This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Smoothing the Score Function to Enhance Generalization in Diffusion Models

Xinyu Zhou Jiawei Zhang Stephen J. Wright
Department of Computer Sciences, University of Wisconsin–Madison
Madison, WI, USA
{xzhou555, jzhang2924, sjwright2}@wisc.edu

Abstract

*Diffusion models achieve remarkable generation quality, yet face a fundamental challenge known as **memorization**, where generated samples can replicate training samples exactly. We develop a theoretical framework to explain this phenomenon by showing that the empirical score function (the score function corresponding to the empirical distribution) is a weighted sum of the score functions of Gaussian distributions, in which the weights are sharp softmax functions. This structure causes individual training samples to dominate the score function, resulting in sampling collapse. In practice, approximating the empirical score function with a neural network can partially alleviate this issue and improve generalization. Our theoretical framework explains why: In training, the neural network learns a smoother approximation of the weighted sum, allowing the sampling process to be influenced by local manifolds rather than single points. Leveraging this insight, we propose two novel methods to further enhance generalization: (1) **Noise Unconditioning** enables each training sample to adaptively determine its score function weight to increase the effect of more training samples, thereby preventing single-point dominance and mitigating collapse. (2) **Temperature Smoothing** introduces an explicit parameter to control the smoothness. By increasing the temperature in the softmax weights, we naturally reduce the dominance of any single training sample and mitigate memorization. Experiments across multiple datasets validate our theoretical analysis and demonstrate the effectiveness of the proposed methods in improving generalization while maintaining high generation quality. Code and models are publicly available at https://github.com/XinyuZhou2001/smoothing_the_score.*

1. Introduction

Diffusion models [8, 21] have emerged as a leading framework in generative modeling, achieving state-of-the-art results across various applications, from text-to-image generation [4, 17] to discovery of protein structures [23, 25] and solving optimization problems [12, 27]. The core mechanism of diffusion models involves gradually perturbing data with Gaussian noise and learning to reverse this process through score-based denoising. But in high dimensions, the data distribution generally occupies a limited region under the Manifold Hypothesis [5], which means that the score matching [22] will be ineffective due to inaccurate gradient estimation in low-density regions [19]. Consequently, diffusion modeling has been proposed to estimate a series of marginal distributions (with wider support) via neural network approximations, ensuring that the score function can be accurately learned along the entire sampling path, and enabling reliable generation through iterative denoising.

However, a growing body of research [2, 16, 18, 24] reveals that although diffusion models can generate high-quality samples, some of these samples are identical to training examples. This problem is called ***memorization***. Memorization can be explained theoretically from the distribution evolution perspective [14], as follows. The Fokker -Planck equation for a diffusion process indicates that if we use the empirical score function to solve the reversed process, each sampled marginal distribution will be the same as its corresponding forward one. Thus, we will eventually go back to the training distribution and cannot produce any new samples once the neural network learns the empirical score function perfectly. Although this theory suggests that perfect learning of the objective function would limit the approach to replicating, novel samples are obtained in practice. This inconsistency motivates us to explore a fundamental question:

> *How can neural networks **partially** resolve the memorization issue?*

This question is crucial since it is directly related to how we can further improve the ***generalization*** properties of the diffusion approach.

The first thing to note is that the score function $s_\theta(x, \sigma)$ learned by a neural network will be slightly different from the empirical score function $s^*(x, \sigma)$. So, a study of the difference could be key to answering this question. To

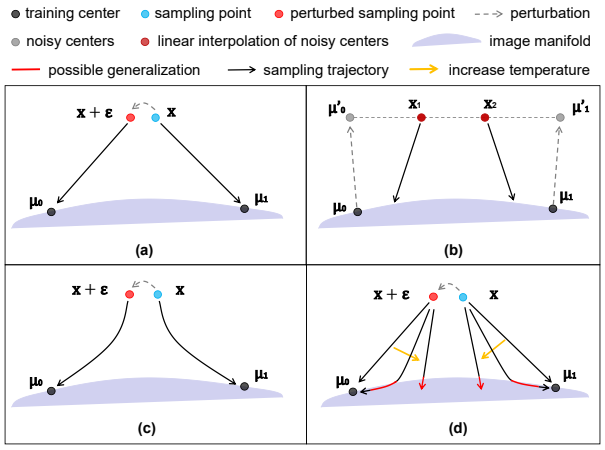


Figure 1. Illustration of sampling behaviors with different score functions. (a) Standard diffusion empirical score function leads to memorization, where trajectories collapse directly to training points. (b) The neural network-learned score function enables generalization. (Interpolation experiment) (c) The noise unconditioning allows smoother score function weights and delayed collapse. (d) The temperature-based score function promotes manifold exploration and prevents premature collapse to training points.

be more precise, let $p_{\text{data}}(x)$ be the ground-truth distribution, and $\{\mu_j\}_{j=1}^M$ be a set of i.i.d samples drawn from it. The empirical distribution can be represented as a delta distribution $p^*(x) \triangleq \frac{1}{M}\sum_{j=1}^M \delta(x-\mu_j)$, where we have $p_{\text{data}}(x) \approx p^*(x)$ if M is sufficiently large. The noise-perturbed distribution then will be a series of Gaussian mixture distributions

$$p_i^*(x) := \frac{1}{M}\sum_{j=1}^{M} \mathcal{N}(x;\mu_j,\sigma_i^2\mathbf{I}), \quad i \in [N],$$

where $\{\sigma_i\}_{i=1}^N$ is the noise schedule. For this setting we can explicitly calculate the empirical score functions

$$\nabla_x \log p_i^*(x) = -\sum_{j=1}^{M} w_j(x)\frac{x-\mu_j}{\sigma_i^2}, \quad i \in [N],$$

where $w_{ij}(x) = \frac{\mathcal{N}(x;\mu_j,\sigma_i^2\mathbf{I})}{\sum_{l=1}^M \mathcal{N}(x;\mu_l,\sigma_i^2\mathbf{I})}$ is the score function weight, which also represents the probability that x is generated by the Gaussian component centered at μ_j.

