Generalization and Memorization in Rectified Flow

Mingxing Rao and Daniel Moyer*

Vanderbilt University, Nashville TN 37235, USA
{mingxing.rao,daniel.moyer}@vanderbilt.edu

Abstract. Generative models based on the Flow Matching objective, particularly Rectified Flow, have emerged as a dominant paradigm for efficient, high-fidelity image synthesis. However, while existing research heavily prioritizes generation quality and architectural scaling, the underlying dynamics of how RF models memorize training data remain largely underexplored. In this paper, we systematically investigate the memorization behaviors of RF through the test statistics of Membership Inference Attacks (MIA). We progressively formulate three test statistics, culminating in a complexity-calibrated metric ($T_{\text{mc\_cal}}$) that successfully decouples intrinsic image spatial complexity from genuine memorization signals. This calibration yields a significant performance surge—boosting attack AUC by up to 15% and the privacy-critical TPR@1%FPR metric by up to 45%—establishing the first non-trivial MIA specifically tailored for RF. Leveraging these refined metrics, we uncover a distinct temporal pattern: under standard uniform temporal training, a model's susceptibility to MIA strictly peaks at the integration midpoint, a phenomenon we justify via the network's forced deviation from linear approximations. Finally, we demonstrate that substituting uniform timestep sampling with a Symmetric Exponential (U-shaped) distribution effectively minimizes exposure to vulnerable intermediate timesteps. Extensive evaluations across three datasets confirm that this temporal regularization suppresses memorization while preserving generative fidelity.

Keywords: Model Memorization · Membership Inference · Rectified Flow

1 Introduction

Flow Matching (FM) [30] has emerged as a powerful, simulation-free paradigm to overcome the slow sampling bottlenecks of Diffusion Models (DMs) [18, 20, 39, 41, 42]. Among its variants [4, 30, 43], Rectified Flow (RF) [32] stands out by connecting data and noise distributions via straight-line trajectories. This linearizes the generation process, enabling highly efficient, few-step sampling. Due to its stability and efficiency, RF underpins contemporary state-of-the-art foundation models such as Stable Diffusion 3 [11], FLUX [25], and Ideogram

* Corresponding author.

4 [2]. However, while prior work heavily optimizes RF for generation quality [36, 43] and architectural scaling, its memorization and generalization dynamics are relatively underexplored.

Conventional indicators of overfitting, such as the standard generalization gap between training and validation losses, fail to accurately capture the complex, time-dependent memorization behaviors inherent to RF (shown in Appendix C). Even if loss gaps are small, it may be possible to recover the specific training data from model weights, outputs, or behavior/activations during generation. This process is known as model inversion [12, 37, 38]. A specific sub-task of model inversion is classifying samples as drawn from the training set or otherwise using those same model statistics; this is known as membership inference [7], and particular methods referred to as membership inference attacks (MIAs).

The ability to perform an MIA against a flow matching model is indicative of memorization; if we can discriminate between training and non-training data using the model itself, even though both training and non-training sets are ostensibly drawn from the same underlying population, there is some training set specific information left in the model.

In this work we construct an MIA for rectified flow. We first establish a baseline test criterion, T_{naive}, derived from the theoretical Flow Matching (FM) objective. We further formulate a Monte Carlo estimator, T_{mc}, based on the practical and commonly used Conditional Flow Matching (CFM) objective. We provide detailed motivation and show significant performance improvement for the transition from T_{naive} to T_{mc}. We then propose a correction/calibration $T_{\text{mc\_cal}}$ for an image-domain specific phenomenon in generative modelling, spatial complexity biases [34, 40]).

Utilizing these refined metrics, we uncover a non-trivial temporal pattern in the memorization dynamics of continuous-time models. When employing a standard uniform temporal sampling strategy $t \sim \mathcal{U}(0, 1)$, the model's susceptibility to MIA is heavily concentrated in the intermediate integration stages, and maximal at the midpoint ($t = 0.5$) of the flow (Fig 1. We provide mathematical justification for this phenomenon by analyzing the deviation between the non-linear regressor and a Linear Minimum Mean Square Error (LMMSE) baseline. Specifically, we show that at $t = 0.5$, the noisy input state becomes statistically orthogonal to the target velocity. Consequently, the network is forced to bypass trivial linear approximations and exploit its non-linear capacity to memorize sample-specific features, thereby maximizing the memorization problem at this exact juncture.

Finally, we tackle the persistent utility-privacy trade-off that typically exists in classic Differential Privacy (DP) mechanisms in generative modeling (see Section 2.3). To mitigate the accumulation of privacy risks without degrading the visual quality of generated samples, we intervene directly in the training dynamics. By substituting the conventional uniform timestep sampling with a Symmetric Exponential (U-shaped) distribution [26], we deliberately minimize the model's exposure to the highly vulnerable intermediate timesteps. Our the-

oretical and empirical analyses confirm that this temporal regularization substantially impedes memorization while preserving generative fidelity (Fig. 4).

We conduct all experiments on CIFAR-10 [24], SVHN [35], and TinyImageNet [8] datasets. Three datasets consistently justify our proposed MIA test statistics and analysis of memorization.

In summary, our contributions are the following:

1. We progressively construct three test statistics (T_{naive}, T_{mc}, $T_{\text{mc\_cal}}$) for membership inference attack and justify each one with our motivation and derivation. We also empirically shows its practical usage.
2. With the MIA test statistics, we substantiate our claim that, under reasonable assumptions, the expected memorization risk for well-trained rectified flow across the entire continuous timesteps is strictly upper-bounded by the memorization evaluated at the midpoint.
3. We replace the standard uniform sampling of t with the Symmetric Exponential distribution proposed by Lee et al. [26], providing a novel and comprehensive justification from the dynamics of model generalization and memorization.

All data splits, model checkpoints, training, and testing code are released on our GitHub repository `https://github.com/mx-ethan-rao/rf_gen_mem.git`.

