
Unveiling Impact of Frequency Components on Membership Inference Attacks for Diffusion Models

Puwei Lian

Southeast University, China
lianpuwei@outlook.com

Yujun Cai

The University of Queensland, Australia
yujun.cai@uq.edu.au

Songze Li*

Southeast University, China
songzeli@seu.edu.cn

Abstract

Diffusion models have achieved tremendous success in image generation, but they also raise significant concerns regarding privacy and copyright issues. Membership Inference Attacks (MIAs) are designed to ascertain whether specific data were utilized during a model’s training phase. As current MIAs for diffusion models typically exploit the model’s image prediction ability, we formalize them into a unified general paradigm which computes the membership score for membership identification. Under this paradigm, we empirically find that existing attacks overlook the inherent deficiency in how diffusion models process high-frequency information. Consequently, this deficiency leads to member data with more high-frequency content being misclassified as hold-out data, and hold-out data with less high-frequency content tend to be misclassified as member data. Moreover, we theoretically demonstrate that this deficiency reduces the membership advantage of attacks, thereby interfering with the effective discrimination of member data and hold-out data. Based on this insight, we propose a plug-and-play high-frequency filter module to mitigate the adverse effects of the deficiency, which can be seamlessly integrated into any attacks within this general paradigm without additional time costs. Extensive experiments corroborate that this module significantly improves the performance of baseline attacks across different datasets and models.

1 Introduction

Diffusion models [13, 35] have achieved significant success in areas such as image generation [4, 14, 33] and video generation [1, 40], and have been widely applied in real life. However, this success has brought increasing attention to copyright and privacy issues from both academia and industry [38, 39, 45]. Recent research shows that diffusion models exhibit a strong memory effect regarding images in the training set, making the risk of privacy leakage a serious concern [8, 39].

Membership Inference Attacks (MIAs) are crucial for assessing model privacy. Their core objective is to determine whether specific data were utilized during a model’s training phase [32]. Generally, MIAs exploit the overfitting characteristics of models. They achieve this by capturing the discrepancies in how the model fits the training data compared to other data, thereby enabling the execution of these attacks [42]. In the field of image generation, there has been extensive prior research on MIAs targeting Variational Autoencoders (VAEs) [12] and Generative Adversarial Networks (GANs) [3, 11]. However, due to the distinct training and generation mechanisms of diffusion models, these established attacks are mostly ineffective when applied to diffusion models [8].

*Corresponding Author.

Recently, MIAs for diffusion models have gradually emerged as a research hotspot. Matsumoto et al. [24] proposed a query-based attack strategy that determines member data by analyzing the model’s loss function. However, this approach overlooks the uncertainty introduced by Gaussian noise. SecMI [8] is based on DDIM inversion [15, 35], obtaining intermediate outputs during generation through a deterministic inversion process. Nevertheless, this method demands a large number of queries, resulting in a significant increase in time costs. Kong et al. [16] introduced a proximal initialization method that utilizes model predictions to obtain initial noise. Although this approach somewhat reduces time overhead, the approximation of initial noise introduces errors that limit the attack performance. Zhai et al. [44] studied text-to-image diffusion models, exploring the associative memory between texts and images. In summary, the core mechanism of these methods is to quantify the model’s image recovery ability by calculating the pixel-wise error and use this as the basis for constructing the membership inference decision logic.

From the perspective of frequency principles, diffusion models exhibit a distinctive generation process: they first denoise low-frequency signals representing overall structure and subsequently incorporate high-frequency details into the samples [41]. This fundamental processing asymmetry means diffusion models handle low-frequency components with greater fidelity and consistency, while high-frequency components show more variation in their reconstruction. While current membership inference attacks have significantly advanced the field, they have not explicitly considered how this frequency-dependent processing affects their effectiveness. This gap is important because a diffusion model’s varying behaviour across frequency bands directly influences its distinctive processing of training versus non-training images, the exact signal MIAs seek to detect. Our analysis within a general pixel-wise error-based MIA paradigm revealed that high-frequency components introduce substantial variance in membership scores, often leading to the misclassification of certain samples. We term this phenomenon "high-frequency deficiency".

Based on the above observations and analyses, we propose a plug-and-play high-frequency filter module. This module exhibits broad applicability and can be integrated into all attacks within the defined general paradigm. Specifically, we transform existing attacks into a metric for quantifying the distance between the target and predicted images. We leverage the Fourier transform to convert the image from the spatial domain to the frequency domain and subsequently apply a filtering operation to selectively remove the high-frequency information. By eliminating the standard deviation caused by high-frequency deficiency, we effectively enhance the performance of existing attacks with negligible additional time overhead. Our contributions can be summarized as follows:

- To the best of our knowledge, this study is the first to explore the impact of frequency domain information on MIAs targeting diffusion models. We formalize a general paradigm for existing pixel-wise error-based attacks and conduct an in-depth analysis of the impact of frequency domain information. The results reveal that existing attacks generally overlook the standard deviation induced by high-frequency deficiency, which restricts their attack performance.
- To address this issue, we proposed a plug-and-play high-frequency filter module. This module effectively suppresses high-frequency deficiency, and we theoretically demonstrated its capacity to improve attack intensity. This module can be seamlessly integrated into all pixel-wise error-based attacks with negligible additional time overhead.
- We conducted extensive experiments to validate the effectiveness of our method. The results indicate that the high-frequency filter significantly improves the performance of existing attacks, achieving substantial improvements in key metrics such as Attack Success Rate (ASR), Area Under the Curve (AUC), and the True Positive Rate at 1% False Positive Rate (TPR@1% FPR).

2 Related Works

Membership Inference Attacks. Shokri et al. [32] proposed the membership inference attacks, which primarily targeted classification models in machine learning. As the evolution of membership inference attacks continues, they can be classified into two categories: black-box attacks [5, 31, 36] and white-box attacks [19, 25], determined by the degree of access granted to the target model. In a white-box setting, the attacker can access the model parameters, whereas in a black-box setting, the attacker only receives the final output of the model. Moreover, noteworthy advancements have been achieved in membership inference attacks targeting generative models. Hayes et al. [11] demonstrated that membership can be effectively discerned through the logits of the discriminator

in GANs. Hilprecht et al. [12] introduced a Monte Carlo scoring methodology incorporating the reconstruction loss term to facilitate attacks on VAEs.

Membership Inference Attacks on Diffusion Models. Recently, membership inference attacks on diffusion models have garnered increasing attention. In white-box settings, Pang et al. [27] proposed executing an attack through the utilization of gradient information extracted from loss. In grey-box settings, the attacker can only access the intermediate and final outputs [8]. Matsumoto et al. [24] pioneered the approach of employing diffusion loss to perform query-based membership inference. Duan et al. [8] introduced an attack leveraging DDIM inversion to retrieve intermediate outputs from the models. Meanwhile, Kong et al. [16] proposed a proximal initialization technique to acquire the deterministic initial noise. The attack is realized by the prediction of this noise from the model. In addition, attacks leveraging the correlation between texts and images have also achieved advancements [22, 39, 44]. Moreover, there has also been a growing interest in attacks for diffusion models in black-box settings [20, 26].

3 Preliminaries

To understand how frequency information affects diffusion models’ behaviour in membership inference contexts, we first establish the fundamental mechanisms of diffusion models and how images can be represented in the frequency domain.

Denoising Diffusion Implicit Model (DDIM). DDIM [35] upgrades the DDPM [13] framework by incorporating a non-Markovian process, which effectively decouples x_{t-1} from x_t . This innovation allows for skipping timesteps, significantly accelerating the sampling process. DDIM redefines the denoising distribution as follows:

$$p_{\theta}(x_{t-1}|x_t) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}}x_0 + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t}x_0}{\sqrt{1 - \bar{\alpha}_t}}, \sigma_t^2 I\right). \quad (1)$$

where $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$ and $(\alpha_1, \dots, \alpha_T)$ are the predefined noise schedules, $\epsilon_{\theta}(x_t, t)$ is predicted by the diffusion models, $\epsilon \sim \mathcal{N}(0, I)$. The denoising process defined by DDIM is outlined as follows:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} \left(\frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_{\theta}(x_t, t)}{\sqrt{\bar{\alpha}_t}} \right) + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2} \epsilon_{\theta}(x_t, t) + \sigma_t \epsilon, \quad (2)$$

where $\sigma_t = \eta \sqrt{(1 - \bar{\alpha}_{t-1}) / (1 - \bar{\alpha}_t)} \sqrt{1 - \bar{\alpha}_t / \bar{\alpha}_{t-1}}$ is the variance schedule, $\eta \in [0, 1]$. The case $\eta = 0$ corresponds to the DDIM, while $\eta = 1$ corresponds to the deterministic DDPM. *More details about the DDPM is provided in Appendix A.*

Frequency Domain Representation of Images. Frequency domain analysis decomposes an image according to a set of basis functions. We focus on the Fourier transform. For simplicity, we only introduce the formulation for grey images, while it is extendable to multi-channel images. Low-frequency components generally correspond to an image’s overall structure and smooth regions, while high-frequency components represent details and edges. Given a $H \times W$ input signal $\mathbf{x} \in \mathbb{R}^{H \times W}$, Discrete Fourier Transform (DFT) projects it onto a collection of sine and cosine waves of different frequencies and phases:

$$\mathbf{X}(u, v) = FFT(\mathbf{x}) = \sum_{x=1}^H \sum_{y=1}^W \mathbf{x}(x, y) e^{-j2\pi(\frac{u}{H}x + \frac{v}{W}y)}, \quad (3)$$

where $\mathbf{x}(x, y)$ is the pixel value at (x, y) ; $\mathbf{X}(u, v)$ represents complex value at frequency (u, v) ; e and j are Euler’s number and the imaginary unit. The inverse Fourier transform is denoted as:

$$\mathbf{x}(x, y) = IFFT(\mathbf{x}) = \frac{1}{HW} \sum_{u=1}^H \sum_{v=1}^W \mathbf{X}(u, v) e^{j2\pi(\frac{u}{H}x + \frac{v}{W}y)}. \quad (4)$$

4 Methodology

4.1 Formalization of MIAs for Diffusion Models

Threat Model. Membership inference focuses on determining whether specific data were utilized in the training. Formally, consider a model f_{θ} parameterized by weights θ and a dataset $D =$

$\{x_1, \dots, x_n\}$ sampled from data distribution q_{data} . Following established conventions [2, 8, 30], D is split into two subsets, D_M and D_H . D_M is the member set of f_θ and D_H is the hold-out set, such that $D = D_M \cup D_H$, $\emptyset = D_M \cap D_H$. So, f_θ is trained on D_M . Each sample x_i is equipped with a membership identifier m_i , where $m_i = 1$ if $x_i \sim D_M$; otherwise, $m_i = 0$. The attackers have access to the f_θ and D but lack knowledge of the specific partitioning between D_M and D_H . The goal is to design an attack algorithm \mathcal{A} that predicts the membership identifier m_i for any given sample x_i :

$$\mathcal{A}(x_i, \theta) = \mathbb{1} [\mathbb{P}(m_i = 1 \mid \theta, x_i) \geq \tau], \quad (5)$$

where $\mathcal{A}(x_i, \theta) = 1$ means x_i comes from D_M , $\mathbb{1}[A] = 1$ if A is true, and τ is the threshold. For generative models, we extend this framework by denoting the generator as G_θ with weights θ and the generative distribution as $p_\theta(x)$, where the generated samples $x \sim p_\theta(x)$.