We obtain further insights by combining algebraic and geometric methods. In high dimensions ($x \in \mathbb{R}^d$ for large d), most of the probability mass of a Gaussian distribution concentrates in a thin shell. Given an empirical marginal distribution $p_i^*(x) = \frac{1}{M}\sum_{j=1}^M \mathcal{N}(x;\mu_j,\sigma_i^2\mathbf{I})$, we can interpret it geometrically as a collection of M ring shells. Each shell is centered at a training point μ_j with radius approximately $\sigma_i\sqrt{d}$ and thickness $3\sqrt{2}\sigma_i$. Since the sampling point x can be regarded as drawn from one of the marginal distributions p_i^*, it should be contained in at least one shell with the noise level σ_i. Moreover, at late stages of sampling, the value of σ_i is small enough that x is unlikely to be in the overlap of different shells, so the score function weight will be dominated by a single Gaussian component (Sec. 3.1). Thus, $\nabla_x \log p_i^*(x) \approx -\frac{x-\mu_j}{\sigma_i^2}$ for a particular value of j, so that subsequent denoising steps collapse to the corresponding center μ_j — the memorization issue.

We also find that even a small perturbation to x can significantly alter the empirical score function, as its weight concentration may transfer from one center $\boldsymbol{\mu}_j$ to another. This function is extremely sharp in regions where the shells overlap (Sec. 3.2), as illustrated in Fig. 1 (a). However, due to capacity and regularizations [1, 26], the neural network tends to learn a smoother function than the spiky empirical score function. This claim is also supported by the interpolation experiments in diffusion models [8, 21], illustrated in Fig. 1 (b). Specifically, two training points μ_0 and μ_1 are first corrupted with noise to obtain perturbed versions μ'_0 and μ'_1. The interpolated sampling points are then created by taking linear combinations $x_1 = \lambda_1\mu'_0 + (1-\lambda_1)\mu'_1$ and $x_2 = \lambda_2\mu'_0 + (1-\lambda_2)\mu'_1$, where $\lambda_1, \lambda_2 \in (0,1)$ are coefficients sufficiently away from the endpoints. After denoising sampling, the interpolated points will converge to the manifold between μ_0 and μ_1 rather than collapsing to either of them. Based on these observations, we believe that we can resolve the question posed earlier as follows:

> *The neural network can resolve the memorization since it smooths out the weights of the empirical score function. Thus, in the late sampling stages, the score function is determined by local manifolds approximated by several nearby training samples, rather than by a single training point.*

We justify this claim in the remainder of the paper.

Nevertheless, as the model capacity increases, the neural network is more able to capture the "spikiness" of the empirical score function. Therefore, if a collapse point is isolated or replicated in the dataset, memorization may still occur. These are several explanations as to why memorization consistently happens in large models and why memorized samples are typically either far from other training points or repeated in the dataset [6, 26]. It follows that by relaxing the objective constraints and smoothing the objective score function weights, we may be able to improve generalization (and decrease memorization). Based on the formula of the empirical score, we propose two approaches for smoothing.

Noise Unconditioning. In most diffusion models, the noise levels are designed as continuous, which means the shells pack the space. Therefore, we know from Sec. 3.1 that for the fixed center μ and position x, there is an "optimal shell" corresponding to some noise level σ_* that will dominate the score function weight over all noise levels.

However, in standard diffusion models, the noise level is fixed at each sampling stage, which means the sampling point may not lie on the optimal shells around most training points, leading to sub-optimal score function weights (much smaller) for these points. So, our first smoothing method is to *remove conditioning on noise*, allowing each training point to adaptively find its optimal shell at the given position x. This ensures that other training points can contribute to the score function, preventing a single point from dominating, and thus mitigating memorization. In addition, this noise unconditioning enables us to regard the sampling as a gradient ascent process, where the optimal solutions are the training points. In this case, even though the sampling may still collapse to the centers, the "collapse time" to a particular center μ_j will be delayed, since other centers still make significant contributions during the sampling. For illustration, compare Fig. 1 (c) to Fig. 1 (a).

Temperature Smoothing. Given that the score function weights $w_{ij}(x)$ are defined as softmax functions, another smoothing approach becomes apparent: Modify the definition of these weights by introducing a temperature parameter to smooth them. The effect of smoothing is to delay the collapse of the denoising process; see Fig. 1 (d). By using higher temperatures at smaller noise levels, the sampling will have chances to continuously explore the local manifold rather than collapsing to a single point. This enables the neural network to sample from combinations over local manifold containing nearby samples, making it more likely to generalize rather than memorize. If the temperature is not sufficiently high, collapse may still occur eventually, but early stopping may prevent full collapse while maintaining the exploration benefits (see the red regions in Fig. 1 (d)). However, we must also avoid excessively high temperatures as this would introduce influence from training points in an excessively large neighborhood, causing the generated point to deviate significantly from the image manifold.

Overall, we make the following contributions.

