

# Score-based Membership Inference on Diffusion Models

Anonymous authors

Paper under double-blind review

## Abstract

Membership inference attacks (MIAs) against Diffusion Models (DMs) raise pressing privacy concerns by revealing whether a sample was part of the training set. While existing methods typically rely on measuring reconstruction error across multiple denoising steps as a test statistic, they often incur significant computational overhead. In this work, we present a simple yet successful attack statistic using only the predicted noise vectors from the DM’s denoiser, or equivalently, the score. Specifically, we show that the expected denoiser output points toward a kernel-weighted local mean of nearby training samples, such that its norm encodes proximity to the training set and thereby reveals membership. Building on this observation, we propose SimA, a single-query attack that provides a principled, efficient alternative to existing multi-query methods. SimA consistently achieves superior performance across variants of DMs and the Latent Diffusion Models (LDMs) on eight different datasets. Its Monte Carlo variant (SimA-MC) exhibits state-of-the-art performance across all experiments, significantly outperforming baseline methods in terms of TPR@1%FPR. These results demonstrate that complex reconstruction trajectories are unnecessary for effective membership inference, establishing SimA as a highly efficient benchmark for auditing privacy in DMs and LDMs.

## 1 Introduction

Generative image models leave evidence of their specific training data at deployment time in their generative process. While making draws from approximations of  $p(x)$  or  $p(x|y)$ , they leave biases of the training samples (finite and fixed realizations from the real  $p(x)$  or  $p(x|y)$ ). These biases may be used in theory to reconstruct the training data, a process known as model inversion (Zhu et al., 2016; Creswell & Bharath, 2018; Carlini et al., 2023; Somepalli et al., 2023a;b; Gu et al., 2023).

The ability to invert these models raises concerns in privacy and intellectual property spaces for specific use-cases of generative models, but also possibly provides unique perspectives into the idiosyncrasies of the generative models themselves. If the models were perfect, they would sample from a distribution indistinguishable from the data generating process; the ways in which they deviate from this distribution inform upon their structure.

A critical precursor to model inversion is the *membership inference attack* (MIA), which determines whether a given image was included in the training set. MIA effectively constructs a classifier for identifying training examples, setting aside the problem of searching the domain for high-likelihood examples. While this may be the easier sub-problem of full model inversion, it is no less important, as without successful MIA, model inversion is impossible.

Recent MIA statistics on diffusion models Duan et al. (2023); Zhai et al. (2024); Kong et al. (2023); Matsumoto et al. (2023); Carlini et al. (2023) rely on either measuring reconstruction error along the Stochastic Differential Equation (SDE) trajectory or likelihood estimation Fu et al. (2023). We argue that it incurs an unnecessary computational overhead. We demonstrate that using solely the output of DM’s denoiser  $\hat{\epsilon}_\theta$  is sufficient to match or even surpass current state-of-the-art (SoTA) attack performance. Our method is based on a simple observation that the empirical distribution of an over-trained model will have peaks at each training point, and its gradient vanishes at the smoothed density (Figure 1), which exposes the membership

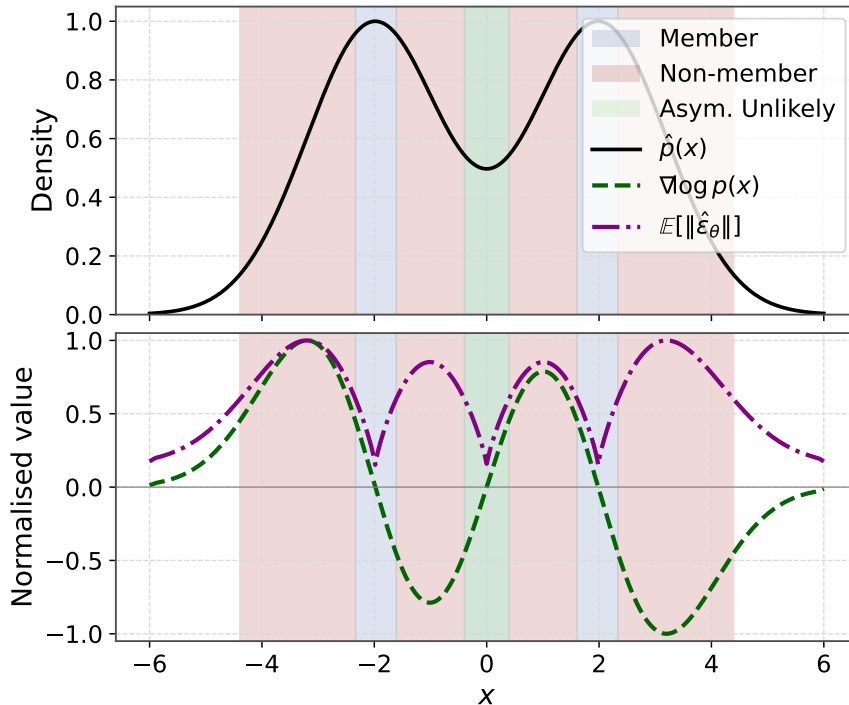


Figure 1: A diagram of our Membership Inference method in one dimension. In blue are regions of high membership likelihood, corresponding to low  $\|\hat{\epsilon}_\theta\|$ , plotted in purple. The green region is unlikely to be sampled in high dimensions (c.f. Sec. 3).

signal. Unlike other generative models, this observation is particularly pertinent to diffusion-based models, where the model output  $\hat{\epsilon}_\theta$  explicitly reveals the score function ( $\hat{\epsilon}_\theta \approx -(1/\sigma_t)\nabla_{x_t} \log p_t(x_t)$ ) by design Song et al. (2020); Ho et al. (2020). Hence, our attack statistic is simply the norm of the estimated score at the test point  $x$  across diffusion times. More intuitions and a theoretical justification are described in Section 3.

Our empirical results show this simple method (**SimA**) has AUC scores at or above the previous SoTA for DDPM-weightsets Ho et al. (2020) on common smaller datasets, and well above the previous SoTA for the related Guided Diffusion model Dhariwal & Nichol (2021) and vanilla LDM Rombach et al. (2022) on ImageNet-1k. For large-scale LDMs such as Stable Diffusion, it achieves comparable results to the SoTA. To enhance stability, we further propose its Monte Carlo extension (**SimA-MC**). While involving a higher computational cost due to additional model queries, SimA-MC further advances upon the current SoTA across all experiments, notably boosting the TPR@1%FPR.

In summary, our contributions are the following:

1. A detailed derivation and theoretical justification of a simple score-based Membership Inference Attack method (SimA and SimA-MC), which is a reduction of other methods into a general framework.
2. An empirical demonstration that SimA and SimA-MC provide top performance on standard datasets and independently trained base models.

All data splits, model checkpoints, training/fine-tuning scripts, and testing code are released on our GitHub repository ([withheld for review](#)).

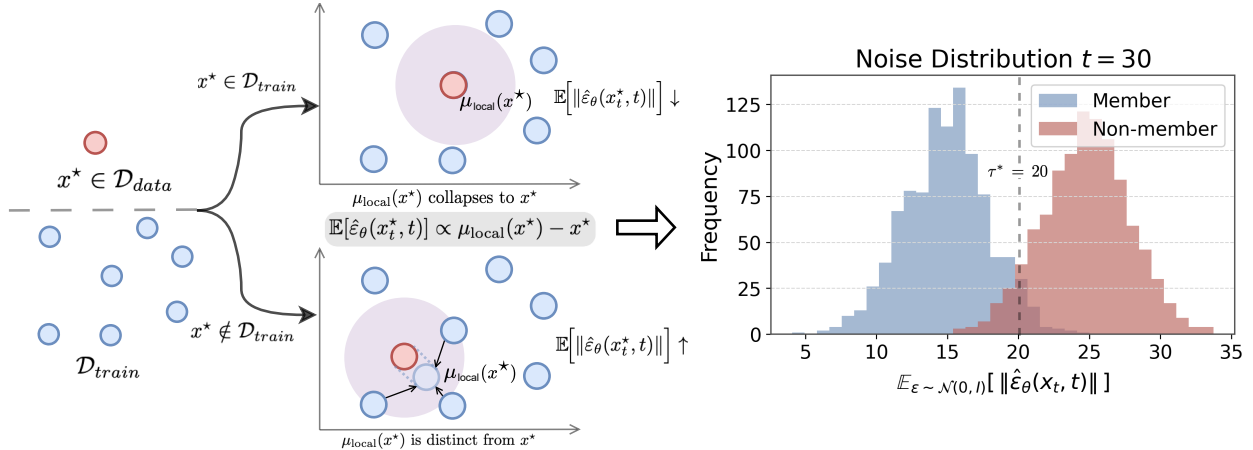


Figure 2: **Score-based MIA intuition with local-mean geometry.** In a small neighborhood (“local ball”) around a query  $x^*$ , let  $\mu_{\text{local}}(x^*)$  be the kernel-weighted mean of nearby training samples. The model’s predicted noise (score) points from  $x^*$  toward this local mean,  $\mathbb{E}[\hat{\epsilon}_\theta(x_t^*, t)] \propto \mu_{\text{local}}(x^*) - x^*$ . For members ( $x^* \in \mathcal{D}_{\text{train}}$ ), the local mean  $\mu_{\text{local}}(x^*)$  collapses to training sample  $x^*$ , producing small norms, whereas for non-members ( $x^* \notin \mathcal{D}_{\text{train}}$ ),  $\mu_{\text{local}}(x^*)$  deviates from  $x^*$ , yielding larger norms. Right: the histogram at  $t = 30$  shows the separation in  $\mathbb{E}_{\epsilon \sim \mathcal{N}(0, I)}[\|\hat{\epsilon}_\theta(x_t, t)\|]$ .

