Understanding Generalizability of Diffusion Models Requires Rethinking the Hidden Gaussian Structure

Xiang Li[1], Yixiang Dai[1], Qing Qu[1]
[1]Department of EECS, University of Michigan,
forkobe@umich.edu, yixiang@umich.edu, qingqu@umich.edu

Abstract

In this work, we study the generalizability of diffusion models by looking into the hidden properties of the learned score functions, which are essentially a series of deep denoisers trained on various noise levels. We observe that as diffusion models transition from memorization to generalization, their corresponding nonlinear diffusion denoisers exhibit increasing linearity. This discovery leads us to investigate the linear counterparts of the nonlinear diffusion models, which are a series of linear models trained to match the function mappings of the nonlinear diffusion denoisers. Interestingly, these linear denoisers are approximately the optimal denoisers for a multivariate Gaussian distribution characterized by the empirical mean and covariance of the training dataset. This finding implies that diffusion models have the inductive bias towards capturing and utilizing the Gaussian structure (covariance information) of the training dataset for data generation. We empirically demonstrate that this inductive bias is a unique property of diffusion models in the generalization regime, which becomes increasingly evident when the model's capacity is relatively small compared to the training dataset size. In the case where the model is highly overparameterized, this inductive bias emerges during the initial training phases before the model fully memorizes its training data. Our study provides crucial insights into understanding the notable strong generalization phenomenon recently observed in real-world diffusion models.

1 Introduction

In recent years, diffusion models [1–4] have become one of the leading generative models, powering the state-of-the-art image generation systems such as Stable Diffusion [5]. To understand the empirical success of diffusion models, several works [6–12] have focused on their sampling behavior, showing that the data distribution can be effectively estimated in the reverse sampling process, assuming that the score function is learned accurately. Meanwhile, other works [13–18] investigate the learning of score functions, showing that effective approximation can be achieved with score matching loss under certain assumptions. However, these theoretical insights, grounded in simplified assumptions about data distribution and neural network architectures, do not fully capture the complex dynamics of diffusion models in practical scenarios. One significant discrepancy between theory and practice is that real-world diffusion models are trained only on a finite number of data points. As argued in [19], theoretically a perfectly learned score function over the empirical data distribution can only replicate the training data. In contrast, diffusion models trained on finite samples exhibit remarkable generalizability, producing high-quality images that significantly differ from the training examples. Therefore, a good understanding of the remarkable generative power of diffusion models is still lacking.

In this work, we aim to deepen the understanding of generalizability in diffusion models by analyzing the inherent properties of the learned score functions. Essentially, the score functions can be interpreted as a series of deep denoisers trained on various noise levels. These denoisers are then chained together to progressively denoise a randomly sampled Gaussian noise into its corresponding

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

clean image, thus, understanding the function mappings of these diffusion denoisers is critical to demystify the working mechanism of diffusion models. Motivated by the linearity observed in the diffusion denoisers of effectively generalized diffusion models, we propose to elucidate their function mappings with a linear distillation approach, where the resulting linear models serve as the linear approximations of their nonlinear counterparts.

Contributions of this work: Our key findings can be highlighted as follows:

- **Inductive bias towards Gaussian structures (Section 3).** Diffusion models in the *generalization regime* exhibit an inductive bias towards learning diffusion denoisers that are close (but not equal) to the optimal denoisers for a multivariate Gaussian distribution, defined by the empirical mean and covariance of the training data. This implies the diffusion models have the inductive bias towards capturing the Gaussian structure (covariance information) of the training data for image generation.
- **Model Capacity and Training Duration (Section 4)** We show that this inductive bias is most pronounced when the model capacity is relatively small compared to the size of the training data. However, even if the model is highly overparameterized, such inductive bias still emerges during early training phases, before the model memorizes its training data. This implies that early stopping can prompt generalization in overparameterized diffusion models.
- **Connection between Strong Generalization and Gaussian Structure (Section 5).** Lastly, we argue that the recently observed strong generalization [20] results from diffusion models learning certain common low-dimensional structural features shared across non-overlapping datasets. We show that such low-dimensional features can be partially explained through the Gaussian structure.