- We theoretically prove that memorization stems from the sharpness of empirical score functions, and experimentally verify that neural networks achieve generalization by implicitly smoothing the score-function weights across different centers, i.e., increasing the influence of samples other than the nearest one.
- We propose two novel methods to mitigate memorization: (1) *Noise Unconditioning*, a framework that unifies distributions across noise levels and is equivalent to explicit score matching on a unified Gaussian mixture, so that sampling can be interpreted as gradient ascent on this fixed objective, enabling optimization-based strategies for generation. (2) *Temperature Smoothing*, a plug-and-play modification of the training objective that explicitly controls the smoothness of score weights and improves generalization with only modest extra cost.
- Our explanation framework bridges the gap between diffusion theory and practice, providing theoretical foundations for many previously confusing experimental phenomena. This theoretical grounding may facilitate further advances in generative modeling.

2. Preliminaries

2.1. Forward SDEs and Backward ODEs

Stochastic differential equations (SDEs) provide a powerful framework for describing diffusion processes [21]. A general forward SDE in $\mathbb{R}^d$ can be written as:

$$\mathrm{d}x_t = \mu(x_t, t)\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\mathbf{w}_t, \tag{1}$$

where $x_t \in \mathbb{R}^d$ is the state of the system at time $t \in [0, T]$, $\mu(\cdot,\cdot)$ and $\sigma(\cdot)$ are the drift and diffusion coefficients respectively, and $\mathbf{w}_t$ represents the standard Brownian motion. This forward SDE describes how data is progressively perturbed into a noise-like distribution as t increases. A remarkable property of this SDE is the existence of an ordinary differential equation (ODE), dubbed the Probability Flow ODE [21], whose solution trajectories sampled at t are distributed according to:

$$\mathrm{d}x_t = \left[\mu(x_t, t) - \frac{1}{2}\sigma(t)^2 \nabla_x \log p_t(x)\right]\mathrm{d}t, \tag{2}$$

where $\nabla_x \log p_t(x)$ is the score function of the perturbed distribution at time t, namely the learning target of a neural network $s_\theta(x, t)$ via *score matching* [9, 19, 22].

2.2. Noise Unconditioning: ODE as Gradient Flow

Note that we let $\mu(x, t) = 0$ and $\sigma(t) = \sqrt{2t}$ following the variance exploding SDE [11, 21]. This choice ensures that we have $p_t(x) = p^*(x) \otimes \mathcal{N}(0, t^2\mathbf{I})$, i.e., $p_i^*(x) = p^*(x) \otimes \mathcal{N}(0, \sigma_i^2\mathbf{I})$, where $\otimes$ denotes the convolution operation. When we remove the noise conditioning, i.e., $s_\theta(x, t) \to s_\theta(x)$, all marginal distributions will be unified into one space. This unified distribution can be approximated by an $M \times N$ Gaussian mixture (Appendix A.1):

$$p_{\text{MN}}(x) = \frac{1}{ZMN}\sum_{j=1}^{M}\sum_{i=1}^{N}\lambda(\sigma_i)\mathcal{N}(x; \mu_j, \sigma_i^2\mathbf{I}), \tag{3}$$

where Z is the normalization constant and $\lambda(\sigma_i)$ represents the weighting function for different noise levels. Following standard practice in diffusion models [19, 21], we choose $\lambda(\sigma_i) = \sigma_i^2$ during training.

For the unified distribution p_{MN}, the regions close to training points (Gaussian centers) naturally exhibit the highest probability density in high dimensions. So sampling these centers can be regarded as maximizing $\log p_{\text{MN}}(x)$,

thereby the sampling process can be reformulated as gradient ascent flow of this fixed objective function $\log p_{\text{MN}}$:

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \frac{\eta}{2}\nabla_x \log p_{\text{MN}}(x), \tag{4}$$

where dt now parameterizes the optimization trajectory (rather than reversed diffusion time, since $\log p_{\text{MN}}(x)$ is time-independent) and $\eta > 0$ is the step size. This ODE represents continuous gradient flow on $\log p_{\text{MN}}(x)$ to maximize the likelihood.

3. From Memorization to Generalization

We now analyze why memorization occurs and how it is related to the sharpness of the score functions. Furthermore, we explain how the neural networks and our methods can generalize through comparing the expansiveness of the gradient ascent operators associated with their score functions.

3.1. Why Memorize?

For a Gaussian mixture distribution p_{MN} as in Eq. (3), we can explicitly calculate its score function:

$$\nabla_x \log p_{\text{MN}}(x) = -\sum_{j=1}^{M}\sum_{i=1}^{N} w_{ij}(x)\frac{x-\mu_j}{\sigma_i^2}.$$

where

$$w_{ij}(x) = \frac{\sigma_i^2 \mathcal{N}(x;\mu_j,\sigma_i^2\mathbf{I})}{\sum_{l=1}^{M}\sum_{k=1}^{N}\sigma_k^2\mathcal{N}(x;\mu_l,\sigma_k^2\mathbf{I})}.$$

and this score function weight can be written as a function $f(x,\mu_j,\sigma_i) \triangleq -(d-2)\ln\sigma_i - \frac{||x-\mu_j||^2}{2\sigma_i^2}$ with softmax:

$$\begin{aligned} w_{ij}(x) &= \frac{\exp(f(x,\mu_j,\sigma_i))}{\sum_{l=1}^{M}\sum_{k=1}^{N}\exp(f(x,\mu_l,\sigma_k))} \\ &= \text{Softmax}_{j,i}(f(x,\mu_j,\sigma_i)). \end{aligned}$$

We have the following conclusion for this score function:

Proposition 2. In high dimensions d, given a position x that is not far from any Gaussian centers, the score function $\nabla_x \log p_{\text{MN}}(x)$ is predominantly determined by the Gaussian component $\mathcal{N}(x;\mu_*,\sigma_*^2\mathbf{I})$, where μ_* and σ_* are defined as follows:

$$\mu_* = \arg\min_{\mu_j} \|x-\mu_j\|_2, \quad j=1,\dots,M,$$

$$\sigma_* = \arg\min_{\sigma_1,\sigma_2,\dots,\sigma_N} \left|\frac{\sigma_{opt}^2}{\sigma_i^2} - 1\right|.$$

where $\sigma_{opt} := \frac{\|x-\mu_*\|}{\sqrt{d-2}} \approx \frac{\|x-\mu_*\|}{\sqrt{d}}$.