2 Background and Motivation

2.1 Membership Inference Attack

The goal of the Membership Inference Attack (MIA) is to determine whether a specific data record x was sampled from the underlying population distribution to be included in the empirical training set $\mathcal{D}_{train}$ of a target model f_θ. Carlini et al. [7] formally cast membership inference as a rigorous statistical hypothesis testing problem. A variety of test statistics have been extensively developed for diffusion-based generative paradigms [10, 13, 23, 38, 49]. However, the research on the diffusion model does not extend to the model trained via flow matching [30, 32]. We proceed from first principles (the training objectives) to mathematically derive a novel test statistic inherently tailored to the Rectified Flow [32] framework. Because models suffering from severe overfitting naturally exhibit high susceptibility to MIA, this derived statistic also serves as a quantitative metric to evaluate the dynamics between model generalization and training data memorization [37].

2.2 Generalization and Memorization

The divergence between training and validation loss—formally known in learning theory as the generalization gap—is the most fundamental statistical signature of overfitting. This method is extensively used on memorization analysis of VAE [6], GAN [17, 46] and Diffusion Model [5, 16, 19, 28, 33, 48]. However, a trivial training/val loss gap proves inadequate for capturing the nuanced

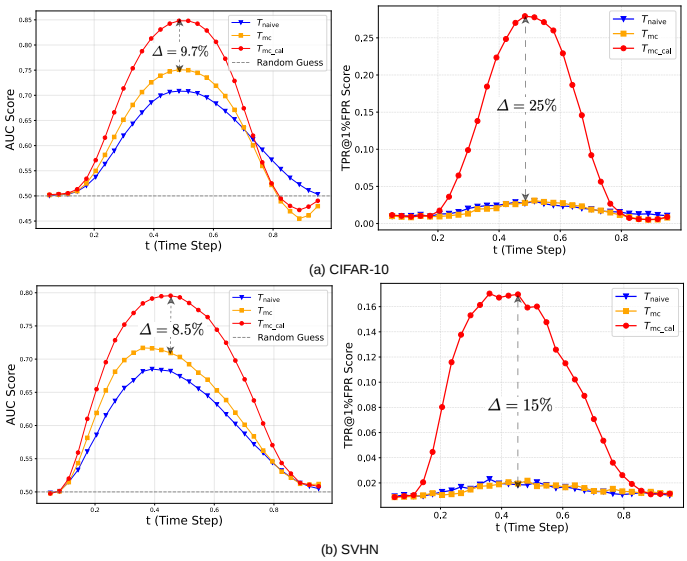


Fig. 1: MIA performance (AUC and TPR@1%FPR, at **left** and **right** respectively) across integration timesteps t on CIFAR-10 and SVHN (**top** row and **bottom** row). Our complexity-calibrated statistic ($T_{\text{mc\_cal}}$) demonstrates a substantial performance surge over the baselines (T_{naive} and T_{mc}).

memorization dynamics inherent to Rectified Flows (RF) [32], as evidenced by the empirical loss trajectories in Appendix C. To establish a rigorous analytical framework, we repurpose the test statistic formulation traditionally utilized in Membership Inference Attacks to precisely quantify sample-specific memorization within RF. By using this metric across different timestep t, we uncover that memorization is heavily concentrated at the intermediate t, whereas the vulnerability remains relatively benign at the temporal extremes (i.e., early and late stages of t). We derive a closed-form expression for the exact timestep at which model memorization peaks under the overfitting assumption. We elaborate this time-dependent memorization in Section 4.

2.3 Differential Privacy

A classic framework to prevent the deep learning model from memorizing individual training samples is Differential privacy [1]. Abadi et al. [1] proposed Differential Privacy-Schocastic Gradient Descent (DP-SGD) and momentum accountant that mathematically guarantee that privacy loss accumulates significantly slower over thousands of SGD iterations. However, enforcing differential privacy on generative models inherently introduces a severe privacy-utility (e.g. generative fidelity) trade-off [9, 44]. This creates a fundamental tension: from a

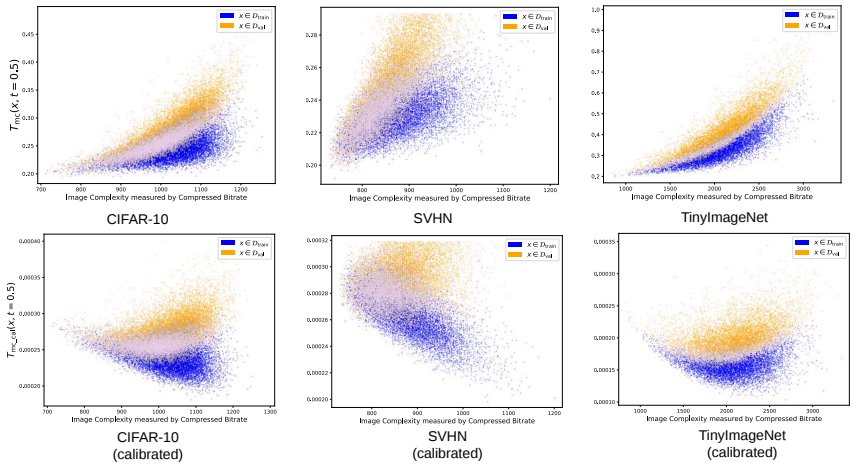


Fig. 2: Top row: The test statistics T_{mc} without calibration. **Bottom row:** The test statistics T_{mc} with calibration. The calibrated version provides more separable points between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$

privacy perspective, repeated exposure to an individual sample exacerbates the risk of catastrophic memorization; yet, from a generative modeling standpoint, the network strictly requires extensive exposure to high-quality data to generate good examples. This raises a critical question: *Can we slow down the privacy loss accumulation without sacrificing generative fidelity?* In Section 4.3, by distorting the sampling distribution (originally, uniform distribution) of the timestep t during the training phase, we empirically show that the model's memorization dynamics are significantly disrupted with the same number of SGD steps. We also provide a theoretical interpretation in section 4 for our insight.

