound understanding of model responses to various samples. We subject these models to rigorous testing across a range of parameters, including training steps, timestep sampling frequency, diffusion steps, and data variance. Across all experimental settings, our method consistently demonstrated near-flawless attack performance, with attack success rate approaching 100% and attack AUCROC near 1.0. We also evaluated our attack against common defense mechanisms, and observed our attacks continue to exhibit commendable performance. We provide access to our code[1].

Keywords

machine learning privacy, membership inference attack

1 Introduction

Recently, diffusion models have gained significant attention, and various applications are emerging. These models [21, 39, 42, 43, 48, 53, 56] rely on a progressive denoising process to generate images, resulting in improved image quality compared to previous models like GANs [7, 10] and VAEs [32]. Leading models primarily fall into two categories. The first category encompasses diffusion-based architectures such as GLIDE [39], Stable Diffusion model [45], DALL-E 2 [42], and Imagen [48]. The second category comprises representative sequence-to-sequence models like DALL-E [43], Parti [66], and CogView [11]. Current text-to-image models possess the capability to generate exquisite and intricately detailed images based on textual inputs, finding extensive applications across various domains such as graphic design and illustration. While diffusion models can be employed to synthesize distinct artistic styles, they often necessitate training on extensive sets of sensitive data. Thus, investigating membership inference attacks (MIAs) [52], which aim to determine whether specific samples are present in the diffusion model's training data, is of paramount importance.

Numerous studies have been conducted on classification models [3, 31, 33, 50, 52, 63, 64], GANs [5, 19, 20, 26, 37], and others. However, due to the unique training and inference method of diffusion models, previous attack methods [64] are no longer suitable. For instance, in classification models, the model's final output is generally used as the attack feature, relying on the model's overfitting to the training data, which leads to differences in classification confidence. Additionally, previous work on generative models such as GANs focused on utilizing the discriminator for determination [37]. Since the diffusion model does not have a discriminator, which makes it different from GANs, a new attack method must be specifically designed for diffusion models.

Some preliminary efforts have been devoted to conducting MIA on diffusion models [4, 27, 35]. However, it merits our attention that these investigations, akin to many others in this domain, predominantly concentrate on loss- and threshold-based attacks. We postulate that different layers in a neural network learn distinct features and, therefore, store varying amounts of information [65]. Evaluations based solely on loss could potentially overlook substantial information [37]. Consequently, a more comprehensive perspective of the model's response to a sample could be attained by considering gradient information from each layer post-backpropagation in addition to the loss incurred by the model.

The main challenges of utilizing gradients for MIAs are the excessive computation overhead and the overfitting issue of training the attack model (given the large size of diffusion models, gradients could have millions of dimensions). We carefully analyze ways to reduce dimensionality and propose a framework incorporating subsampling and aggregation. We call our framework Gradient attack based on Subsampling and Aggregation (GSA) and initiate

[1] https://github.com/py85252876/GSA

Proceedings on Privacy Enhancing Technologies 2025(2), 398–415
© 2025 Copyright held by the owner/author(s).
https://doi.org/10.56553/popets-2025-0068

two instances, GSA$_1$ and GSA$_2$, demonstrating different trade-offs within the GSA framework.

To ensure the comprehensiveness and integrity of our investigation, we conduct experiments on the fundamental unconditional Denoising Diffusion Probabilistic Models (DDPM) [21] and the state-of-the-art Imagen model [48], which presently leads the text-to-image domain. CIFAR-10 and ImageNet datasets are utilized to train the unconditional diffusion models, while the MS COCO dataset is employed to train the Imagen model. We further explore the influence of varying parameters on the effectiveness of the attack. Ultimately, we validate the effectiveness of our attack strategy with a near 100% success rate, thus underscoring the imperative need for addressing the security aspects of diffusion models.

The contributions of our work are two-fold:

- We have analyzed membership inference attacks on diffusion models in existing research. Moreover, we have conceptualized our attack for new practical scenarios and conducted analyses across various dimensions, such as timesteps and model layers.
- We conducted experiments on three datasets using the traditional DDPM model and the cutting-edge text-to-image model, Imagen. Our results demonstrate extremely high accuracy across four evaluation metrics, underscoring the effectiveness of using gradients as attack features.

Roadmap. In Section 2, we introduce the background of diffusion models and delve into membership inference attacks. We also discuss the challenges we encountered and review existing attacks on diffusion models. In Section 3, we present our attack strategy. The experimental setup is detailed in Section 4, while Section 5 showcases the results of these experiments. In Section 6, we apply our GSA framework at the model layer level, demonstrating a further reduction in computational time. Section 7 illustrates the performance of our attack under various defense strategies. The limitations of our attack are discussed in Section 8. Section 9 touches upon related works, and finally, we conclude in Section 10.

2 Background

2.1 Diffusion Models

The work of the Denoising Diffusion Probabilistic Models [21] (DDPM) has drawn considerable attention and led to the recent development of diffusion models [53, 56], which are characteristically described as "progressively denoising to obtain the true image". There are two categories of diffusion models: unconditional diffusion models, which do not incorporate any guiding input for image output, and conditional diffusion models, which were developed subsequently and generate images based on provided inputs information, such as labels [10, 22], text [23, 39, 42, 45, 48], or low-resolution images [47, 49].

Unconditional Diffusion Models. A diffusion model has two phases. First, during the forward process, the model progressively adds standard Gaussian noise to the true image x_0 through T steps. The image at time t is given by

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t \tag{1}$$

where ϵ_t represents the standard Gaussian noise obtained from the reparameterization trick. Furthermore, $\bar{\alpha}_t$ is defined as the product $\prod_{i=1}^{t} \alpha_i$, with each parameter α_i monotonically decreasing and lying in the interval $[0, 1]$.

Second, the reverse process begins with the noise image x'_T, where $x'_T \sim \mathcal{N}(0, I)$, and it progressively denoises to yield $x'_{T-1}, x'_{T-2}, \ldots, x'_0$ through the neural network (e.g., U-Net) ϵ_θ, parameterized by θ. Specifically, ϵ_θ takes a image x'_t and a timestep t as inputs, and predicts the noise, represented by $\epsilon_\theta(x'_t, t)$ that should be eliminated at step t. The final goal is to maximize the similarity between each pair of original image x_0 and the denoised image x'_0.

During the training phase, the objective is to minimize the loss, which is defined as the expected squared ℓ_2 error. This error is evaluated overall ϵ_t and the training sample x_0, as given by:

$$L_t(\theta) = \mathbb{E}_{x_0, \epsilon_t} \left[\| \epsilon_t - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_t, t) \|_2^2 \right] . \tag{2}$$

More details can be found in Appendix A.

Conditional Diffusion Models. As the study of diffusion models deepens, it has been discovered that classifiers can be utilized to guide the diffusion model generation [10]. Specifically, given a pre-trained classifier M and a target class c, one can derive 'directional information', $\nabla_{x_t} \log M(x_t|y)$, for an image x_t and fuse it to the generation process of unconditional diffusion models.

In the text domain, Imagen employs T5, a significant language model [41], as a text encoder to guide the generation process through text embeddings [48]. Specifically, a distinct time embedding vector is constructed and modified during each timestep to align with the image's dimensions. The text embedding extracted from T5 is then incorporated with the time embedding and image to generate the conditional image.

2.2 Membership Inference Attack

Membership inference attack (MIA) tries to predict if a given sample was part of the training set used to train the target model. It has been widely applied to different deep learning models, including classification models [3, 31, 33, 50, 52, 63, 64], generative adversarial networks (GANs) [5, 19, 20, 26, 37], and diffusion models [4, 12, 27, 30, 35, 62]. MIA exploits the differential responses exhibited by machine learning models to training data. Specifically, these models react differently to samples they have been trained on, termed 'member samples', versus unfamiliar 'non-member samples'.

Shokri et al. [52] first proposed the technique of shadow training. This involves training shadow models to imitate the behavior of the target model. An attack model is then trained based on the output of the shadow models. This transforms membership inference into a classification problem.

Considering the increased computational overhead of training a machine learning model as an attack model, Yeom et al. [64] proposed a more streamlined and resource-efficient approach—the *threshold-based* MIA. This method begins with the computation of loss values from the model's output prediction vector. These calculated loss metrics are subsequently compared against a chosen threshold to infer the membership status of a data record.

Carlini et al. [3] argue that while threshold-based attacks are effective for non-membership inference, they lack precision for member sample classification. This discrepancy arises as the approach simplifies the comparison process by scaling all samples based on their loss values, potentially omitting crucial sample-specific properties. To address this, Carlini et al. propose an alternative approach called Likelihood Ratio Attack (LiRA), which derives two distributions from the model's confidence values. These distributions are then used to determine the membership status of a given sample, thereby offering a more balanced evaluation of both member and non-member sets.

2.3 Problem Formulation

In this paper, we investigate MIA in diffusion models. We are given a target model. The task is to predict whether a certain sample is part of the training dataset. MIA on diffusion models (compared to classifiers) presents distinct challenges: Classic classifier models yield vectors. Thus people can use its prediction vector as a feature for MIA [3, 33, 50, 52, 63, 64], which constitutes a black-box attack. Diffusion models produce images as outputs, making it challenging to launch an attack on a diffusion model using only its output, i.e., the image. The current state-of-the-art attacks on diffusion models are predominantly white-box, relying on the loss generated during the evaluation process, as noted in Table 1. Our work is mainly focused on exploring how to get effective attack features. After getting the attack features (gradient data), we use it to train a machine learning model (i.e., XGBoost, MLP) as the attack model to identify the data sample.

Threat Model. We operate under the assumption that an attacker possesses white-box access to the target model, encompassing its architectural intricacies and specific parameter details. In the context of conditional diffusion models, we assume that the attacker knows all modalities (for instance, image-text pairs) pertaining to the victim models. The same assumption has also been adopted in several existing works, which we will discuss in detail later [4, 27, 35]. As more people openly share their model architectures and pre-trained checkpoints (like in HuggingFace[2]), the scenario is realistic. A motivating example is an artist who suspects his artwork is being used without permission to train a diffusion model. This model is subsequently uploaded to the HuggingFace website. As a result, others can use it to generate images that mimic the artist's unique style. Clearly, this constitutes a severe violation of the artist's intellectual property rights. The artist, as a result, downloads the model and checks whether their artwork is used to train the model.

2.4 Existing Work

Existing White-Box Attacks to Diffusion Models. A key challenge in applying MIA is selecting the appropriate information/features to distinguish member and non-member samples. Most effective attacks on diffusion models predominantly employ white-box techniques [4, 27, 35].

Hu et al. [27] and Matsumoto et al. [35] suggested utilizing the loss, as defined in Equation 2, at each timestep t as a feature in conjunction with a threshold-based MIA. Leveraging the loss directly as an attack vector presents the most intuitive attack approach. However, the loss value differences between member and non-member samples vary across different timesteps. For each model, additional computation is required to identify the most effective range of timesteps, which greatly increases pre-computational cost and becomes impractical. Additionally, since the loss value is a scalar, it may lead to unstable attack accuracy due to insufficient information for reliable differentiation. In contrast, gradient data can effectively differentiate between member and non-member samples without requiring prior timestep selection. As high-dimensional data, it also enhances the accuracy and robustness of the attack.

Table 1: Compared with existing work, we argue that with white-box access, using gradients is more effective. We also evaluated more comprehensively on larger datasets.