The proposition can be proved according to two properties of the score function weights:

Property 1. σ **domination:** Given a certain position x and any fixed center μ_j, there exists an optimal noise level σ_j^* such that its corresponding Gaussian component dominates the score function weight over all other noise levels. Mathematically, σ_j^* is the global maximum of the function $f(\sigma) = -(d-2)\ln\sigma - \frac{||x-\mu_j||^2}{2\sigma^2}$, with $f(\sigma)$ demonstrating pronounced sensitivity to deviations from σ_j^*.

Under the noise scheduling that the boundaries of Gaussian shells touch (Appendix A.3), namely a kind of log-uniform distribution as in [20], the ratio of the maximum to second-maximum score function weights across noise levels is (derivation in Appendix A.4.1):

$$\gamma_\sigma = \frac{w_{ij}(x)}{w_{i+1,j}(x)} \approx e^{18(1-2\alpha)}, \quad \text{for some } \alpha \in [0, 0.5).$$

We give an example to build intuition about how large the ratio can be even for modest values of α. For the case of $\alpha = \frac{1}{3}$ (that is, σ_{opt} is only slightly closer to σ_i than to σ_{i+1}), we have $\gamma_\sigma \approx e^6 \approx 403$, so $w_{ij}(x)$ dominates its closest neighbor. Its dominance over all other weights will be even greater. Based on this property, we can then simplify the score function, representing the weight of a center μ_j by only using its optimal noise level σ_j^*:

$$w_j^*(x) := \frac{\sigma_j^{*2}\mathcal{N}(x;\mu_j,\sigma_j^{*2}\mathbf{I})}{\sum_{l=1}^{M}\sigma_l^{*2}\mathcal{N}(x;\mu_l,\sigma_l^{*2}\mathbf{I})}. \tag{5}$$

Property 2. μ **domination:** Given a position x, the score function weight of a center decreases exponentially as the distance between the center and x increases. We have the following conclusion for the case that two training points μ_j and μ_l are a similar distance from x, namely $\sigma_j^* = \sigma_l^*$ but $\|x-\mu_l\| \geq \|x-\mu_j\|$:

$$\gamma_\mu \approx \frac{w_j^*}{w_l^*} = \exp\left(\frac{\delta||\mu_j-\mu_l||^2}{\sigma_j^{*2}}\right) \tag{6}$$

where $\delta := \frac{||x-\mu_l||\cos\angle x\mu_l\mu_j}{||\mu_j-\mu_l||} - \frac{1}{2} \geq 0$ represents the relative distance advantage of μ_j over μ_l with respect to x (see derivation in Appendix A.4.2) and γ_μ captures the ratio of the weights corresponding to the two centers μ_j and μ_l. Typically, $\delta \lesssim O(\frac{1}{\sqrt{d}})$ maintains the condition $\sigma_j^* = \sigma_l^*$. However, during the sampling, the progressive reduction of σ_j^* causes γ_μ to grow. When σ_j^* is sufficiently small, the single center μ_j will dominate the score function weight, resulting in collapse and subsequent memorization.

3.2. Generalization Evaluation

From Property 2 in Sec. 3.1, we know that memorization occurs when σ is small because the closest training point dominates the score function weight, even when it is not much closer than other points. However, in practice, we generate

samples using *neural networks* to approximate the empirical score function, and this often results in novel samples. These observations suggest that the score function learned by the neural network must differ from the empirical score function. A natural explanation is that the neural network fits the score function quite well, but implicitly smooths its weights at small noise levels, preventing the dominance of individual training points and avoiding sampling collapse. Following this insight, we analyze the generalization ability of neural networks, standard diffusion, noise conditioning, and temperature smoothing by comparing the expansiveness of their sampling process at points where the shells from different training points overlap. We simplify the analysis by considering only the two training points closest to x, which is sufficient for our purposes because Proposition 2 indicates that the closest training samples have the greatest (dominant) influence on the flow.

Denoting the two closest training points by μ_0 and μ_1, assume that the sampling point x is similarly close to both of them, with the same shell radius $\sigma^*\sqrt{d}$ for both. We assume $\|x-\mu_0\| = C\|x-\mu_1\| = C\sigma^*\sqrt{d}$, where $C \geq 1$. We then slightly perturb x along the direction $(\mu_0 - \mu_1)$ to obtain position $\mathbf{y}$, such that $\|\mathbf{y}-\mu_1\| = C\|\mathbf{y}-\mu_0\| = C\sigma^*\sqrt{d}$. After one sampling step for both x and $\mathbf{y}$, we arrive at $\mathbf{x}'$ and $\mathbf{y}'$ respectively. We then evaluate the expansion factor

$$\gamma_{ex} := \frac{\|\mathbf{y}'-\mathbf{x}'\|}{\|\mathbf{y}-x\|}.$$

This ratio serves as an indicator of generalization because 1) memorization can induce an arbitrarily large ratio (formally unbounded in the limit); 2) a bounded ratio (i.e., a locally non-expansive sampling map, in the same sense as non-expansive updates in optimization) preserves the local connectivity of the sampled distribution, and thus serves as a quantitative proxy for generalization.