2.4 Input Complexity Bias in Generative Models

Inspired by Out-of-Distribution detection, likelihoods computed from generative models are notoriously susceptible to the inherent complexity of the input data [14, 34, 40]. Nalisnick et al. [34] first exposed this vulnerability by demonstrating that generative models (Glow [22], PixelCNN [45], and VAE [21]) often assign higher likelihoods to visually simpler OOD samples (e.g., SVHN) than to the complex in-distribution data they were trained on (e.g., CIFAR-10). Serrà et al. [40] formally identified input complexity—often measurable via standard compression bitrates—as the primary confounding factor, revealing that likelihood estimations are heavily dominated by low-level spatial redundancies rather than high-level semantic content. This complexity bias has since been shown to persistently plague modern generative paradigms, including Denoising Diffusion Probabilistic Models [14]. In continuous-time frameworks such as Rectified

Flows, the likelihood is computed through the instantaneous change of variables theorem [15, 30, 31]. For held-out dataset $x \in \mathcal{D}_{\text{data}} \setminus \mathcal{D}_{\text{train}}$, we confirmed the same phenomenon (See Appendix A)on Rectified Flow. The likelihood also served as an important MIA test statistic. This provides insight that the proposed MIA test statistic is heavily biased towards input complexity as well.

3 MIA Test Statistics for Rectified Flow

In this section, we formalize our proposed methodology. We derive our test statistics for membership inference attack from first principles based on training objectives (Section 3.1) and introduce a calibration mechanism conditioned on input image complexity (Section 3.2). We justify this calibration strategy and demonstrate its empirical benefits in Section 3.2.

Preliminaries: The flow matching [30] framework defines a system of ordinary differential equations (ODEs) which transport particles from a "base" distribution π_0 (usually chosen to be simple, e.g., standard Gaussian noise) to an empirical target distribution π_1 (e.g., real images). Let $X_0 \sim \pi_0$ and $X_1 \sim \pi_1$ denote the source and target random variables, respectively. Flow matching implicitly defines an interpolent path between these two distributions over a continuous time variable $t \in [0, 1]$ by defining flow fields $v(x, t)$

$$dX_t = v(X_t, t)dt \tag{1}$$

so that $X_{t=0}$ has distribution π_0, and that $X_{t=1}$ has distribution π_1. A common choice of field is the Rectified Flow [32], which is defined implicitly by drawing straight lines between pairs of points and calculating the interpolant field:

$$X_t = tX_1 + (1 - t)X_0 \tag{2}$$

While our theoretical analysis extends to other Flow Matching variants (detailed in Appendix D), we mainly focus on Rectified Flow, as it is conceptually simple, quite performant, and thereby has become the most popular in contemporary work.

The time derivative of linear path flows yield a constant vector field pointing directly from X_0 to X_1, given by $\frac{\mathrm{d}X_t}{\mathrm{d}t} = X_1 - X_0$. During inference, the high level objective of flow matching is to fit a neural network $v_\theta(X_t, t)$ to that vector field condition, fitting an explict function that for any given timestep t and position x can produce the appropriate vector. The original flow matching procedure fit v_θ by least-squares optimization of the linear path criterion [30]:

$$\mathcal{L}_{\text{FM}}(\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,1),X_0\sim\pi_0,X_1\sim\pi_1}\left[\|(X_1 - X_0) - v_\theta(X_t, t)\|^2\right]. \tag{3}$$

where $\mathcal{U}(0, 1)$ denotes the uniform sampling of $t \in [0, 1]$.

Directly optimizing the original FM objective is computationally intractable due to the integration over a data distribution required to compute the marginal

vector field. To circumvent this limitation, Lipman et al. [30] introduced Conditional Flow Matching (CFM). By conditioning on individual image samples x_1, the CFM objective utilizes conditional vector fields that can be efficiently computed and optimized over on a per-sample basis. The CFM objective is

$$\mathcal{L}_{\text{CFM}}(\theta|x_1) = \mathbb{E}_{t\sim\mathcal{U}(0,1),X_0\sim\pi_0}\left[\|(X_1 - X_0) - v_\theta(X_t,t)\|^2 \mid X_1 = x_1\right]. \quad (4)$$

A crucial observation is that $\nabla_\theta \mathcal{L}_{FM}(\theta) = \nabla_\theta \mathcal{L}_{CFM}(\theta|x_1)$ [30, 31], meaning that the CFM objective has the same optima as the original FM objective, and that θ optimized for one objective have the same relative quality for the other objective.

3.1 Deriving MIA Test Criteria

Naive attack: Based on the FM objective (Eq. 3), we can construct a basic model inversion attack. Given a test image $\boldsymbol{x}$, we sample Gaussian noise ε. We write the "naive" test criterion for MIA

$$T_{\text{naive}}(\boldsymbol{x}, \boldsymbol{t}) = \|\boldsymbol{x} - (v_\theta(\boldsymbol{x_t}, t) + \varepsilon)\|^2 \text{ with } \boldsymbol{x_t} = t\boldsymbol{x} + (1-t)\varepsilon. \quad (5)$$

For models minimizing Eq. 3, as the models overfit, we observe $T_{\text{naive}}(\boldsymbol{x} \in \mathcal{D}_{train}) < T_{\text{naive}}(\boldsymbol{x} \in \mathcal{D}_{val})$. Clearly an "oracle" perfect flow would have the same T value in expectation over training and validation datasets, but as the model becomes specialized toward individual datapoints, a classic overfitting pattern, the flow will fail to generalize, and this gap will widen. As a demonstration, we give the following conceptual experiment: consider a single draw from a Gaussian as the target dataset. The $\mathcal{L}_{FM}$ or $\mathcal{L}_{CFM}$ flow would eventually point towards that draw from all positions, meaning that subsequent draws from the mode of the Gaussian would be pulled away from that mode, even though in expectation they should not move.

The T_{naive} criterion includes a chosen/sampled ε. We can take this one step further and derive a Monte Carlo-amortized criterion T_{mc}. Consider a fully overfit model minimizing the CFM loss, i.e., with $\mathcal{L}_{CFM} = 0$,

$$\mathbb{E}_{t\sim\mathcal{U}(0,1),X_0\sim\pi_0}\left[\|(x_1 - X_0) - v_\theta(X_t,t)\|_2^2\right] = 0. \quad (6)$$

which is equivalent to

$$\mathbb{E}[x_1] - \mathbb{E}[X_0] - \mathbb{E}[v_\theta(X_t,t)] = 0. \quad (7)$$

Since π_0 is the standard Gaussian distribution, $\mathbb{E}[X_0] = 0$. The sample x_1 does not depend on x_0, so $\mathbb{E}[x_1] = x_1$. Thus,

$$x_1 - \mathbb{E}[v_\theta(X_t,t)] = 0. \quad (8)$$

Or in other words, under the assumption that the CFM loss has been minimized, the empirical distribution samples x_1 should approximate $\mathbb{E}[v_\theta(X_t,t)]$, which gives a way to distinguish the training data from the held-out data.