## 2 Background and Related Work

**Diffusion Models:** Our membership inference work is specific to diffusion-based generative image models. Originally introduced as score-based generative models (without the explicit connection to the Diffusion model) in Song & Ermon (2019), a very large number of publications have explored variations of these models since that point (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2020; Nichol & Dhariwal, 2021; Dhariwal & Nichol, 2021; Rombach et al., 2022).

While each contribution has its own particular training paradigm and architecture, our attack applies to the broad class of models that estimate a gradient flow field  $\hat{\epsilon}(x, t)$  at points  $x$  for a smoothing parameter/diffusion time  $t$  that approximates the gradient of the smoothed log-likelihood,  $\nabla \log(p(x) * \mathcal{K}(t))$  (Kamb & Ganguli (2024), or Appendix A.4), which is often induced by a conceptual and/or training-time “forward” noise process  $x_t = \sqrt{\alpha_t}x_0 + \sigma_t\epsilon$  and a “backward” de-noising process similar to denoising auto-encoder processes (Alain & Bengio, 2014). These may be variance-preserving or variance-exploding (Song et al., 2020), based on the exact parameterization of the noise schedule. In this work we directly use weights from the following models: DDPM (Ho et al., 2020), Guided Diffusion (Dhariwal & Nichol, 2021), Latent Diffusion Model (Esser et al., 2021), and Stable Diffusion (Rombach et al., 2022). Each of these directly estimate the noise vector  $\epsilon$  using a neural network  $\hat{\epsilon}(x, t)$ . Our method manipulates  $x$  and  $t$  and analyze the networks’ outputs, but otherwise is agnostic to the exact architecture and weights.

Latent Diffusion Models (Esser et al., 2021; Rombach et al., 2022) perform their forward process on the latent space of some encoder-decoder structure (usually a Variational Auto-encoder (Kingma et al., 2013)). Rao et al. (2025) has shown, this encoder-decoder structure appears to be closely related to membership inference.

The analytic tractability of these models permits theory about their behavior (Kamb & Ganguli, 2024; Lukoianov et al., 2025), even pre-dating the publication of the DDPM, e.g., Alain & Bengio (2014). Our method is conceptually linked to results and intuition from Alain & Bengio (2014) and Kamb & Ganguli (2024).

**Membership Inference Attack threat models:** Model inversion and membership inference attacks pre-date the introduction of the DDPM and LDM generative image models; MIA was defined originally for general classification tasks (Shokri et al., 2017). A non-trivial amount of literature since that point has

focused on MIA for generative image models (Chen et al., 2020) starting with GANs, due to its attack surface (pixel arrays) and its clear privacy and intellectual property implications.

Much of the terminology and structure were defined in the security context, where a threat model defines the scope and allowable resources for an attack vector. So-called “black box” attacks from the original MIA literature are performed without knowledge of the model weights or structure, and can only rely on input and output pairs from a deployed model. In contrast, “white box” attacks (Pang et al., 2023) have access to the full architecture/weight set. We consider the most common class of diffusion model MIAs, “grey-box” attacks (Duan et al., 2023; Zhai et al., 2024; Kong et al., 2023; Matsumoto et al., 2023; Carlini et al., 2023; Fu et al., 2023), which span a range of options between those two extrema; in general they have access to weights and/or internal representations, and may query the model for particular test points. We review each of these grey box attacks in detail in Section 3 and compare to each, with the exception of Zhai et al. (2024) as it performs membership inference on conditional generative models instead of the unconditioned case.

### 3 Methodology

#### Method Overview

The predicted noise  $\hat{\varepsilon}_\theta$  is outputted by the neural network, which is a scaled estimator of the score  $-\sigma_t \nabla_{x_t} \log p_t(x_t)$  (Song et al., 2020; Ho et al., 2020; Luo, 2022), for data generating distribution  $p(x)$  and its mollifications  $p_t(x)$  on a noising schedule  $\sigma_t$ . Our Simple Attack (**SimA**) method is

$$\mathcal{A}(x, t) = \|\hat{\varepsilon}_\theta(x, t)\|. \quad (1)$$

Here,  $x$  is the test point, which ostensibly was drawn from  $p(x)$ , but is either in the training data or not, and  $t$  is the diffusion time parameter. We provide a more rigorous derivation and its connections to other MIA methods in the following sections, but we feel that its intuition is also instructive on why such a simple method would work.

Figure 2 visualizes the intuition of our method: the expected norm of the predicted score for a query image  $x^*$  at time  $t$  is effectively the gradient of a Gaussian kernel density estimator (see Appendix D). For well-separated points, these estimators’ implicit distributions will have peaks at each training point; the gradient vanishes at critical points of the smoothed density, which include the peaks corresponding to the original data points (the blue region in the Fig. 1), up to a bias term from the smoothing. Manipulating this fact allows us to form a simple yet successful estimator.

There should be false positive terms at the other critical points (the green region in Fig 1). In high dimension, these should be by-in-large saddle points between maxima. Because these points occur directly between original datapoints at their arithmetic mean, if the data manifold (the support of  $p(x)$ ) has any curvature, these points would be off manifold; empirically we do not seem to encounter many of them, as indicated by the TPR@1%FPR measurements (see Section 4).

#### 3.1 From Forward Diffusion to Score

**Notation:** Let  $\{\beta_t\}_{t=1}^T$  be the variance schedule of the forward diffusion process in DDPM Ho et al. (2020). Define  $\alpha_t := 1 - \beta_t$  and its cumulative product  $\bar{\alpha}_t := \prod_{s=1}^t \alpha_s$ . The total noise variance accumulated up to step  $t$  is  $\sigma_t^2 := 1 - \bar{\alpha}_t$ .

**Forward Diffusion as a Scaled Gaussian Convolution:** With the variance-preserving (VP) schedule of Ho et al. (2020), the forward model is

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sigma_t \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I), \quad \sigma_t^2 = 1 - \bar{\alpha}_t, \quad (2)$$

where  $x_0 \sim p_{\text{data}}$ . Marginalizing  $x_0$  gives the perturbed data distribution (Song & Ermon, 2019). For a *clean* (without noise) sample  $x \in \mathbb{R}^d$ ,

$$p_t(x) = \int_{\mathbb{R}^d} p_{\text{data}}(x_0) \mathcal{N}(x \mid \sqrt{\bar{\alpha}_t} x_0, \sigma_t^2 I) dx_0. \quad (3)$$

Equation 3 is a Gaussian convolution of the original data distribution (see Appendix D for an explicit derivation), followed by a global scaling by  $\sqrt{\bar{\alpha}_t}$ .

$$p_t(x) = \frac{1}{\bar{\alpha}_t^{d/2}} \left( p_{\text{data}} * \mathcal{N}\left(0, \frac{\sigma_t^2}{\bar{\alpha}_t} I\right) \right) \left( \frac{x}{\sqrt{\bar{\alpha}_t}} \right). \quad (4)$$

Therefore,  $p_t(x)$  is the data distribution convolved with scaled Gaussian distribution whose kernel’s covariance  $\frac{\sigma_t^2}{\bar{\alpha}_t} I = (\bar{\alpha}_t^{-1} - 1)I$  grows monotonically with the timestep  $t$ .

**Gradient of  $\mathbf{p}_t(\mathbf{x})$ :** Writing the kernel in standard form  $\mathcal{K}_t(x, x_0) = (2\pi\sigma_t^2)^{-d/2} \exp(-\|x - \sqrt{\bar{\alpha}_t}x_0\|^2/2\sigma_t^2)$ , we can compute its spatial gradient:  $\nabla_x \mathcal{K}_t = -\sigma_t^{-2}(x - \sqrt{\bar{\alpha}_t}x_0)\mathcal{K}_t$ . We then combine this with Eq. 3 to obtain

$$\nabla_x p_t(x) = -\sigma_t^{-2} \int p_{\text{data}}(x_0)(x - \sqrt{\bar{\alpha}_t}x_0)\mathcal{K}_t(x, x_0) dx_0 \quad (5)$$

which requires continuity assumptions on  $p_{\text{data}}$  which are usually assumed by DDPM analyses (Alain & Bengio, 2014).

**Introducing exact likelihood of each datapoint  $q_t(x_0 | x)$ :** Define the exact distribution of the  $x_0$  given an observation from the Gaussian smoothed distribution:

$$q_t(x_0 | x) = \frac{p_{\text{data}}(x_0)\mathcal{K}_t(x, x_0)}{p_t(x)}. \quad (6)$$

Because  $p_t(x)$  normalizes Eq. 3, we can then rewrite Eq. 5 as

$$\nabla_x p_t(x) = -\frac{p_t(x)}{\sigma_t^2} \left[ x - \sqrt{\bar{\alpha}_t} \underbrace{\mathbb{E}_{q_t(x_0|x)}[x_0|x]}_{\mu_t(x)} \right]. \quad (7)$$

We call  $\mathbb{E}_{q_t(x_0|x)}[x_0|x] = \mu_t(x)$  the *denoising mean*; it is the likelihood-weighted average of the positions of the datapoints that could have generated  $x$  at time  $t$  through the forward process, which is the same as the mean optimal solution to the denoising problem.