Empirical vs. Neural Network: Under the above assumption, for the empirical score function of standard diffusion, we have:

$$\gamma_{ex} \approx \frac{\|\eta(\mu_0-\mu_1)\|}{\|y-x\|}\left|1-\frac{2}{a+1}\right| \tag{7}$$

where η is the sampling stepsize and $a > 1$ represents the ratio of the dominant to subdominant score function weight (see derivation in Appendix A.5). This result still holds for $\|y-x\| \to 0$ if $d \to \infty$, which means that $a \to \infty$ and the ratio could be unbounded, showing poor generalization. However, the results of interpolation experiments in diffusion models [8, 21] show that $\|y'-x'\| \to 0$ if $\|y-x\| \to 0$. Our experiments (5.1.2) also show that the learned score function has a much smaller ratio than the empirical one.

Conditioning vs. Unconditioning. Under our assumption, the unconditioning modeling has the same ratio formula as in Eq. (7) but with a different coefficient a. From the property of σ-domination, we know that the unconditioning case has a smoother score function weight, leading to a smaller ratio γ_{ex} and therefore better generalization.

Temperature. We modify the weight calculations by introducing a temperature vector $T \in \mathbb{R}^N$ whose i-th component T_i contains the temperature for shell i. We define T_j^* to be the value of T_i for which $\sigma_i = \sigma_j^*$. Generalizing Eq. (5), the temperature-based score function weight is defined as:

$$w_j^*(x;T) = \frac{\exp\left(\frac{f(x,\mu_j,\sigma_j^*)}{T_j^*}\right)}{\sum_{l=1}^{M}\exp\left(\frac{f(x,\mu_l,\sigma_l^*)}{T_l^*}\right)}.$$

(We recover Eq. (5) by setting $T_i = 1, i = 1, 2, \ldots, N$.) As the T_i increase to ∞, the temperature smoothing reduces the dominance ratio a, resulting in smaller γ_{ex} and better generalization.

4. Learning the Smoothed Score Function

We next describe our methods for training the neural network score functions $\mathbf{s}_\theta$, parametrized by weight vector θ. Our noise scheduling follows the variance exploding SDE in [21], where the noise scale σ is sampled from a discrete set of N points that approximate a log-uniform distribution over the range $[\sigma_{\min}, \sigma_{\max}]$. The probability of selecting each σ_i is equally $\frac{1}{N}$. We denote sampling from this noise schedule as $\sigma_i \sim p_\sigma$.

4.1. Unconditioning Score Matching

The loss function in standard noise conditioning score-based neural networks (NCSN) [19] is

$$\mathcal{L}_{\text{c}} = \mathbb{E}_{\sigma_i\sim p_\sigma}\mathbb{E}_{\mu\sim p^*}\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma_i^2\mathbf{I})}\left[\frac{\sigma_i^2}{2}\left\|\mathbf{s}_\theta(x,\sigma_i)+\frac{x-\mu}{\sigma_i^2}\right\|_2^2\right]. \tag{8}$$

We have a similar objective function for the unconditioning modeling:

$$\mathcal{L}_{\text{u}} = \mathbb{E}_{\sigma_i\sim p_\sigma}\mathbb{E}_{\mu\sim p^*}\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma_i^2\mathbf{I})}\left[\frac{\sigma_i^2}{2}\left\|\mathbf{s}_\theta(x)+\frac{x-\mu}{\sigma_i^2}\right\|_2^2\right]. \tag{9}$$

The only difference is that we remove the noise as an input to the neural networks. Following the denoising score matching [22], we show in Appendix A.6 that optimizing this loss function is equivalent to performing explicit score matching for the distribution p_{MN}.

4.2. Temperature-based Score Matching

From Eq. (6) and the analysis in Section 3, we know that when the noise level is small, training samples close to x will dominate the score function weights. Therefore, we set a threshold σ_{collapse} to determine when we should introduce

temperature scaling. When $\sigma_i \leq \sigma_{\text{collapse}}$, for a noisy point x drawn from $\mathcal{N}(\mu, \sigma_i^2 \mathbf{I})$ where $\mu \sim p^*$, μ should be the closest training sample to x. We can then approximate the score function using the top-K nearest training samples to μ, denoted by $\mu_{(j)}$, $j = 1, 2, \ldots, K$, as follows:

$$\nabla_x \log p_{\text{MN}}(x; T) \approx \sum_{j=1}^{K} w^*_{(j)}(x; T) \left(-\frac{x - \mu_{(j)}}{\sigma^{*2}_{(j)}} \right) \quad (10)$$

where

$$w^*_{(j)}(x; T) = \frac{\exp\left(\frac{f(x, \mu_{(j)}, \sigma^*_{(j)})}{T^*_{(j)}}\right)}{\sum_{l=1}^{K} \exp\left(\frac{f(x, \mu_{(l)}, \sigma^*_{(l)})}{T^*_{(l)}}\right)}.$$

We thus define the score matching loss to be:

$$\mathcal{L}_{\text{T}} = \mathbb{E}_{x \sim p_{\text{MN}}} \left[\frac{1}{2} \left\| \mathbf{s}_\theta(x) - \nabla_x \log p_{\text{MN}}(x; T) \right\|_2^2 \right].$$

where practically we use Eq. (10) to approximate $\nabla_x \log p_{\text{MN}}(x; T)$.

To sum up, the complete temperature-based training loss adaptively combines both approaches based on the noise level of each sample:

$$\mathcal{L} = \mathbb{E}_{\sigma_i \sim p_\sigma} \left[\begin{cases} \mathcal{L}_{\text{u}}, & \text{if } \sigma_i > \sigma_{\text{collapse}} \\ \mathcal{L}_{\text{T}}, & \text{if } \sigma_i \leq \sigma_{\text{collapse}} \end{cases} \right] \quad (11)$$

Experimentally, we set $T_i = \max(\frac{\sigma_{\text{collapse}}}{\sigma_i}, 1)$. Note that temperature-based score matching is not rigorously learning the score function of p_{MN}, but its smoothed proxy version.