Attack	Feature	Victim target	Training dataset
[4]	Loss (LiRA)	Unconditional Conditional	CIFAR-10
[27]	Loss (Threshold)	Unconditional	FFHQ DRD
[35]	Loss (Threshold)	Unconditional	CIFAR-10 CelebA
Ours	Gradient (ML model)	Unconditional Conditional	CIFAR-10 ImageNet MS COCO

Carlini et al. [4] also opted to employ loss and use the LiRA framework. In the context of the LiRA online framework, the attack strategy necessitates utilizing target points for the training of several shadow models, a process that is both computationally demanding and time-intensive. Subsequently, it constructs the $\mathbb{D}_{\text{in}}$ and $\mathbb{D}_{\text{out}}$ distributions at each timestep. In the original experiments reported in the paper, 16 shadow models were trained to generate distributions for each timestep. For more sophisticated models, such as Stable Diffusion [45], retraining a large cohort of shadow models to generate loss distributions poses a considerable challenge. In our work, we aim to use fewer shadow models to execute the attack while maintaining effectiveness and efficiency. More details about LiRA can be found at Appendix B.

Other Attacks. Several studies have utilized the properties of DDIM [29, 54, 57] (as detailed in Appendix A) for attacks [12, 30]. However, these attacks are contingent on the deterministic reverse process of DDIM, and cannot be directly applied to DDPM. Detailed discussions of these attacks are deferred to Appendix D.1 and Appendix D.2.

Prior to diffusion models, there are also MIAs for GANs [5, 19, 20, 26, 37]. Note that GANs and diffusion models differ in their overall architecture; therefore, white-box attacks toward GANs are not directly applicable to diffusion models. On the other hand, black-box attacks share similarities as both GANs and diffusion models are generative models. In particular, inspired by the attacks of GAN-Leaks [5], Matsumoto et al. [35] proposed an attack that is based on the reconstruction error of the target image and a set of generated samples. We will present the details at Appendix D.3, but the attack shows limited effectiveness.

Meanwhile, Wu et al. [62] carried out black-box attacks on pre-trained text-to-image diffusion models, launching attacks at both the pixel-level and semantic-level. However, their method does not employ the shadow model technique as proposed in [52], instead

[2] A multitude of model cards can be found on the Hugging Face website. https://huggingface.co/models

conducting all experiments directly on the target model and selecting the training set of the pre-trained model as the member set. Consequently, this attack strategy is not universally effective for every victim model.

Hu et al. [27] also initiated an interesting threat model (the so-called grey-box model or query-based model) where the attacker sees the intermediate denoised images and proposes an attack based on the similarities (likelihood) between pairs of these intermediate samples and those in the forward pass (details also in Appendix D.4).

3 Methodology

Previous works on attacking the diffusion model encompass black-box [35, 62], gray-box [12, 27, 30], and white-box approaches [4, 27, 35]. Upon comparing their accuracy and considering practical implications, we contend that white-box attacks on diffusion models are the most effective.

3.1 Theoretical Foundation and Challenge

Current white-box attacks often manipulate the loss at different timesteps through various methods (e.g., threshold [27, 35] or distribution [4]). However, it often necessitates a substantial amount of time to identify the timestep where the loss can most distinctly differentiate between the member and non-member set samples. We argue that *rather than relying on the loss information, given white-box access, it could be more insightful to leverage gradient information* that better reflects the model's different responses to member samples and non-member samples. The intuition using gradients is, as gradients are generally very high-dimensional (than losses), it offers a more nuanced representation of its response to an input target point compared to mere loss values.

Figure 1 shows the general idea of our attack. It is important to note that, owing to the specific architecture of the diffusion model, a single query point can yield multiple loss values originating from different timesteps. Subsequently, based on the loss L, we can derive the gradients using the standard back-propagation technique and use the gradients as features to train a machine-learning model to execute MIA.

In the diffusion model, the training loss function is defined as Equation 2. For each sample, noise is added using Equation 1, generating a noised sample x_t. The trained U-Net modules then predict the noise ϵ_t that needs to be denoised at timestep t, based on x_t and t. The existing methods [4, 27, 35] assume that the loss value of a member sample is typically smaller than that of a non-member sample, which indicates intuitively

$$x \in \mathcal{D}_m \text{ if and only if } \|\epsilon_t - \epsilon_\theta(x_t, t)\|_2^2 < \tau$$

where $\epsilon_\theta(x_t, t)$ represents the predicted noise at t-th step and ϵ_t is the ground true noise sample. However, we have observed that this approach can lead to misjudgments. For example, inherently complex member samples might exhibit higher loss values compared to simpler non-member samples, a phenomenon also observed in GAN-Leaks [5]. This indicates that relying solely on loss as the attack feature may introduce some degree of bias. Carlini et al. [4] also found that using loss values as the sole criterion for determining membership is inadequate.

In our work, we propose using gradient values as attack features to better capture the model's reaction to samples. Unlike loss values, which are scalars and provide limited information, gradient data offer a more comprehensive view. Additionally, even when two samples have identical loss values, their corresponding gradients can differ, as gradients depend on the specific inputs within the computational graph. For instance, the diffusion model ϵ_θ (with parameter θ) calculates gradients for a query sample x at t-th step; the gradients can be expressed as:

$$\nabla_\theta L_t(\theta, x) = \nabla_\theta \|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 \tag{3}$$

According to the definition of the Euclidean norm squared, we can expand the squared term in Equation 3:

$$\begin{aligned}\|\epsilon_t - \epsilon_\theta(x_t, t)\|^2 &= (\epsilon_t - \epsilon_\theta(x_t, t))^\top (\epsilon_t - \epsilon_\theta(x_t, t)) \\ &= \|\epsilon_t\|^2 - 2\epsilon_t^\top \epsilon_\theta(x_t, t) + \|\epsilon_\theta(x_t, t)\|^2.\end{aligned}$$

Then, we proceed to compute the derivatives of each of the three expanded terms with respect to θ:

$$\begin{aligned}\nabla_\theta L_t(\theta, x) &= \nabla_\theta \left(\|\epsilon_t\|^2 - 2\epsilon_t^\top \epsilon_\theta(x_t, t) + \|\epsilon_\theta(x_t, t)\|^2\right) \\ &= \nabla_\theta \|\epsilon_t\|^2 - 2\nabla_\theta \left(\epsilon_t^\top \epsilon_\theta(x_t, t)\right) + \nabla_\theta \|\epsilon_\theta(x_t, t)\|^2 \\ &= 0 - 2\epsilon_t^\top \nabla_\theta \epsilon_\theta(x_t, t) + 2\epsilon_\theta(x_t, t)^\top \nabla_\theta \epsilon_\theta(x_t, t) \\ &= -2\left(\epsilon_t - \epsilon_\theta(x_t, t)\right)^\top \nabla_\theta \epsilon_\theta(x_t, t) \\ &= 2\left(\epsilon_\theta(x_t, t) - \epsilon_t\right)^\top \nabla_\theta \epsilon_\theta(x_t, t)\end{aligned} \tag{4}$$

From Equation 4, we show the gradient depends on both the value of the training loss $(\epsilon_\theta(x_t, t) - \epsilon_t)$ and the specific query sample being computed $(\nabla_\theta \epsilon_\theta(x_t, t))$. For member and non-member samples that produce the same numerical loss value, gradients can still use $\nabla_\theta \epsilon_\theta(x_t, t)$ to discriminate them. We also present the experimental evidence to support our finding in Appendix C.

Intuitively, during the training phase, the model fits to member samples. Therefore, when encountering a training sample, the already converged model requires less parameter adjustment compared to a non-member sample, leading to smaller gradients. Based on this intuition, we use the model's gradient values as features for detecting query sample membership, as expressed by:

$$x \in \mathcal{D}_m \text{ if and only if } \nabla_\theta \|\epsilon_t - \epsilon_\theta(x_t, t)\|_2^2 < \tau$$

The above findings demonstrate that even when the loss values are equal, the gradient information obtained from different samples still varies. We believe that this characteristic of gradient data represents the model's response to the query sample more effectively than the attack features used in existing methods [4, 27, 35], thereby enabling more successful attacks.. However, the key challenge of using gradients for MIA is utilizing gradient information effectively. Considering the substantial number of parameters in the diffusion model (for instance, in our experiments, the Imagen model boasts close to 250 million trained parameters, while the DDPM model approaches 114 million), training the attack model by using the gradient of each model parameter for every image is both computationally impractical and prone to overfitting, despite its potential to maximally differentiate between member and non-member samples. Moreover, in diffusion models, the diffusion process typically involves T timesteps (usually set to 1000). For each timestep t in the

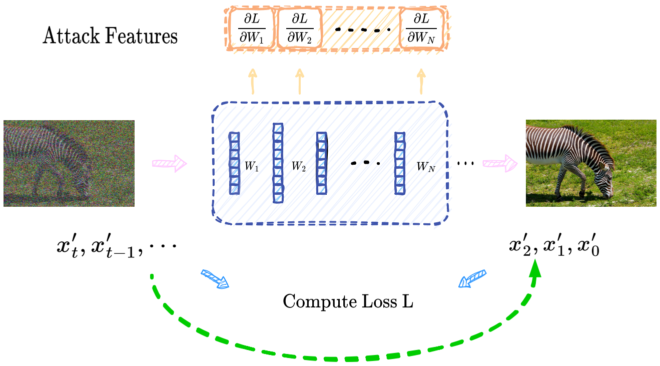

Attack Features
$\frac{\partial L}{\partial W_1}$
$\frac{\partial L}{\partial W_2}$
$\frac{\partial L}{\partial W_N}$
W_1
W_2
W_N
$x'_t, x'_{t-1}, \cdots$
x'_2, x'_1, x'_0
Compute Loss L

Figure 1: High-level pipeline of our attack: Given the target sample x_0, we first add noise based on Equation 1 and feed it to the target model shaded in blue. At each sample step, we can compute a loss L using Equation 2 to derive the gradients. Gradients from all sample steps (with appropriate subsampling and aggregation operations) are used as features to train the attack model for MIA.

Table 2: Impact of three different timestep-level sampling methods on attack accuracy and their respective time consumption.

Method	ASR	AUC	TPR@1%FPR	TPR@0.1%FPR	Time (seconds)
Effective	0.947	0.992	0.663	0.311	21587
Poisson	0.801	0.882	0.270	0.053	2422
Equidistant	0.932	0.981	0.641	0.304	2398

range $1, \ldots, T$, a separate loss and set of gradients are generated, further increasing the dimensionality of the overall gradients.

3.2 Gradient Dimensionality Reduction

We propose a general attack framework for reducing the dimensionality of the gradients while trying to keep the useful information for differentiating members vs non-members. It is composed of two common techniques: (1) subsampling, which chooses the most informative gradients in a principled way, and (2) aggregation, which combines/compresses those informative gradients data. We name the framework Gradient attack based on Subsampling and Aggregation (GSA).

We then present a three-level taxonomy outlining where these two techniques can be applied: at the timestep level, across different layers within the target model, and within specific gradients of each layer, as detailed below.

(1) Timestep Level: As corroborated by prior studies [4, 12, 27, 30, 35], diffusion models display distinct reactions to member and non-member samples *depending on the timestep*. For instance, Carlini et al. [4] identified a 'Goldilock's zone', which yielded the most effective results in their attack, to be within the range $t \in [50, 300]$. We believe that the importance of gradient data also varies across different timesteps. Therefore, sampling the timesteps that contain the most useful information will undoubtedly result in more accurate attack outcomes. We refer to the attacks conducted on the most effective gradient data within the 'Gold zone' as *effective sampling*. However, implementing *effective sampling* requires detecting the 'Gold zone' in the target model each time, and the optimal timesteps for achieving the best attack accuracy may vary across different models. As a result, we propose two alternative sampling methods: *equidistant sampling* and *poisson sampling*. In *equidistant sampling*, the denoising steps are selected at intervals of $T/|K|$ (K refer to the sampled timesteps set) for any given model. In *poisson sampling*, an average rate parameter λ ($|K|/T$) is used to randomly generate intervals following an exponential distribution, thereby selecting $|K|$ steps from a total of T steps. We then present a simple case study to test and compare these three different sampling methods.