**Obtaining the exact score:** Dividing Eq. equation 7 by  $p_t(x)$  yields the score of distribution  $p_t(x)$  at  $x$  (Lemma 6 of Pidstrigach (2022), or, alternatively, applying the chain rule to  $\nabla_x \log p_t(x)$  and then substituting in values):

$$s_t(x) = \nabla_x \log p_t(x) = -\frac{x - \sqrt{\bar{\alpha}_t} \mu_t(x)}{\sigma_t^2}. \quad (8)$$

This score function  $s_t(x)$  is the desired output of the  $\varepsilon$ -parameterization of Score-based Denoising (Song et al., 2020) and the original DDPM (Ho et al., 2020). During training the UNet is asked to predict the standard noise (Ho et al., 2020):

$$\hat{\varepsilon}_\theta(x, t) \approx \frac{x_t - \sqrt{\bar{\alpha}_t} \mu_t(x)}{\sigma_t} = -\sigma_t \nabla_x \log p_t(x) \quad (9)$$

This is consistent with Eq. 151 of Luo (2022). For  $x \in \text{supp } p_t$  (with or without noise), the estimator  $\hat{\varepsilon}_\theta(x, t)$  should approximate the negative score of  $p_t(x)$ . However, in practice the data distribution  $p_{\text{data}}$  becomes the empirical distribution  $p_{\text{training}}$ , which is a finite sample of points. The noised distribution  $p_t(x)$  is then a kernel smoothing of that empirical distribution, and its finite-sample denoising mean is described in Kamb & Ganguli (2024) and refined in Lukoianov et al. (2025). Equation 3 of Lukoianov et al. (2025) states it as:

$$\mu_t^{\text{finite}}(x) = \sum_{i=1}^N w_i(x, t) x_0^{(i)}, \quad w_i(x, t) = \text{softmax}_i \left\{ -\frac{1}{2\sigma_t^2} \|x - \sqrt{\bar{\alpha}_t} x_0^{(j)}\|_2^2 \right\}_{j \in [N]} \quad (10)$$

where the  $x_0^{(i)}$  are training data, and  $\text{softmax}_i$  is the  $i^{\text{th}}$  index of a softmax function over the  $N$  training data points. The discrepancy between the finite sample optimal denoised  $x$  distribution and the large sample limit  $q_t(x_0|x)$  gives rise to our membership inference attack; the model “overfits” to the training set, and that overfitting gap is the discrepancy which **SimA** seeks to exploit.

### 3.2 Membership Inference Attack

Given a datapoint  $x \in \mathbb{R}^d$  and a  $t = 1, \dots, T$ , our membership decision criterion  $\mathcal{A}$  is defined as

$$\mathcal{A}(x, t) = \|\hat{\varepsilon}_\theta(x, t)\|_p. \quad (11)$$

Using  $\ell_p$  norms other than  $p = 2$  provide slightly improved performance. This trend is also found in another MIA method, Kong et al. (2023). While this is somewhat mysterious, the  $\ell_4$  norm appears in sum-of-squares computations (Barak et al., 2015), spherical harmonics (Stanton & Weinstein, 1981), and blind source separation (Hyvarinen, 1997) with surprising regularity.

Following the Bayes-optimal loss-threshold formulation of membership inference in classification models by Sablayrolles et al. (2019), we recast the decision rule for diffusion models. Specifically, we define

$$\mathcal{M}_{\text{opt}}(x, t) = \mathbb{1}[\mathcal{A}(x, t) \leq \tau], \quad (12)$$

where  $\tau$  is a threshold calibrated on a held-out validation set and  $\mathcal{A}(x, t) = \|\hat{\varepsilon}_\theta(x, t)\|_2$  is the estimated noise at  $x$  for diffusion step  $t$ . If the predicted noise norm is *smaller* than  $\tau$ , the sample is inferred to be a *member*; otherwise, it is classified as a *non-member*.

**Monte Carlo variant of SimA:** Notably, SimA is a scaled point estimate ( $\varepsilon = 0$ ) of our intuition in Figure 2. Although it is analytically derived in Section 3.1, point estimate of this nature is inherently numerically unstable. Given that both the score function and its estimator are smooth, we can make it more stable and overall better by local averaging. This leads to the MC solution. Formally, given a test image  $x$  and  $N$  Monte Carlo (MC) samples  $\varepsilon_n \sim \mathcal{N}(0, I)$ , the Monte Carlo Variant of SimA (**SimA-MC**) is defined as

$$\mathcal{A}_{\text{mc}}(x, t) = \frac{1}{N} \sum_n \|\hat{\varepsilon}_\theta(x_t, t)\| \quad x_t = \sqrt{\alpha_t} x_0 + \sigma_t \varepsilon_n. \quad (13)$$

We originally thought that such a high-dimensional Monte Carlo would perform poorly until a very large number of samples. However, this method turns out to be surprisingly tractable. The SimA-MC method provides SoTA performance within 30 samples, often within 10. For sufficient ( $n \in [10, 30]$ ) samples, it improves all performance metrics on all datasets of our experiments. The results is summarized in lower block of all tables in Section 4.

### 3.3 Theoretical Justification on Three Cases

The attack criterion  $\mathcal{A}$  can be applied to three cases, which we expand upon below (see appendix E for detailed derivation of the first two cases).

**Case 1 — Member of the Training Set:** Let  $x^{(k)}$  denote one of the training images  $\{x^{(i)}\}_{i=1}^N$ . As  $t \rightarrow 0$ , the finite sample denoising mean collapses to the input (full derivation is in case 1 of Appendix E):

$$\mu_t^{\text{finite}}(x^{(k)}) \xrightarrow[t \rightarrow 0]{} x^{(k)} \quad (14)$$

Consequently, the estimated noise vector shrinks to zero as well, meaning our criterion  $\mathcal{A}(x^{(k)}, t \rightarrow 0)$  should be small:

$$\hat{\varepsilon}_\theta(x^{(k)}, t) = \frac{x^{(k)} - \sqrt{\alpha_t} x^{(k)}}{\sigma_t} \sim \frac{\sigma_t}{2} x^{(k)} \xrightarrow[t \rightarrow 0]{} 0. \quad (15)$$

While the actual value at zero is undefined, and values for small  $t < 10$  are empirically unstable for non-member  $x^{(k)}$  (leading to a poor estimator, see Appendix F for the detailed reason), we find that for  $t \in [10, 300]$ , these values will still be smaller than the case 2. These time frames are unfortunately noise schedule and data dependent.

**Case 2 — Held-out but On-Manifold:** Here  $x^\dagger$  is sampled from the same data distribution  $p_{\text{data}}$  as the training set, yet was *never* shown during training. We can consider only the local means (i.e.  $\mu_t^{\text{finite}}(x^\dagger) \approx \mu_t^{\text{local}}(x^\dagger)$ ) because we already established that the diffusion model acts as a KDE (Eq. 10 in Section 3.1). This means that many points will be “far” from our chosen point, and thus have exponentially decreasing influence. The only high influence points will be “close” points, which is definitionally the local neighborhood.

We use the notation of local moment matching (Bengio et al., 2012; Alain & Bengio, 2014) to describe these points. Consider the local mean defined with respect to that  $B_r(x^\dagger)$ :

$$\mu_t^{\text{local}}(x^\dagger) = m_r(x^\dagger) = \int_{B_r(x^\dagger)} xp_t(x|x^\dagger)dx \quad (16)$$

where  $r \asymp \sigma_t/\sqrt{\bar{\alpha}_t}$ . Theorems 2 and 3 of Alain & Bengio (2014) put together produce a statement about a term that is equivalent to  $\hat{\varepsilon}$  (Eq. 28 of Alain & Bengio (2014)) (Full derivation is in case 2 of Appendix E):

$$m_r(x^\dagger) - x^\dagger \approx \frac{r^2}{d+2} \left. \frac{\partial \log p_t(x)}{\partial x} \right|_{x^\dagger} = \frac{r^2}{d+2} \left( \frac{x^\dagger - \sqrt{\bar{\alpha}_t} \mu_t(x^\dagger)}{\sigma_t^2} \right) = -\frac{r^2}{\sigma_t(d+2)} \hat{\varepsilon}_\theta(x^\dagger, t) \quad (17)$$

We claim that for in-support regions of Gaussian mollified empirical distributions with well separated points, these regions will generally not be flat, and thus  $\|\hat{\varepsilon}_\theta(x^\dagger, t)\|$  will tend away from zero. The magnitude to which  $\|\hat{\varepsilon}_\theta(x^\dagger, t)\|$  diverges from zero clearly depends on the maximal density of the dataset, but as high dimensional spaces have exponentially larger volumes than lower dimensional spaces, even for large datasets (e.g. ImageNet in ResNet resolution) we can expect these voids to be non-trivially large.