General Paradigm. Attacks such as Naive [24], SecMI [8] and PIA [16], relying on pixel-wise errors, have demonstrated effectiveness in diffusion models. These attacks share common characteristics, utilizing the model’s image prediction capability to execute attacks. They can be unified as follows: given the image to be tested x_i , calculate the distance between the image $x_{i,t}$ predicted by the model at step t and the target result $x_{i,t}^{target}$ of x_i at step t , then using the distance as the membership score. $x_{i,t}$ and $x_{i,t}^{target}$ are obtained in different ways depending on the algorithm. Furthermore, setting a threshold τ . If the score is less than τ , the image is classified as member data; otherwise, it is classified as hold-out data. Formally, these attacks can be formulated as the general paradigm:

$$\mathcal{A}(x_i, \theta) = \mathbb{1} [\|x_{i,t} - x_{i,t}^{target}\|_q \leq \tau], \quad (6)$$

where q represents the type of norm. We prove in Appendix B that the pixel-wise error-based attacks can be translated into the general paradigm expressed in Eq. 6.

4.2 Frequency Perspective of MIAs for Diffusion Models

Frequency Characteristics of Diffusion Models. Existing attack studies on diffusion models mainly concentrate on pixel-wise errors, yet they neglect an important aspect: analyzing models’ information processing from the frequency domain. Diffusion models’ operational mechanism features distinct frequency hierarchical properties. They first denoise low-frequency signals according to the learned distribution, then utilize specific low-frequency information as prior knowledge to process high-frequency details [28]. Recovering high-frequency information effectively relies not only on the model’s learned distribution but also closely ties to the image’s inherent structure and contours during denoising. Previous research [41] has shown that diffusion models exhibit more variation and uncertainty in handling high-frequency information.

Current pixel-wise error-based attacks mainly evaluate a model’s ability to process individual image. However, high- and low-frequency content varies greatly among images, and diffusion models have different mechanisms for handling such information. These two factors raise an interesting question: Does the high- and low-frequency content within a single image impact existing attack algorithms?

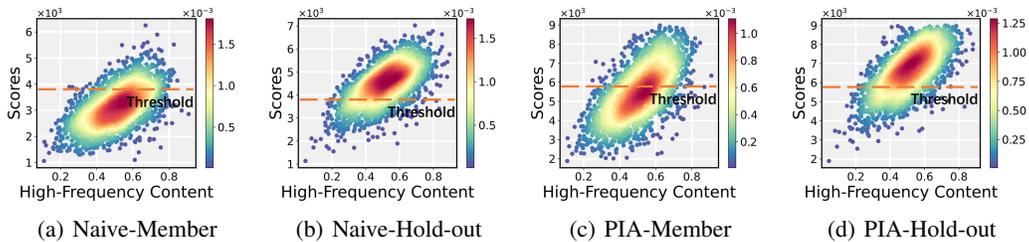


Figure 1: Statistical plots of membership scores versus high-frequency content for the MS-COCO dataset. Horizontal coordinates indicate high-frequency content and vertical coordinates indicate membership scores. We used red to indicate areas with the highest data density.

Frequency Effects on MIAs for Diffusion. To study how frequency domain information affects MIAs, we visualized the relationship between high-frequency content and existing attacks’ membership scores. First, we transform images to the frequency domain using the Fourier transform and divide the high- and low-frequency regions by setting the frequency domain radius = 5 as the high-low frequency boundary. Then, we calculated the percentage of high-frequency content by

summing the squared frequency components in both regions. As shown in Fig. 1, visualizing the scores of attacks reveals a trend that as the high-frequency content of the images increased, so did the membership scores. This shows current attacks are biased, giving higher scores to images with more high-frequency content. Higher scores mean a lower degree of fitting, making it more likely to be classified as hold-out data. We term the phenomenon "high-frequency deficiency".

Table 1: High-frequency content statistics for failed samples. In the failed samples, the high-frequency content of the member data is significantly larger than that of the hold-out data.

| Method | STLU-10 | | Tiny-IN | | MS-COCO | | Flickr | |
|--------|---------|----------|---------|----------|---------|----------|--------|----------|
| | Member | Hold-out | Member | Hold-out | Member | Hold-out | Member | Hold-out |
| Naive | 0.598 | 0.413 | 0.607 | 0.445 | 0.604 | 0.424 | 0.612 | 0.415 |
| PIA | 0.601 | 0.412 | 0.599 | 0.402 | 0.616 | 0.422 | 0.623 | 0.409 |
| SecMI | 0.629 | 0.504 | 0.602 | 0.419 | 0.632 | 0.517 | 0.622 | 0.500 |

To dig deeper into the influence of high-frequency deficiency on attacks, we analyzed the high-frequency content of common attack failure cases. As shown in Tab. 1, in the member dataset, images with more high-frequency content are often misclassified as hold-out data. In contrast, in the hold-out dataset, images with less high-frequency content tend to be wrongly labelled as member data. Additionally, we visualized pixel-level distance analysis of images, which helped us accurately assess the contribution of different image components to attack scores. The results show that high-frequency components of images have a greater impact on scores. Specifically, scores fluctuate significantly with changes in high-frequency content, indicating a strong correlation. Due to space limitations, we provide the visualizations in Appendix D.2.

4.3 High-Frequency Filter Design for Enhanced Diffusion MIAs

Based on the above observations, the following conclusion can be drawn: existing MIAs have not adequately accounted for the effect of high-frequency deficiency, which leads to their failure in certain specific scenarios. Intuitively, the inherent deficiency of diffusion models in handling high-frequency components leads to confusion in distinguishing between images with different high-frequency content. Further, we explore the effects of high-frequency deficiency from a theoretical perspective. Definition 1 gives an advantage measure that characterizes how well an algorithm can distinguish between member data and hold-out data. Theorem 1 characterizes the advantage of score-based MIAs for diffusion models.

Definition 1. [42] *The membership advantage of algorithm \mathcal{A} is defined as:*

$$Adv^M(\mathcal{A}) = \Pr[\mathcal{A} = 1|m = 1] - \Pr[\mathcal{A} = 1|m = 0]. \quad (7)$$

where $\Pr[\mathcal{A} = 1|m = 1]$ indicates the probability that algorithm \mathcal{A} identifies the data as a member when the data is a member. $\Pr[\mathcal{A} = 1|m = 0]$ indicates the probability that if the data is hold-out data, algorithm \mathcal{A} will identify it as a member.

Theorem 1. *Assume membership scores s follow a normal distribution. Let membership scores of member data $s_1 \sim N(0, \sigma_M^2)$ and hold-out data $s_2 \sim N(0, \sigma_H^2)$, where $\sigma_H > \sigma_M$. The membership advantage of algorithm \mathcal{A} is:*

$$Adv^M(\mathcal{A}) = erf\left(\frac{\sigma_H}{\sigma_M} \sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right) - erf\left(\sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right), \quad (8)$$

where $erf(x) = \frac{1}{\sqrt{\pi}} \int_{-x}^x e^{-t^2} dt$.

Proof. We provide the proof of Theorem 1 in Appendix C.

The advantage is 0 when $\sigma_H = \sigma_M$ and increases with the σ_H/σ_M . When $\sigma_H < \sigma_M$ means that there is no member advantage. Mathematically, the high-frequency deficiency contributes equally to a high standard deviation in the scores of both member and hold-out data, which leads to a decrease in σ_H/σ_M , weakening the advantages of identifying member data and consequently interfering with the effective differentiation between member and hold-out data.

Enhanced MIAs Based on High-Frequency Filter. Having established that high-frequency content introduces variability that masks the membership signal, we propose a simple yet effective solution: selectively filtering out this confounding high-frequency information while preserving the more reliable low-frequency components that carry stronger membership signals. Mathematically, this operation is performed as follows:

$$\mathcal{F}(x_{i,t}) = IFFT(FFT(x_{i,t}) \odot \beta_{i,t}(r)), \quad (9)$$

where \odot denotes element-wise multiplication, and $\beta_{i,t}(r)$ is a mask designed as a filtering factor for frequency:

$$\beta_{i,t}(r) = \begin{cases} s & \text{if } r > r_t, \\ 1 & \text{otherwise.} \end{cases} \quad (10)$$

where s serves to implement the frequency-dependent filtering factor, r denotes the frequency domain radius, and r_t is the high-frequency threshold radius. Therefore, our improvement to the general paradigm can be expressed as:

$$\mathcal{A}'(x_i, \theta) = \mathbf{1} [\|\mathcal{F}(x_{i,t}) - \mathcal{F}(x_{i,t}^{target})\|_q \leq \tau], \quad (11)$$

Proposition 1. Denote the original standard deviations of membership scores in member and hold-out data as σ_M and σ_H , and the standard deviations after removing the high-frequency components are σ'_M and σ'_H . The standard deviation of membership scores in the high-frequency components is h_M/h_H , and the low-frequency components is l_M/l_H in member and hold-out data. Let $l_H - l_M = \Delta$, $h_M = k \cdot h_H$ with $k > 0$. If $k^2 > 1 + \frac{2\Delta}{h_H^2}(l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2})$, we have:

$$\sigma'_H/\sigma'_M > \sigma_H/\sigma_M. \quad (12)$$

Proof Sketch. Let $\sigma_H = \sigma'_H + \Delta_H$ and $\sigma_M = \sigma'_M + \Delta_M$, Eq. 12 is equivalent to:

$$\Delta_M - \Delta_H > 0. \quad (13)$$

We can rewrite Eq. 13 in terms of h_H , l_M , Δ and k . We obtain:

$$(k^2 - 1)h_H^2 > 2l_M\Delta + 2\Delta^2 - 2\Delta\sqrt{(l_M + \Delta)^2 + k^2h_H^2}. \quad (14)$$

Some algebraic manipulation gets that:

$$k^2 > 1 + \frac{2\Delta}{h_H^2}(l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2}). \quad (15)$$

We provide the detailed proof of Proposition 1 in Appendix D.