Relationship with Prior Arts. Recent research [20–24] demonstrates that diffusion models operate in two distinct regimes: (*i*) a memorization regime, where models primarily reproduce training samples and (*ii*) a generalization regime, where models generate high-quality, novel images that extend beyond the training data. In the generalization regime, a particularly intriguing phenomenon is that diffusion models trained on non-overlapping datasets can generate nearly identical samples [20]. While prior work [20] attributes this "*strong generalization*" effect to the structural inductive bias inherent in diffusion models leading to the optimal denoising basis (geometry-adaptive harmonic basis), our research advances this understanding by demonstrating diffusion models' inductive bias towards capturing the Gaussian structure of the training data. Our findings also corroborate with observations of earlier study [25] that the learned score functions of well-trained diffusion models closely align with the optimal score functions of a multivariate Gaussian approximation of the training data.

2 Preliminary

Basics of Diffusion Models. Given a data distribution $p_{\text{data}}(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{R}^d$, diffusion models [1–4] define a series of intermediate states $p(\boldsymbol{x}; \sigma(t))$ by adding Gaussian noise sampled from $\mathcal{N}(\mathbf{0}, \sigma(t)^2 \boldsymbol{I})$ to the data, where $\sigma(t)$ is a predefined schedule that specifies the noise level at time $t \in [0, T]$, such that at the end stage the noise mollified distribution $p(\boldsymbol{x}; \sigma(T))$ is indistinguishable from the pure Gaussian distribution. Subsequently, a new sample is generated by progressively denoising a random noise $\boldsymbol{x}_T \sim \mathcal{N}(\mathbf{0}, \sigma(T)^2 \boldsymbol{I})$ to its corresponding clean image $\boldsymbol{x}_0$. Following [4], this forward and backward diffusion process can be expressed with a probabilistic ODE:

$$d\boldsymbol{x} = -\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma(t))dt. \tag{1}$$

In practice the score function $\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma(t))$ can be approximated by

$$\nabla_{\boldsymbol{x}} \log p(\boldsymbol{x}; \sigma(t)) = (\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t)) - \boldsymbol{x})/\sigma(t)^2, \tag{2}$$

where $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$ is parameterized by a deep network with parameters $\boldsymbol{\theta}$ trained with the denoising score matching objective:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \boldsymbol{I})} \left[\|\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\epsilon}; \sigma(t)) - \boldsymbol{x}\|_2^2 \right]. \tag{3}$$

In the discrete setting, the reverse ODE in (1) takes the following form:

$$\boldsymbol{x}_{i+1} \leftarrow (1 - (t_i - t_{i+1})\frac{\dot{\sigma}(t_i)}{\sigma(t_i)})\boldsymbol{x}_i + (t_i - t_{i+1})\frac{\dot{\sigma}(t_i)}{\sigma(t_i)}\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_i; \sigma(t_i)), \tag{4}$$

where $\boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}, \sigma^2(t_0)\boldsymbol{I})$. Notice that at each iteration i, the intermediate sample $\boldsymbol{x}_{i+1}$ is the sum of the scaled $\boldsymbol{x}_i$ and the denoising output $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_i; \sigma(t_i))$. Obviously, the final sampled image is largely determined by the denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$. If we can understand the function mapping of these diffusion denoisers, we can demystify the working mechanism of diffusion models.

Optimal Diffusion Denoisers under Simplified Data Assumptions. Under certain assumptions on the data distribution $p_{\text{data}}(\boldsymbol{x})$, the optimal diffusion denoisers $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$ that minimize the score matching objective (3) can be derived analytically in closed-forms as we discuss below.