5. Experiments

Experimental Setup. To evaluate the effectiveness of our smoothing methods, we adopt VE-SDE [21] as the baseline, which is known as the first time-reverse SDE framework of diffusion models. For fair comparison, we reproduce the VE-SDE setup and apply our smoothing methods with the same implementations. For example, our unconditioning approach uses the same NN architecture as the baseline, with only the time embedding layers removed. (see Appendix B.1 for other details) In addition to the four commonly used datasets, CIFAR-10 [13], CelebA 64×64 [15], ImageNet 64×64 [3] and CelebA-HQ 256×256 [10], we also collect a small 64×64 dataset for ablations, consisting of 1,000 cats and 200 caracals. Caracals differ from pet cats in having long ears and ferocious faces (see Fig. 2), so that we can clearly see any possible generalizations.

KNN Spaces. The curvature of the image manifold in pixel space is large, so when K is large, or the temperature is high, the learned score function may guide samples deviating from the manifold. Therefore, we also apply KNN in the feature space (with significantly smaller curvature). Specifically, we use a pre-trained ResNet-18 first to map all samples to a 512-dimensional latent space, then calculate the KNN and return their indexes for later explicit score function calculation, still in the pixel space. We find that under the same temperature and K, the feature space-based KNN will lead to obviously better generation quality than the pixel space-based one. This phenomenon further verifies our former explanation that neural networks achieve generalization via learning a smoother score function determined by a local manifold. Note that in this section, all illustrated samples are obtained by pixel space-based KNN to support our claim. We show more results and comparison in the Appendix B.4.2.

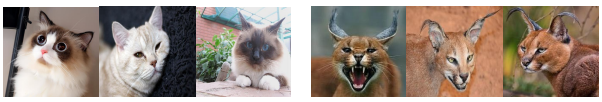


Figure 2. Illustration for the cat (**left**) and caracal (**right**). **Note** that there are no yellow or long-eared pet cats in the dataset.

Quality Metric. Besides calculating the Fréchet Inception Distance (FID) [7] between generated samples G and training samples $\mathcal{T}_{\text{tr}}$, we also calculate the FID between generated samples and test samples $\mathcal{T}_{\text{test}}$. We randomly select the same number of samples per class from the training set as in the test set to calculate FID. For CelebA, we randomly select 10k images from both the training set and test set.

Sampling for Unconditioning. We first try the existing samplers [21] by replacing the conditioning score function $s_\theta(x, t)$ with our unconditioning one $s_\theta(x)$. We find that SDE samplers still succeed, while ODE samplers may fail catastrophically. Under our framework, this occurs because at late sampling stages, the unconditional score function adapts to the true noise level $\sigma_{n*} := \frac{||x_n - \mu_*||}{\sqrt{d}}$, where x_n is current sample and μ_* is its closest training point. However, the sampling schedule still uses the predetermined noise level σ_n. In ODE samplers, we empirically observe that $\sigma_n \gg \sigma_{n*}$ as x_n approaches training data, creating a critical mismatch. We prove in Appendix A.7.1 that this mismatch causes ODE samplers to take excessively large steps $\propto \sigma_n^2 / \sigma_{n*}^2$, leading to catastrophic overshoot. In contrast, SDE samplers remain stable because their stochastic noise term creates a self-correcting mechanism that keeps σ_{n*} reasonably close to σ_n (Appendix A.7.2). To address this mismatch, one way is to fix the implicit noise level after every ODE step like the Predictor-Corrector (PC) sampling in [21]. Another way is to directly replacing the predetermined σ_n by σ_{n*}:

$$x_{n+1} = x_n + \alpha \sigma_{n*}^2 s_\theta(x_n)$$

where α controls the step size. Although computing σ_{n*} requires finding the nearest training point which may be

expensive for large-scale datasets, our unconditional score function enables bigger step sizes than standard diffusion models, resulting in significantly fewer function evaluations (NFE) that often compensate for this overhead. This is because the unconditioning score function is well-defined across the entire support of p_{MN}, and so avoids the error accumulation of large time steps in standard diffusion models.

5.1. Ablation Studies on Cat Caracal Dataset

5.1.1. Qualitative Experiments

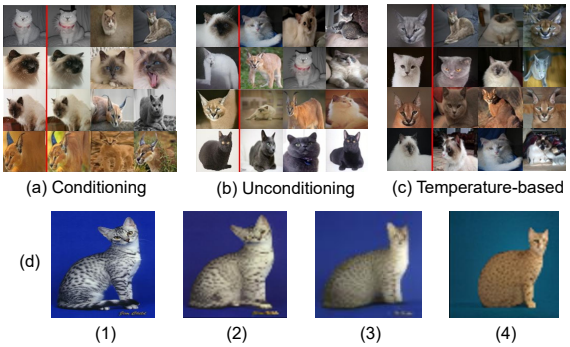


Figure 3. Qualitative comparison with different methods. (a, b, c) illustrates the generated images (left of red line) and their top-3 nearest neighbors of training samples in pixel space. (d) shows the generated images (2, 3, 4) and their corresponding closest training sample (1). (closest in both pixel and feature spaces)

Since the cat-caracal dataset uses only 1,200 images for training, there are relatively few regions where the empirical score function is sharp, i.e., where different training points compete. Therefore, a neural network with sufficient capacity can capture this sharpness, leading to pure memorization, as we see in Fig. 3 (a). However, either unconditioning modeling or temperature smoothing (here $T_i = 7/\sigma_i$ for $i = 1, 2, \ldots, N$ and K=10) can generalize well, illustrated by Fig. 3 (b) and (c) respectively. We can see obvious generalization in (c), e.g., the first generated cat, which has a caracal face but short ears and gray color. Meanwhile, in Fig. 3 (d), we compare the generated images with their same collapse training sample (1). Conditioning modeling (2) can only replicate the source image. In contrast, the unconditioning modeling (3) will not fully memorize, and temperature smoothing (4) shows significant generalization.