**Case 3 — Out-of-Distribution (OOD).** Since no training data support is available in out-of-distribution (OOD) regions, the diffusion model lacks information about these areas. Consequently, the learned score field in such regions is necessarily an extrapolation, and the theoretical derivations established under the in-distribution assumption no longer hold. We do not expect either the diffusion model or our attack criterion to perform well in these regions. First, even though the theoretical finite-sample optimal denoiser is well defined (Eq. 10), a neural network approximation to it will have very little training data in these regions. Second, a trial datapoint  $x^*$  is by-definition not from  $p_{\text{data}}$  in these regions.

### 3.4 SimA in Comparison to Other Diffusion MIA Models

A standard concept in MIA is the use of the training loss function evaluated on the data points in question as the member/non-member decision criterion. This is the Loss criterion, and is proposed in Matsumoto et al. (2023). They use a stochastic sample  $\varepsilon$  to estimate this criterion, adding it to the test point  $x^*$ .

$$\text{Loss} = \|\varepsilon - \hat{\varepsilon}_\theta(\sqrt{\bar{\alpha}_t}x^* + \sqrt{1 - \bar{\alpha}_t}\varepsilon, t)\| \quad (\text{Matsumoto et al., 2023}) \quad (18)$$

In theory this should be evaluated across a large number of  $\varepsilon$  measurements, but for each point they choose to use only one. This method is predicated on the idea that for points in the training set, the noise estimation will be better or even overfit in comparison to points not in the training set. SimA is mathematically the scaled evaluation of this method at  $\varepsilon = 0$ , which is the mean and mode of the  $\varepsilon$  distribution, but inherently very different and performs much better. Effectively, Matsumoto et al. (2023) are measuring draws from lower likelihood areas, which may not exhibit the overfit phenomenon as well as  $\varepsilon = 0$ . Fu et al. (2023) provides this same loss estimate as the selection criterion, but increases the Monte Carlo sampling to 20 and performs it only for a single step, sampling  $\varepsilon_t$  the stepwise noise instead of sampling  $\varepsilon$ .

SecMI of Duan et al. (2023) takes this one step further, evaluating not only a score term but also a single step term, which measures sensitivity to single step differences in  $t$ :

$$\text{SecMI} = \|\sqrt{1 - \bar{\alpha}}(\hat{\varepsilon}_\theta(x^*, t) - \hat{\varepsilon}_\theta(\sqrt{\bar{\alpha}_{t+1}}x^* + \sqrt{1 - \bar{\alpha}_{t+1}}\varepsilon, t + 1))\| \quad (19)$$

SecMI is dependent on sampling  $\varepsilon$ 's as well; the authors prescribe using  $N = 12$  samples.

Table 1: Performance of benchmark methods on **DDPM** across four datasets. In the upper block, the **bold** values indicate the best results (baselines and SimA). In the lower block, the **bold** values indicate the new SoTA achieved by SimA-MC compared to the whole upper block.

Method	#Query↓	CIFAR-10 (%)			CIFAR-100 (%)			STL10-U (%)			CelebA (%)		
		ASR↑	AUC↑	TPR↑†	ASR↑	AUC↑	TPR↑†	ASR↑	AUC↑	TPR↑†	ASR↑	AUC↑	TPR↑†
PIA	2	<b>85.06</b>	<b>91.86</b>	29.54	82.97	<b>90.27</b>	29.77	79.25	86.59	23.41	83.65	91.53	41.21
PFAMI <sub>met</sub>	20	73.64	80.32	8.16	70.84	77.45	8.26	64.7	71.05	5.73	71.75	79.62	16.55
SecMI <sub>stat</sub>	12	82.71	89.72	33.44	81.41	88.7	33.52	78.08	85.25	22.02	83.62	91.27	32.00
Loss	1	77.67	84.73	24.14	75.42	82.69	20.35	72.83	79.45	17.21	79.16	87.29	32.38
SimA (★)	1	83.62	90.45	<b>35.86</b>	<b>83.12</b>	90.05	<b>37.77</b>	<b>79.47</b>	<b>86.72</b>	<b>29.1</b>	<b>83.97</b>	<b>91.81</b>	<b>42.39</b>
SimA-MC (#mc=10)	10	84.16	91.23	<b>43.58</b>	82.79	90.2	<b>43.14</b>	<b>79.42</b>	<b>86.86</b>	<b>33.84</b>	<b>86.9</b>	<b>94.13</b>	<b>59.23</b>
SimA-MC (#mc=30)	30	<b>85.21</b>	<b>92.16</b>	<b>44.74</b>	<b>83.65</b>	<b>91.01</b>	<b>44.66</b>	<b>80.55</b>	<b>87.97</b>	<b>37.19</b>	<b>88.17</b>	<b>95.04</b>	<b>60.00</b>

† True Positive Rate at 1% False Positive Rate.

Table 2: Attack performance of four baseline methods used to evaluate memorization of **LDMs**. In the upper block, the **bold** values indicate the best results (baselines and SimA). In the lower block, the **bold** values indicate the new SoTA achieved by SimA-MC compared to the whole upper block.

Method	#Query↓	CIFAR-10 (%)			CelebA (%)			ImageNet-1K (%)		
		AUC↑	ASR↑	TPR@1%FPR↑	AUC↑	ASR↑	TPR@1%FPR↑	AUC↑	ASR↑	TPR@1%FPR↑
PIA	2	85.73	78.19	15.90	83.36	75.87	9.41	66.09	62.52	2.95
PFAMI <sub>mat</sub>	20	74.71	69.86	7.81	75.50	68.31	7.98	67.44	63.05	3.74
SecMI <sub>stat</sub>	12	87.42	80.12	19.11	83.09	75.85	10.53	68.21	63.44	3.71
Loss	1	73.28	67.48	7.86	68.65	63.47	6.39	67.49	63.16	<b>4.09</b>
SimA (★)	1	<b>89.10</b>	<b>81.63</b>	<b>19.88</b>	<b>84.66</b>	<b>77.04</b>	<b>11.09</b>	<b>69.62</b>	<b>64.92</b>	3.87
SimA-MC (#mc=10)	10	<b>90.68</b>	<b>82.71</b>	<b>39.86</b>	<b>91.55</b>	<b>83.56</b>	<b>39.76</b>	<b>71.13</b>	<b>65.95</b>	<b>7.44</b>

The closely related PIA method computes this same loss quantity again, but using a  $t = 0$  term instead of the  $\varepsilon$  sample.

$$\text{PIA} = \|\hat{\varepsilon}_\theta(x^*, t = 0) - \hat{\varepsilon}_\theta(\sqrt{\alpha_t}x^* + \sqrt{1 - \alpha_t}\hat{\varepsilon}_\theta(x^*, t = 0), t)\| \quad (20)$$

While in theory diffusion models might not be well defined at  $t = 0$ , in practice they often can extrapolate as they are trained on nearby  $t$ ; in the continuous time case they are trained on the interval  $[0, 1]$ . Our method has similar components to this method, manipulating  $t$  around the test point, but again replacing its “ground-truth” noise with  $\varepsilon = 0$  (here replaced by  $\hat{\varepsilon}_\theta(x^*, t = 0)$ ).

**Remarkably, unlike the three aforementioned attack statistics that rely on reconstruction along the SDE trajectory, SimA uniquely leverages the behavior of the score function based the observation in Figure 1.**

Table 3: Performance of public checkpoint of **Guided Diffusion** pretrained on ImageNet1K. In the upper block, the **bold** values indicate the best results (baselines and SimA). *Member* set: ImageNet-1K (train split); *Held-out*: ImageNetV2 (train split).

Method	#Query↓	ImageNet		
		ASR↑	AUC↑	TPR@1%FPR↑
PIA	2	64.65	66.44	9.93
PFAMI <sub>Met</sub>	20	67.85	72.22	3.77
SecMI <sub>stat</sub>	12	77.97	82.55	<b>34.73</b>
Loss	1	57.27	60.38	7.00
SimA (★)	1	<b>85.73</b>	<b>89.77</b>	21.73
SimA-MC (#mc=10)	10	81.13	85.86	10.73
SimA-MC (#mc=30)	30	83.28	87.33	17.63

Table 4: Attack performance of four baseline methods used to evaluate memorization of **Stable Diffusion**. In the upper block, the **bold** values indicate the best results (baselines and SimA). In the lower block, the **bold** values indicate the new SoTA achieved by SimA-MC compared to the whole upper block.