Monte Carlo attack: Given a test image x and N Monte Carlo (MC) samples $\varepsilon_n \sim \pi_0$, a Monte Carlo version of the naive test statistics is defined

$$T_{\text{mc}}(x,t) = \|x - \frac{1}{N}\sum_n \left[v_\theta(tx + (1-t)\varepsilon_n, t)\right]\|_2^2 \tag{9}$$

Similar to the naive method, $T_{\text{mc}}(x \in \mathcal{D}_{train}) < T_{\text{mc}}(x \in \mathcal{D}_{val})$, if the model is over-fit.

3.2 Complexity-Calibrated Test Criteria

Likelihood estimates for out-of-distribution data derived from deep generative models are biased towards the inherent complexity of the input image [34, 40]. Serrà et al. [40] demonstrated that likelihood scores are positively correlated to the input spatial complexity, often quantified via standard compression bitrates, both within distribution and out-of-distribution. For held-out dataset $x \in \mathcal{D}_{\text{data}} \setminus \mathcal{D}_{\text{train}}$, we observe the same phenomena in rectified flow (Appendix A).

Our proposed test statistics, T_{naive} and T_{mc}, also exhibit these biases. As illustrated in Fig. 2, visually simple samples systematically yield lower T_{mc} values, whereas complex samples produce spuriously high T_{mc} values across both the CIFAR-10, SVHN, and TinyImageNet datasets. It is thus valuable to decouple image complexity from actual memorization signal by calibrating our test statistics. We formalize our complexity-calibrated test statistic, $T_{\text{mc\_cal}}$:

$$T_{\text{mc\_cal}}(x,t) = \frac{T_{\text{mc}}(x,t)}{C(x)} \text{ with } x_t = tx + (1-t)\varepsilon. \tag{10}$$

The input complexity, denoted as $C(x)$, is flexibly defined; while its best theoretical definition, Kolmogorov complexity [27], is not computable in most practical settings, we could also use the byte length of the input image x compressed into a bitstream by any popular compression method. These form an upper bound for the uncomputable Kolmogorov complexity [27]. A parameterized calibration equation is $T_{\text{mc}}/\text{Complexity}^{\beta}$, where β is the single parameter.

This can be transformed to a log-linear relationship: $log(T_{\text{mc}}) \sim \beta log(\text{complexity}) + c$. We choose JPEG as the compression method, as it results in $\beta = 1$ across datasets, removing the need for an additional calibration step to select β. Additional discussion is relegated to Appendix B, as other options are available, but those options also result in additional calibration steps.

By integrating complexity calibration, we achieve substantial empirical gains in distinguishing training members ($x \in \mathcal{D}_{\text{train}}$) from non-members ($x \in \mathcal{D}_{\text{val}}$), as detailed in our Experiments section 5.3. Furthermore, we demonstrate in Appendix E that calibrating T_{mc} yields a significantly more pronounced performance improvement compared to applying an equivalent calibration to the naive baseline, T_{naive}.

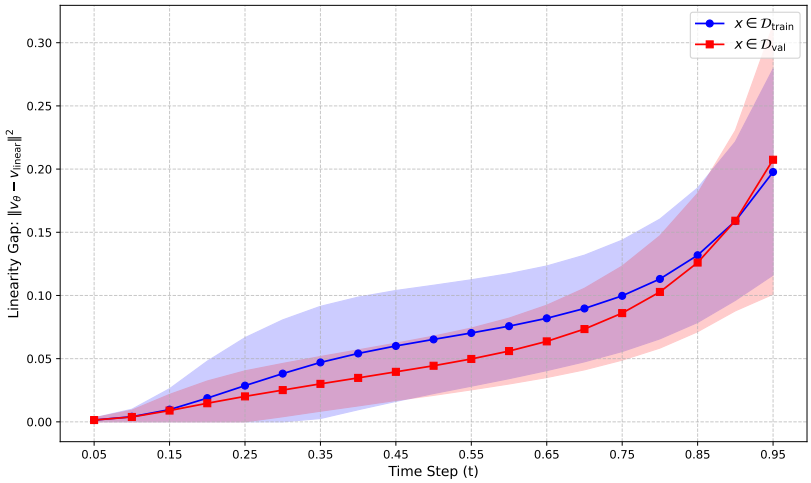


Fig. 3: The gap between v_θ and v_{linear} for $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. Solid lines denote the mean, while the shaded areas indicate $\pm$ standard deviation.

4 Memorization Dynamics across Different Timesteps t

The continuous timestep t, as an important input of our proposed test statistics, serves as a critical lens in reflecting overall model memorization. In our empirical evaluations (Fig. 1), we surprisingly discover that, under a uniform temporal sampling strategy $t \sim \mathcal{U}(0, 1)$, the model's vulnerabilities to MIA peaks distinctly at the midpoint, around $t = 0.5$. This phenomenon is not a spurious empirical artifact; rather, we show that it is intrinsically rooted in the underlying architectural properties of RF. In this section, we theoretically analyze the model memorization (is equivalent to the vulnerability to attacks) across varying timesteps t. Moreover, in Section 4.3, we propose a mitigation strategy designed to enhance robustness against catastrophic overfitting without compromising generative fidelity, as evaluated by the Fréchet Inception Distance (FID).