5.1.2. Quantitative Experiments

To empirically support our argument in Sec. 3.2, we calculate γ_{ex} under different noise levels using the learned score functions (NN Conditioning and Unconditioning), the empirical conditioning score function, the unconditioning (T=1), and the temperature-based ones (T=10,100,1000). Specifically, we randomly choose 10k data pairs in the cat-caracal dataset, and obtain x, y, x', y' as the assumption in Sec. 3.2, where we let $||x - y|| = 0.2\sigma^*$ and $\eta = 0.1$.

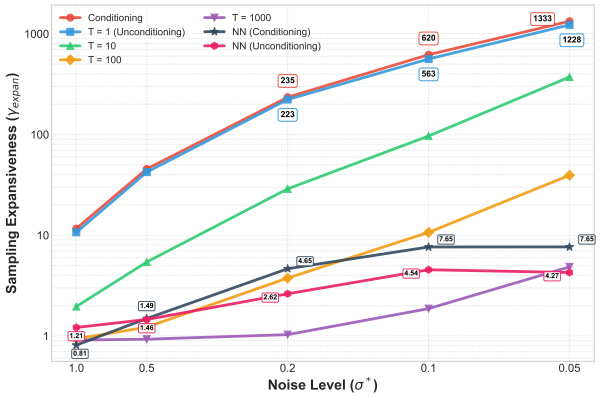


Figure 4. Sampling expansiveness ratio γ_{ex} at different noise levels. Empirical conditioning scores are highly expansive at low noise levels, while unconditioning, temperature smoothing, and NN approximation all yield much smaller γ_{ex}.

We can see the results in Fig. 4. They are consistent with our predictions of the behavior of γ_{ex} for the various score functions and their NN approximations. We observe that larger temperature values yield smaller expansion ratios γ_{ex} for the empirical score function variants, and that the NN behavior is similar to that of the temperature-smoothed empirical score function. Additionally, as the noise level decreases (toward the right of the chart), γ_{ex} tends to increase, as the points x and y collapse more rapidly toward their respective (different) training points. We note too the large difference in behavior between the unconditioned empirical score function and its NN approximation; the latter has much smaller values of γ_{ex} and therefore better generalization properties.

5.2. Sample Quality

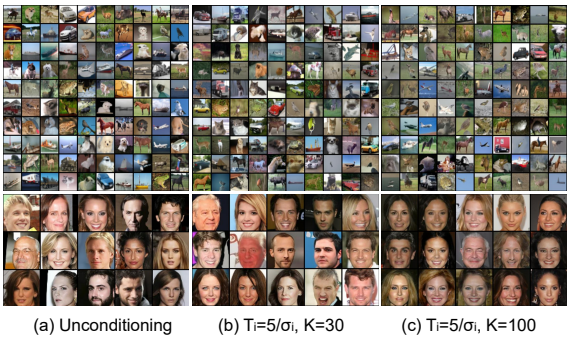


Figure 5. Generated images (top to bottom) for variants of SDE 1K NFE on datasets CIFAR-10 and CelebA. (pixel space KNN)

To further verify the effectiveness of our methods, we also implement experiments on some commonly used datasets. We first present ablation studies evaluated by FID on the CIFAR-10 and CelebA datasets. As shown in Tab. 1,

although the FID score slightly deteriorates at some settings, this is accompanied by observable and meaningful generalization. As illustrated in Fig. 5 (b) and (c), for example, the generated images on CelebA exhibit clear generalization: the red face in (b) and a similar countenance in (c), both of which indicate that the NN has captured the common characteristics of training samples in the local manifold. In practice, we can simply choose an appropriate temperature to achieve strong generalization while maintaining comparable quality.

Table 1. Ablation studies of our smoothing methods on the CIFAR-10 and CelebA datasets. "→" indicates the performance change from pixel space-based KNN to feature space-based KNN.

Method	FID($G, \mathcal{T}_{\text{tr}}$)	FID($G, \mathcal{T}_{\text{test}}$)
CIFAR-10 32×32		
SDE (PC) 1K NFE		
Conditioning	**6.49**	**6.56**
Unconditioning	7.33	7.34
$T_i = 1/\sigma_i$, K=30	8.32 → 8.08	8.25 → 8.02
$T_i = 1/\sigma_i$, K=100	8.67 → 7.97	8.61 → 7.89
$T_i = 5/\sigma_i$, K=30	8.56 → 8.12	8.60 → 8.15
$T_i = 5/\sigma_i$, K=100	13.25 → 8.35	13.41 → 8.30
$T_i = 7/\sigma_i$, K=30	21.28 → 8.26	21.34 → 8.22
$T_i = 7/\sigma_i$, K=100	50.81 → 7.96	51.08 → 7.98
ODE (PC) 1K NFE		
Conditioning	8.48	8.50
Unconditioning	8.84	8.88
$T_i = 1/\sigma_i$, K=30	8.75 → **8.45**	8.67 → **8.43**
CelebA 64×64		
SDE (PC) 1K NFE		
Conditioning	7.25	7.81
Unconditioning	**7.07**	**7.34**
$T_i = 5/\sigma_i$, K=30	9.58 → 8.78	9.32 → 8.54
$T_i = 5/\sigma_i$, K=100	36.63 → 9.47	35.01 → 9.10
$T_i = 10/\sigma_i$, K=30	9.88 → 8.64	9.48 → 8.56
$T_i = 10/\sigma_i$, K=100	61.97 → 8.40	60.91 → 8.19
ODE (PC) 1K NFE		
Conditioning	**7.65**	**7.87**
Unconditioning	7.71	7.88
$T_i = 5/\sigma_i$, K=30	9.36 → 8.67	9.50 → 8.80