Method	#Query↓	Pokémon (%)				MS-COCO (%)			Flickr30K (%)		
		AUC↑	ASR↑	TPR@1%FPR↑	FPR↑	AUC↑	ASR↑	TPR@1%FPR↑	AUC↑	ASR↑	TPR@1%FPR↑
PIA	2	<b>94.93</b>	<b>90.52</b>	32.85		92.51	85.72	24.76	68.88	64.61	2.67
PFAMI <sub>met</sub>	20	50.0	23.18	0.0		86.06	78.44	12.38	64.81	60.87	2.27
SecMI <sub>stat</sub>	12	88.12	81.04	29.02		91.35	84.00	<b>39.16</b>	<b>71.81</b>	<b>66.54</b>	<b>5.29</b>
Loss	1	92.03	85.47	<b>40.77</b>		83.26	75.80	13.48	63.13	59.74	2.35
SimA (*)	1	93.50	87.87	20.38		<b>93.71</b>	<b>87.34</b>	29.80	70.04	65.96	2.59
SimA (#mc=10)	10	<b>96.75</b>	<b>91.47</b>	<b>63.79</b>		<b>93.21</b>	<b>85.76</b>	<b>45.36</b>	70.26	65.65	4.47
SimA (#mc=30)	30	<b>97.01</b>	<b>92.31</b>	<b>70.50</b>		<b>94.24</b>	<b>86.66</b>	<b>52.48</b>	<b>72.23</b>	<b>66.85</b>	<b>6.33</b>

## 4 Experiments

### 4.1 Environment

All experiments were conducted on a dedicated computing node running the Ubuntu 22.04.4 LTS operating system. The hardware infrastructure is equipped with an AMD EPYC 7453 28-Core Processor, paired with 256 GB of system RAM. Hardware acceleration for model training and evaluation was provided by eight NVIDIA A40 GPUs. The software environment was built using Python 3.11.3, with all deep learning workflows and tensor computations implemented via PyTorch version 2.10.0, utilizing CUDA toolkit 12.8. More software library details are provided on our online resource in the Appendix ??.

### 4.2 Setup

We evaluated our attack on **15 member-held-out pairs** drawn from **11 datasets** (see Appendix B). The experiments were conducted on the following target models:

**Denosing Diffusion Probabilistic Model:** For CIFAR-10, CIFAR-100 (Krizhevsky et al., 2009), STL10-U (unlabeled split) (Coates et al., 2011), and CelebA (Liu et al., 2015), we trained a vanilla DDPM (Ho et al., 2020) from scratch on the *member* set. From each training split we subsample  $n$  images and partition them equally into a *member* set and a *held-out* set.

**Latent Diffusion Model:** For CIFAR-10, CelebA (Liu et al., 2015) and ImageNet-1K (Russakovsky et al., 2015), we trained a vanilla LDM (Rombach et al., 2022) from scratch on the *member* set. Likewise, we subsample  $n$  images from each training split and partition them equally into a *member* set and a *held-out* set.

**Pre-trained Guided Diffusion:** We examined the publicly released Guided Diffusion model<sup>1</sup> (Dhariwal & Nichol, 2021) trained on ImageNet-1K (Russakovsky et al., 2015). Since there is no confirmed information that this checkpoint was previously trained on ImageNet-1K validation set, ImageNetV2 (Recht et al., 2019), collected to mirror the original distribution (same data collection process and same year range), serves as the *held-out* set.

**Pre-trained Latent Diffusion Models:** For Pokémon<sup>2</sup>, COCO2017-Val Lin et al. (2014), and Flickr30k Young et al. (2014), we fine-tune Stable Diffusion v1-4<sup>3</sup> on a randomly selected subset of each training split, reserving an equally sized subset as the **held-out** set. We also studied the original Stable Diffusion v1-5<sup>4</sup> checkpoint, pre-trained on LAION-Aesthetics v2 5+ (Schuhmann et al., 2022) (*member* set). Here we sampled 2500 images from LAION-2B-MultiTranslated<sup>5</sup> as non-members, respectively. Notably, the images from LAION-2B-MultiTranslated are filtered with attributes  $p_{\text{watermark}} < 0.5$ ;  $p_{\text{prediction}}(\text{aesthetic\_score}) > 5.0$

<sup>1</sup><https://github.com/openai/guided-diffusion>

<sup>2</sup><https://huggingface.co/datasets/lambdalabs/pokemon-blip-captions>

<sup>3</sup><https://huggingface.co/CompVis/stable-diffusion-v1-4>

<sup>4</sup><https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>

<sup>5</sup><https://huggingface.co/datasets/laion/laion2B-multi-joined-translated-to-en>

and  $similarity > 0.3$ .  $pwatermark$  and  $prediction$  are to minimize the domain shift between the member set and the held-out set. And  $similarity$  is to ensure the alignment of the text-image pairs.

Full statistics including dataset splits, resolutions, conditions, etc. are summarized in Table 5 of Appendix B.

**Baselines:** Earlier membership-inference attacks (MIAs) aimed at GANs and VAEs—e.g., (Chen et al., 2020; Hilprecht et al., 2019; Hu & Pang, 2021)—do not transfer well to diffusion models, as shown by Duan et al. (2023). Consequently, we restricted our evaluation to attacks specifically designed for diffusion models. We compared our method with four baselines: SecMI (Duan et al., 2023), PIA (Kong et al., 2023), PFAMI (Fu et al., 2023), Loss (Matsumoto et al., 2023). We omitted CLiD (Zhai et al., 2024), a text-conditioned MIA, because its text-supervision is incompatible with the setup used in most of our experiments, and the provided code is not usable.

**Evaluation Metrics:** We evaluated attack performance using several metrics: ASR (attack success rate, i.e., membership inference accuracy), AUC (Area Under ROC Curve), TPR at 1% FPR (TPR@1%FPR), and the number of queries per attack (#Query). The TPR@1%FPR is computed by selecting the threshold  $\tau$  at which the false positive rate falls just below 0.01, and reporting the corresponding true positive rate at that operating point. The ASR is defined as the maximum accuracy achieved over all thresholds, i.e.  $\tau^* = \max_{\tau} \frac{1}{2}(\text{TPR}(\tau) + 1 - \text{FPR}(\tau))$  in the balanced setting. The AUC is computed as the trapezoidal integral of TPR( $\tau$ ) against FPR( $\tau$ ) across all thresholds.

**Implementation details:** Some baselines (Duan et al., 2023; Kong et al., 2023; Zhai et al., 2024) augmented their score- or feature-vector statistic with an auxiliary neural classifier; to focus on the statistic itself we evaluated only the norm-based versions, which were SecMI<sub>stat</sub> and PFAMI<sub>Met</sub>. Of the two variants (PIA and PIAN) introduced by Kong et al. (2023), we benchmarked only PIA, as PIAN showed no statistically significant gain in general from their experiments.

To minimise re-implementation error, our codebase reused the official releases of SecMI, PIA, and Guided Diffusion wherever possible. We failed to re-use PFAMI’s code as the provided code was inoperable. A reimplemented copy is provided in our code base. PFAMI, in several cases, failed to attack the victim model (ASR  $\approx$  50%). We hypothesize that this degradation arose from the sensitivity of its Monte Carlo estimator to the effective sample size, which varied with dataset characteristics and latent dimensionality, yielding high-variance estimates. For each method, we followed the hyperparameter suggestion in their original paper. Notably,  $l_2$ -norm,  $l_4$ -norm and  $l_2$ -norm were used for SecMI, PIA and Loss as suggested.  $l_4$ -norm was used for SimA as it achieved the best performance in general. For the timestep-dependent attacks (SimA, SecMI, PIA, Loss), we swept  $t = 0:300$  and reported the best-performing value for each method (PFAMI is timestep-free and therefore needed no sweep). More implementation details can be found in Section 4.3.

### 4.3 Main results

**In Table 1**, all DDPM baselines were trained on a member split and evaluated on an held-out split of equal sizes: 25k/25k for CIFAR-10 and CIFAR-100, 50k/50k for STL10-U, and 30k/30k for CelebA, all at a spatial resolution of  $32 \times 32$ . We adopted the public checkpoints and splits released by SecMI for CIFAR-10/100, and retrained STL10-U and CelebA for 18k and 60k steps, respectively. Across all datasets and metrics, SimA achieved the best in almost all the experiments over three metrics (ASR, AUC, TPR@1%FPR) while requiring the fewest queries, underscoring its efficiency and practical advantage. Sim-MC reaches the performance of SimA within 10 MC samples and outperforms within 30 MC samples. The performance of SimA-MC is expected to asymptotically approach that of SimA and eventually outperform it as the number of Monte Carlo samples increases. Notably, the Monte Carlo variant boost the performance at the critical region, as suggested by TPR@1%FPR.

**In Table 2**, the setting is mostly identical to Table 1. We increased the spatial resolution of CelebA to  $64 \times 64$ . For ImageNet-1K, the LDM was trained on 100k images of its train split. We take another 100k images as the *held-out* set. SimA achieves the best performance with a single query, and Sim-MC advances the SoTA within 10 Monte Carlo Samples.

**In Table 3**, we further evaluated Guided Diffusion (Dhariwal & Nichol, 2021) at their public class-conditional ImageNet-1K checkpoints ( $256 \times 256$ ). SimA has the highest ASR and AUC by a wide margin, while SecMI

has a very high TPR@1%FPR. SimA-MC exhibits suboptimal performance in this experiment, likely due to the vast sampling space ( $3 \times 256 \times 256$ ) in the image domain. Effectively covering such a high-dimensional manifold requires a substantially larger number of Monte Carlo samples, which poses a significant challenge in terms of computational tractability. However, Sim-MC works pretty well in a latent space diffusion model as shown in our other experiments.

**Table 4** follows the experimental protocol of Zhai et al. (2024) and evaluates membership-inference attacks on Stable Diffusion under two scenarios:

*Fine-tuning.* A Stable Diffusion v1-4 checkpoint is fine-tuned on the designated *member* split of each target dataset; attacks are launched on paired *member/held-out* splits.