Since $l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2} < 0$, $\sigma'_H/\sigma'_M > \sigma_H/\sigma_M$ is hold constantly when $k \geq 1$. The key insight of Proposition 1 is that when the ratio of h_M/h_H exceeds a certain threshold, filtering the membership scores derived from high-frequency components will amplify the membership advantage. It provides theoretical validation for our high-frequency filtering approach. To further validate the practical applicability of the theoretical framework, we systematically investigated the constraint conditions of k . As shown in Tab. 2, k satisfies its constraint conditions under normal circumstances.

Table 2: Let $f = 1 + \frac{2\Delta}{h_H^2}(l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2})$, we compared the values of k^2 and f in different datasets and methods, k always satisfies its constraint conditions.

| Method | Tiny-IN | | STLU-10 | | MS-COCO | | Flickr | |
|--------|---------|---------|---------|---------|---------|---------|--------|---------|
| | k^2 | f | k^2 | f | k^2 | f | k^2 | f |
| Naive | 0.988 | > 0.921 | 0.976 | > 0.932 | 1.139 | > 0.924 | 0.910 | > 0.892 |
| PIA | 0.976 | > 0.927 | 1.290 | > 0.904 | 1.290 | > 0.914 | 1.317 | > 0.909 |
| SecMI | 1.102 | > 0.920 | 0.959 | > 0.897 | 1.778 | > 0.919 | 1.102 | > 0.911 |

Table 3: Under the default training settings, attack performance of baselines in DDIM. High-frequency filter results in a significant improvement in baseline performance.

| Method | STL10-U | | | CIFAR-100 | | | Tiny-IN | | |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|---------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 73.60 | 79.60 | 5.96 | 70.41 | 77.01 | 7.13 | 74.60 | 81.97 | 7.96 |
| Naive+F | 77.98 | 84.40 | 7.50 | 75.01 | 82.36 | 9.73 | 80.69 | 87.60 | 13.29 |
| SecMI | 81.14 | 87.39 | 11.11 | 80.56 | 87.21 | 16.50 | 82.91 | 89.60 | 13.96 |
| SecMI+F | 86.51 | 91.39 | 14.63 | 88.09 | 93.74 | 24.32 | 90.31 | 93.82 | 25.79 |
| PIA | 80.43 | 87.45 | 9.98 | 77.51 | 84.80 | 12.27 | 80.87 | 86.30 | 14.66 |
| PIA+F | 86.81 | 92.11 | 19.57 | 85.05 | 92.20 | 23.34 | 89.12 | 93.23 | 32.91 |
| Avg+ | +5.38 | +4.49 | +4.48 | +6.56 | +6.43 | +7.16 | +7.25 | +5.59 | +11.80 |

5 Experiments

5.1 Experiment Setup

Datasets and Models. We adhere to the stringent assumption that both member and hold-out data reside within the same distribution. For DDIM, we used the STL10-U [7], CIFAR-100 [17], and Tiny-ImageNet(Tiny-IN) [7] datasets for training, respectively. Specifically, we randomly selected 50% of the training set to serve as member data, while the remaining 50% was designated as hold-out data. For text-to-image diffusion models, we employed 416/417 samples on Pokémon [18], 2500/2500 samples on MS-COCO [23], and 1000/1000 samples on Flickr [43] as the member/hold-out dataset, utilizing stable diffusion v1-4 [6] for fine-tuning. Furthermore, for pre-trained diffusion models, we selected stable diffusion v1-4 and v1-5 [29] as attack targets. We adhere to the settings in [9, 44], employing Laion-MI [9] dataset to ensure that both member and hold-out data exhibit the same distribution. *Detailed training information is provided in Appendix E.2.*

Evaluation Metrics. We use the established metrics employed in prior research [8, 16, 24], which include the Attack Success Rate (ASR), the Area Under the Curve (AUC), and the True Positive Rate (TPR) at 1% False Positive Rate (FPR) (denoted as TPR@1%FPR).

Baselines. We employed the SecMI [8], PIA [16], and Naive [24] as our baselines for comparison. These approaches are characterized as query-based attacks and operate effectively within a grey-box setting. We adopt the parameters advised in their respective publications.

Implementation Details. Both training (fine-tuning) and inference are conducted on a single RTX 3090 GPU(24G). We set $s = 0.2$, $r_t = 5$ in our method and use ℓ_2 norm under the general paradigm.

Table 4: Under the default training settings, attack performance in fine-tuned stable diffusion.

| Method | Pokémon | | | MS-COCO | | | Flickr | | |
|----------------|--------------|--------------|---------------|---------------|--------------|---------------|--------------|--------------|---------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 79.50 | 86.97 | 6.49 | 80.29 | 87.85 | 4.80 | 79.29 | 86.14 | 16.59 |
| Naive+F | 87.88 | 94.14 | 41.25 | 93.60 | 98.32 | 41.99 | 90.90 | 96.82 | 67.60 |
| SecMI | 76.37 | 83.16 | 12.74 | 82.09 | 89.37 | 16.79 | 71.49 | 77.31 | 6.19 |
| SecMI+F | 83.75 | 89.73 | 31.25 | 91.00 | 95.74 | 27.40 | 80.10 | 85.95 | 21.20 |
| PIA | 72.27 | 76.76 | 7.75 | 68.19 | 72.88 | 5.20 | 64.60 | 67.95 | 5.79 |
| PIA+F | 80.87 | 85.44 | 39.25 | 76.00 | 83.08 | 16.59 | 69.30 | 74.62 | 19.60 |
| Avg+ | +8.12 | +7.47 | +28.26 | +10.01 | +9.01 | +19.73 | +8.31 | +8.66 | +26.61 |

5.2 Overall Performance

Denosing Diffusion Implicit Models. For DDIM, we compared the performance of all baselines before and after adding the high-frequency filter, with the relevant results detailed in Tab. 3. We evaluated the average performance improvements of various baselines in different metrics. The experimental results clearly indicate that the filter significantly improves the performance of all baselines. Moreover, the higher the complexity of dataset, the more pronounced this performance improvement becomes. Taking Tiny-IN as an example, after adding the filter, the ASR and AUC

improved by 7.25% and 5.59% on average. The increase in the TPR@1% FPR metric was even more significant, with an average improvement of 11.80% and a maximum improvement of 18.25%.

Stable Diffusion Models. The experimental results for the fine-tuned stable diffusion attacks are presented in Tab. 4. Based on the analysis of the average improvement over baselines, we observed that the ASR improved by 10.01%, AUC improved by 9.01%, and the improvement in the TPR@1% FPR reached as high as 19.73% on the MS-COCO dataset. Notably, on Flickr, our method achieved the highest TPR@1% FPR improvement of 51.01% in Naive. These results strongly indicate that our method significantly enhances the attack efficacy of the baselines across diverse data distributions and scales by mitigating high-frequency deficiency. Moreover, we have conducted tests on the pre-trained stable diffusion, and the results indicate that the filter exhibits only a modest effect, possibly due to the inherent shortcomings of the baselines. *A detailed discussion is provided in Appendix E.3.*

5.3 In-depth Analysis of Attack Performance

We have theoretically proven that removing high-frequency deficiency can amplify the distinction between member and hold-out data. To further validate our conjectures, we visualize the membership scores of the baselines before and after applying the filter. As illustrated in Fig. 2, it is evident that the distribution gap between member data and hold-out data has increased noticeably after applying our method. Blue boxes mark the areas where member and hold-out data interleave, indicating member/hold-out data are indistinguishable by thresholds. After applying the filter, we observe a significant reduction in sample interleavings. This compelling evidence validates our conjectures and demonstrates the filter’s effectiveness. *More visualizations will be presented in Appendix E.4.*

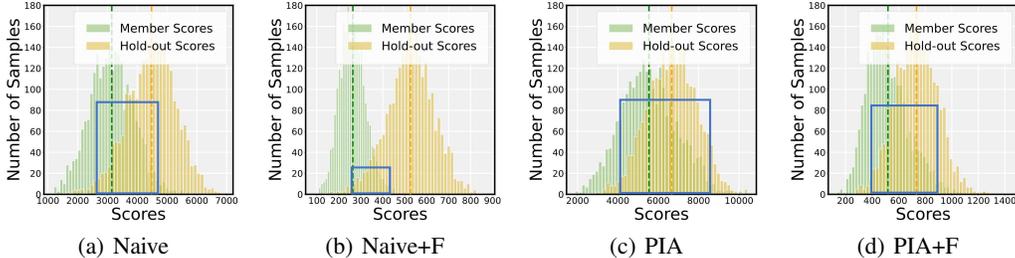


Figure 2: Membership score distribution of member and hold-out data in the MS-COCO dataset. The score distribution gap between member data and hold-out data has noticeably increased.

5.4 Robustness Analysis

Weaker Overfitting. When fine-tuning stable diffusion, the number of iterations is typically determined by the user’s specific needs. In some scenarios, extensive iterations are not required to fine-tune the model. To simulate a model with a lower degree of overfitting, we halved the number of iterations across all datasets, as referenced in [44]. As shown in Tab. 5, the filter demonstrates nice effectiveness. Taking the Flickr dataset as an example, after applying the filter, the ASR improved by an average of 4.34%, with a maximum improvement of 9.25% in Naive. The AUC increased by an average of 7.04%, with the Naive demonstrating the most significant improvement of 14.53%. Compared to the default settings, the filter’s effect is diminished, which is directly linked to the performance decline of the baseline under the weaker overfitting. Therefore, the results can still prove the effectiveness of the high-frequency filter.