(2) Layer-wise Selection and Aggregation: Beyond timesteps, the layers within the model present another pivotal dimension for subsampling and aggregation. Recognizing the nuances captured across layers—from basic patterns in shallower layers to intricate details in deeper ones—it is deemed essential to selectively harness gradients from these layers, especially the informative ones, to optimize the attack model's training.

(3) Gradients within Each Layer: Within each layer of a neural network, there is typically no specific ordering of the gradient data. Therefore, it is more reasonable to treat these gradients as a set [15].

Case Study. Since existing attacks [4, 12, 27, 30, 35] heavily focus on timestep-level selection, we designed a case study to better examine how different subsampling methods impact attack performance. We evaluated the attack accuracy using three sampling methods: *effective sampling*, *equidistant sampling*, and *poisson sampling*. For *effective sampling*, it is necessary to first identify the 'Gold zone'. To achieve this, we recorded the attack results in every 20 step across the T denoising steps. The timestep with the best attack performance, along with the 10 surrounding timesteps, was then selected as the sampling points for effective sampling. For *equidistant sampling*, we set step 1 as the initial step and then sample timesteps at fixed intervals of $T/|K|$. In contrast, *poisson sampling* uses $|K|/T$ as the parameter λ to sample from the T steps.

We select 5000 samples from CIFAR-10 dataset to train DDPM as target model. For each sampling method, we set the number of sampling steps ($|K|$) to 10. In Table 2, we found that *effective sampling* achieves the highest attack accuracy, while *poisson sampling* has the lowest. This result aligns with our initial assumption that using gradient data sampled from the 'Gold zone'—the interval yielding the best attack results on individual timesteps—would lead to optimal performance. In contrast, *poisson sampling*'s randomness may lead to poor attack outcomes if the sampled timesteps cannot effectively discriminate between members and non-members.

However, in Table 2, we also present the time consumption for implementing different sampling methods. We found that although *effective sampling* achieves high attack accuracy, it takes nearly 8 times longer compared to *equidistant* and *poisson sampling*. This is because *effective sampling* requires precomputing the attack performance for numerous timesteps to identify the 'Gold zone'. Meanwhile, *equidistant sampling* only slightly reduces the ASR by 0.015 and the AUC by 0.011 compared to *effective sampling*, while being

Algorithm 1 GSA$_1$

```
Input: Target model denoted as $\epsilon_\theta$ with $N$ layers, a equidistantly
    selected set of timesteps $K$, and a sample $x$.
for $t \in K$ do
    Sample $\epsilon_t$ from Gaussian distribution
    Compute $x_t$ based on Equation 1
    Compute loss $L_t$ from Equation 2
end for
$\bar{L} \leftarrow \frac{1}{|K|}\sum_{t \in K} L_t$
$\mathcal{G} \leftarrow \left[\left\|\frac{\partial \bar{L}}{\partial W_1}\right\|_2^2, \cdots \left\|\frac{\partial \bar{L}}{\partial W_N}\right\|_2^2\right]$
Output: $\mathcal{G}$
```

Algorithm 2 GSA$_2$

```
Input: Target model denoted as $\epsilon_\theta$ with $N$ layers, a equidistantly
    selected set of timesteps $K$, and a sample $x$.
$\mathcal{G} \leftarrow [\ ]$
for $t \in K$ do
    Sample $\epsilon_t$ from Gaussian distribution
    Compute $x_t$ based on Equation 1
    Compute loss $L_t$ from Equation 2
    $\mathcal{G}_t = \left[\left\|\frac{\partial L_t}{\partial W_1}\right\|_2^2, \cdots \left\|\frac{\partial L_t}{\partial W_N}\right\|_2^2\right]$
end for
$\mathcal{G} \leftarrow \frac{1}{|K|}\sum_{t \in K} \mathcal{G}_t$
Output: $\mathcal{G}$
```

more time-efficient. To balance effectiveness and efficiency, we use *equidistant sampling* to derive a subsampled timestep set K from the total diffusion steps T. Following this, for the timesteps in K, we can aggregate the gradients or losses generated at each timestep using statistical methods such as the mean, median, or trimmed mean to produce the final output. If the values being aggregated are the gradients from each timestep, the output can be directly used as the final output. However, if the aggregated values are the losses, the processed loss value needs to be used in backpropagation to extract gradient information for the final output.

3.3 Our Instantiations

We present two exemplary instantiations of the attack within the framework, representing two extreme points in the trade-off space between efficiency and effectiveness. We call them GSA$_1$ and GSA$_2$. GSA$_1$ performs more reduction, gaining efficiency but losing information. GSA$_2$ does less reduction, retaining effectiveness but at a cost to efficiency. In the GSA$_1$ method, although we equidistantly sample $|K|$ timesteps from T, only a single gradient computation is required. This outcome is realized by in GSA$_1$ computing the loss, L_t, for each timestep present in K. Subsequently, we take the mean of these individual losses, represented as $\bar{L}$, to perform backpropagation. This process eventually yields a solitary gradient vector. On the other hand, GSA$_2$ entails performing backpropagation and computing gradients for each timestep in K, and then using the mean of all gradient vectors, denoted as $\mathcal{G}$, as the final output.

Note that we only slightly optimize our two instantiations in this paper because they are already very effective. We leave more detailed investigations of the design space and more effective proposals as future work.

Based on our detailed analysis of existing white-box attacks [4, 27, 35], we first find that the optimal timesteps for mounting the most effective attacks vary depending on the specific dataset and diffusion model in question.

Consequently, we adopt the *equidistant sampling* strategy to select sample timesteps from the range $[1, T]$, denoted by a set K. This approach is designed to encompass timesteps that can distinctly differentiate between member and non-member samples, avoiding an exclusive focus on timesteps that are either too early or too late.

After getting loss from each selected step, we use backpropagation to compute the gradients for the model. Given the diverse nature of gradients within a layer, we aggregate the model's gradient information on a per-layer basis. That is, once the gradient information for a layer's parameters is obtained, the ℓ_2-norm of these gradients is used as the representation for that layer's gradient information. This approach offers a dual advantage: it substantially reduces computational overhead while also holistically encapsulating that layer's gradient information.

This forms the basis of GSA$_2$ (given in Algorithm 2): for each timestep t in the set K, we calculate the per-layer gradient using the ℓ_2-norm, and then find their average.

However, this approach can still incur substantial computational costs when applied to large diffusion models and datasets — taking nearly 6 hours to execute on Imagen. To address this inefficiency, we preprocess the loss values from multiple timesteps before doing gradient computation. In light of this challenge, we introduce GSA$_1$ (outlined in Algorithm 1), which reduces the gradient extraction time for the Imagen model to less than 2 hours, significantly decreasing the computational time required.

4 Experimental Setup

4.1 Datasets

We use CIFAR-10, ImageNet, and MS COCO datasets. The use of CIFAR-10 allows for an easier comparison of attack results as it has been frequently employed in previous work [4, 12, 30, 35]. Both ImageNet and MS COCO serve as significant target datasets in the domain of image generation, with MS COCO used for training in various tasks, such as VQ-diffusion [18], Parti Finetuned [66], U-ViT-S/2 [2], and Imagen [48].

ImageNet dataset is a large-scale and diverse collection of images designed for image classification and object recognition tasks in the fields of machine learning and computer vision. When conducting experiments with the ImageNet dataset, researchers typically utilize a specific subset consisting of 1.2 million images for training and 50,000 images for validation, while an additional 100,000 images are reserved for testing. Considering the constraints on training resources and to ensure diversity in the training images, we opt

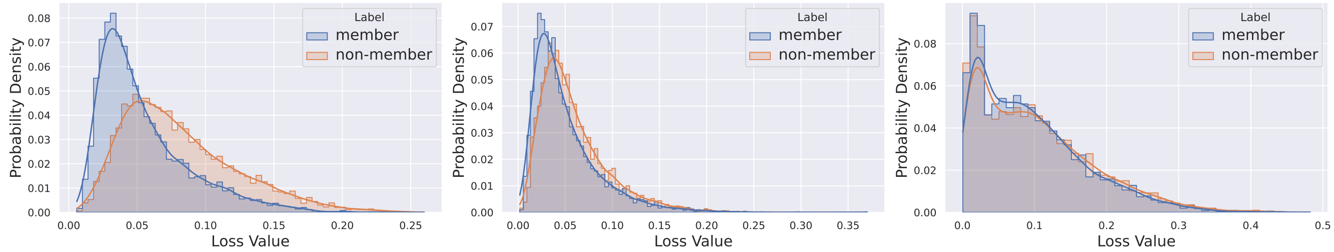


Figure 2: Loss distribution for member vs. non-member samples across CIFAR-10, ImageNet, and MS COCO (from left to right), used by existing work [4, 27, 35]. Models use default settings from Table 3.

to utilize the ImageNet test set as the training set for training the models in our work.

CIFAR-10 dataset comprises 10 categories of 32×32 color images, with each category containing $6,000$ images. These categories include airplanes, automobiles, birds, cats, deer, dogs, frogs, horses, ships, and trucks. In total, the dataset consists of $60,000$ images, of which $50,000$ are designated for training and $10,000$ for testing. The CIFAR-10 dataset is commonly employed as a benchmark for image classification and object recognition tasks in the fields of machine learning and computer vision.

MS COCO dataset contains over $200,000$ labeled high-resolution images collected from the internet, with a total of 1.5 million object instances and 80 different object categories. The categories cover a wide range of common objects, including people, animals, vehicles, and household items, among others. The MS COCO dataset is noteworthy for its diversity and the complexity of its images and annotations. Images in the MS COCO dataset depict a wide variety of scenes and object layouts. In this experiment, we utilize all images from the MS COCO training set for model training. The first caption from the five associated with each image is selected as the corresponding textual description.

4.2 Training Setup

Table 3: Default parameters used for the experiments.

Parameters	Unconditional Diffusion	Unconditional Diffusion	Imagen
Channels	128	128	128
Diffusion steps	1000	1000	1000
Dataset	CIFAR-10	ImageNet	MS COCO
Training data size	8000	8000	30000
Resolution	32	64	64
Learning rate	$1e-4$	$1e-4$	$1e-4$
Batch size	64	64	64
Noise schedule	linear	linear	linear, cosine
Learning rate schedule	cosine	cosine	cosine
Training time	400 epochs	400 epochs	$600,000$ steps

We tabulated the default training parameters for the unconditional diffusion model on CIFAR-10 and ImageNet, and for Imagen on MS COCO, in Table 3. Given that we have employed ASR (Accuracy) as our evaluation metric, we endeavor to maintain a balance between the quantities of the member set and the non-member set to ensure the precision of model validation. The structure for the unconditional diffusion model aligns with those from the diffusers library [60] in Huggingface. Imagen is based on the open-source implementation by Phil Wang et al.[3], and we have retained consistency in its configuration. All experiments were conducted using two NVIDIA A100 GPUs.

4.3 Metrics

In the process of comparing experimental results, we employ Attack Success Rate (ASR) [6], Area Under the ROC Curve (AUC), and True-Positive Rate (TPR) values under fixed low False-Positive Rate (FPR) as evaluation metrics.

In our experiments, we ensure an equal number of member and non-member image samples. Given the balanced nature of our dataset and the stability of ASR in such contexts, we employ ASR as our primary evaluation metric.