*Pre-training.* A pre-trained Stable Diffusion v1-5 model is attacked directly, without additional fine-tuning. The *member* set is a subset of LAION-Aesthetics v2 5+ that was used during pre-training, while the *held-out* set is an equally-sized split drawn from the target dataset.

For every dataset we create five random *member/held-out* partitions. For LAION-2B-MultiTranslated, we provided the results for both unconditional and text-conditional cases. All other experiments are run in unconditional modes. The unconditional baseline is obtained by passing an empty string to the CLIP text encoder. The main paper reports results for Pokémon(fine-tuning), MS-COCO (fine-tuning), Flickr30k (fine-tuning); LAION-2B-MultiTranslated (pre-training) appear in Table 6 of the Appendix.

This series of experiments indicates that Diffusion processes on their own can already be solved for MIA to a high degree of fidelity with a very simple estimator.

## 5 Conclusion

In the present work we have described a simple membership inference estimator, giving theoretical justification for its performance, and for the performance of similar estimators in the literature which previously lacked a unified theoretical backing. We demonstrate that this estimator has competitive performance on many baselines. Overall, we believe SimA can serve as a lightweight auditing tool to evaluate the model memorization and privacy leakage.

## References

- Guillaume Alain and Yoshua Bengio. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.
- Boaz Barak, Jonathan A Kelner, and David Steurer. Dictionary learning and tensor decomposition via the sum-of-squares method. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 143–151, 2015.
- Yoshua Bengio, Guillaume Alain, and Salah Rifai. Implicit density estimation by local moment matching to sample from auto-encoders. *arXiv preprint arXiv:1207.0057*, 2012.
- Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX security symposium (USENIX Security 23)*, pp. 5253–5270, 2023.
- 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*, pp. 343–362, 2020.
- Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 215–223. JMLR Workshop and Conference Proceedings, 2011.
- Antonia Creswell and Anil Anthony Bharath. Inverting the generator of a generative adversarial network. *IEEE transactions on neural networks and learning systems*, 30(7):1967–1974, 2018.

- Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.
- 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*, pp. 8717–8730. PMLR, 2023.
- Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.
- Wenjie Fu, Huandong Wang, Chen Gao, Guanghua Liu, Yong Li, and Tao Jiang. A probabilistic fluctuation based membership inference attack for diffusion models. *arXiv preprint arXiv:2308.12143*, 2023.
- Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023.
- Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. Monte carlo and reconstruction membership inference attacks against generative models. *Proceedings on Privacy Enhancing Technologies*, 2019.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Hailong Hu and Jun Pang. Membership inference attacks against gans by leveraging over-representation regions. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2387–2389, 2021.
- Aapo Hyvarinen. A family of fixed-point algorithms for independent component analysis. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pp. 3917–3920. IEEE, 1997.
- Mason Kamb and Surya Ganguli. An analytic theory of creativity in convolutional diffusion models. *arXiv preprint arXiv:2412.20292*, 2024.
- Diederik P Kingma, Max Welling, et al. Auto-encoding variational bayes, 2013.
- 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.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- 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 *European conference on computer vision*, pp. 740–755. Springer, 2014.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.
- Artem Lukoianov, Chenyang Yuan, Justin Solomon, and Vincent Sitzmann. Locality in image diffusion models emerges from data statistics. *arXiv preprint arXiv:2509.09672*, 2025.
- Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.
- Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. Membership inference attacks against diffusion models. In *2023 IEEE Security and Privacy Workshops (SPW)*, pp. 77–83. IEEE, 2023.
- Elizbar A Nadaraya. On estimating regression. *Theory of Probability & Its Applications*, 9(1):141–142, 1964.
- Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

- 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.
- Jakiw Pidstrigach. Score-based generative models detect manifolds. *Advances in Neural Information Processing Systems*, 35:35852–35865, 2022.
- Mingxing Rao, Bowen Qu, and Daniel Moyer. Latent diffusion inversion requires understanding the latent space. *arXiv preprint arXiv:2511.20592*, 2025.
- Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.
- Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.
- Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
- 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*, pp. 5558–5567. PMLR, 2019.
- Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in neural information processing systems*, 35:25278–25294, 2022.
- 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)*, pp. 3–18. IEEE, 2017.
- Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6048–6058, 2023a.
- Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36: 47783–47803, 2023b.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- 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.
- Robert J Stanton and Alan Weinstein. On the  $l_4$  norm of spherical harmonics. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 89, pp. 343–358. Cambridge University Press, 1981.
- Geoffrey S Watson. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 359–372, 1964.
- 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.
- 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.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European conference on computer vision*, pp. 597–613. Springer, 2016.

## A Appendix

### B Datasets and splits

Datasets and splits used for our experiments are summarized in Table 5.

### C Stable Diffusion pre-trained on LAION-Aesthetics v2 5+.

We also studied the original Stable Diffusion v1-5<sup>6</sup> checkpoint, pre-trained on LAION-Aesthetics v2 5+ (Schuhmann et al., 2022) (*member set*). Here we sampled 2500 images from LAION-2B-MultiTranslated<sup>7</sup> as non-members, respectively. Notably, the images from LAION-2B-MultiTranslated are filtered with attributes  $p_{\text{watermark}} < 0.5$ ;  $p_{\text{prediction}}(\text{aesthetic\_score}) > 5.0$  and  $\text{similarity} > 0.3$ .  $p_{\text{watermark}}$  and  $p_{\text{prediction}}$  are to minimize the domain shift between the member set and the held-out set. And  $\text{similarity}$  is to ensure the alignment of the text-image pairs. We provided both unconditional and text-conditional cases in Table 6. PIA performs the best at this setting, and SimA achieved comparable results.

### D From marginal density to explicit Gaussian convolution

Let  $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$  and  $\sigma_t^2 = 1 - \bar{\alpha}_t$ . The VP forward marginal is

$$p_t(x) = \int_{\mathbb{R}^d} p_{\text{data}}(x_0) \mathcal{N}(x \mid \sqrt{\bar{\alpha}_t} x_0, \sigma_t^2 I) dx_0. \quad (21)$$

Set  $u = \sqrt{\bar{\alpha}_t} x_0$  so that  $x_0 = u/\sqrt{\bar{\alpha}_t}$  and  $dx_0 = \bar{\alpha}_t^{-d/2} du$ . Then

$$p_t(x) = \bar{\alpha}_t^{-d/2} \int_{\mathbb{R}^d} p_{\text{data}}\left(\frac{u}{\sqrt{\bar{\alpha}_t}}\right) \mathcal{N}(x \mid u, \sigma_t^2 I) du. \quad (22)$$

Introduce  $\tilde{x} := x/\sqrt{\bar{\alpha}_t}$  and  $\tilde{u} := u/\sqrt{\bar{\alpha}_t}$ , so  $u = \sqrt{\bar{\alpha}_t} \tilde{u}$  and  $du = \bar{\alpha}_t^{d/2} d\tilde{u}$ . Substituting into equation 22 cancels the Jacobians and yields

$$p_t(x) = \int_{\mathbb{R}^d} p_{\text{data}}(\tilde{u}) \mathcal{N}(\sqrt{\bar{\alpha}_t}(\tilde{x} - \tilde{u}) \mid 0, \sigma_t^2 I) d\tilde{u}. \quad (23)$$

Use the Gaussian scaling identity

$$\mathcal{N}(az \mid 0, \sigma^2 I) = a^{-d} \mathcal{N}\left(z \mid 0, \frac{\sigma^2}{a^2} I\right) \quad (\text{for } a > 0),$$

with  $a = \sqrt{\bar{\alpha}_t}$  and  $z = \tilde{x} - \tilde{u}$ . Then equation 23 becomes

$$p_t(x) = \bar{\alpha}_t^{-d/2} \int_{\mathbb{R}^d} p_{\text{data}}(\tilde{u}) \mathcal{N}\left(\tilde{x} - \tilde{u} \mid 0, \frac{\sigma_t^2}{\bar{\alpha}_t} I\right) d\tilde{u}. \quad (24)$$

The integral in equation 24 is a (Euclidean) convolution evaluated at  $\tilde{x}$ :

$$(p_{\text{data}} * \mathcal{N}(0, \frac{\sigma_t^2}{\bar{\alpha}_t} I))(\tilde{x}) = \int_{\mathbb{R}^d} p_{\text{data}}(\tilde{u}) \mathcal{N}\left(\tilde{x} - \tilde{u} \mid 0, \frac{\sigma_t^2}{\bar{\alpha}_t} I\right) d\tilde{u}.$$

<sup>6</sup><https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5>

<sup>7</sup><https://huggingface.co/datasets/laion/laion2B-multi-joined-translated-to-en>

Table 5: Datasets and splits used for our experiments.