- *Multi-delta distribution of the training data.* Suppose the training dataset contains a finite number of data points $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$, a natural way to model the data distribution is to represent it as a multi-delta distribution: $p(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N} \delta(\boldsymbol{x} - \boldsymbol{y}_i)$. In this case, the optimal denoiser is

$$\mathcal{D}_{\text{M}}(\boldsymbol{x}; \sigma(t)) = \frac{\sum_{i=1}^{N} \mathcal{N}(\boldsymbol{x}; \boldsymbol{y}_i, \sigma(t)^2\boldsymbol{I})\boldsymbol{y}_i}{\sum_{i=1}^{N} \mathcal{N}(\boldsymbol{x}; \boldsymbol{y}_i, \sigma(t)^2\boldsymbol{I})}, \tag{5}$$

  which is essentially a softmax-weighted combination of the finite data points. As proved in [24], such diffusion denoisers $\mathcal{D}_{\text{M}}(\boldsymbol{x}; \sigma(t))$ can only generate exact replicas of the training samples, therefore they have no generalizability.

- *Multivariate Gaussian distribution.* Recent work [25] suggests modeling the data distribution $p_{\text{data}}(\boldsymbol{x})$ as a multivariate Gaussian distribution $p(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean $\boldsymbol{\mu}$ and the covariance $\boldsymbol{\Sigma}$ are approximated by the empirical mean $\boldsymbol{\mu} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{y}_i$ and the empirical covariance $\boldsymbol{\Sigma} = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{y}_i - \boldsymbol{\mu})(\boldsymbol{y}_i - \boldsymbol{\mu})^T$ of the training dataset. In this case, the optimal denoiser is:

$$\mathcal{D}_{\text{G}}(\boldsymbol{x}; \sigma(t)) = \boldsymbol{\mu} + \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T(\boldsymbol{x} - \boldsymbol{\mu}), \tag{6}$$

  where $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ is the SVD of the empirical covariance matrix, with singular values $\boldsymbol{\Lambda} = \text{diag}\,(\lambda_1, \cdots, \lambda_d)$ and $\tilde{\boldsymbol{\Lambda}}_{\sigma(t)} = \text{diag}\left(\frac{\lambda_1}{\lambda_1 + \sigma(t)^2} \cdots \frac{\lambda_d}{\lambda_d + \sigma(t)^2}\right)$. With this linear Gaussian denoiser, as proved in [25], the sampling trajectory of the probabilistic ODE (1) has close form:

$$\boldsymbol{x}_t = \boldsymbol{\mu} + \sum_{i=1}^{d} \sqrt{\frac{\sigma(t)^2 + \lambda_i}{\sigma(T)^2 + \lambda_i}} \boldsymbol{u}_i^T(\boldsymbol{x}_T - \boldsymbol{\mu})\boldsymbol{u}_i, \tag{7}$$

  where $\boldsymbol{u}_i$ is the i^{th} singular vector of the empirical covariance matrix. While [25] demonstrate that the Gaussian scores approximate learned scores at high noise variances, we show that they are nearly the best linear approximations of learned scores across a much wider range of noise variances.

Generalization vs. Memorization of Diffusion Models. As the training dataset size increases, diffusion models transition from the memorization regime—where they can only replicate its training images—to the generalization regime, where the they produce high-quality, novel images [17]. While memorization can be interpreted as an overfitting of diffusion models to the training samples, the mechanisms underlying the generalization regime remain less well understood. This study aims to explore and elucidate the inductive bias that enables effective generalization in diffusion models.

3 Hidden Linear and Gaussian Structures in Diffusion Models

In this section, we study the intrinsic structures of the learned score functions of diffusion models in the generalization regime. Through various experiments and theoretical investigation, we show that

Diffusion models in the generalization regime have inductive bias towards learning the Gaussian structures of the dataset.