We note that most instances MIAs on diffusion models use the AUC metric for evaluation [4, 12, 27, 30, 35]. Likewise, in assessing the merits of our work in Section 5.1, we will also use AUC as one of our assessment metrics. Additionally, as Carlini et al. [3] argued that TPR under a low FPR scenario is a key evaluation criterion, we also use TPR at 1% FPR and 0.1% FPR, respectively.

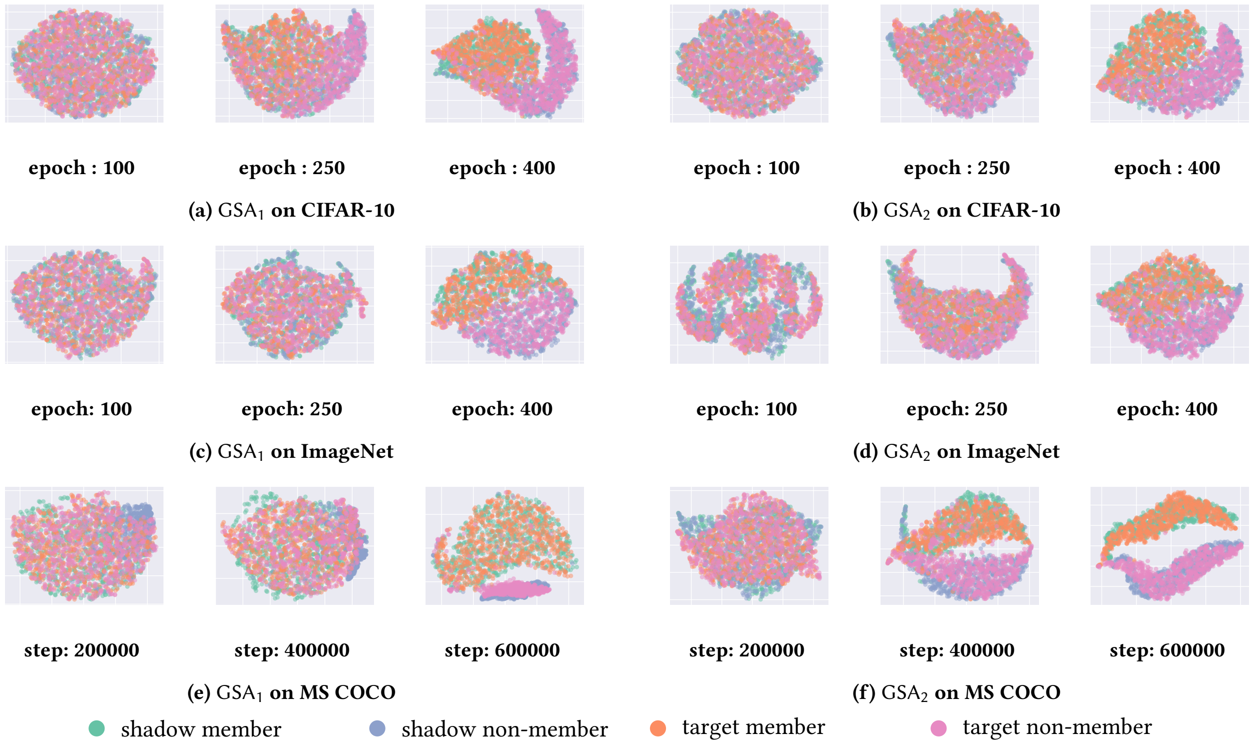
5 Evaluation Results

5.1 Comparison with Existing Methods

Table 4: Existing white-box attacks on the CIFAR-10 dataset are benchmarked using four distinct metrics. LiRA*, LSA*, GSA$_1$, and GSA$_2$ are all obtained under the same conditions.

Attack method	CIFAR-10			
	ASR↑	**AUC↑**	**TPR@1%FPR(%)↑**	**TPR@0.1%FPR(%)↑**
Baseline	0.736	0.801	5.65	–
LiRA	–	0.982	5(5M) 99(102M)	7.1
Strong LiRA	–	0.997	–	29.4
LiRA*	0.626	0.71	1.45	0.25
LSA*	0.83	0.909	13.77	0.925
GSA$_1$	**0.993**	**0.999**	**99.7**	**82.9**
GSA$_2$	**0.988**	**0.999**	**97.88**	**58.57**

[3] Code available at https://github.com/lucidrains/imagen-pytorch



(a) GSA$_1$ on CIFAR-10

(b) GSA$_2$ on CIFAR-10

(c) GSA$_1$ on ImageNet

(d) GSA$_2$ on ImageNet

(e) GSA$_1$ on MS COCO

(f) GSA$_2$ on MS COCO

Figure 3: The left and right columns display the visualization of high-dimensional gradient information using t-SNE after GSA$_1$ and GSA$_2$ have respectively executed attacks on the three datasets (using the output from the last layer of our attack model). For all six attacks, it is observed that member and non-member samples are distinctly differentiated when reaching the training steps defined by the default settings (as referenced in Table 3).

We benchmark GSA$_1$ and GSA$_2$ against existing methodologies, maintaining all other model parameters consistent. Contrasting traditional loss-based white-box attacks such as LiRA [4] and others techniques [27, 35], we provide a thorough evaluation highlighting the superior efficacy of GSA$_1$ and GSA$_2$. The baseline approach [27, 35, 64] depicted in Table 4 is the most intuitive, which predicts the sample as a non-member if the loss exceeds a certain value and vice versa [35]. This also represents the most traditional judgment method in MIA, utilized here as the baseline.

5.1.1 *Feature Informative.*

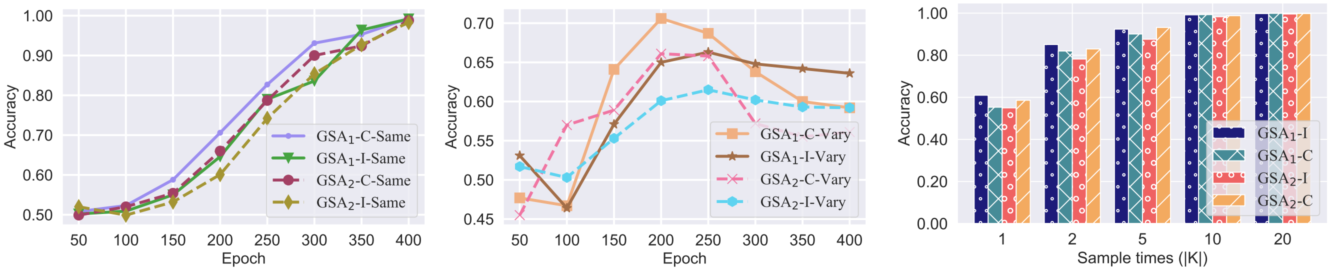
LSA* refers to the results of training the attack model using the loss under the same training conditions and sampling frequency as GSA$_1$ and GSA$_2$. The sole distinction between LSA* and GSA lies in their features: while LSA* utilizes loss as its attack feature, GSA employs the gradient. Comparative results between them substantiate that the gradient information of the diffusion model is more aptly suited as attack features.

It is apparent from Table 4 that both GSA$_1$ and GSA$_2$ exceed other techniques in terms of all evaluation metrics. Under the AUC criterion, LiRA [4] also attains a high attack accuracy, attributed to excessive training steps and many shadow models. However, when ensuring an equivalent quantity of shadow models and training epochs for the LiRA* based on the LiRA framework, its ASR, TPR, and AUC scores are significantly lower compared to GSA$_1$ and GSA$_2$. In the original paper, the LiRA framework achieves TPRs of 5% after training for 200 epochs, with the FPRs fixed at 1%. Remarkably, after training for 4080 epochs, the TPR increases to 99%. In contrast, for GSA$_1$ and GSA$_2$, TPRs of 99.7% and 78.75% are respectively achieved after only 400 epochs, underscoring a more efficient attack strategy. This essentially corroborates our core proposition that gradient information of the model exhibits a more pronounced response to member set samples than loss.

5.1.2 *Timestep Selection.*

Moreover, the 'time zone' demonstrating discernible differences in the loss distribution between members and non-members vary across different models and datasets [4, 12, 27, 35]. Consequently, to achieve a more potent attack, it becomes imperative to extract the loss and establish thresholds or distributions for each timestep using shadow models, aiming to pinpoint the most efficacious 'time zone'. In contrast, both GSA$_1$ and GSA$_2$ execute attacks by solely harnessing the gradient information derived from *equidistant sampling* timesteps across the T diffusion steps, achieving similar attack accuracy in just **one-thirtieth** of the time. Given a consistent dataset size and model architecture, extracting loss across T steps takes 36 hours. In contrast, GSA$_1$ and GSA$_2$ achieve the **same accuracy level** in less than 1 hour by extracting gradients from 10 equidistant sampling timesteps.

To further demonstrate that the optimal timestep for distinguishing between member and non-member samples using loss varies across different datasets and models. We plot the loss distribution



(a) Impact of training epoch **(b) Impact of |K|**

Figure 4: "-I-" and "-C-" denote experiments with ImageNet and CIFAR-10 datasets. Panel (a) (left) reveals that attacks are more effective when shadow and target models closely fit the training data; (right) however, increased fitting disparities between them weaken the attack. Panel (b) shows that greater sampling frequency boosts the attack's effectiveness, possibly due to acquiring finer data and getting more informative timestep.

for three distinct datasets used in our experiment: CIFAR-10, ImageNet, and MS COCO. Following the methodology of LiRA in attacking diffusion models [4], we identified the optimal timestep for each of the three distinct datasets that best distinguishes member from non-member samples. For this, we equidistantly sampled 10 timesteps from shadow models (the training times of these shadow models align with those presented in Table 3). However, we observed that the identified timesteps across the three datasets were not consistent. Upon visualizing the loss distribution at these specific timesteps in Figure 2, we found that even at these optimal points, the loss distribution did not effectively differentiate between member and non-member samples. DDPM trained on the CIFAR-10 dataset clearly differentiates between member and non-member loss distributions. However, such a difference is not pronounced for models trained on ImageNet and MS COCO datasets. For models to execute attacks on the ImageNet and MS COCO datasets, it is essential to compute the loss distribution across a broader range of timesteps and increase their training time.

Using the same model parameters and sampling frequency as in Figure 2, we tried attacks with GSA_1 and GSA_2. The attack features were derived from the gradients of timesteps sampled from T using the same sampling frequency as previously employed. We visualized this high-dimensional gradient information using t-SNE [59] in Figure 3. It can be observed quite intuitively, that across all datasets, both GSA_1 and GSA_2 can effortlessly differentiate between target member and target non-member data using the features derived from the gradients of shadow models.

5.2 Attacking Unconditional Diffusion Model

In this section, we trained six shadow models to facilitate the attack. We focus on unconditional diffusion models and test on CIFAR-10 and ImageNet datasets.

Training on Different Epochs. Our first goal is to understand how varying training epochs for target and shadow models influence our attacks. We considered two possible scenarios.

- In the first scenario, the attacker knows the target model's training epochs and matches the shadow model's training accordingly.
- In the second scenario, the attacker is unaware of the target model's training details and varies only the shadow model's training epochs for experimentation.

In Figure 4a, we present the experimental results under the first scenario. These findings indicate that as the training epochs for both the target and shadow models increase, the attack success rate for GSA_1 and GSA_2 consistently improves. In this context, the suffixes "-I-" and "-C-" refer to experiments on ImageNet and CIFAR-10, respectively. We postulate that with an increasing number of epochs, the model tends to fit the training data more closely after convergence. This amplifies the gradient discrepancy between member and non-member samples, subsequently bolstering the efficacy of the attack.