Moreover, the results in Tab. 1 show that the feature space-based KNN consistently outperforms the pixel space-based KNN across all settings. Operating in feature space enables more aggressive smoothing (larger temperatures and K) while maintaining strong performance. For instance, on CIFAR-10 with $T_i = 7/\sigma_i$ and $K = 100$, pixel-space KNN collapses (FID increases to 50.81), whereas feature-space KNN still attains an FID of 7.96. This observation also supports our hypothesis that the neural network generalizes by smoothing the local manifolds, as lower local curvature (feature space) allows for stronger score smoothing without driving samples off the manifold. The consistent improvements suggest that feature space provides a more appropriate geometric structure for our smoothing methods. These results motivate us to extend our approach to latent diffusion models, which may further enhance generalization while preserving computational efficiency. In the Appendix, we present results on other high-resolution datasets. We also provide detailed proofs, training cost analysis, and extensive experiments to validate our theoretical approximations and claims.

6. Related Work

Memorization and Generalization. Recent work has revealed memorization behaviors in diffusion models, where generated samples can closely replicate training data [2, 18, 24]. [2] demonstrated systematic extraction of training images from large-scale models like Stable Diffusion [17], raising privacy and copyright concerns. [18] analyzed the connection between data duplication and memorization, proposing mitigation strategies through caption diversification. To investigate the causes of memorization, studies like [6] provided empirical evidence showing that the optimal solution of denoising score matching can only generate training data copies; [26] proposed the memorization–generalization dichotomy, arguing that generalization occurs when models fail to memorize, contrasting with overfitting in supervised learning [28]. Recent work [1] further elaborates on how neural networks learn the empirical score function: the model first fits an approximation to the population score and later converges to the highly irregular empirical score that leads to exact memorization.

7. Conclusion

This work establishes a theoretical framework for understanding memorization in diffusion models through the analysis of several variants of the empirical score function and its NN approximations. It reveals that memorization arises from the dominance of individual training samples in softmax-weighted score functions, while generalization emerges through smoothed weight distributions that enable local manifold exploration. By exploring the fundamental properties of the score function, we identify the cause of memorization and propose two smoothing methods — Noise Unconditioning and Temperature Smoothing — which elegantly control the concentration of the score function weight and effectively reduce memorization while maintaining high generation quality.

We developed mathematical analysis to explain a theory-practice inconsistency, and made experimental validations, revealing how neural networks achieve remarkable generalization. Our framework rationalizes previously observed memorization phenomena and unifies prior unconditioning approaches from an optimization perspective.

Acknowledgement

The first and third authors wish to acknowledge support from NSF grants 2224213 and 2023239. We also thank Qin Li and Sixu Li for valuable discussions.

References

[1] Tony Bonnaire, Raphaël Urfin, Giulio Biroli, and Marc Mézard. Why diffusion models don't memorize: The role of implicit dynamical regularization in training. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 2, 8

[2] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *USENIX Security Symposium (USENIX Security 23)*, 2023. 1, 8

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[4] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. *International Conference on Machine Learning (ICML)*, 2024. 1

[5] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016. 1

[6] Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. On memorization in diffusion models. *arXiv preprint arXiv:2310.02664*, 2023. 2, 8

[7] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 6

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1, 2, 5

[9] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 3

[10] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *International Conference on Learning Representations (ICLR)*, 2018. 6

[11] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 3

[12] Siddarth Krishnamoorthy, Satvik Mehul Mashkaria, and Aditya Grover. Diffusion models for black-box optimization. *International Conference on Machine Learning (ICML)*, 2023. 1

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[14] Sixu Li, Shi Chen, and Qin Li. A good score does not lead to a good generative model. *arXiv preprint arXiv:2401.04856*, 2024. 1

[15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015. 6

[16] Jie Ren, Yaxin Li, Shenglai Zeng, Han Xu, Lingjuan Lyu, Yue Xing, and Jiliang Tang. Unveiling and mitigating memorization in text-to-image diffusion models through cross attention. In *European Conference on Computer Vision (ECCV)*, 2024. 1

[17] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 8

[18] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 1, 8

[19] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. 1, 3, 5

[20] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 4

[21] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3, 5, 6

[22] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. 1, 3, 5

[23] Joseph L Watson, David Juergens, Nathaniel R Bennett, Brian L Trippe, Jason Yim, Helen E Eisenach, Woody Ahern, Andrew J Borst, Robert J Ragotte, Lukas F Milles, et al. De novo design of protein structure and function with rfdiffusion. *Nature*, 2023. 1

[24] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024. 1, 8

[25] Kevin E Wu, Kevin K Yang, Rianne van den Berg, Sarah Alamdari, James Y Zou, Alex X Lu, and Ava P Amini. Protein structure generation via folding diffusion. *Nature Communications*, 2024. 1

[26] TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K Ryu. Diffusion probabilistic models generalize when they fail to memorize. In *ICML Workshop on Structured Probabilistic Inference and Generative Modeling*, 2023. 2, 8

[27] Boyang Zhang, Zhiguo Wang, and Ya-Feng Liu. A gradient guided diffusion framework for chance constrained programming. *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. 1

[28] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *International Conference on Learning Representations (ICLR)*, 2017. 8