Table 5: Attack performance with weaker overfitting assumption in fine-tuned stable diffusion.

| Method | Pokémon | | | MS-COCO | | | Flickr | | |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 70.23 | 74.85 | 5.00 | 60.79 | 63.51 | 1.60 | 67.50 | 68.65 | 5.49 |
| Naive+F | 79.41 | 84.98 | 12.60 | 70.30 | 75.43 | 5.40 | 76.75 | 83.18 | 14.00 |
| SecMI | 70.31 | 74.46 | 4.25 | 67.19 | 71.99 | 3.60 | 71.24 | 77.09 | 4.59 |
| SecMI+F | 72.26 | 76.44 | 5.80 | 69.10 | 73.79 | 4.79 | 72.25 | 79.09 | 6.00 |
| PIA | 66.39 | 67.77 | 3.50 | 55.09 | 53.64 | 0.80 | 60.00 | 59.76 | 2.90 |
| PIA+F | 68.71 | 71.89 | 5.20 | 57.04 | 57.10 | 2.20 | 62.25 | 64.35 | 3.50 |
| Avg+ | +4.48 | +5.41 | +3.61 | +4.46 | +5.73 | +2.13 | +4.34 | +7.04 | +3.51 |

5.5 Ablation Study

To investigate the impact of the filter under different hyperparameter settings, we adjusted the high-frequency threshold r_t and the filtering parameter s . Experiments were conducted with various values for r_t and s on the MS-COCO dataset using Naive attack, and the results are presented in Tab. 6. From the experimental analysis, we recommend a value range for r_t of $[3, 10]$ and for s of $[0.0, 0.3]$. Within this range, the filter achieves optimal performance, significantly enhancing the baseline performance while exhibiting low sensitivity to changes in hyperparameters, further demonstrating the robustness of the filter. When $r_t = 1$, the high-frequency threshold is set too low, leading to most frequency components of the image being filtered. This contradicts our intention to only suppress high-frequency deficiency, resulting in a significant decline in attack performance. When $s = 0.4$ and $s = 0.5$, the filtering effect on the high-frequency components is weakened. Although the baseline performance improves significantly, it does not reach the optimal state. *More ablation experiments will be presented in Appendix E.5.*

Table 6: Naive’ attack performance at different s and r_t in MS-COCO dataset. Our method has lower sensitivity to hyperparameters and performs excellently under most parameter settings.

| s/r_t | $s = 0.0$ | | $s = 0.1$ | | $s = 0.2$ | | $s = 0.3$ | | $s = 0.4$ | | $s = 0.5$ | |
|------------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|
| | ASR | AUC |
| $r_t = 1$ | 69.40 | 73.93 | 81.49 | 89.15 | 82.99 | 90.24 | 80.80 | 88.12 | 80.50 | 87.87 | 80.30 | 87.48 |
| $r_t = 3$ | 93.70 | 97.63 | 93.59 | 97.59 | 92.69 | 96.47 | 89.20 | 94.22 | 84.79 | 91.93 | 83.39 | 90.04 |
| $r_t = 5$ | 94.10 | 97.89 | 94.59 | 97.85 | 93.60 | 98.32 | 90.60 | 95.23 | 87.59 | 93.61 | 84.89 | 91.50 |
| $r_t = 7$ | 93.60 | 97.27 | 93.40 | 96.88 | 91.39 | 96.50 | 91.20 | 95.75 | 88.80 | 94.13 | 86.40 | 92.49 |
| $r_t = 9$ | 92.00 | 96.48 | 91.69 | 96.61 | 91.00 | 95.71 | 90.60 | 95.13 | 88.20 | 94.02 | 86.90 | 92.58 |
| $r_t = 10$ | 91.90 | 96.30 | 91.80 | 96.04 | 90.70 | 95.68 | 89.99 | 94.78 | 87.80 | 93.91 | 87.00 | 92.57 |

5.6 Impact of Data Augmentation

In the training process, data augmentation techniques are typically employed by default to prevent overfitting. For example, DDIM uses RandomHorizontalFlip during training, and stable diffusion employs Random-Crop and Random-Flip by default during fine-tuning [10]. Based on this context, this study refers to the experimental strategies regarding the impact of data augmentation on MIAs from [8, 44], eliminating data augmentation from the training process and attacking the trained models. As indicated in Tab. 7, removing data augmentation while fine-tuning the stable diffusion enhances the attack performance of the baselines, and the filter still shows excellent improvement. Notably, this improvement is particularly pronounced for Naive, where the TPR@1% FPR increased by 40.01%, 54.81%, and 58.70% across different datasets, while the AUC reached 96.81%, 98.42%, and 98.21%, respectively. Moreover, other baselines also exhibited significant performance improvements. *Detailed analysis of the data augmentation in DDIM will be provided in Appendix E.7.*

Table 7: After removing data augmentation, the attack performance in fine-tuned stable diffusion.

| Method | Pokémon | | | MS-COCO | | | Flickr | | |
|----------------|--------------|--------------|---------------|---------------|--------------|---------------|--------------|--------------|---------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 80.00 | 86.45 | 5.99 | 80.09 | 87.61 | 5.99 | 79.50 | 86.17 | 12.80 |
| Naive+F | 92.00 | 96.81 | 46.00 | 95.60 | 98.42 | 60.80 | 93.50 | 98.21 | 71.50 |
| SecMI | 81.08 | 86.67 | 5.30 | 80.90 | 88.23 | 11.20 | 80.69 | 87.55 | 14.80 |
| SecMI+F | 85.00 | 90.42 | 23.00 | 86.50 | 92.61 | 13.60 | 85.50 | 91.76 | 35.50 |
| PIA | 75.49 | 82.64 | 10.60 | 69.90 | 74.36 | 6.40 | 68.40 | 71.68 | 7.00 |
| PIA+F | 87.00 | 92.54 | 28.00 | 79.10 | 86.37 | 25.00 | 76.50 | 82.44 | 31.00 |
| Avg+ | +9.14 | +8.00 | +25.03 | +10.10 | +9.07 | +25.27 | +8.97 | +9.00 | +34.47 |

6 Conclusion

In this paper, we define a general paradigm for the pixel-wise error-based MIAs for diffusion models. Under this general paradigm, we find that the current attacks ignore the intrinsic deficiency of the diffusion model in handling the high-frequency components, which results in limited attack performance. To address this, we introduce a simple and efficient method which mitigates the negative

impact of high-frequency deficiency on MIAs by filtering images' high-frequency information. Experimental results reveal that our method can be seamlessly incorporated into attacks within the general paradigm, significantly enhancing attack performance across diverse settings.

Limitations. Our method exhibits limited efficacy in the pre-training setting, likely due to the overall substandard performance of existing attacks in this setting. This issue may arise from the fact that the pre-trained model does not align with the current assumptions related to overfitting. Future investigations need to thoroughly explore MIAs in the pre-training setting.

References

- [1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22563–22575, 2023.
- [2] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE, 2022.
- [3] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In *Proceedings of the 2020 ACM SIGSAC conference on computer and communications security*, pages 343–362, 2020.
- [4] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.
- [5] Christopher A Choquette-Choo, Florian Tramèr, Nicholas Carlini, and Nicolas Papernot. Label-only membership inference attacks. In *International conference on machine learning*, pages 1964–1974. PMLR, 2021.
- [6] CompVis. Stable diffusion v1-4, 2024. URL <https://huggingface.co/CompVis/stable-diffusion-v1-4>.
- [7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [8] Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *International Conference on Machine Learning*, pages 8717–8730. PMLR, 2023.
- [9] Jan Dubiński, Antoni Kowalczyk, Stanisław Pawlak, Przemysław Rokita, Tomasz Trzciniński, and Paweł Morawiecki. Towards more realistic membership inference attacks on large diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4860–4869, 2024.
- [10] Hugging Face. Fine-tuning stable diffusion, 2024. URL https://github.com/huggingface/diffusers/blob/main/examples/text_to_image/train_text_to_image.py.
- [11] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. Logan: Membership inference attacks against generative models. *arXiv preprint arXiv:1705.07663*, 2017.
- [12] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019.
- [13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [14] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024.
- [15] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2426–2435, 2022.

- [16] Fei Kong, Jinhao Duan, RuiPeng Ma, Hengtao Shen, Xiaofeng Zhu, Xiaoshuang Shi, and Kaidi Xu. An efficient membership inference attack for the diffusion model by proximal initialization. *arXiv preprint arXiv:2305.18355*, 2023.
- [17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- [18] Lambda. Pokemon blip captions, 2023. URL <https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions>.
- [19] Klas Leino and Matt Fredrikson. Stolen memories: Leveraging model memorization for calibrated {White-Box} membership inference. In *29th USENIX security symposium (USENIX Security 20)*, pages 1605–1622, 2020.
- [20] J. Li, J. Dong, T. He, and J. Zhang. Towards black-box membership inference attack for diffusion models. *arXiv preprint arXiv:2405.20771*, 2024.
- [21] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.
- [22] Qiao Li, Xiaomeng Fu, Xi Wang, Jin Liu, Xingyu Gao, Jiao Dai, and Jizhong Han. Unveiling structural memorization: Structural membership inference attack for text-to-image diffusion models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 10554–10562, 2024.
- [23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer vision—ECCV 2014: 13th European conference, zurich, Switzerland, September 6–12, 2014, proceedings, part v 13*, pages 740–755. Springer, 2014.
- [24] Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*, pages 77–83. IEEE, 2023.
- [25] Milad Nasr, Reza Shokri, and Amir Houmansadr. 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)*, pages 739–753. IEEE, 2019.
- [26] Yan Pang and Tianhao Wang. Black-box membership inference attacks against fine-tuned diffusion models. *arXiv preprint arXiv:2312.08207*, 2023.
- [27] Yan Pang, Tianhao Wang, Xuhui Kang, Mengdi Huai, and Yang Zhang. White-box membership inference attacks against diffusion models. *arXiv preprint arXiv:2308.06405*, 2023.
- [28] Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving average sampling in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8911–8920, 2024.
- [29] RunwayML. Stable diffusion v1-5, 2024. URL <https://huggingface.co/runwayml/stable-diffusion-v1-5>.
- [30] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. White-box vs black-box: Bayes optimal strategies for membership inference. In *International Conference on Machine Learning*, pages 5558–5567. PMLR, 2019.
- [31] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.
- [32] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
- [33] Chenyang Si, Ziqi Huang, Yuming Jiang, and Ziwei Liu. Freeu: Free lunch in diffusion u-net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4733–4743, 2024.
- [34] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. pmlr, 2015.