4.1 The Linearity Gap between v_θ and v_{linear} across t

In Rectified Flow, the neural network v_θ is trained to regress the complex, non-linear vector field. To study the generalization and memorization on rectified flow, we draw inspiration from recent theoretical studies that analyze complex diffusion denoisers through the linear filter, Wiener filter [19, 29, 33, 47], We use the Linear Minimum Mean Square Error (LMMSE) estimator as the linear regressor (v_{linear}) representing the absolute absence of non-linear feature extraction or memorization. Let the global mean of the data distribution be $\mu_{x_1} = \mathbb{E}[x_1]$ and its global covariance matrix be $\Sigma_{x_1} = \text{Cov}(x_1)$. Recap RF forward process, the state x_t is defined as:

$$x_t = tx_1 + (1 - t)x_0$$

where $x_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian noise independent of x_1. For non-linear regerssor v_θ, The input information is x_t and the target velocity vector is $v = x_1 - x_0$. To formulate the LMMSE baseline, we formulate it as a constrained optimization problem.

The LMMSE estimator seeks an optimal weight matrix $W^* \in \mathbb{R}^{D\times D}$ and bias vector $b^* \in \mathbb{R}^D$ that minimize the expected squared ℓ_2-norm of the prediction error:

$$W^*, b^* = \arg\min_{\mathbf{W},b} \mathbb{E}_{(v,x_t)} \left[\|v - (Wx_t + b)\|_2^2\right] \tag{11}$$

By solving this constrained objective via the orthogonality principle, we obtain the closed-form global linear predictor:

$$W^* = \Sigma_{vx_t}\Sigma_{x_tx_t}^{-1}, \quad b^* = \mathbb{E}[v] - W^*\mathbb{E}[x_t] \tag{12}$$

Substituting the time-dependent moments of the Rectified Flow forward process into this general solution (see Appendix F for detailed derivations of the covariance terms), we obtain the closed-form global linear baseline for any input $\boldsymbol{x}_t$:

$$v_{\text{linear}}(x_t) = \mu_{x_1} + (t\Sigma_{x_1} - (1-t)\mathbf{I})\left(t^2\Sigma_{x_1} + (1-t)^2\mathbf{I}\right)^{-1}(x_t - t\mu_{x_1}) \tag{13}$$

We define the *Linearity Gap* Δv as the mean squared error between the network's output and the analytical LMMSE prediction:

$$\Delta v(x_t, t) = v_\theta(x_t, t) - v_{\text{linear}}(x_t) \tag{14}$$

A lower $\|\Delta v(x_t, t)\|^2$ suggests a more linear behavior of the vector field $v_\theta(x_t, t)$. By empirically plotting $\|\Delta v\|^2$ across the trajectory $t \in [0, 1]$ in Fig. 3, it is clear that $\Delta v|_{x\in\mathcal{D}_{val}}$ is lower than $\Delta v|_{x\in\mathcal{D}_{train}}$ and the difference is maximized around intermediate timesteps $t \in [0.45, 0.55]$. This timeframe corresponds to that of peak MIA vulnerability in Fig.1. We explain this phenomenon in the next subsection.

4.2 Orthogonality between x_t and v at $t = 0.5$

Σ_{vx_t} is calculated as follows:

$$\Sigma_{v,x_t} = \mathbb{E}[(v - \mathbb{E}[v])(x_t - \mu)^T] \tag{15}$$

Previously, we explicitly compute μ_{x_1} and Σ_{x_1} on $\mathcal{D}_{train}$. For analytical tractability, we assume the data distribution is normalized such that $\mu_{x_1} = \mathbf{0}$ and the covariance $\Sigma_{x_1} = \boldsymbol{I}$. This assumption is reasonable as the data is normalized before being sent to the model. Therefore,

$$\Sigma_{vx_t} = \mathbb{E}\left[vx_t^\top\right] = \mathbb{E}\left[(x_1 - x_0)(tx_1 + (1-t)x_0)^\top\right] \tag{16}$$

Given the independence of the signal and noise for unseen data ($\mathbb{E}[x_0x_1^\top] = 0$), this expands to:

$$\Sigma_{vx_t} = t\mathbb{E}[x_1x_1^\top] - (1-t)\mathbb{E}[x_0x_0^\top] \tag{17}$$

Since x_0 is standard Gaussian ($\Sigma_{x_0} = \boldsymbol{I}$):

$$\Sigma_{x_t,v} = t\boldsymbol{I} - (1-t)\boldsymbol{I} = (2t-1)\boldsymbol{I} \tag{18}$$

Observation: From Eq. 18, it is evident that at precisely $t = 0.5$, the covariance $\Sigma_{x_{0.5},v} = \mathbf{0}$. At this timestep, the input $x_{0.5} = 0.5(x_1 + x_0)$ is statistically orthogonal to the target velocity $v = x_1 - x_0$.

At $t = 0.5$ the input provides zero linear information regarding the target and v_{linear} degenerates towards mean-prediction (substitute $t = 0.5$ to Eq. 13, $v_{\text{linear}}(x_{0.5}) = \mu_{x_1} = 0$). For training samples $x \in \mathcal{D}_{\text{train}}$, the intermediate timesteps (around $t = 0.5$) represent the primary regime where catastrophic memorization occurs. In this heavily mixed state, the model v_θ can no longer rely on the trivial linear structure of the input. Consequently, during the training phase, the network memorizes the sample-specific structures to minimize the vector field estimation error. At the inference stage, the independence assumption between the target x_1 and the model parameters θ is violated since the network leverages its overfitted weights to condition its prediction on the specific identity of training sample. Conversely, for unseen validation samples $x \in \mathcal{D}_{\text{val}}$, successful generalization relies entirely on whether these samples share these learned modes [19]. In the absence of such shared structures, the theoretically optimal predictor reduces to the baseline LMMSE estimator, v_{linear}, causing the network's output to fully collapse to this uninformative state. However, at the temporal boundaries ($t \to 0$ or $t \to 1$), the dynamics shift. Here, v_θ retains near-perfect visibility of either the pure noise or the original data.

Peak Memorization in Rectified Flows: Assume the integration timestep is uniformly sampled, $t \sim \mathcal{U}(0,1)$, and the target data variable x_1 is standardized such that its expectation $\mu_{x_1} = \mathbf{0}$ and covariance $\Sigma_{x_1} = \boldsymbol{I}$. Under these structural conditions, the expected memorization risk across the entire continuous trajectory is strictly upper-bounded by the memorization evaluated at the trajectory midpoint. That is, the maximal memorization occurs at $t = 0.5$.