In the second scenario setting, when the training epochs of the target model are fixed at 200 epochs, the attack accuracy is optimal when the shadow model's training epochs closely match those of the target model. Furthermore, observations from Figure 4a suggest that the overall efficacy of membership inference attacks is closely tied to the consistency in the degree of fit between the shadow models and their training data as compared to that of the target model with its training data. When shadow models exceed the target model in data fitting, it does not invariably lead to an improved attack performance. Contrarily, the attack's success rate might diminish due to disparities in their fitting levels.

Then, our experiments explore the influence of the degree of overfitting in both shadow and target models on attack accuracy. Moreover, we examine the impact of discrepancies in data-fitting levels between the target and shadow models on the performance of the attack.

Sampling Frequency Variation Analysis. In both GSA_1 and GSA_2, the term 'sample times' ($|K|$) refers to the number of elements in the set K, derived through the *equidistant sampling* of timesteps from T. GSA_1 and GSA_2 employ statistical methods on distinct pieces of information; the former determines the mean loss over the $|K|$ timesteps, while the latter computes the average gradient value. Our initial hypothesis was that an increased number of sampling instances, providing the attack model with more information and potentially capturing distinct timesteps that clearly differentiate between member and non-member samples, would lead to improved attack accuracy.

Figure 4b confirms our initial hypothesis that collecting more gradient information from a single sample enhances the attack's success rate. In all attacks, we maintained a constant setting of 1000 diffusion steps and conducted equidistant sampling across these steps. Our focus was on understanding how varying the sampling frequencies during the evaluation process of a single sample affects the attack's accuracy.

From our experimental data, we observed that the attack's success rate was lowest when gradient information was collected only once per sample. This limited data collection blurred the distinction between member and non-member set samples. Notably, the precision saw a significant boost when the collection frequency increased. However, after reaching a threshold of ten collections per sample, further increases in frequency showed diminishing returns in precision. Thus, we inferred that, for both attack strategies across these two datasets, collecting gradient information ten times from each sample is optimal for distinguishing between member and non-member sets. In other experiments, to strike a balance between efficiency and precision, we will adopt this **ten-times-per-sample** information collection formula as the default setting.

Different Diffusion Steps and Training Image Resolution. In the context of diffusion models, increasing the number of diffusion steps can potentially enhance image quality. This improvement stems from the model's refined capability to capture detailed image nuances by reducing noise over more steps. However, as we add more diffusion steps, the optimization challenge might become more complex. This complexity can slow down the convergence speed and require more detailed hyperparameter adjustments to find the optimal model setup.

When contemplating membership inference attacks, their genesis primarily stems from overfitting during the training phase, leading to discrepancies between the member and non-member samples. We theorize that if adding diffusion steps slows model convergence, it might reduce the overfitting phenomenon, affecting the attack's success. We set the total diffusion steps from 500 to 2000, kept other parameters unchanged, and retrained the model on both ImageNet and CIFAR-10 datasets.

In Figure 5a, we observe that increasing the number of diffusion steps significantly influences our attack success rate, which aligns with our hypothesis. For models trained on CIFAR-10, both GSA_1 and GSA_2 achieve an attack accuracy close to 1.00 after training with 300 epochs. However, as the number of denoising steps increases, the attack accuracy decreases by nearly 10% when the denoising step is set to 2000. The increase in denoising steps leads to a decrease in attack accuracy. This pattern is also observed for models trained on ImageNet when attacked with GSA_1 and GSA_2. We think this is because MIA relies on exploiting the model's overfitting. However, increasing the denoising steps slows down the model's convergence, thereby impairing the effectiveness of the attack.

Moreover, input data resolution also plays an important role in determining attack success rates. High-resolution images help in distinguishing between member and non-member samples due to their intricate details, but they also require more computational resources and longer training times. Such images may also decelerate the convergence rate of the model, potentially mitigating the extent of overfitting and necessitating additional epochs to achieve equivalent attack outcomes as before.To investigate the impact of high-resolution images on attack performance, we conducted the experiments using both GSA_1 and GSA_2 on images with resolutions ranging from 64 to 256 pixels.

In Figure 6, we observed that the highest attack accuracy was achieved with GSA_1 and GSA_2 when the image resolution was set to 128×128. The results indicate that lower-resolution samples do not necessarily lead to better attack performance. Increasing the resolution from 64 to 128 allows the model to capture more granular details, improving the distinction between member and non-member samples. However, when the resolution is further increased to 256, a noticeable decline in success rate occurs. We believe this is because higher-resolution images require more training steps for the model to converge. Therefore, when the training time is fixed but the resolution increases, the overfitting phenomenon to the training data diminishes. This reduction in overfitting causes the attack to become less effective. Additionally, both excessively high and low resolutions can negatively impact the final attack performance. An optimal resolution exists where the model can capture sufficient details without requiring extensive training, achieving a balanced fit.

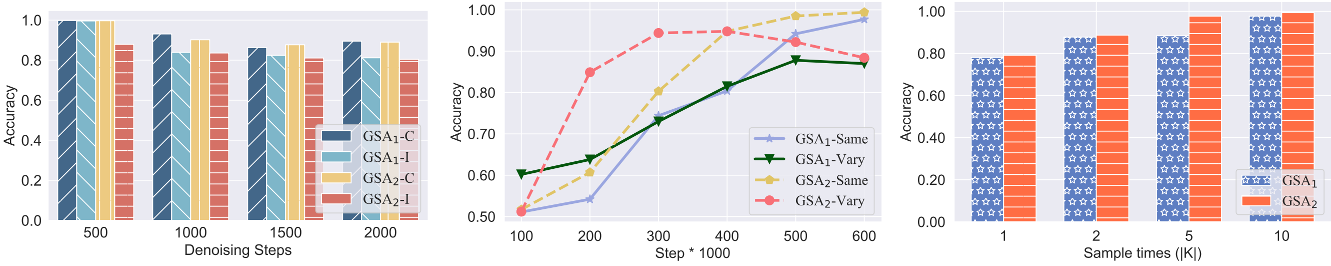
> *Takeaways:* In settings where unconditional diffusion models serve as the target model, overfitting is considered foundational for MIAs. Moreover, distinctions between member and non-member samples can vary at different timesteps. Given these factors, we have investigated several elements that could influence the attack's success rate. These factors encompass the number of training epochs, number of sampling timesteps from a single instance (represented as $|K|$), the total diffusion steps, and the resolution of the images. Results from these explorations are presented in the aforementioned figures, with ASR adopted as the evaluation metric.

5.3 Attacking Conditional Diffusion Model

In this section, we design experiments with Imagen, a state-of-the-art generation model in the text-to-image field. We train two shadow models from scratch, using the MS COCO dataset in this part for training purposes.

Training on Different Epochs. In Figure 5b, consistent with the two attack scenarios posited in Section 5.2, we analyze the effect of training steps on the attack success rate for Imagen models. Our categorization is premised on the attacker's knowledge of the target model's training steps. Notably, when the attacker is uncertain about the number of training steps of the target model, we set the training steps of the target model to a fixed value (in this instance, $400,000$ steps). This experimental setup aligns with that of Section 5.2.

Consistent with previous experiments using the unconditional diffusion models, a large proportion of the attack success rate for the Imagen model is influenced by the training steps of the target



(a) Impact of diffusion steps

(b) Impact of training epoch

(c) Impact of |K|

Figure 5: Notations “-I-” and “-C-” are consistent with those in Figure 4a. Panel (a) suggests that increasing the number of diffusion steps, which decelerates convergence, results in a reduced attack success rate. Panel (b) reinforces findings from Figure 4a: enhanced data-fitting by both the shadow and target models boosts the attack’s efficacy. However, when there are disparities in the data fitting, the efficacy diminishes. Panel (c) shows that augmenting the sampling steps for Imagen—thus acquiring more information—significantly improves the attack’s success rate.

Table 5: The table presents the performance results of GSA_1 **and** GSA_2**, trained on three different datasets and evaluated using four distinct evaluation metrics.**

Attack method	ASR↑			AUC↑			TPR@1%FPR↑			TPR@0.1%FPR↑		
	CIFAR-10	ImagetNet	MS COCO	CIFAR-10	ImagetNet	MS COCO	CIFAR-10	ImagetNet	MS COCO	CIFAR-10	ImagetNet	MS COCO
LSA	0.822	0.702	0.684	0.896	0.766	0.746	0.146	0.034	0.073	0.021	0.004	0.011
GSA_1	0.993	0.992	0.977	0.999	0.999	0.997	0.997	0.995	0.954	0.829	0.937	0.627
GSA_2	0.988	0.983	0.994	0.999	0.999	0.999	0.979	0.964	0.998	0.586	0.743	0.976

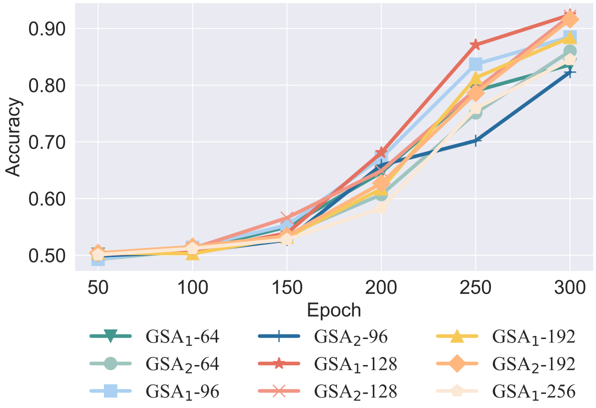


Figure 6: Results from ImageNet represent the resolution of the image can influence the attack’s training accuracy by affecting the model’s convergence rate.

model and shadow models. Precisely, the more the target model overfits the data, the higher the success rate of the MIA, even if the overfitting phenomenon during the shadow model’s training is not notably pronounced. For example, Figure 5b shows that when deploying the GSA_2 method with the shadow model trained for 200,000 steps, an attack success rate of up to 84.9% can be achieved if the target model has been trained for 400,000 steps. However, if the target model’s training steps are only 200,000, the attack success rate drops to merely 60.7%, representing a nearly 25% decrease in accuracy. Hence, the degree to which the target model fits the data profoundly influences the effectiveness of the attack. Surprisingly, when the training steps of the shadow models exceed those of the target model, further increasing the training steps for the shadow models leads to a decline in the success rate of MIA attacks. This finding is similar to the phenomenon observed in Section 5.2 (i.e., the efficacy of the attack is intimately linked to the disparity in data-fitting degrees between the shadow models and their training datasets and the target model with its respective training data.).

Sampling Frequency Variation Analysis. It is evident, as depicted in Figure 5c, that the frequency of information extraction from a single sample by the model plays a pivotal role in influencing the success rate of the attack. Specifically for Imagen, when both shadow models have undergone extensive training iterations, the attack model trained with $|K| = 10$ achieves a remarkable accuracy of 99.4%. More intriguingly, when the FPR is controlled at 1% and 0.1%, the TPR is recorded at 99.78% and 97.52% respectively. These remarkable findings highlight a substantial increase in accuracy, forming a significant discrepancy compared to the basic accuracy level of 78.1% achieved with $|K| = 1$.

Through the utilization of two approaches, GSA_1 and GSA_2, we seek to elucidate the impact of equidistant timestep sampling frequency on MIA, mainly when applied to large-scale models such as

Imagen. The ultimate goal is to ascertain if we can conserve computational resources without compromising attack effectiveness.