- [35] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [36] Liwei Song and Prateek Mittal. Systematic evaluation of privacy risks of machine learning models. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2615–2632, 2021.
- [37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.
- [38] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N Metaxas, and Shiqing Ma. Diagnosis: Detecting unauthorized data usages in text-to-image diffusion models. *arXiv preprint arXiv:2307.03108*, 2023.
- [39] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.
- [40] Tianxing Wu, Chenyang Si, Yuming Jiang, Ziqi Huang, and Ziwei Liu. Freeinit: Bridging initialization gap in video diffusion models. In *European Conference on Computer Vision*, pages 378–394. Springer, 2024.
- [41] Xingyi Yang, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Diffusion probabilistic model made slim. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 22552–22562, 2023.
- [42] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st computer security foundations symposium (CSF)*, pages 268–282. IEEE, 2018.
- [43] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the association for computational linguistics*, 2:67–78, 2014.
- [44] Shengfang Zhai, Huanran Chen, Yinpeng Dong, Jiajun Li, Qingni Shen, Yansong Gao, Hang Su, and Yang Liu. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. *Advances in Neural Information Processing Systems*, 37:74122–74146, 2024.
- [45] Gong Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1755–1764, 2024.

A Denoising Diffusion Probabilistic Model (DDPM)

DDPM [13] consists of a forward process and a denoising process. In the forward process, DDPM transitions from an intractable data distribution, represented as $x_0 \sim q_0(x_0)$, to a Gaussian distribution $q_T(x_T) \sim \mathcal{N}(x_T; 0, I)$. This transition is achieved by progressively adding Gaussian noise to the original image x_0 . Consequently, the transition distribution at timestep t is defined as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1}, (1 - \alpha_t)I), \quad (\text{A.1})$$

where $\alpha_1, \dots, \alpha_T$ are the predefined noise schedules. Leveraging the properties of chained Gaussian processes, DDPM defines $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$. Consequently, the value of x_t is computed in a single step:

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (\text{A.2})$$

DDPM allows for a simplification of the optimization objective:

$$L_{sample} = \mathbb{E}_{x_0, \epsilon} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|^2 \right], \quad (\text{A.3})$$

where $\epsilon_\theta(x_t, t)$ is predicted by the diffusion models. The denoising (or reverse) process shares the same functional form as the forward process [34]. It is expressed as a Gaussian transition characterized by a learned mean and a fixed variance:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}\left(x_{t-1}; \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right), \sigma_t^2 I\right). \quad (\text{A.4})$$

The denoising process defined by DDPM is outlined as follows:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}} \epsilon_\theta(x_t, t) \right) + \sigma_t z, \quad (\text{A.5})$$

where $z \sim \mathcal{N}(0, I)$, which brings uncertainty and diversity to the denoising process.

B General Paradigm

Naive [24] determines member data through the loss function, specifically judging based on the distance between the added noise and the predicted noise. SecMI [8] leverages diffusion inversion to obtain the ground truth at step t and step $t - 1$. Based on this, the model predicts x_{t-1} from the ground truth at step t and assesses whether a sample is a member by examining the distance between the predicted x_{t-1} and its ground truth. PIA [16] retrieves an initial noise through a proximal initialization process, then uses the model to predict noise for samples containing that initial noise. Finally, it evaluates the membership status based on the distance between the predicted noise and the initial noise. In this section, we will demonstrate that the baselines Naive, PIA and SecMI can be translated into the general paradigm we have defined, and here we use the ℓ_1 norm as an example.

Naive: According to Eq. A.2, x_0 can be expressed as:

$$x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon}{\sqrt{\bar{\alpha}_t}}, \quad \epsilon \sim \mathcal{N}(0, I). \quad (\text{B.1})$$

Naive recognizes membership based on the following inequality:

$$\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\| < c, \quad (\text{B.2})$$

where $\epsilon_\theta(\cdot)$ indicates the predicted noise. Multiply $\sqrt{1 - \bar{\alpha}_t}$ on both sides of Eq. B.2:

$$\|\sqrt{1 - \bar{\alpha}_t}\epsilon - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\| < \sqrt{1 - \bar{\alpha}_t}c, \quad (\text{B.3})$$

which is equivalent to:

$$\|\sqrt{1 - \bar{\alpha}_t}\epsilon - x_t + x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)\| < \sqrt{1 - \bar{\alpha}_t}c. \quad (\text{B.4})$$

Divide both sides of Eq. B.4 by $\sqrt{\bar{\alpha}_t}$:

$$\left\| \frac{\sqrt{1 - \bar{\alpha}_t}\epsilon - x_t + x_t - \sqrt{1 - \bar{\alpha}_t}\epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, t)}{\sqrt{\bar{\alpha}_t}} \right\| < \frac{\sqrt{1 - \bar{\alpha}_t}c}{\sqrt{\bar{\alpha}_t}}. \quad (\text{B.5})$$

According to Eq. B.1, we can set:

$$x_0^{target} = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon}{\sqrt{\bar{\alpha}_t}}, x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta (\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, t)}{\sqrt{\bar{\alpha}_t}}. \quad (\text{B.6})$$

Therefore, Eq. B.5 can be converted to the distance between the original image and the target image:

$$\|x_0 - x_0^{target}\| \leq \tau, \quad (\text{B.7})$$

where $\tau = c\sqrt{1 - \bar{\alpha}_t}/\sqrt{\bar{\alpha}_t}$.

PIA: In order to reduce the error caused by random noise, the authors used proximal initialization to obtain the initial noise so as to improve Naive. It is expressed as:

$$\|\epsilon_\theta(x_0, 0) - \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_0, 0), t)\| < c, \quad (\text{B.8})$$

where $\epsilon_\theta(x_0, 0)$ represents the noise prediction for x_0 . In the same way, Eq. B.8 can be converted to:

$$\left\| \frac{\sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_0, 0) - x_t + x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_0, 0), t)}{\sqrt{\bar{\alpha}_t}} \right\| < \frac{\sqrt{1 - \bar{\alpha}_t} c}{\sqrt{\bar{\alpha}_t}}. \quad (\text{B.9})$$

According to Eq. B.1, we can set:

$$x_0^{target} = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_0, 0)}{\sqrt{\bar{\alpha}_t}}, x_0 = \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_0, 0), t)}{\sqrt{\bar{\alpha}_t}}. \quad (\text{B.10})$$

Therefore, Eq. B.9 can be converted to the distance between the original image and the target image:

$$\|x_0 - x_0^{target}\| \leq \tau, \quad (\text{B.11})$$

where $\tau = c\sqrt{1 - \bar{\alpha}_t}/\sqrt{\bar{\alpha}_t}$.

SecMI: Inspired by recent works on deterministic reversing and sampling from diffusion models [15, 37], SecMI used DDIM and DDIM inversion deterministic sampling in the forward and backward processes for the samples to be tested:

$$x_{t+1} = \phi_\theta(x_t, t) = \sqrt{\bar{\alpha}_{t+1}} \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} + \sqrt{1 - \bar{\alpha}_{t+1}} \epsilon_\theta(x_t, t), \quad (\text{B.12})$$

$$x_{t-1} = \psi_\theta(x_t, t) = \sqrt{\bar{\alpha}_{t-1}} \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \epsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}} + \sqrt{1 - \bar{\alpha}_{t-1}} \epsilon_\theta(x_t, t). \quad (\text{B.13})$$

Denote by $\Phi_\theta(x_s, t)$ the deterministic reverse, i.e., from x_s to x_t ($s < t$):

$$x_t = \Phi_\theta(x_s, t) = \phi_\theta(\dots \phi_\theta(\phi_\theta(x_s, s), s+1), t-1), \quad (\text{B.14})$$

and $\Psi_\theta(x_t, s)$ the deterministic denoise process, i.e., from x_t to x_s :

$$x_s = \Psi_\theta(x_t, s) = \psi_\theta(\dots \psi_\theta(\psi_\theta(x_t, t), t-1), s+1). \quad (\text{B.15})$$

Then, SecMI define t -error as the approximated posterior estimation error at step t , The algorithm is defined as:

$$\|\psi_\theta(\phi_\theta(\tilde{x}_t, t), t) - \tilde{x}_t\| < c, \quad (\text{B.16})$$

where $\tilde{x}_t = \Phi_\theta(x_0, t)$. A series of transformations are designed to fetch x_{t+1}^{target} and x_t^{target} and predict x_t by x_{t+1}^{target} . Therefore, Eq. B.16 can be expressed as:

$$\|x_t - x_t^{target}\| \leq \tau, \quad (\text{B.17})$$

where $\tau = c$, $x_t = \psi_\theta(\phi_\theta(\tilde{x}_t, t), t)$, $x_t^{target} = \tilde{x}_t$.

C Proof of Theorem 1

Theorem 1. Assume membership scores s follow a normal distribution. Let membership scores of member data $s_1 \sim N(0, \sigma_M^2)$ and hold-out data $s_2 \sim N(0, \sigma_H^2)$, where $\sigma_H > \sigma_M$. The membership advantage of algorithm \mathcal{A} is:

$$Adv^M(\mathcal{A}) = erf\left(\frac{\sigma_H}{\sigma_M} \sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right) - erf\left(\sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right). \quad (\text{C.1})$$

Proof. For a given sample x_i , the membership score can be obtained according to the general paradigm we have defined:

$$s = \|x_{i,t} - x_{i,t}^{target}\|. \quad (\text{C.2})$$

Following [42], let the error conditional probability density functions $f(s|m=1)$ and $f(s|m=0)$:

$$f(s|m=1) = \frac{1}{\sqrt{2\pi}\sigma_M} e^{-s^2/(2\sigma_M^2)}, \quad (\text{C.3})$$

$$f(s|m=0) = \frac{1}{\sqrt{2\pi}\sigma_H} e^{-s^2/(2\sigma_H^2)}. \quad (\text{C.4})$$

After knowing $f(s|m=0)$ and $f(s|m=1)$, the attacker selects the maximum likelihood of $f(s|m)$ as the output. This is based on determining the probability density function to decide whether the data point x comes from the member data ($m=1$) or the hold-out data ($m=0$). Given $f(s|m=0) = f(s|m=1)$, the formula is equivalent to:

$$\frac{1}{\sqrt{2\pi}\sigma_M} e^{-s^2/(2\sigma_M^2)} = \frac{1}{\sqrt{2\pi}\sigma_H} e^{-s^2/(2\sigma_H^2)}. \quad (\text{C.5})$$