In order to model more general cases where Σ_{x_1} departs from $\boldsymbol{I}$, we need to generalize Eq. 18 from a zero variance criterion to min variance, i.e., $\min_t \text{Tr}[t\Sigma_{x_1} + (1-t)\Sigma_{x_0}]$. The optimal t is the point at which the t-weighted mean of the eigenvalues of Σ_{x_1} and Σ_{x_0} is minimal. We demonstrate these cases in Appendix H.

4.3 U-shape Sampling Distribution for Timesteps t

To operationalize our insights regarding the memorization dynamics, we replace the conventional uniform sampling $t \sim \mathcal{U}(0,1)$ with a Symmetric Exponential distribution. This temporal prior explicitly assigns higher probability density to the boundary regions ($t \to 0$ and $t \to 1$) while suppressing the intermediate regime. Formally, for a given concentration parameter $\alpha > 0$, the probability density function (PDF) is defined as:

$$p(t;\alpha) = \frac{\alpha}{2(1-e^{-\alpha})}\left(e^{-\alpha t} + e^{-\alpha(1-t)}\right), \quad t \in [0,1] \tag{19}$$

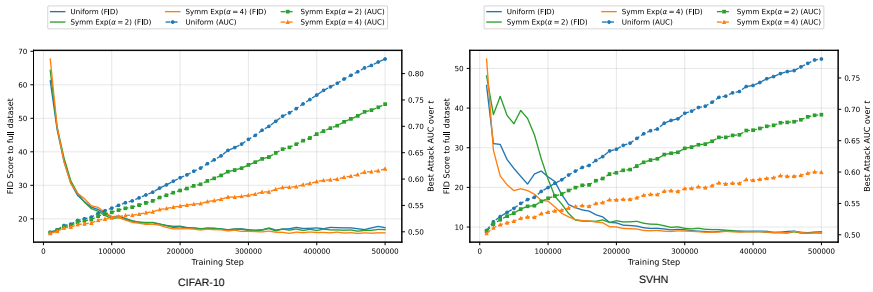
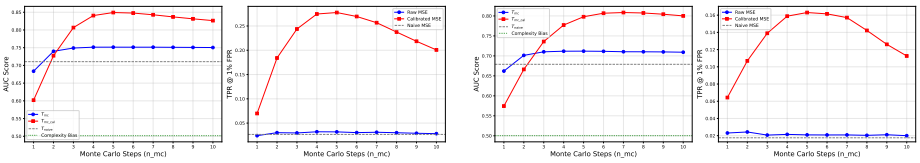


Fig. 4: Training dynamics (FID and peak AUC) when employing a Symmetric Exponential distribution for timestep sampling. Increasing the concentration parameter α (shown in order of Blue [none], Green, Orange [most]) effectively decelerates and suppresses model memorization while fully preserving generative fidelity.



(a) CIFAR-10 **(b)** SVHN

Fig. 5: The dynamic with increasing number of Monte Carlo samples on T_{mc} and $T_{\text{mc\_cal}}$

Note that as $\alpha \to 0$, $p(t;\alpha)$ degenerates to the standard uniform distribution. In practice, to ensure numerical stability, we linearly map the sampled t from $[0,1]$ to a tightly bounded interval $[t_{\min}, t_{\max}]$, where $t_{\min} = 10^{-5}$ and $t_{\max} = 1 - 10^{-5}$. While this technique was initially introduced by Lee et al. [26] based on the empirical observation that the loss of 2-Rectified Flow [32] is high at the boundaries and low in the middle. We justify this with full reasons, that is, from the perspective of model generalization and memorization. For completeness, we show $\int_0^1 p(t;\alpha)\,dt = 1$ in Appendix G.

While early stopping is a standard regularization technique to mitigate memorization, determining the optimal stopping criterion for generative models remains challenging. Practitioners typically wait for the FID to stabilize; however, as illustrated in Fig. 4, moderate memorization occurs before this convergence. Alternatively, while differential privacy guarantees protection against per-sample memorization, it inherently degrades generative fidelity. Hence, one need for techniques that decelerate the rate of memorization without sacrificing sample quality. In Section 5.4, we empirically demonstrate that our proposed modification successfully achieves this optimal balance.

Table 1: Correlation analysis between Input Complexity (Compressed Bitrate) and T_{mc} (before calibration). We report Pearson (ρ) and Spearman (r_s) correlation coefficients for both training and validation sets.

Split	CIFAR-10		SVHN		TinyImageNet64	
	Pearson ρ	Spearman r_s	Pearson ρ	Spearman r_s	Pearson ρ	Spearman r_s
Train	0.6030	0.5917	0.8520	0.8699	0.5275	0.5535
Val	0.8318	0.8561	0.8842	0.9060	0.7837	0.8111

5 Experiments

5.1 Setup

We tested our findings on three public datasets, CIFAR-10, SVHN, TinyImageNet, MSCOCO with resolution 32^2, 32^2, 64^2, 512^2 (see Appendix J for detailed dataset descriptions and train/validation splits). All experiments are executed on eight NVIDIA A40 GPUs.

Evaluation Metrics: We compute the proposed test statistics for both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. To quantify memorization, we evaluate the separability between the distributions of $T(x \in \mathcal{D}_{\text{train}})$ and $T(x \in \mathcal{D}_{\text{val}})$. This is measured by the Area Under the ROC Curve (AUC), where we assign binary labels to the samples (e.g., class 1 for $\mathcal{D}_{\text{train}}$ and class 0 for $\mathcal{D}_{\text{val}}$). We report TPR at a 1% FPR (TPR@1%FPR) because standard AUC can be misleadingly high even if the attack fails to identify any specific member with high confidence. For computational purposes, the number of queries to compute T_{naive}, T_{mc}, and $T_{\text{mc\_cal}}$ (#Query). We evaluated the generative image fidelity standard metric, Fréchet Inception Distance.

Dataset Configuration: We follow standard dataset construction protocols for MIA. For CIFAR-10, we evenly split the original training set into 25K/25K samples for $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$, respectively, and train the RF model for 500K steps on $\mathcal{D}_{\text{train}}$. Similarly, we construct 20K/20K splits for SVHN (trained for 500K steps), 50K/50K splits for TinyImageNet (trained for 100K steps) and 3K/3K for MSCOCO (trained for 45K steps). To maintain data diversity, we ensure a balanced class distribution across both $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$. All models are trained using the standard RF scripts from `https://github.com/lucidrains/rectified-flow-pytorch.git`.