Based on the *linearity* observed in diffusion denoisers trained in the generalization regime, we propose to investigate their intrinsic properties through a *linear distillation* technique, with which we train a series of linear models to approximate the nonlinear diffusion denoisers (Section 3.1). Interestingly, these linear models closely resemble the optimal denoisers for a multivariate Gaussian distribution characterized by the empirical mean and covariance of the training dataset (Section 3.2). This implies diffusion models have the inductive bias towards learning the Gaussian structure of the training dataset. We theoretically show that the observed Gaussian structure is the optimal solution to the denoising score matching objective under the constraint that the model is linear (Section 3.3). In the subsequent sections, although we mainly demonstrate our results using the FFHQ datasets, our findings are robust and extend to various architectures and datasets, as detailed in Appendix G.

3.1 Diffusion Models Exhibit Linearity in the Generalization Regime

Our study is motivated by the emerging linearity observed in diffusion models in the generalization regime. Specifically, we quantify the linearity of diffusion denoisers at various noise level $\sigma(t)$ by jointly assessing their ”Additivity” and ”Homogeneity” with a linearity score (LS) defined by the cosine similarity between $\mathcal{D}_{\boldsymbol{\theta}}(\alpha \boldsymbol{x}_1 + \beta \boldsymbol{x}_2; \sigma(t))$ and $\alpha \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1; \sigma(t)) + \beta \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_2; \sigma(t))$:

$$\mathtt{LS}(t) \;=\; \mathbb{E}_{\boldsymbol{x}_1, \boldsymbol{x}_2 \sim p(\boldsymbol{x}; \sigma(t))} \left[\left| \left\langle \frac{\mathcal{D}_{\boldsymbol{\theta}}(\alpha \boldsymbol{x}_1 + \beta \boldsymbol{x}_2; \sigma(t))}{\|\mathcal{D}_{\boldsymbol{\theta}}(\alpha \boldsymbol{x}_1 + \beta \boldsymbol{x}_2; \sigma(t))\|_2}, \frac{\alpha \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1; \sigma(t)) + \beta \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_2; \sigma(t))}{\|\alpha \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1; \sigma(t)) + \beta \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_2; \sigma(t))\|_2} \right\rangle \right| \right],$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2 \sim p(\boldsymbol{x}; \sigma(t))$, and $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}$ are scalars. In practice, the expectation is approximated with its empirical mean over 100 samples. A more detailed discussion on this choice of measuring linearity is deferred to Appendix A.

Following the EDM training configuration [4], we set the noise levels $\sigma(t)$ within the continuous range [0.002,80]. As shown in Figure 1, as diffusion models transition from the memorization regime to the generalization regime (increasing the training dataset size), the corresponding diffusion denoisers $\mathcal{D}_{\boldsymbol{\theta}}$ exhibit increasing linearity. This phenomenon persists across diverse datasets[1] as well as various training configurations[2]; see Appendix B for more details. This emerging linearity motivates us to ask the following questions:

- *To what extent can a diffusion model be approximated by a linear model?*
- *If diffusion models can be approximated linearly, what are the underlying characteristics of this linear approximation?*

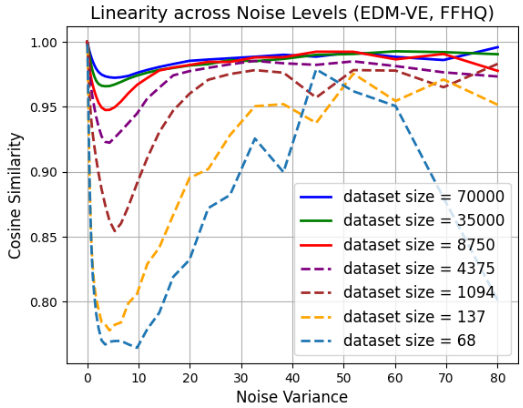


Figure 1: **Linearity scores of diffusion denoisers.** Solid and dashed lines depict the linearity scores across noise variances for models in the generalization and memorization regimes, respectively, where $\alpha = \beta = 1/\sqrt{2}$.