By the properties of logarithms, we obtain:

$$\ln 1 - \ln(\sqrt{2\pi}\sigma_M) + \ln\left(e^{-s^2/(2\sigma_M^2)}\right) = \ln 1 - \ln(\sqrt{2\pi}\sigma_H) + \ln\left(e^{-s^2/(2\sigma_H^2)}\right), \quad (\text{C.6})$$

which is equivalent to:

$$-\frac{s^2}{2\sigma_M^2} - \ln(\sigma_M) = -\frac{s^2}{2\sigma_H^2} - \ln(\sigma_H). \quad (\text{C.7})$$

Let $\pm s_{eq}$ be the points at which these two probability density functions are equal:

$$s_{eq} = \sigma_H \sqrt{\frac{2\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}. \quad (\text{C.8})$$

If $\sigma_M < \sigma_H$, $f(s|m=1) > f(s|m=0)$ if and only if $|s| < s_{eq}$. The membership advantage, as defined in Definition 1, can be rewritten as:

$$Adv^M(\mathcal{A}) = \Pr[|s| < s_{eq}|m=1] - \Pr[|s| < s_{eq}|m=0]. \quad (\text{C.9})$$

According to the relationship between the error function and the standard normal distribution, Eq. C.9 can be converted to:

$$Adv^M(\mathcal{A}) = erf\left(\frac{s_{eq}}{\sqrt{2}\sigma_M}\right) - erf\left(\frac{s_{eq}}{\sqrt{2}\sigma_H}\right). \quad (\text{C.10})$$

Substituting s_{eq} into the Eq. C.10 yields:

$$Adv^M(\mathcal{A}) = erf\left(\frac{\sigma_H}{\sigma_M} \sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right) - erf\left(\sqrt{\frac{\ln(\sigma_H/\sigma_M)}{(\sigma_H/\sigma_M)^2 - 1}}\right). \quad (\text{C.11})$$

Therefore, Theorem 1 is proved.

D Proof of Proposition 1

Proposition 1. Denote the original standard deviations of membership scores in member and hold-out data as σ_M and σ_H , and the standard deviations after removing the high-frequency components are σ'_M and σ'_H . The standard deviation of membership scores in the high-frequency components is h_M/h_H , and the low-frequency components is l_M/l_H in member and hold-out data. Let $l_H - l_M = \Delta$, $h_M = k \cdot h_H$ with $k > 0$. If $k^2 > 1 + \frac{2\Delta}{h_H^2}(l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2})$, we have:

$$\sigma'_H/\sigma'_M > \sigma_H/\sigma_M. \quad (\text{D.1})$$

Proof. We follow the premise of Theorem 1, the standard deviation before and after filtering high-frequency information satisfies $\sigma_H > \sigma_M$ and $\sigma'_H > \sigma'_M$. $\frac{\sigma'_H}{\sigma'_M} > \frac{\sigma_H}{\sigma_M}$ is equivalent to:

$$\sigma'_H\sigma_M > \sigma_H\sigma'_M. \quad (\text{D.2})$$

Let $\sigma_H = \sigma'_H + \Delta_H$ and $\sigma_M = \sigma'_M + \Delta_M$. Substituting these into Eq. D.2, we obtain:

$$\sigma'_H\sigma'_M + \sigma'_H\Delta_M > \sigma'_H\sigma'_M + \Delta_H\sigma'_M. \quad (\text{D.3})$$

Since $\sigma'_H > \sigma'_M > 0$, Eq. D.3 holds constantly if and only if:

$$\Delta_M - \Delta_H > 0. \quad (\text{D.4})$$

Let the errors of the high- and low-frequency are independent, denoted as $\text{cov}(l_M, h_M) \approx \text{cov}(l_H, h_H) \approx 0$. From the principle of normal distribution superposition, this gives:

$$\Delta_M = \sqrt{l_M^2 + h_M^2 + \text{cov}(l_M, h_M)} - \sqrt{l_M^2}, \Delta_H = \sqrt{l_H^2 + h_H^2 + \text{cov}(l_H, h_H)} - \sqrt{l_H^2}. \quad (\text{D.5})$$

Eq. D.4 can be expressed as:

$$\sqrt{l_M^2 + h_M^2 + \text{cov}(l_M, h_M)} - \sqrt{l_H^2 + h_H^2 + \text{cov}(l_H, h_H)} > l_M - l_H. \quad (\text{D.6})$$

Since $l_H - l_M = \Delta = \sigma'_H - \sigma'_M > 0$ and $\text{cov}(l_M, h_M) \approx \text{cov}(l_H, h_H) \approx 0$, we obtain:

$$\sqrt{l_M^2 + h_M^2} > \sqrt{(l_M + \Delta)^2 + h_H^2} - \Delta. \quad (\text{D.7})$$

Square both sides of Eq. D.7 to obtain the following:

$$l_M^2 + h_M^2 > (l_M + \Delta)^2 + h_H^2 - 2\Delta\sqrt{(l_M + \Delta)^2 + h_H^2} + \Delta^2. \quad (\text{D.8})$$

Substitute $h_M = k \cdot h_H$ into Eq. D.8, then:

$$(k^2 - 1)h_H^2 > 2l_M\Delta + 2\Delta^2 - 2\Delta\sqrt{(l_M + \Delta)^2 + k^2h_H^2}. \quad (\text{D.9})$$

Add $(l_M + \Delta)^2 + h_H^2$ on both sides of Eq. D.9:

$$(l_M + \Delta)^2 + k^2h_H^2 > 2l_M\Delta + 2\Delta^2 - 2\Delta\sqrt{(l_M + \Delta)^2 + k^2h_H^2} + (l_M + \Delta)^2 + h_H^2. \quad (\text{D.10})$$

Let $t = (l_M + \Delta)^2 + k^2h_H^2$, substitute t into Eq. D.10:

$$t > 2l_M\Delta + 2\Delta^2 - 2\Delta\sqrt{t} + (l_M + \Delta)^2 + h_H^2, \quad (\text{D.11})$$

which is equivalent to:

$$(\sqrt{t} + \Delta)^2 > (l_M + 2\Delta)^2 + h_H^2. \quad (\text{D.12})$$

Therefore, we have:

$$\sqrt{t} > \sqrt{(l_M + 2\Delta)^2 + h_H^2} - \Delta. \quad (\text{D.13})$$

Substitute back $t = (l_M + \Delta)^2 + k^2h_H^2$ to Eq. D.13, we obtain:

$$(l_M + \Delta)^2 + k^2h_H^2 > (l_M + 2\Delta)^2 + h_H^2 - 2\Delta\sqrt{(l_M + 2\Delta)^2 + h_H^2} + \Delta^2. \quad (\text{D.14})$$

We ultimately obtain:

$$k^2 > 1 + \frac{2\Delta}{h_H^2}(l_M + 2\Delta - \sqrt{(l_M + 2\Delta)^2 + h_H^2}). \quad (\text{D.15})$$

Therefore, Proposition 1 is proved.

E Complementary Experiments

E.1 More Frequency Domain Analysis

As shown in Fig. 3, on the Flickr dataset, membership scores statistics align with our conjecture: they are positively correlated with high-frequency content, increasing as the latter rises.

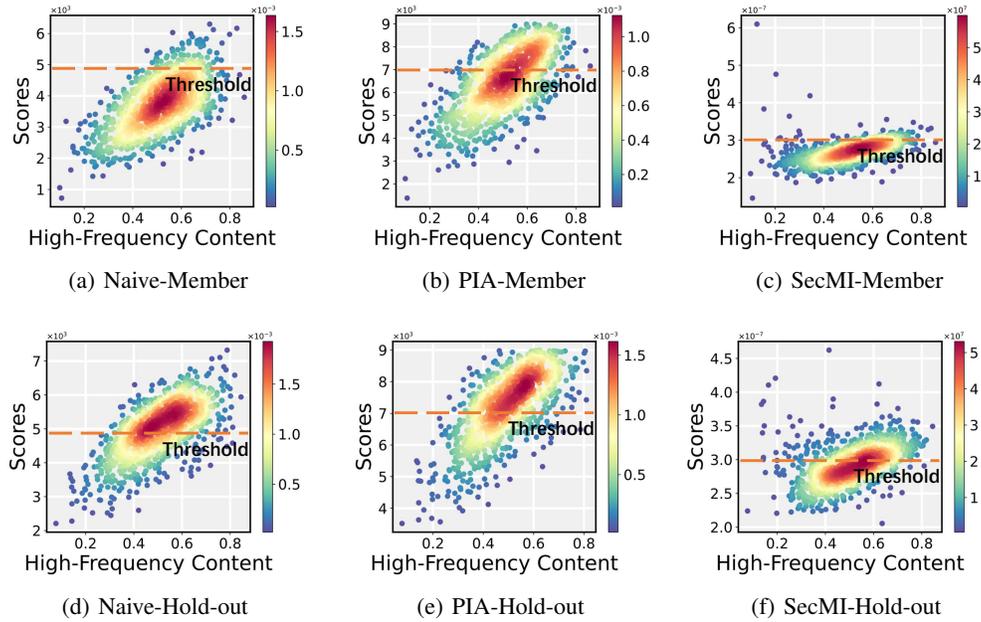


Figure 3: Statistical plots of membership scores versus high-frequency content on the Flickr dataset.

In addition, we visualize the distribution of membership scores contributions in Fig. 4 and Fig. 5, comparing the original images with the pixel-wise errors. Color depth characterizes the magnitude of errors; the deeper the color, the larger the corresponding error at that location. We have observed that areas of high error often coincide with areas of high-frequency information. Due to the variability in high-frequency content across different images, the extent of errors displays significant differences.