Model	Member	Held-out	Pre-trained	Fine-tuned	Splits	Resolution	Cond.
DDPM	CIFAR-10	CIFAR-10	No	–	25k/25k	32	–
	CIFAR-100	CIFAR-100	No	–	25k/25k	32	–
	STL10-U	STL10-U	No	–	50k/50k	32	–
	CelebA	CelebA	No	–	30k/30k	32	–
Latent Diffusion Model	CIFAR-10	CIFAR-10	No	–	25k/25k	32	–
	CIFAR-100	CIFAR-100	No	–	25k/25k	32	–
	STL10-U	STL10-U	No	–	50k/50k	32	–
	CelebA	CelebA	No	–	30k/30k	64	–
	ImageNet-1k	ImageNet-1k	No	–	100k/100k	256	class
	ImageNet-1k	ImageNetv2	No	–	3k/3k	256	class
Guided Diffusion	ImageNet-1k	ImageNetV2	Yes	No	3k/3k	256	class
Stable Diffusion V1-4	Pokémon	Pokémon	Yes	Yes	416/417	512	text
	COCO2017-Val	COCO2017-Val	Yes	Yes	2.5k/2.5k	512	text
	Flickr30k	Flickr30k	Yes	Yes	10k/10k	512	text
Stable Diffusion V1-5	LAION-Aesthetics v2 5+	LAION-2B-MultiTranslated	Yes	No	2.5k/2.5k	512	text
	LAION-Aesthetics v2 5+	COCO2017-Val	Yes	No	2.5k/2.5k	512	text

Table 6: Performance of **Stable Diffusion** that is pretrained on LAION-Aesthetics v2 5+. LAION-2B-MultiTranslated is the pre-training setting. The performance difference of the text-conditional attack from the unconditional one is in blue. Format:  $uncond.\% (\Delta = cond.\% - uncond.\%)$ .

Method	#Query↓	LAION-2B-MultiTranslated (%)		
		ASR↑	AUC↑	TPR@1%FPR↑
PIA	2	<b>60.6(-0.0)</b>	<b>63.8(+0.0)</b>	<b>4.12(+0.08)</b>
PFAMI <sub>Met</sub>	20	50.0(+0.0)	40.8(+0.2)	0.80(+0.08)
SecMI <sub>stat</sub>	12	56.7(+0.1)	58.6(+0.0)	<u>2.92(-0.16)</u>
Loss	1	50.1(+0.0)	37.0(+0.0)	0.80(+0.00)
SimA	1	<u>60.3(-0.0)</u>	<u>63.3(-0.0)</u>	<u>3.20(-0.08)</u>

Therefore,

$$p_t(x) = \bar{\alpha}_t^{-d/2} \left( p_{\text{data}} * \mathcal{N}\left(0, \frac{\sigma_t^2}{\bar{\alpha}_t} I\right) \right) \left( \frac{x}{\sqrt{\bar{\alpha}_t}} \right) \quad (25)$$

with  $\sigma_t^2/\bar{\alpha}_t = \bar{\alpha}_t^{-1} - 1$ .

## E MIA in three cases

When the *data prior* is the empirical distribution constructed from the training set  $\{x^{(i)}\}_{i=1}^N$ . We have  $p_{\text{train}}(x_0) = \frac{1}{N} \sum_i \delta(x_0 - x^{(i)})$ . For any test image  $x$ , the UNet predicts

$$\hat{\varepsilon}_\theta(x, t) \approx \frac{x - \sqrt{\bar{\alpha}_t} \mu_t(x)}{\sigma_t}, \quad (26)$$

We are interested in the quantity

$$\begin{aligned} \mu_t^{\text{finite}}(x) &= \mathbb{E}_{q_t(x_0|x)}[x_0] = \sum_{i=1}^N w_i(x, t) x^{(i)}, \\ w_i(x, t) &= \frac{\exp[-\|x - \sqrt{\bar{\alpha}_t} x^{(i)}\|^2 / (2\sigma_t^2)]}{\sum_{j=1}^N \exp[-\|x - \sqrt{\bar{\alpha}_t} x^{(j)}\|^2 / (2\sigma_t^2)]}. \end{aligned} \quad (27)$$

Equations equation 27 express the posterior mean as a weighted average of the training samples, where each kernel weight  $w_i(x, t)$  depends on the Euclidean distance between the noisy query  $x$  and the down-scaled datum  $\sqrt{\bar{\alpha}_t} x^{(i)}$ .

### E.1 Case 1 — Training member

Pick  $x = x^{(k)} \in \{x^{(i)}\}$ . Set  $x = x^{(k)}$  in equation 27. Define the squared distances

$$d_{ik}(t) := \|x^{(k)} - \sqrt{\bar{\alpha}_t} x^{(i)}\|^2, \quad \Delta_{ik}(t) := \frac{d_{ik}(t) - d_{kk}(t)}{2\sigma_t^2}. \quad (28)$$

Using equation 27 and equation 28 we obtain

$$w_k(x^{(k)}, t) = \left[ 1 + \sum_{i \neq k} \exp(-\Delta_{ik}(t)) \right]^{-1}, \quad (29)$$

$$w_{i \neq k}(x^{(k)}, t) = \exp(-\Delta_{ik}(t)) w_k(x^{(k)}, t). \quad (30)$$

**Small-noise limit**  $t \rightarrow 0$ . Because  $\sigma_t^2 = 1 - \bar{\alpha}_t \rightarrow 0$  and

$$1 - \sqrt{\bar{\alpha}_t} = O(\sigma_t^2), \quad d_{kk}(t) = (1 - \sqrt{\bar{\alpha}_t})^2 \|x^{(k)}\|^2 = O(\sigma_t^4),$$

we have, for  $i \neq k$ ,

$$\Delta_{ik}(t) \sim \frac{\|x^{(k)} - x^{(i)}\|^2}{2\sigma_t^2} \xrightarrow{t \rightarrow 0} +\infty, \quad \Delta_{kk}(t) = 0.$$

Hence

$$w_k(x^{(k)}, t) \xrightarrow{t \rightarrow 0} 1, \quad w_{i \neq k}(x^{(k)}, t) \xrightarrow{t \rightarrow 0} 0. \quad (31)$$

This implies

$$\boxed{\mu_t^{\text{finite}}(x^{(k)}) \xrightarrow{t \rightarrow 0} x^{(k)}}. \quad (32)$$

Moreover, substituting into the estimator equation 26,

$$\|\hat{\varepsilon}_\theta(x^{(k)}, t)\|_2 \approx \left\| \frac{x^{(k)} - \sqrt{\bar{\alpha}_t} x^{(k)}}{\sigma_t} \right\|_2 = \frac{|1 - \sqrt{\bar{\alpha}_t}|}{\sigma_t} \|x^{(k)}\|_2. \quad (33)$$

To see the asymptotic form using Taylor expansion around  $\sigma_t^2 = 0$ , note that

$$\sqrt{\bar{\alpha}_t} = \sqrt{1 - \sigma_t^2} = 1 - \frac{1}{2}\sigma_t^2 - \frac{1}{8}\sigma_t^4 + O(\sigma_t^6),$$

so that

$$1 - \sqrt{\bar{\alpha}_t} = \frac{1}{2}\sigma_t^2 + O(\sigma_t^4).$$

Therefore

$$\frac{1 - \sqrt{\bar{\alpha}_t}}{\sigma_t} = \frac{1}{2}\sigma_t + O(\sigma_t^3),$$

and hence

$$\|\hat{\varepsilon}_\theta(x^{(k)}, t)\|_2 \sim \frac{\sigma_t}{2} \|x^{(k)}\|_2 \xrightarrow{t \rightarrow 0} 0. \quad (34)$$

## E.2 Case 2 — Held-out but On-Manifold

Consider a test data point  $x^\dagger$  which is not in our original dataset (i.e.,  $x^\dagger \notin \{x^{(i)}\}_{i=1}^N$ ), but sampled from the same generating distribution  $p_{\text{data}}$ . Under the diffusion process assumptions, the local weighted mean  $\mu_t(x)$  in equation 27 has the same algebraic form as a kernel regression (Nadaraya, 1964; Watson, 1964), using training dataset  $\{x^{(i)}\}_{i=1}^N$  with (effective) Gaussian bandwidth

$$h(t) := \frac{\sigma_t}{\sqrt{\alpha_t}}. \quad (35)$$

The weights  $w_i(x, t)$  in equation 27 are proportional to  $\exp(-\|x^\dagger - \sqrt{\alpha_t}x^{(i)}\|_2^2 / (2h(t)^2))$ , so  $\mu_t^{\text{finite}}(x^\dagger)$  coincides with a Gaussian-kernel local average of “nearby” training points, where “nearby” is on the order of bandwidth  $h(t)$ . The kernel-weighted local mean with radius  $r$  around  $x^\dagger$  is defined as

$$m_r(x^\dagger) := \frac{\int_{B_r(x^\dagger)} u K_r(u - x^\dagger) p(u) du}{\int_{B_r(x^\dagger)} K_r(u - x^\dagger) p(u) du} \approx \mu_t^{\text{finite}}(x^\dagger), \quad (36)$$

$$K_r(z) \propto \exp\left(-\frac{\|z\|_2^2}{2r^2}\right), \quad (37)$$

where  $r \asymp h(t)$  and  $p(u)$  is the empirical distribution. This is a normalization of the local mean defined in Eq. 16 of this paper, and Appendix 6.4.1 of Alain & Bengio (2014). By local moment matching (Bengio et al., 2012; Alain & Bengio, 2014), combining Theorems 2 and 3 of Alain & Bengio (2014) (see their Eq. (28)) yields the second-order expansion

$$m_r(x^\dagger) - x^\dagger = \frac{r^2}{d+2} \left. \frac{\partial \log p_t(x)}{\partial x} \right|_{x^\dagger} + o(r^3), \quad (38)$$

where  $p_t$  is the Gaussian-mollified density at scale  $\sigma_t$ . Using equation 9, we obtain the proportionality stated in the main text:

$$\boxed{m_r(x^\dagger) - x^\dagger \approx \frac{r^2}{d+2} \left. \frac{\partial \log p_t(x)}{\partial x} \right|_{x^\dagger} = \frac{r^2}{d+2} \left( \frac{x^\dagger - \sqrt{\alpha_t} \mu_t(x^\dagger)}{\sigma_t^2} \right) = -\frac{r^2}{\sigma_t(d+2)} \hat{\varepsilon}_\theta(x^\dagger, t)} \quad (39)$$

Notably, this equation should *not* be correct if  $x^\dagger$  is in a low density region of the support of  $p_{\text{data}}$ , but these points should generally be rarely sampled by definition. Moreover, the  $\hat{\varepsilon}$  estimate of the score will also be extrapolating at those points.