In Figure 5c, we maintain consistent training iterations for both the target and shadow models. This graph depicts how different equidistant timestep sampling frequencies affect the success rate of GSA_1 and GSA_2. We experimented with four distinct frequencies: 1, 2, 5, and 10. Evidently, when restricted to one sampling time, the attack success rate plummets to the lowest. When the sampling frequency doubles, the attack success rate sees a notable increase. The outcome difference between two and five sampling times is minimal for GSA_1. Nevertheless, at a frequency of five times, GSA_2 achieves a success rate comparable to GSA_1 with ten sample times. Impressively, ten sampling times boosts GSA_2's success rate to nearly 100%, indicating a marked improvement. Given the high accuracy achieved by sampling ten times for each sample, further sampling appears unnecessary.

> *Takeaways:* We tested our two attacks primarily on the large-scale model, Imagen, taking into account two factors: the number of training epochs and the timestep sampling frequency. We have examined how overfitting and timestep selection frequency affect the efficacy of our attack strategies.

6 Ablation Study

Following the framework described in Section 3.2, our approach effectively subsamples and aggregates gradients across various dimensions. As evident in Table 5, both GSA_1 and GSA_2 demonstrate exemplary performance on all experiments. Subsequently, we further explore the potential for subsampling and aggregating information from the model layer dimension. We aim to ascertain how gradient data from the model layer influences the attack success rates of GSA_1 and GSA_2. Initially, both GSA_1 and GSA_2 extracted gradient information from every layer of the model for the training of the attack model. However, with the increasing size of dataset and growing model complexity, the computational overhead also rises. Thus, we aim to investigate whether it is feasible to ensure the attack success rates of GSA_1 and GSA_2 without necessarily extracting gradient information from all layers of the model.

Pursuant to this idea, We once again conducted experiments using GSA_1 and GSA_2 on datasets, including CIFAR-10, ImageNet, and MS COCO, while maintaining all other settings according to the default configuration in Table 3. We gradually increased the depth of layers from which we collected gradient information. As illustrated in Figure 9, the x-axis denotes the cumulative number of layers from which gradients are gathered, starting from the top layer. The y-axis employs the True Positive Rate (TPR) at a False Positive Rate (FPR) of 0.1% as the evaluative criterion. The results indicate that as we collect gradient information from increasing layers, the attack success rate correspondingly escalates due to enhanced information accessibility. Remarkably, attaining the highest attack success rate can be achieved merely by gathering gradient data from the top 80% layers of the models. Accordingly, it may not be essential to extract gradient information from each distinct layer of the model, potentially leading to significant computational resource savings.

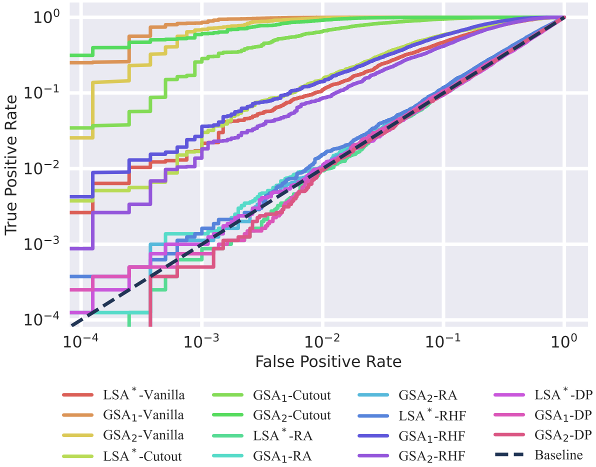


Figure 7: The performance of LSA*, GSA_1 and GSA_2 under varying defensive strategies is displayed. 'Vanilla' refers to the model without any defense methods. 'RA' represents RandAugment, and 'RHF' denotes RandomHorizontalFlip.

7 Defenses

Membership inference attacks are significantly fueled by the overfitting of models to their training data. Thus, mitigating overfitting, such as through data augmentation, could reduce the success rate of these attacks. We employed various methods of data augmentation [8, 9] methods and DP-SGD [1, 13], a strong privacy-preserving method, as defensive mechanisms against the LSA*, GSA_1 and GSA_2 attacks. The results following the implementation of these defense mechanisms are presented in Table 6.

Firstly, fundamental data augmentation techniques such as Cutout [9] and RandomHorizontalFlip (RHF) were employed as defensive measures. All experiments against LSA*, GSA_1, and GSA_2 were conducted using DDPM [21] trained on the CIFAR-10 dataset. In these experiments, the model parameters for LSA* were identical to those for GSA_1 and GSA_2, with the only difference being that LSA* used the loss value as attack features. As shown in Table 6, without any added defense mechanisms, all three attacks achieved high success rates, with GSA_1 and GSA_2 outperforming LSA* (aligned with Section 5.1). When Cutout and RandomHorizontalFlip were applied, LSA* was much more affected than GSA_1 and GSA_2. Specifically, LSA*'s ASR and AUC dropped to around 50% with RHF, while GSA_1 and GSA_2 maintained ASR near 0.80 and AUC scores are above 0.80. This represents that when defending against fundamental data augmentations, the gradient-based GSA_1 and GSA_2 are more robust compared to the loss-based LSA*.

Then, we evaluated the attack performance of LSA*, GSA_1, and GSA_2 using more powerful defensive strategies: DP-SGD [1, 13] and RandAugment [8]. DP-SGD, a widely used method, protects training datasets in machine learning by adding noise to the gradient of each sample, thereby ensuring data privacy. In our experiment, we set the clipping bound C to 1 and the failure probability δ to 1×10^{-5}, keeping the experimental settings consistent with the

Table 6: Efficacy of various defensive measures against LSA*, GSA$_1$, and GSA$_2$. Specifically, DP-SGD and RandAugment significantly hindered the attacks' effectiveness.

Method	DP-SGD		RandAugment		RandomHorizontalFlip		Cutout		No Defense	
	ASR↑	AUC↑	ASR↑	AUC↑	ASR↑	AUC↑	ASR↑	AUC↑	ASR↑	AUC↑
LSA*	0.504	0.508	0.505	0.501	0.524	0.536	0.765	0.846	0.830	0.909
GSA$_1$	0.506	0.511	0.512	0.518	0.793	0.874	0.923	0.977	0.993	0.999
GSA$_2$	0.501	0.502	0.504	0.507	0.737	0.811	0.979	0.997	0.988	0.999

defaults in Table 3. The results show that both DP-SGD and RandAugment effectively defend against LSA* as well as our GSA$_1$ and GSA$_2$, reducing the attack ASR and AUC to levels similar to random guessing. The defense effects are also visualized in Figure 7.

8 Limitation

As shown in Table 5, while GSA$_1$ and GSA$_2$ can yield satisfactory results with limited computational resources, they are still constrained by their time consumption. Even after implementing subsampling and aggregation across three dimensions, the process of gradient extraction remains time-intensive for larger datasets and more intricate models compared to simply computing the loss. Future studies are anticipated to explore these areas further and identify additional dimensions for reduction. Additionally, the methods employed in this study, GSA$_1$ and GSA$_2$, necessitate gradient information from the model for a successful attack. This suggests that requiring complete parameters of the target model during the attack is a rather stringent condition.

9 Related Work

Diffusion Model. Diffusion model is an emergent generative network originally inspired by diffusion processes from non-equilibrium thermodynamics [53]. Distinguished from previous Generative Adversarial Networks (GANs) [7, 10] and Variational Autoencoders (VAEs) [32], the objective of the diffusion model is to approximate the actual data distribution by engaging a parameterized reverse process that aligns with a simulated diffusion process.

Diffusion models can be connected with score-based models [57], generating samples by estimating the gradients of the data distribution and utilizing this gradient information to guide the process of noise addition, thereby producing samples of superior quality. Moreover, the diffusion model showcases the capability to generate images conditioned on specific inputs [10, 36, 42, 47].

Apart from generating images, diffusion models are capable of performing specific area retouching in images according to given specifications, hence effectively accomplishing inpainting [34] tasks. Nowadays, advancements in diffusion models have granted them the ability to generate not only static images but also videos [24] and 3D scenes [17].

Membership Inference Attack. Membership inference attacks, primarily steered by the seminal work of Homer et al. [25], have become an integral part of privacy attack research. The nature of these attacks is typically determined by the depth of information obtained about the target model, whether they are black-box [6, 28, 46, 50, 52, 55, 58, 64] or white-box [38, 44]. The primary objective lies in determining whether a sample is part of the target model's training set using various metrics functions such as loss [46, 64], confidence [50], entropy [50, 55], or difficulty calibration [61].

Defense. As the popularity of diffusion models continues to rise, a growing body of research quickly unfolds around the privacy and security protections associated with these models. Attacks on diffusion models currently extend beyond mere training data leakage [4, 12, 14, 27, 35, 62] to include the potential use of sensitive data for training [51], as well as model theft [40]. Consequently, effective defense mechanisms against these novel attack types have started to emerge. To prevent the leakage of training data from the target model, privacy distillation [14] methods can be employed. Using this approach, a secure diffusion model can be trained on data generated by the target model after sensitive information has been filtered out. This effectively prevents the leakage of sensitive information during the model training process. For artists concerned about their artwork being used to train diffusion models to generate similar styles, GLAZE [51] teams suggest adding a watermark to the original art pieces to prevent them from being mimicked by diffusion models. Simultaneously, for every institution, a diffusion model trained using computational resources can be considered one of the company's assets. As such, the desired target model can be fine-tuned to learn a unique diffusion process [40], which in turn, contributes to the model's protection.

10 Conclusion

In this work, we propose a membership inference attack framework that utilizes the norm of the gradient information in diffusion models and presents two specific attack examples, namely GSA$_1$ and GSA$_2$. We find that the attack performance on the DDPM and Imagen, trained with the CIFAR-10, ImageNet, and MS COCO datasets, is quite remarkable according to all four evaluation metrics. We posit that a diffusion model's gradient information is more indicative of overfitting to a data point than its loss, hence employing gradient information in MIA could lead to higher success rates. This assertion aligns with the nuanced understanding of model dynamics in the machine learning field. Compared to existing white box loss-based attack methodologies [4, 27, 35], our proposed approach demonstrates superior performance under identical model configurations, showcasing efficiency and stability across various datasets and models. This paper introduces the perspective of leveraging gradients for MIA and hopes to inspire valuable follow-up works in this direction.

Acknowledgment

We thank the reviewers for their valuable comments and suggestions. This work is partially supported by NSF CNS-2350332, CNS-2350333, and CNS-2220433.

References

[1] Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (2016), pp. 308–318.

[2] Bao, F., Nie, S., Xue, K., Cao, Y., Li, C., Su, H., and Zhu, J. All are worth words: A vit backbone for diffusion models, 2023.

[3] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership inference attacks from first principles. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), IEEE, pp. 1897–1914.

[4] Carlini, N., Hayes, J., Nasr, M., Jagielski, M., Sehwag, V., Tramer, F., Balle, B., Ippolito, D., and Wallace, E. Extracting training data from diffusion models. *arXiv preprint arXiv:2301.13188* (2023).

[5] Chen, D., Yu, N., Zhang, Y., and Fritz, M. 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.

[6] Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only membership inference attacks. In *International conference on machine learning* (2021), PMLR, pp. 1964–1974.

[7] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. Generative adversarial networks: An overview. *IEEE signal processing magazine 35*, 1 (2018), 53–65.

[8] Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (2020), pp. 702–703.

[9] DeVries, T., and Taylor, G. W. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552* (2017).

[10] Dhariwal, P., and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems 34* (2021), 8780–8794.

[11] Ding, M., Yang, Z., Hong, W., Zheng, W., Zhou, C., Yin, D., Lin, J., Zou, X., Shao, Z., Yang, H., et al. Cogview: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems 34* (2021), 19822–19835.

[12] Duan, J., Kong, F., Wang, S., Shi, X., and Xu, K. Are diffusion models vulnerable to membership inference attacks?, 2023.