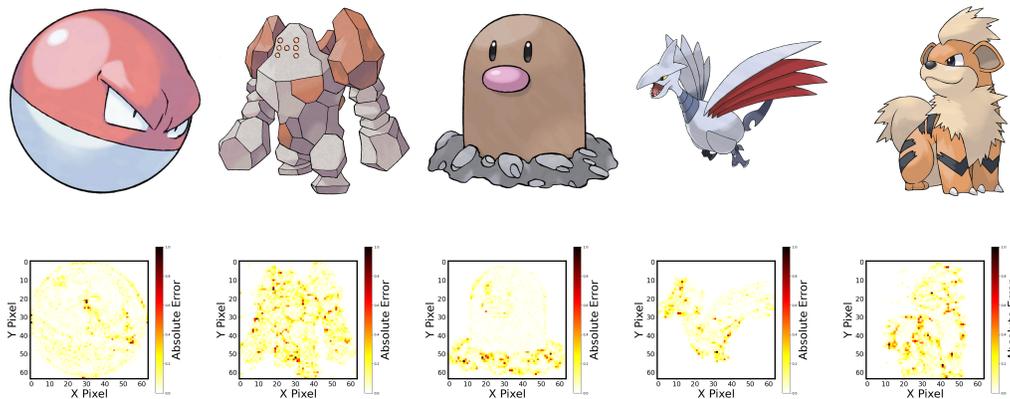


Figure 4: Naive pixel-wise errors distribution visualization, with the top half being the original image and the bottom half being the error visualization. The areas of high error often coincide with areas of high-frequency information.

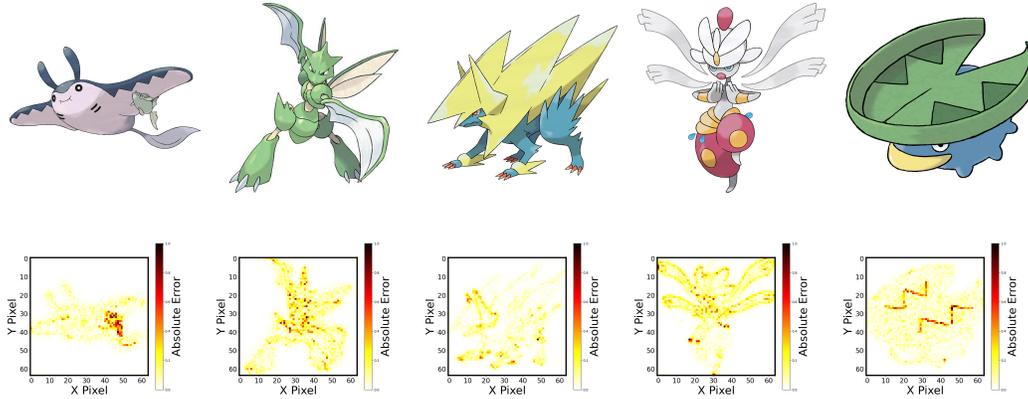


Figure 5: PIA pixel-wise errors distribution visualization.

E.2 Detailed settings

As shown in Tab. 8, we present the segmentation of member and hold-out data across all datasets, ensuring that both member and hold-out data are independently and identically distributed, with equal quantities. Furthermore, we specify the batch size and the number of iterations for training across different datasets. Attacking under pre-trained conditions does not require additional training of the models.

Table 8: Detailed dataset settings and model settings.

| Model | Dataset | Resolution | Member | Hold-out | Batch-size | Iterations |
|-------------------------|---------------|------------|--------|----------|------------|------------|
| DDIM | CIFAR-100 | 32 | 25000 | 25000 | 128 | 800000 |
| | STL10-U | 32 | 50000 | 50000 | 128 | 1600000 |
| | Tiny-ImageNet | 32 | 50000 | 50000 | 128 | 1600000 |
| Stable Diffusion v1.4 | Pokémon | 512 | 416 | 417 | 1 | 15000 |
| | Flickr | 512 | 1000 | 1000 | 1 | 60000 |
| | MS-COCO | 512 | 2500 | 2500 | 1 | 150000 |
| Stable Diffusion v1.4/5 | Laion-MI | 512 | 2500 | 2500 | / | / |

E.3 Attack for Pre-trained Stable Diffusion

As shown in Tab. 9, when we tried to attack the pre-trained models, the effect of the filter was weak. This phenomenon can be attributed to the fact that the baselines completely failed in the pre-training setting. Their performance on ASR and AUC metrics was nearly equivalent to random guessing. The likely reason is that the assumptions of current attacks are not well-suited to the pre-training configuration. As a result, even when our filter was applied, it was difficult to achieve significant performance improvements.

Table 9: Under the pre-trained settings, the attack performance in stable diffusion.

| Method | SD1.4 | | | SD1.5 | | |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 53.19 | 52.82 | 0.20 | 53.86 | 52.69 | 0.20 |
| Naive+F | 53.50 | 53.09 | 0.20 | 54.10 | 53.07 | 0.20 |
| SecMI | 53.29 | 50.99 | 0.60 | 52.88 | 52.99 | 1.40 |
| SecMI+F | 54.50 | 51.85 | 1.60 | 54.30 | 53.43 | 1.80 |
| PIA | 52.99 | 52.04 | 0.20 | 53.67 | 53.22 | 0.40 |
| PIA+F | 53.10 | 52.11 | 0.20 | 53.70 | 53.24 | 0.40 |
| Avg+ | +0.54 | +0.40 | +0.33 | +0.56 | +0.28 | +0.13 |

E.4 Membership Scores Distribution for Samples from Member and Hold-out Set.

As illustrated in Fig. 6, we further present the distribution of membership scores on the Pokémon and Flickr datasets. Additionally, we conducted a statistical analysis of the σ_H/σ_M . As shown in Tab. 10, the results strongly validate the effectiveness of our method: after applying the high-frequency filter, the σ_H/σ_M exhibits varying degrees of improvement, which aligns closely with our theoretical expectations.

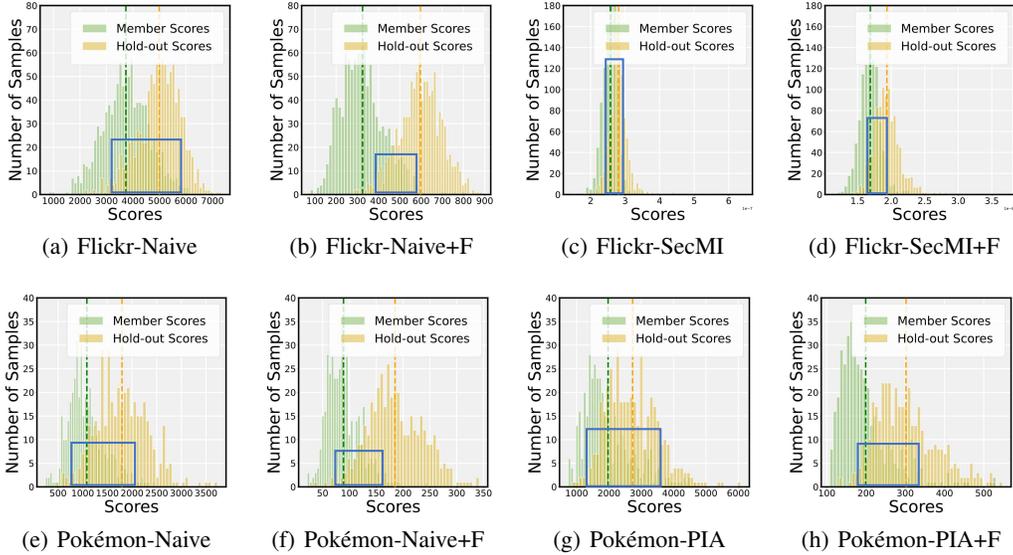


Figure 6: Membership scores distribution for samples from member set and hold-out set.

Table 10: Statistics of σ_H/σ_M before and after applying filter to the baselines.

| Dataset | Metric | Naive | Naive+F | PIA | PIA+F | SecMI | SecMI+F |
|---------|---------------------|--------|---------|--------|--------|--------|---------|
| Pokémon | σ_H/σ_M | 1.3541 | 1.5819 | 1.0453 | 1.1049 | 1.0841 | 1.3591 |
| MS-COCO | σ_H/σ_M | 1.0896 | 1.7722 | 1.0121 | 1.2826 | 1.1969 | 1.6674 |

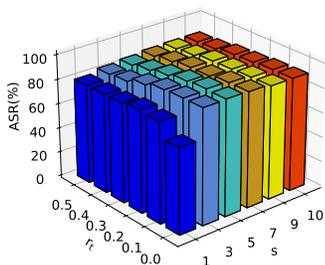
E.5 Ablation Study on DDIM

As shown in Tab. 11, ablation experiments were performed on DDIM using the Tiny-IN dataset, and the experimental phenomenon is similar to the results in fine-tuned stable diffusion. we recommend a value range for r_t of $[3, 10]$ and for s of $[0.0, 0.3]$. Within this range, the filter achieves optimal performance, significantly enhancing the baseline performance while exhibiting low sensitivity to changes in hyperparameters.

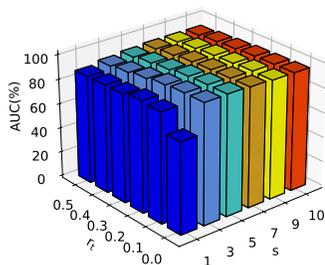
Table 11: Naive’attack performance at different s and r_t in Tiny-IN dataset.

| s/r_t | $s = 0.0$ | | $s = 0.1$ | | $s = 0.2$ | | $s = 0.3$ | | $s = 0.4$ | | $s = 0.5$ | |
|------------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|-----------|-------|
| | ASR | AUC |
| $r_t = 1$ | 70.64 | 77.41 | 80.16 | 87.79 | 78.90 | 86.41 | 78.19 | 85.62 | 77.89 | 85.27 | 77.77 | 85.09 |
| $r_t = 3$ | 83.14 | 90.32 | 83.65 | 90.95 | 83.98 | 91.38 | 83.12 | 90.68 | 81.86 | 89.40 | 80.57 | 88.13 |
| $r_t = 5$ | 85.05 | 92.20 | 85.13 | 92.28 | 85.03 | 92.22 | 84.43 | 91.87 | 83.57 | 91.11 | 82.56 | 90.06 |
| $r_t = 7$ | 84.62 | 91.87 | 84.59 | 91.83 | 84.39 | 91.66 | 83.87 | 91.33 | 83.36 | 90.78 | 82.59 | 90.02 |
| $r_t = 9$ | 82.68 | 90.14 | 82.62 | 90.09 | 82.51 | 89.94 | 82.21 | 89.67 | 81.84 | 89.29 | 81.31 | 88.78 |
| $r_t = 10$ | 81.74 | 89.21 | 81.66 | 89.16 | 81.54 | 89.02 | 81.29 | 88.81 | 80.93 | 88.49 | 80.56 | 88.07 |

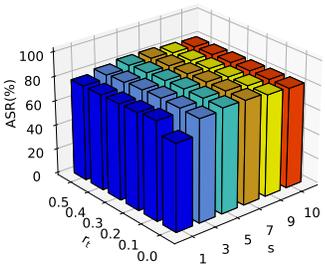
In addition, we visualize ASR and AUC with different parameter settings. As illustrated in Fig. 7 our method demonstrates extreme robustness and is highly insensitive to hyperparameter variations.