5.2 Calibration Analysis

To demonstrate that standard test statistics are highly biased by input image complexity, we plot the likelihood estimation against input complexity in Appendix A, and T_{mc} against input complexity in the first row of Fig. 2. A corresponding correlation coefficient analysis is provided in Table 1. Because this inherent bias obscures the true extent of memorization, we decouple the

Table 2: MIA Performance comparison across three datasets. All values are reported in percentage (%). Δ represents the improvement over the method immediately above.

Method	CIFAR-10		SVHN		TinyImageNet	
	AUC↑	TPR@1%FPR↑	AUC↑	TPR@1%FPR↑	AUC↑	TPR@1%FPR↑
T_{naive}	70.98	2.70	68.04	1.89	72.33	4.10
T_{mc} (#mc=5)	75.12	3.05	70.92	1.54	76.44	5.33
Δ (vs. T_{naive})	+4.14	+0.35	+2.88	-0.35	+4.11	+1.23
$T_{\text{mc\_cal}}$ (#mc=5)	**84.89**	**27.88**	**79.43**	**16.46**	**92.96**	**50.03**
Δ (vs. T_{mc})	+9.77	+24.83	+8.51	+14.92	+16.52	+44.70

Table 3: Comparison of membership inference attack (MIA) performance on MSCOCO across different timesteps t, for a Rectified Flow model trained in the latent space of a VAE All values are reported in percentage (%). Δ represents the improvement over the method immediately above.

t	0.10	0.20	0.30	0.40	0.50	0.60	0.70	0.80
T_{naive}	57.94	70.63	75.50	**77.04**	76.14	73.16	68.94	48.46
T_{mc} (#mc=5)	59.11	73.15	80.25	85.14	**85.33**	83.45	75.63	64.37
Δ (vs. T_{naive})	+1.17	+2.52	+4.75	+8.10	+9.19	+10.29	+6.69	+15.91

test score from input complexity. As shown in the second row of Fig. 2, our calibrated test statistic ($T_{\text{mc\_cal}}$) achieves a significantly cleaner separation between $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$.

5.3 MIA Experiments

We initially evaluate the baseline statistic T_{naive} and observe that FM objective-based statistics tend to underestimate memorization. To validate our proposed T_{mc}, we evaluate its performance by increasing the number of Monte Carlo samples (N_{mc}) from 1 to 10. As illustrated in Fig. 5, T_{mc} underperforms T_{naive} at $N_{mc} = 1$, but surpasses it at $N_{mc} \geq 2$ and continues to improve. This behavior suggests that as we sample more noise vectors ε, $\mathbb{E}[\boldsymbol{x_0}]$ approaches zero, allowing the velocity network v_θ to predict the test image $\boldsymbol{x_1}$ more directly. We prescribe $N_{mc} = 5$ for subsequent experiments to balance computational overhead and AUC performance. Furthermore, because reconstruction error is heavily influenced by image complexity, adopting our calibrated statistic $T_{\text{mc\_cal}}$ yields a significant performance surge, particularly in TPR@1%FPR. The final results are summarized in Table 2.

For the dataset with high-resolution images such as MSCOCO, we trained an RF in the latent space of a Variational Autoencoder (VAE). We directly employ the publicly released autoencoder model `stabilityai/sd-vae-ft-mse` [3] without further fine-tuning. We swept different t for this experiment and the results

are summarized in Table 3. For T_{naive}, we observe a slight shift towards $t = 0$ for optimal t as the images exhibit spatial auto-correlation. For T_{mc}, optimal t perfectly remains on 0.5. VAE spaces have regularized covariances, so on average Σ_{x_1} will be closer to identity. Notably, $T_{\text{mc\_cal}}$ is not applicable for latent RF, but we can use the encoder/decoder metric for the calibration. This is described in [37].

5.4 Impact of Symmetric Exponential Time Sampling

To validate the theoretical insights from Section 4.3, we train additional RF models using a Symmetric Exponential Distribution for time sampling with $\alpha = 2$ and $\alpha = 4$. Larger values of α concentrate the sampling density toward the boundaries ($t \to 0$ and $t \to 1$). The results are presented in the third column of Fig. 1, with supplementary TinyImageNet experiments detailed in Appendix I. Note that we report the maximum AUC over t, which typically occurs near $t = 0.5$. The figures clearly demonstrate that introducing this sampling technique significantly decelerates memorization during training, while maintaining or even reducing the FID score. This empirical evidence directly supports our analysis and claims regarding memorization dynamics in Section 4.

6 Conclusion and Discussion

In this paper, we present a series of test statistics for MIA on RF and provide a theoretical justification for each refinement. Our analysis of memorization dynamics across timesteps t reveals that memorization is primarily concentrated at the midpoint of ODE trajectory. We justify this observation by establishing a link to the Linear Minimum Mean Square Error (LMMSE) estimator. Leveraging this insight, we adjust the sampling distribution of t, a modification that decelerates memorization without compromising generative image fidelity.

7 Acknowledgments

This work was supported in part by NSF 2321684 and the ARPA-H ALISS project.