[13] Dwork, C. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation* (2008), Springer, pp. 1–19.

[14] Fernandez, V., Sanchez, P., Pinaya, W. H. L., Jacenków, G., Tsaftaris, S. A., and Cardoso, J. Privacy distillation: Reducing re-identification risk of multimodal diffusion models, 2023.

[15] Ganju, K., Wang, Q., Yang, W., Gunter, C. A., and Borisov, N. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security* (2018), pp. 619–633.

[16] Grathwohl, W., Chen, R. T., Bettencourt, J., Sutskever, I., and Duvenaud, D. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367* (2018).

[17] Gu, J., Gao, Q., Zhai, S., Chen, B., Liu, L., and Susskind, J. Learning controllable 3d diffusion models from single-view images. *arXiv preprint arXiv:2304.06700* (2023).

[18] Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., and Guo, B. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 10696–10706.

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

[20] Hilprecht, B., Härterich, M., and Bernau, D. Monte carlo and reconstruction membership inference attacks against generative models. *Proc. Priv. Enhancing Technol. 2019*, 4 (2019), 232–249.

[21] Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems 33* (2020), 6840–6851.

[22] Ho, J., Saharia, C., Chan, W., Fleet, D. J., Norouzi, M., and Salimans, T. Cascaded diffusion models for high fidelity image generation. *The Journal of Machine Learning Research 23*, 1 (2022), 2249–2281.

[23] Ho, J., and Salimans, T. Classifier-free diffusion guidance, 2022.

[24] Ho, J., Salimans, T., Gritsenko, A., Chan, W., Norouzi, M., and Fleet, D. J. Video diffusion models. *arXiv preprint arXiv:2204.03458* (2022).

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

[26] Hu, H., and Pang, J. Membership inference attacks against gans by leveraging over-representation regions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (2021), pp. 2387–2389.

[27] Hu, H., and Pang, J. Membership inference of diffusion models. *arXiv preprint arXiv:2301.09956* (2023).

[28] Hui, B., Yang, Y., Yuan, H., Burlina, P., Gong, N. Z., and Cao, Y. Practical blind membership inference attack via differential comparisons. *arXiv preprint arXiv:2101.01341* (2021).

[29] Kim, G., Kwon, T., and Ye, J. C. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 2426–2435.

[30] Kong, F., Duan, J., Ma, R., Shen, H., Zhu, X., Shi, X., and Xu, K. An efficient membership inference attack for the diffusion model by proximal initialization. *arXiv preprint arXiv:2305.18355* (2023).

[31] Li, J., Li, N., and Ribeiro, B. 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.

[32] Liang, D., Krishnan, R. G., Hoffman, M. D., and Jebara, T. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference* (2018), pp. 689–698.

[33] Liu, Y., Zhao, Z., Backes, M., and Zhang, Y. Membership inference attacks by exploiting loss trajectory. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022), pp. 2085–2098.

[34] Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., and Van Gool, L. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 11461–11471.

[35] Matsumoto, T., Miura, T., and Yanai, N. Membership inference attacks against diffusion models, 2023.

[36] Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.-Y., and Ermon, S. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations* (2021).

[37] Mukherjee, S., Xu, Y., Trivedi, A., Patowary, N., and Ferres, J. L. privgan: Protecting gans from membership inference attacks at low cost to utility. *Proc. Priv. Enhancing Technol. 2021*, 3 (2021), 142–163.

[38] Nasr, M., Shokri, R., and Houmansadr, A. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE symposium on security and privacy (SP)* (2019), IEEE, pp. 739–753.

[39] Nichol, A., Dhariwal, P., Ramesh, A., Shyam, P., Mishkin, P., McGrew, B., Sutskever, I., and Chen, M. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741* (2021).

[40] Peng, S., Chen, Y., Wang, C., and Jia, X. Protecting the intellectual property of diffusion models by the watermark diffusion process, 2023.

[41] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research 21*, 1 (2020), 5485–5551.

[42] Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., and Chen, M. Hierarchical text-conditional image generation with clip latents, 2022.

[43] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In *International Conference on Machine Learning* (2021), PMLR, pp. 8821–8831.

[44] Rezaei, S., and Liu, X. Towards the infeasibility of membership inference on deep models. *arXiv preprint arXiv:2005.13702* (2020).

[45] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models, 2022.

[46] Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., and Jégou, H. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning* (2019), PMLR, pp. 5558–5567.

[47] Saharia, C., Chan, W., Chang, H., Lee, C., Ho, J., Salimans, T., Fleet, D., and Norouzi, M. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 Conference Proceedings* (2022), pp. 1–10.

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

[49] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. Image super-resolution via iterative refinement. *IEEE Transactions on Pattern Analysis and Machine Intelligence 45*, 4 (2022), 4713–4726.

[50] Salem, A., Zhang, Y., Humbert, M., Berrang, P., Fritz, M., and Backes, M. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246* (2018).

[51] Shan, S., Cryan, J., Wenger, E., Zheng, H., Hanocka, R., and Zhao, B. Y. Glaze:

Protecting artists from style mimicry by text-to-image models, 2023.

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

[53] Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning* (2015), PMLR, pp. 2256–2265.

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

[55] Song, L., and Mittal, P. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)* (2021), pp. 2615–2632.

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

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

[58] Truex, S., Liu, L., Gursoy, M. E., Yu, L., and Wei, W. Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing 14*, 6 (2019), 2073–2089.

[59] van der Maaten, L., and Hinton, G. E. Visualizing data using t-sne. *Journal of Machine Learning Research 9* (2008), 2579–2605.

[60] von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., and Wolf, T. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

[61] Watson, L., Guo, C., Cormode, G., and Sablayrolles, A. On the importance of difficulty calibration in membership inference attacks. *arXiv preprint arXiv:2111.08440* (2021).

[62] Wu, Y., Yu, N., Li, Z., Backes, M., and Zhang, Y. Membership inference attacks against text-to-image generation models. *arXiv preprint arXiv:2210.00968* (2022).

[63] Ye, J., Maddi, A., Murakonda, S. K., Bindschaedler, V., and Shokri, R. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022), pp. 3093–3106.

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

[65] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? *Advances in neural information processing systems 27* (2014).

[66] Yu, J., Xu, Y., Koh, J. Y., Luong, T., Baid, G., Wang, Z., Vasudevan, V., Ku, A., Yang, Y., Ayan, B. K., et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789* (2022).

A Additional Information for Denoising Diffusion Probabilistic Model

The operating mechanism of the diffusion model entails the model learning the posterior probability of the forward process, thereby achieving the denoising process. In the forward noise addition process, assume that there is a sample x_{t-1} at time point $t-1$. Then x_t can be represented as:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \sqrt{1-\alpha_t}\epsilon,\ \epsilon \sim \mathcal{N}(0,1) \tag{5}$$

Since ϵ is a random noise, we can unroll the recursive definition and derive x_t directly from x_0 (the original image) and time step t (and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$,):

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t,\ \epsilon_t \sim \mathcal{N}(0,1) \tag{6}$$

The reverse process can be described as:

$$p_\theta(x_{0:T}) = p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)$$

where $x'_T \sim \mathcal{N}(0, I)$. The image x'_{t-1} at $t-1$ can be restored from x'_t at time t, and can be represented as:

$$p_\theta(x'_{t-1}|x'_t) = \mathcal{N}(x'_{t-1}; \boldsymbol{\mu}_\theta(x'_t, t), \Sigma_\theta(x'_t, t)) \tag{7}$$

In the reverse process, the model aims to use the posterior probability of the forward process to guide the denoising process.

$$q(x_{t-1}|x_t, x_0) = \mathcal{N}(x_{t-1}; \bar{\boldsymbol{\mu}}(x_t, x_0), \bar{\boldsymbol{\beta}}_t\mathbf{I})$$

As the $\bar{\boldsymbol{\beta}}_t$ in the posterior probability is also a determined value, the model only needs to learn $\bar{\boldsymbol{\mu}}(x_t, t)$.

In Equation 7, $\boldsymbol{\mu}_\theta(x'_t, t)$ is the predicted mean of the distribution for the sample x'_{t-1} at the preceding timestep, and $\Sigma_\theta(x'_t, t)$ denotes the covariance matrix of this distribution. In the original study, $\Sigma_\theta(x'_t, t) = \sigma_t^2\mathbf{I}$ is set as untrained time-dependent constants. Consequently, our primary attention is dedicated to the mean $\boldsymbol{\mu}_\theta(x'_t, t)$ of the predictive network p_θ. By expanding the aforementioned posterior probability using a probability density function, we can derive the mean and variance of the posterior probability. Given that the variance in $p_\theta(x'_{t-1}|x'_t)$ is associated with β_t and is a deterministic value, our attention is solely on the mean.

When we express x_0 in terms of x_t (from Equation 6) within the mean $\tilde{\boldsymbol{\mu}}(x_t, x_0)$, the revised $\tilde{\boldsymbol{\mu}}(x_t, x_0)$ then only consists of x_t and random noise ϵ_t. Given that x_t is known at the current time step t, the task can be reformulated as predicting the random variable ϵ_t. The $\tilde{\boldsymbol{\mu}}(x_t, x_0)$ can be represented as:

$$\tilde{\boldsymbol{\mu}}(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t)$$

Concurrently, $\boldsymbol{\mu}_\theta(x'_t, t)$ can be expressed as:

$$\boldsymbol{\mu}_\theta(x'_t, t) = \frac{1}{\sqrt{\alpha_t}}(x'_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x'_t, t))$$

Thus, the initial loss function for calculating the prediction of $\boldsymbol{\mu}_\theta(x'_t, t)$ can be reformulated into an equation predicting the noise $\epsilon_\theta(x_t, t)$.

$$\begin{aligned} &L_t(\theta) \\ =&\mathbb{E}_{x_0,\epsilon}\left[\frac{\beta_t^2}{2\sigma_t^2\alpha_t(1-\alpha_t)}||\epsilon_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_t, t)||^2\right]\end{aligned} \tag{8}$$

It has been observed that DDPM [21] relies solely on the marginals $q(x_t|x_0)$ during sampling and loss optimization, rather than directly utilizing the joint probability $q(x_{1:T}|x_0)$. Given that many joint distributions share the same marginals, DDIM [54] proposed a non-Markovian forward process as an alternative to the Markovian noise addition process inherent in DDPM. However, the final non-Markovian noise addition is structurally identical to that of DDPM, with the only distinction being the sampling process.

$$x'_{t-1} = \sqrt{\bar{\alpha}_{t-1}}f_\theta(x'_t, t) + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x'_t, t) + \sigma_t\epsilon$$

Where α_t and ϵ are consistent with the notations used in DDPM. σ_t represents the variance of the noise. The function

$$f_\theta(x'_t, t) = \left(\frac{x'_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x'_t, t)}{\sqrt{\bar{\alpha}_t}}\right)$$

denotes the prediction of x'_0 at timestep t, given x'_t and the pretrained model ϵ_θ.It is worth noting that when $\sigma_t = 0$, the procedure is referred to as the DDIM sampling process, which deterministically generates a sample from latent variables.