**Implication for the attack statistic.** In in-support regions of the empirical distribution, the local geometry is generally *not* flat, so  $\|m_r(x^\dagger) - x^\dagger\| > 0$  for  $r > 0$  sufficiently small. By equation 39, this implies  $\|\hat{\varepsilon}_\theta(x^\dagger, t)\|$  is bounded away from zero (at fixed  $t$ ), hence  $\mathcal{A}(x^\dagger, t) = \|\hat{\varepsilon}_\theta(x^\dagger, t)\|_p$  exceeds the member case (Case 1). This matches the intuition summarized in the main text: on-manifold, held-out queries denoise less precisely than memorized training points, yielding a moderately larger attack statistic without the divergence seen off-manifold.

## E.3 Case 3 — Out-of-distribution (far off-manifold)

Since no training data support is available in out-of-distribution (OOD) regions, the diffusion model lacks information about these areas. Consequently, the learned score field in such regions is necessarily an extrapolation, and the theoretical derivations established under the in-distribution assumption no longer hold.

Specifically, in the adversarial setup, optimization proceeds in two steps per iteration. First, we update the VAE with the usual reconstruction+KL objective and, after a warm-up, add a non-saturating adversarial term (hinge form; equivalent to  $-\mathbb{E}[D(\hat{x})]$ ) gated to turn on only after a chosen global-step threshold (we set this to 2,000 steps in our runs). The adversarial term is weighted adaptively using a VQGAN-style (Esser

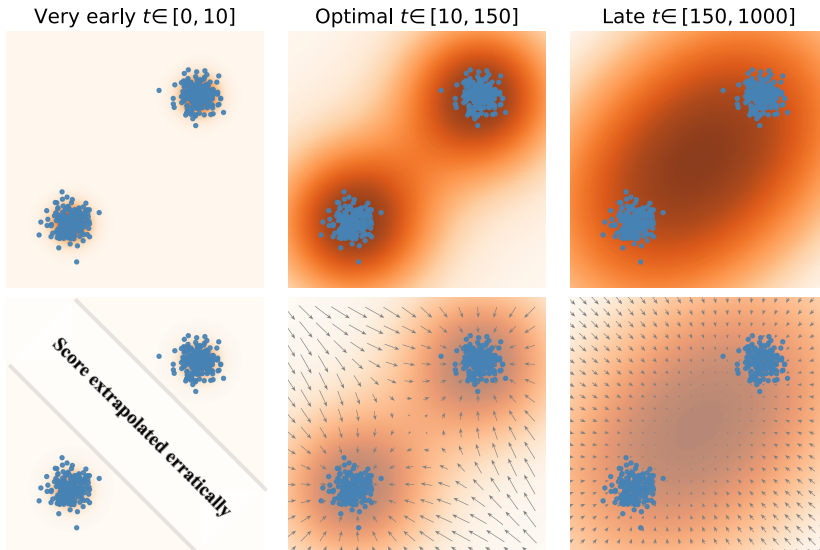


Figure 3: Top: density  $p_t(x)$ ; blue dots are training samples. Bottom: estimated score  $\nabla_x \log p_t(x)$ . For **very early**  $t \in [0, 10]$ , the inter-mode region is low-density, so the score extrapolates erratically (shaded band). **Optimal**  $t \in [10, 300]$ : moderate Gaussian convolution enlarges the support and regularizes the estimator—density bridges the modes and the score points coherently toward them, yielding the strongest separation between members and held-out points. For **late**  $t \in [300, 1000]$ ,  $p_t$  approaches an isotropic Gaussian and the score collapses toward the global mean, diminishing membership signal.

et al., 2021) gradient-norm ratio on the decoder’s last layer and then scaled by a constant factor of 0.8; before the start step, its weight is zero by design. Second, we update the discriminator with a hinge loss on real images  $x$  versus reconstructions  $\hat{x}$  recomputed in a no-grad branch (to avoid generator gradient leakage), using AdamW with the same learning rate/betas/weight-decay as the VAE optimizer.

## F Why Extremely Early and Late $t$ are Problematic:

Intuitively,  $\hat{\varepsilon}_\theta(x, t)|_{t=0}$  is expected to achieve the best performance. Because the noise added to a member at  $t = 0$  is expected to be zero (as shown in **case 1**) and non-zero for a non-member. However, in region of low data density, score-matching lacks sufficient evidence to *reliably* estimate the score function (Song & Ermon, 2019). Song & Ermon (2019) argues that training minimizes the expected value of score estimates (here is  $\mathbb{E}_{p_{t=0}}[\|\hat{\varepsilon}_\theta(x_{t=0}, t) - \varepsilon\|^2]$ ), which provides *inaccurate scores* where  $p_{t=0}(x)$  is infinitesimal. To be specific, for the input  $\{x \in \mathcal{R} \mid \{\mathbf{x}_i\}_{i=1}^N \cap \mathcal{R} = \emptyset, \mathcal{R} \subset \mathbb{R}^d\}$ ,  $\nabla_x p_{t=0}(x)$  extrapolates erratically. Consequently, for **very early timesteps** ( $\sigma_t \ll 1$ ) the learned field outside the tightly supported member set can plateau or even shrink, nullifying the privacy signal. Increasing  $t$  corresponds to *extra Gaussian convolution*, expanding the effective support and regularising the score. Figure 4 plots the average normalised estimator magnitude  $\|\hat{\varepsilon}_\theta(x, t)\|$  for  $t \in [0, 300]$  on the *member* and *held-out* splits across datasets. Transient fluctuations are confined to the very earliest timesteps ( $t \approx 0$ ), and the maximal gap between the curves typically occurs at early—but not initial—timesteps, indicating that moderately early diffusion steps provide the strongest membership signal.

Conversely, for **late steps** ( $\sigma_t \approx 1$ ) the forward process approaches an isotropic Gaussian (data information gradually diminish);  $p_t$  is nearly homogeneous, so the posterior  $\mu_t(x) = q_t(x_0 \sim p_{\text{training}} \mid x)$  collapses to  $\mu_t(x) = q_t(x_0 \sim \mathcal{N}(0, I) \mid x)$ , which depends on test images and membership information is lost. Figure 3 illustrates the phenomenon. The *optimal* timestep  $t^*$  is therefore dataset-specific and also depends on the noise schedule.

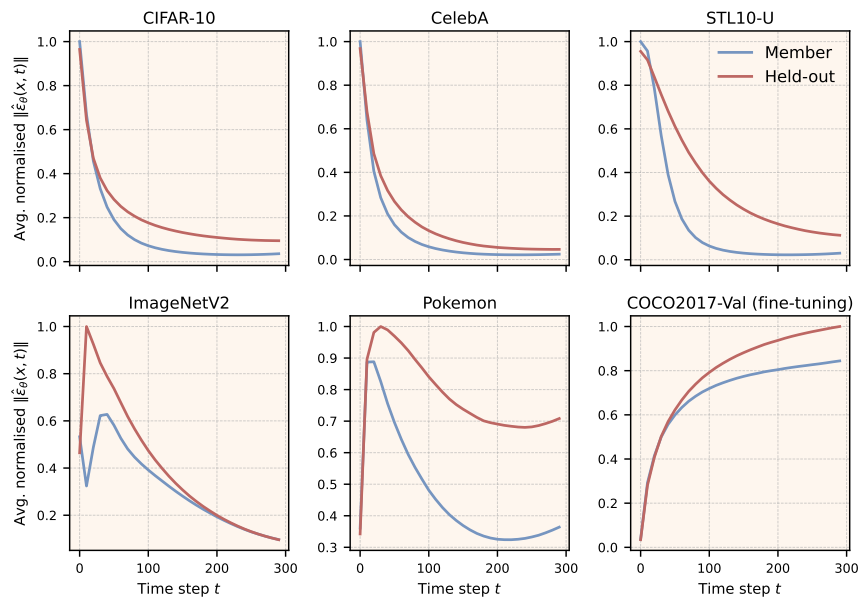


Figure 4: The average normalised estimator magnitude  $\|\hat{\epsilon}_\theta(x, t)\|$  for  $t \in [0, 300]$  on the *member* and *held-out* splits across datasets