Investigating the Linear Structures via Linear Distillation. To address these questions, we investigate the hidden linear structure of diffusion denoisers through *linear distillation*. Specifically, for a given diffusion denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$ at noise level $\sigma(t)$, we approximate it with a linear function (with a bias term) such that:

$$\mathcal{D}_{\mathrm{L}}(\boldsymbol{x}; \sigma(t)) := \boldsymbol{W}_{\sigma(t)} \boldsymbol{x} + \boldsymbol{b}_{\sigma(t)} \;\approx\; \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t)),\ \forall \boldsymbol{x} \sim p(\boldsymbol{x}; \sigma(t)), \tag{8}$$

where the weight $\boldsymbol{W}_{\sigma(t)} \in \mathbb{R}^{d \times d}$ and bias $\boldsymbol{b}_{\sigma(t)} \in \mathbb{R}^{d}$ are learned by solving the following optimization problem with gradient descent:[3]

$$\min_{\boldsymbol{W}_{\sigma(t)}, \boldsymbol{b}_{\sigma(t)}} \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})} \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma(t)^2 \boldsymbol{I})} || \boldsymbol{W}_{\sigma(t)}(\boldsymbol{x} + \boldsymbol{\epsilon}) + \boldsymbol{b}_{\sigma(t)} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\epsilon}; \sigma(t)) ||_2^2. \tag{9}$$

If these linear models effectively approximate the nonlinear diffusion denoisers, analyzing their weights can elucidate the generation mechanism.

While diffusion models are trained on continuous noise variance levels within [0.002,80], we examine the 10 discrete sampling steps specified by the EDM schedule [4]: [80.0, 42.415, 21.108, 9.723, 4.06, 1.501, 0.469, 0.116, 0.020, 0.002] . These steps are considered sufficient for studying the diffusion mappings for two reasons: *(i)* images generated using these 10 steps closely match those generated with more steps, and *(ii)* recent research [30] demonstrates that the diffusion denoisers trained on similar noise variances exhibit analogous function mappings, implying that denoiser behavior at discrete variances represents their behavior at nearby variances.

[1] For example, FFHQ [26], CIFAR-10 [27], AFHQ [28] and LSUN-Churches [29].

[2] For example, EDM-VE, EDM-VP and EDM-ADM.

[3] For the following, the input is the vectorized version of the noisy image and the expectation is approximated using finite samples of input-output pairs $(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i, \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\epsilon}, \sigma(t)))$ with $i = 1, ..., N$ (see distillation details in Appendix C).

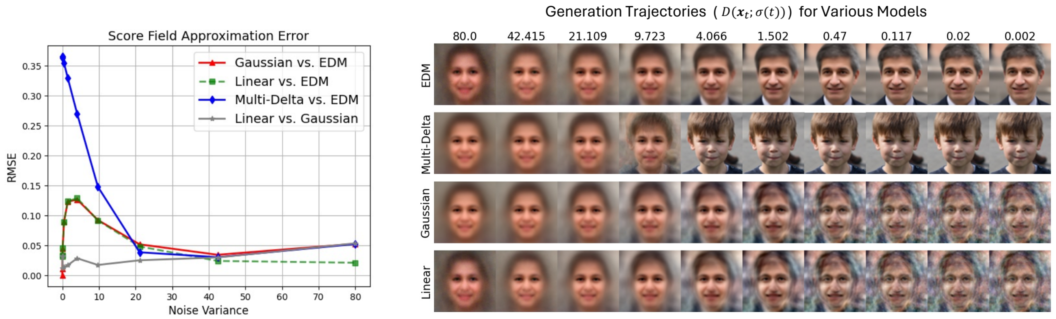


Figure 2: **Score field approximation error and sampling Trajectory.** The left and right figures demonstrate the score field approximation error and the sampling trajectories $\mathcal{D}(\boldsymbol{x}_t; \sigma(t)$ of actual diffusion model (EDM), Multi-Delta model, linear model and Gaussian model respectively. Notice that the curve corresponding to the Gaussian model almost overlaps with that of the linear model, suggesting they share similar funciton mappings.