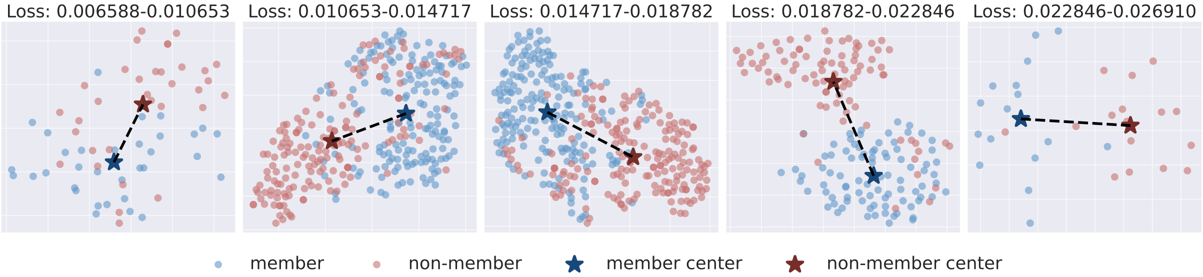


Figure 8: Use t-SNE to represent the member and non-member data pair with the same loss value (rounded to $1e-7$) across five loss intervals. The input to t-SNE is the output of each sample from the last layer of the attack model.

B Additional Likelihood Ratio Attack Details

Carlini et al. [3] contend that it is erroneous to consider the ramifications of misclassifying a sample as a member of the set as identical to those of incorrect non-member set designation. As a result, they proposed a new evaluation metric and introduced their improved method, LiRA, which proved to be far more effective than previous MIA attack methods in experiments, with up to ten times more efficacy under low False Positive Rates (FPRs). The shadow training technique is also needed here, but it involves creating $\mathbb{D}_{\text{in}}$ and $\mathbb{D}_{\text{out}}$ based on each shadow model's response to the same sample depending on whether the sample was used in the model's training or not. This attack method is white-box as it requires access to the model's output loss and some prior knowledge of the target member's dataset, necessitating the use of target points in the shadow model's training.

$$\Lambda = \frac{p(\text{conf}_{\text{obs}} \mid \mathbb{D}_{\text{in}}(x,y))}{p(\text{conf}_{\text{obs}} \mid \mathbb{D}_{\text{out}}(x,y))}$$

The term 'conf$_{\text{obs}}$' refers to the value generated by applying negative exponentiation and logit scaling to the loss produced by the target model for an observed image. '$\mathbb{D}_{\text{in}}$' represents the distribution derived from the processed loss for the member set, while '$\mathbb{D}_{\text{out}}$' stands for the distribution established based on the loss generated for the non-member set samples.

Evidently, the form of LiRA's online attack necessitates retraining the shadow model each time a target point (x, y) is obtained. This approach represents a substantial and arguably uneconomical consumption of resources.

Hence, after proposing this online attack form with many constraints, Carlini et al [3]. suggested an improved offline attack form that does not require target points in shadow models' training and modifies the attack form to:

$$\Lambda = 1 - \Pr[Z > \text{conf}_{\text{obs}}], \text{where } Z \sim \mathbb{D}_{\text{out}}(x,y)) \ .$$

However, the success rate of offline attacks is considerably lower compared to online attacks.

C Additional Information for Methodology

In Section 3, we establish the theoretical foundation for GSA$_1$ and GSA$_2$. Specifically, we emphasize that the loss-based attack faces a challenge: *when member and non-member samples have the same loss value, the attack loses effectiveness.* We demonstrate that, in this situation, the gradient data differ between the two samples.

Therefore, we aim to provide experimental evidence to support this claim in this section. Following the attack pipeline, we continue to use gradient data from the shadow model to train an attack model. Then, we compare the loss values of member and non-member samples in the target model. When the loss values of member and non-member samples are the same, we collect them as a data pair. After collecting all data pairs in the target model member/non-member set, we feed all data pairs into the attack model and extract embeddings from the last layer as inputs to do the t-SNE visualization. In Figure 8, we divide the range of loss values into five intervals and present the data pairs in each interval. It is clear that members and non-members can have different gradients in each data pair. Moreover, the member and non-member samples can form distinct clusters. These results indicate that the challenge posed by identical loss values can be overcome by using gradient data, and that gradient data can serve as better features for the attack.

D Additional Information for Existing Work

D.1 SecMI attack

Drawing from the deterministic reversing and sampling techniques in diffusion models as presented by Song et al. [57] and Kim et al. [29], Duan et al. [12] proposed a query-based method that leverages the sampling process and reverse sampling process error at timestep t as the attack feature. The approximated posterior estimation error can be expressed as:

$$\tilde{\ell}_{t,x_0} = \|\psi_\theta(\phi_\theta(\tilde{x}_t, t), t) - \tilde{x}_t\|^2$$

where

$$\psi_\theta(x_t, t) = \sqrt{\bar{\alpha}_{t-1}} f_\theta(x_t, t) + \sqrt{1-\bar{\alpha}_{t-1}} \epsilon_\theta(x_t, t)$$

represents the deterministic denoising step, and

$$\phi_\theta(x_t, t) = \sqrt{\bar{\alpha}_{t+1}} f_\theta(x_t, t) + \sqrt{1-\bar{\alpha}_{t+1}} \epsilon_\theta(x_t, t)$$

signifies the deterministic reverse step(also called DDIM deterministic forward process [29]) at time t, as defined in the original work [29, 54, 57]. $\tilde{x}_t$ is obtained from the recursive application of ϕ_θ, given by $\phi_\theta(\ldots\phi_\theta(\phi_\theta(x_0, 0), 1), t-1)$.

Based on $\tilde{\ell}_{t,x_0}$, the authors proposed SecMI$_{stat}$ and SecMI$_{NNs}$, which employs the threshold-based attack approach [64] and neural network-based attack method [52], respectively.

D.2 Proximal Initialization Attack (PIA)

Building upon the work of Duan et al. [12], Kong et al. [30] also identified the deterministic properties inherent to the DDIM model [29, 54, 57]. In the DDIM framework, given x_0 and x_k, it is feasible to utilize these two points to predict any other ground truth point x_t [30]. Consequently, this methodology employs the ℓ_p-norm to compute the distance between any ground truth point $x_{t-t'}$ and its predicted counterpart $x'_{t-t'}$. After leveraging the ground truth extraction properties of DDIM [29] and utilizing the sampling formula from [54], the equation to compute the distance is given by:

$$R_{t,p} = \|\epsilon_\theta(x_0, 0) - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_0, 0), t)\|_p.$$

The notation in the above equation is consistent with the DDPM model, where $R_{t,p}$ denotes the distance. Given that ϵ is initialized at $t = 0$, this method is termed the Proximal Initialization Attack (PIA). When normalizing $\epsilon_\theta(x_0, 0)$, it is referred to as PIAN (PIA Normalize). This work employs a threshold-based [64] attack approach.

Compared to SecMI [12], the attack accuracy has seen a notable improvement. Yet, when juxtaposed with white-box attacks [4, 27], the success rate of this model attack remains suboptimal.

D.3 GAN-Leaks

GAN-Leaks [5] is a pivotal work in the realm of MIA against GAN models. This work meticulously breaks down attack scenarios into categories based on the level of access to the latent code, generator, and discriminator. For each category, from full black-box to accessible discriminator, GAN-Leaks presents tailored attack methodologies. This work formalizes MIA as an optimization problem. For a given query sample, the goal is to identify the closest reconstruction by optimizing within the generator's output space. A query sample is deemed a member if its reconstruction error is smaller. This can be represented as:

$$\mathcal{R}(x|\mathcal{G}_v) = G_v(z^*), \text{ where } z^* = \underset{z}{\operatorname{argmin}}\, L(x, G_v(z))$$

where $L(\cdot,\cdot)$ represents the general distance metric, G_v denotes the victim generator, and z^* is the optimal estimate.

GAN-Leaks [5] is a straightforward attack approach that can be universally applied across diverse settings and generative networks. However, its reliability is contingent upon the quality of the reconstructed image, which can be significantly influenced by the complexity of the original image. A complex image, even if it is from the training set, might encompass intricate details leading to a substantial discrepancy between the reconstructed and query images, resulting in misclassification. To address this, the authors employed a calibration technique to rectify such inaccuracies, ensuring commendable attack accuracy for GAN-Leaks on smaller datasets (comprising fewer than 1000 images). Nonetheless, when applied to extensive datasets, the efficacy of GAN-Leaks diminishes.

D.4 Likelihood-based Attack

The log-likelihood of the samples can be used to conduct a membership inference attack. The formula is given by:

$$\log p(x) = \log p_T(x_T) - \int_0^T \nabla \cdot \tilde{\mathrm{f}}_\theta(x_t, t)\, dt.$$

This equation was originally proposed by Song et al. [57]. If the log-likelihood value exceeds the threshold, the sample is inferred as a member. The term $\nabla \cdot \tilde{\mathrm{f}}_\theta(x_t, t)$ is estimated using the Skilling-Hutchinson trace estimator, as suggested by Grathwohl et al. [16].

E Additional Information for Ablation Study

We employed GSA$_1$ and GSA$_2$ on CIFAR-10, ImageNet, and MS COCO to further conduct layer-wise reduction as mentioned in Section 3.2, aiming to reduce computational time and resource consumption. The experimental results are presented in Figure 9.

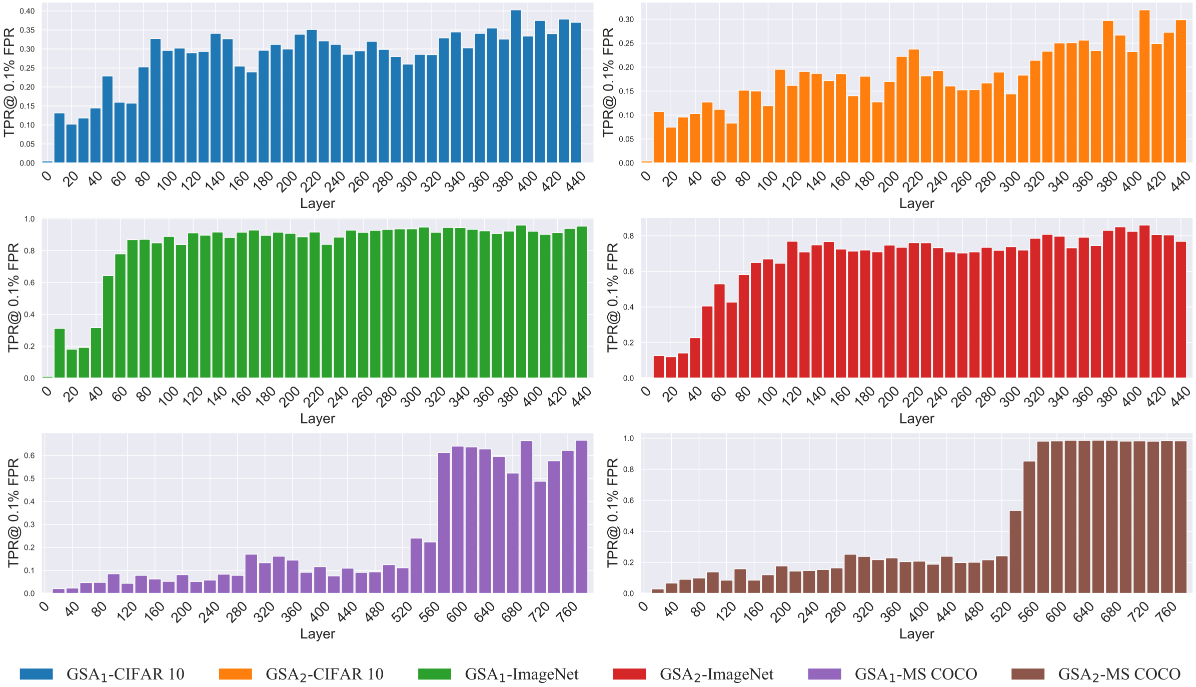


Figure 9: Using GSA$_1$ and GSA$_2$ on CIFAR-10, ImageNet, and MS COCO, we can reduce the layers needed for gradient extraction without compromising attack effectiveness. Notably, for attacks on ImageNet-trained DDPM, only 30% of the layers are required for a successful attack.