Bibliography

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

[2] AI, I.: Ideogram 4. `https://ideogram.ai/blog/ideogram-4.0/` (2026), accessed: 2026-06-26

[3] AI, S.: sd-vae-ft-mse. `https://huggingface.co/stabilityai/sd-vae-ft-mse` (2023), hugging Face model card; autoencoder (VAE) decoder fine-tuned on LM-Humans / LAION training set. Accessed: 2026-06-26

[4] Albergo, M.S., Vanden-Eijnden, E.: Building normalizing flows with stochastic interpolants. arXiv preprint arXiv:2209.15571 (2022)

[5] Bonnaire, T., Urfin, R., Biroli, G., Mézard, M.: Why diffusion models don't memorize: The role of implicit dynamical regularization in training. arXiv preprint arXiv:2505.17638 (2025)

[6] van den Burg, G., Williams, C.: On memorization in probabilistic deep generative models. Advances in neural information processing systems **34**, 27916–27928 (2021)

[7] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE symposium on security and privacy (SP). pp. 1897–1914. IEEE (2022)

[8] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)

[9] Dockhorn, T., Cao, T., Vahdat, A., Kreis, K.: Differentially private diffusion models. arXiv preprint arXiv:2210.09929 (2022)

[10] Duan, J., Kong, F., Wang, S., Shi, X., Xu, K.: Are diffusion models vulnerable to membership inference attacks? In: International Conference on Machine Learning. pp. 8717–8730. PMLR (2023)

[11] Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: Forty-first international conference on machine learning (2024)

[12] Fredrikson, M., Jha, S., Ristenpart, T.: Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC conference on computer and communications security. pp. 1322–1333 (2015)

[13] Fu, W., Wang, H., Gao, C., Liu, G., Li, Y., Jiang, T.: A probabilistic fluctuation based membership inference attack for diffusion models. arXiv preprint arXiv:2308.12143 (2023)

[14] Goodier, J., Campbell, N.D.: Likelihood-based out-of-distribution detection with denoising diffusion probabilistic models. arXiv preprint arXiv:2310.17432 (2023)

[15] Grathwohl, W., Chen, R.T., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. arXiv preprint arXiv:1810.01367 (2018)
[16] Gu, X., Du, C., Pang, T., Li, C., Lin, M., Wang, Y.: On memorization in diffusion models. arXiv preprint arXiv:2310.02664 (2023)
[17] Hayes, J., Melis, L., Danezis, G., De Cristofaro, E.: Logan: Membership inference attacks against generative models. arXiv preprint arXiv:1705.07663 (2017)
[18] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
[19] Kadkhodaie, Z., Guth, F., Simoncelli, E.P., Mallat, S.: Generalization in diffusion models arises from geometry-adaptive harmonic representations. arXiv preprint arXiv:2310.02557 (2023)
[20] Karras, T., Aittala, M., Aila, T., Laine, S.: Elucidating the design space of diffusion-based generative models. Advances in neural information processing systems **35**, 26565–26577 (2022)
[21] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
[22] Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)
[23] Kong, F., Duan, J., Ma, R., Shen, H., Zhu, X., Shi, X., Xu, K.: An efficient membership inference attack for the diffusion model by proximal initialization. arXiv preprint arXiv:2305.18355 (2023)
[24] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
[25] Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: Flux.1 kontext: Flow matching for in-context image generation and editing in latent space (2025), `https://arxiv.org/abs/2506.15742`, accessed: 2026-06-26
[26] Lee, S., Lin, Z., Fanti, G.: Improving the training of rectified flows. Advances in neural information processing systems **37**, 63082–63109 (2024)
[27] Li, M., Vitányi, P., et al.: An introduction to Kolmogorov complexity and its applications, vol. 3. Springer (2008)
[28] Li, P., Li, Z., Zhang, H., Bian, J.: On the generalization properties of diffusion models. Advances in Neural Information Processing Systems **36**, 2097–2127 (2023)
[29] Li, X., Dai, Y., Qu, Q.: Understanding generalizability of diffusion models requires rethinking the hidden gaussian structure. Advances in neural information processing systems **37**, 57499–57538 (2024)
[30] Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. arXiv preprint arXiv:2210.02747 (2022)
[31] Lipman, Y., Havasi, M., Holderrieth, P., Shaul, N., Le, M., Karrer, B., Chen, R.T., Lopez-Paz, D., Ben-Hamu, H., Gat, I.: Flow matching guide and code. arXiv preprint arXiv:2412.06264 (2024)

[32] Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. arXiv preprint arXiv:2209.03003 (2022)
[33] Lukoianov, A., Yuan, C., Solomon, J., Sitzmann, V.: Locality in image diffusion models emerges from data statistics. arXiv preprint arXiv:2509.09672 (2025)
[34] Nalisnick, E., Matsukawa, A., Teh, Y.W., Gorur, D., Lakshminarayanan, B.: Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136 (2018)
[35] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., et al.: Reading digits in natural images with unsupervised feature learning. In: NIPS workshop on deep learning and unsupervised feature learning. vol. 2011, p. 4. Granada (2011)
[36] Pooladian, A.A., Ben-Hamu, H., Domingo-Enrich, C., Amos, B., Lipman, Y., Chen, R.T.: Multisample flow matching: Straightening flows with minibatch couplings. arXiv preprint arXiv:2304.14772 (2023)
[37] Rao, M., Qu, B., Moyer, D.: Latent diffusion inversion requires understanding the latent space. arXiv preprint arXiv:2511.20592 (2025)
[38] Rao, M., Qu, B., Moyer, D.: Score-based membership inference on diffusion models. arXiv preprint arXiv:2509.25003 (2025)
[39] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: 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)
[40] Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J.F., Luque, J.: Input complexity and out-of-distribution detection with likelihood-based generative models. arXiv preprint arXiv:1909.11480 (2019)
[41] Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems **32** (2019)
[42] Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456 (2020)
[43] Tong, A., Fatras, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., Bengio, Y.: Improving and generalizing flow-based generative models with minibatch optimal transport. arXiv preprint arXiv:2302.00482 (2023)
[44] Tramer, F., Boneh, D.: Differentially private learning needs better features (or much more data). arXiv preprint arXiv:2011.11660 (2020)
[45] Van Den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016)
[46] Yang, H., E, W.: Generalization error of gan from the discriminator's perspective. Research in the Mathematical Sciences **9**(1), 8 (2022)
[47] Yang, X., Zhou, D., Feng, J., Wang, X.: Diffusion probabilistic model made slim. In: Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition. pp. 22552–22562 (2023)

[48] Yoon, T., Choi, J.Y., Kwon, S., Ryu, E.K.: Diffusion probabilistic models generalize when they fail to memorize. In: ICML 2023 workshop on structured probabilistic inference {\&} generative modeling (2023)
[49] Zhai, S., Chen, H., Dong, Y., Li, J., Shen, Q., Gao, Y., Su, H., Liu, Y.: Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. Advances in Neural Information Processing Systems **37**, 74122–74146 (2024)