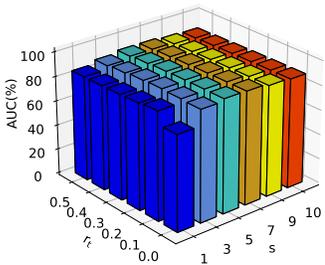
(a) Naive-MS-COCO-ASR



(b) Naive-MS-COCO-AUC



(c) Naive-Tiny-IN-ASR



(d) Naive-Tiny-IN-AUC

Figure 7: The three-dimensional histogram shows the ASR/AUC under different parameter settings. The x-axis represents the parameter r_t , the y-axis represents the parameter s , and the z-axis represents the ASR/AUC.

E.6 Missing Text

Previously, for attacks on stable diffusion, we typically assumed access to the text used for each image during training, which is a strong requirement for attackers. In real-world scenarios, we may not be able to obtain the text, which increases the difficulty of the attack. To simulate more realistic attack conditions, we conducted attacks in scenarios without text and with text generated by BLIP [21]. As shown in Tab. 12, when attacking without text or with the text generated by BLIP, although the performance of the baselines has declined noticeably, the filter still shows satisfactory results. This demonstrates that our method still exhibits good performance under more stringent attack conditions.

Table 12: Attack performance in no-text and BLIP-generated text conditions.

| Method | MS-COCO | | | | Flickr | | | |
|----------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| | No Text | | BLIP Text | | No Text | | BLIP Text | |
| | ASR | AUC | ASR | AUC | ASR | AUC | ASR | AUC |
| Naive | 61.19 | 64.45 | 75.80 | 81.43 | 69.74 | 73.80 | 75.49 | 80.99 |
| Naive+F | 69.80 | 74.56 | 87.72 | 91.51 | 81.75 | 88.23 | 87.75 | 94.77 |
| SecMI | 69.60 | 74.77 | 81.85 | 87.99 | 75.00 | 79.22 | 77.99 | 84.00 |
| SecMI+F | 70.70 | 76.70 | 82.92 | 88.76 | 75.67 | 80.26 | 78.95 | 84.87 |
| PIA | 53.10 | 51.99 | 61.42 | 65.69 | 57.24 | 58.75 | 66.00 | 66.84 |
| PIA+F | 54.70 | 53.18 | 64.94 | 70.81 | 59.61 | 59.25 | 69.00 | 73.29 |
| Avg+ | +3.77 | +4.41 | +5.50 | +5.32 | +5.02 | +5.32 | +5.41 | +7.03 |

E.7 DDIM without data augmentation

As shown in Tab. 13, we tested the DDIM trained without data augmentation. The results indicate that our method still demonstrates significant effects. It is worth noting that on the CIFAR-100 dataset, existing attacks have already achieved outstanding performance in terms of ASR and AUC metrics; therefore, our method shows minor improvements in these two metrics. However, there is an average improvement of 18.58% in the TPR@1% FPR metric.

Table 13: After removing data augmentation from training, the attack performance in DDIM.

| Method | STL10-U | | | CIFAR-100 | | | Tiny-IN | | |
|----------------|--------------|--------------|---------------|--------------|--------------|---------------|--------------|--------------|---------------|
| | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR | ASR | AUC | TPR@1%FPR |
| Naive | 81.82 | 88.75 | 10.92 | 94.02 | 97.98 | 47.65 | 80.00 | 87.61 | 11.75 |
| Naive+F | 85.22 | 91.89 | 13.84 | 96.46 | 99.01 | 66.98 | 86.07 | 93.09 | 21.40 |
| SecMI | 85.34 | 91.95 | 18.20 | 95.70 | 98.93 | 71.19 | 83.77 | 90.96 | 20.63 |
| SecMI+F | 91.93 | 96.82 | 37.13 | 98.83 | 99.85 | 98.16 | 93.56 | 98.04 | 61.24 |
| PIA | 88.88 | 94.98 | 29.63 | 97.54 | 99.54 | 90.48 | 88.06 | 94.49 | 33.88 |
| PIA+F | 93.14 | 97.54 | 45.68 | 99.13 | 99.91 | 99.91 | 94.41 | 98.52 | 72.54 |
| Avg+ | +4.75 | +3.52 | +12.63 | +2.39 | +0.77 | +18.58 | +7.40 | +5.53 | +29.64 |

E.8 ROC curves Visualization

As illustrated in Fig. 8, We visualized the ROC curves for different attacks on MS-COCO and Tiny-IN datasets. The blue curves and green curves represent the baselines before and after applying the high-frequency filter module. We can clearly see the powerful effect of the filter through the ROC curve. At the same time, we visualize the ROC curves for different Naive parameter settings on MS-COCO and Tiny-IN datasets. The results are shown in Fig. 9, which show that our method provides a significant improvement over the baselines at different parameter settings.

E.9 Compute Overhead

In this section, we assess the time cost of the filter. As shown in Tab. 14, we count the time spent on attacking all samples in the dataset, and the additional time overhead is approximately negligible by averaging on a single sample. Therefore, the experimental results show that our method hardly brings any additional time cost.

Table 14: We calculate the runtime of the attacks on the CIFAR-100 and Flickr datasets.

| Model | Dataset | Naive | Naive+F | PIA | PIA+F | SecMI | SecMI+F |
|------------------|-----------|-------|---------|------|-------|-------|---------|
| DDIM | CIFAR-100 | 75s | 80s | 145s | 153s | 1470s | 1480s |
| Stable Diffusion | Flickr | 482s | 506s | 752s | 782s | 3474s | 3492s |

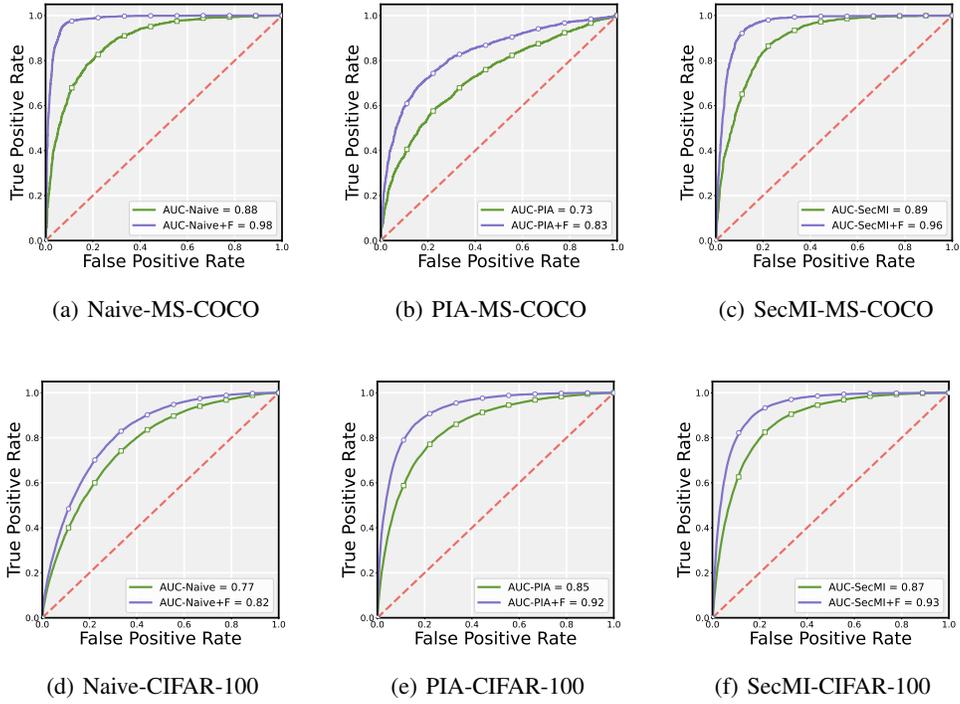


Figure 8: ROC curves before and after applying the high-frequency filter for the baselines.

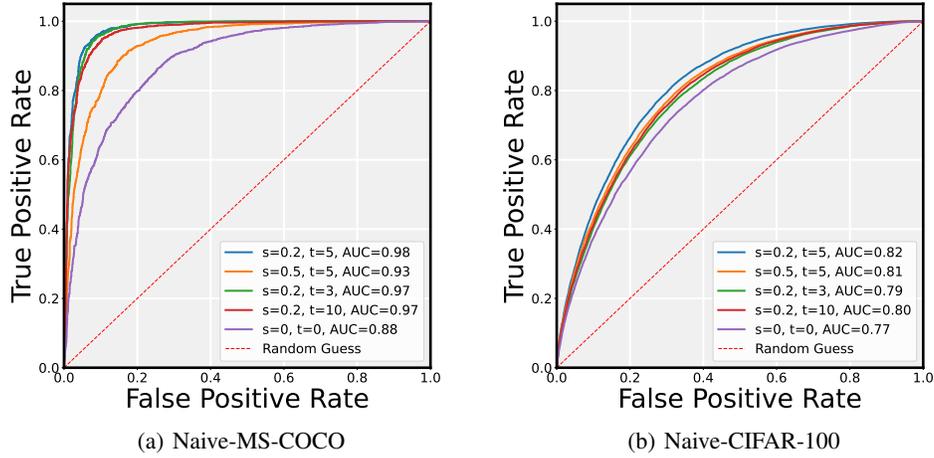


Figure 9: ROC curves of Naive with different parameter settings.

F Ethics Statements

This study proposes a generalized membership inference attacks improvement algorithm aimed at enhancing the ability to infer whether specific samples were used during the training of diffusion models. Membership inference attacks have significant applications in unauthorized data usage audits and are one of the key means of maintaining copyright. Our method is expected to advance the development of copyright protection and model privacy research in the field of image generation. However, we also recognize that this method may pose privacy risks to existing diffusion models. To prevent the misuse of our research, all experiments in this study are conducted based on publicly available datasets and open-source model architectures. Additionally, the code for this research will be released to the public.