After obtaining the linear models $\mathcal{D}_{\mathrm{L}}$, we evaluate their differences with the actual nonlinear denoisers $\mathcal{D}_{\boldsymbol{\theta}}$ with the score field approximation error, calculated using the expectation over the root mean square error (RMSE):

$$\text{Score-Difference}(t) := \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x}),\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0};\sigma(t)^2\boldsymbol{I})} \underbrace{\sqrt{\frac{\|\mathcal{D}_{\mathrm{L}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t)) - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t))\|_2^2}{d}}}_{\text{RMSE of a pair of randomly sampled } \boldsymbol{x} \text{ and } \boldsymbol{\epsilon}}, \quad (10)$$

where d represents the data dimension and the expectation is approximated with its empirical mean. While we present RMSE-based results in the main text, our findings remain consistent across alternative metrics, including NMSE, as detailed in Appendix G.

We perform linear distillation on well trained diffusion models operating in the generalization regime. For comprehensive analysis, we also compute the score approximation error between $\mathcal{D}_{\boldsymbol{\theta}}$ and: *(i)* the optimal denoisers for the multi-delta distribution $\mathcal{D}_{\mathrm{M}}$ defined as (5), and *(ii)* the optimal denoisers for the multivariate Gaussian distribution $\mathcal{D}_{\mathrm{G}}$ defined as (6). As shown in Figure 2, our analysis reveals three distinct regimes:

- *High-noise regime [20,80].* In this regime, only coarse image structures are generated (Figure 2(right)). Quantitatively, as shown in Figure 2(left), the distilled linear model $\mathcal{D}_{\mathrm{L}}$ closely approximates its nonlinear counterpart $\mathcal{D}_{\boldsymbol{\theta}}$ with RMSE below 0.05. Both Gaussian score $\mathcal{D}_{\mathrm{G}}$ and multi-delta score $\mathcal{D}_{\mathrm{M}}$ also achieve comparable approximation accuracy.
- *Low-noise regime [0.002,0.1].* In this regime, only subtle, imperceptible details are added to the generated images. Here, both $\mathcal{D}_{\mathrm{L}}$ and $\mathcal{D}_{\mathrm{G}}$ effectively approximate $\mathcal{D}_{\boldsymbol{\theta}}$ with RMSE below 0.05.
- *Intermediate-noise regime [0.1,20]:* This crucial regime, where realistic image content is primarily generated, exhibits significant nonlinearity. While $\mathcal{D}_{\mathrm{M}}$ exhibits high approximation error due to rapid convergence to training samples—a memorization effect theoretically proved in [24], both $\mathcal{D}_{\mathrm{L}}$ and $\mathcal{D}_{\mathrm{G}}$ maintain relatively lower approximation errors.

Qualitatively, as shown in Figure 2(right), despite the relatively high score approximation error in the intermediate noise regime, the images generated with $\mathcal{D}_{\mathrm{L}}$ closely resemble those generated with $\mathcal{D}_{\boldsymbol{\theta}}$ in terms of the overall image structure and certain amount of fine details. This implies *(i)* the underlying linear structure within the nonlinear diffusion models plays a pivotal role in their generalization capabilities and *(ii)* such linear structure is effectively captured by our distilled linear models. In the next section, we will explore this linear structure by examining the linear models $\mathcal{D}_{\mathrm{L}}$.

3.2 Inductive Bias towards Learning the Gaussian Structures

Notably, the Gaussian denoisers $\mathcal{D}_{\mathrm{G}}$ exhibit behavior strikingly similar to the linear denoisers $\mathcal{D}_{\mathrm{L}}$. As illustrated in Figure 2(left), they achieve nearly identical score approximation errors, particularly in the critical intermediate variance region. Furthermore, their sampling trajectories are remarkably similar (Figure 2(right)), producing nearly identical generated images that closely match those from the actual diffusion denoisers (Figure 3). These observations suggest that $\mathcal{D}_{\mathrm{L}}$ and $\mathcal{D}_{\mathrm{G}}$ share

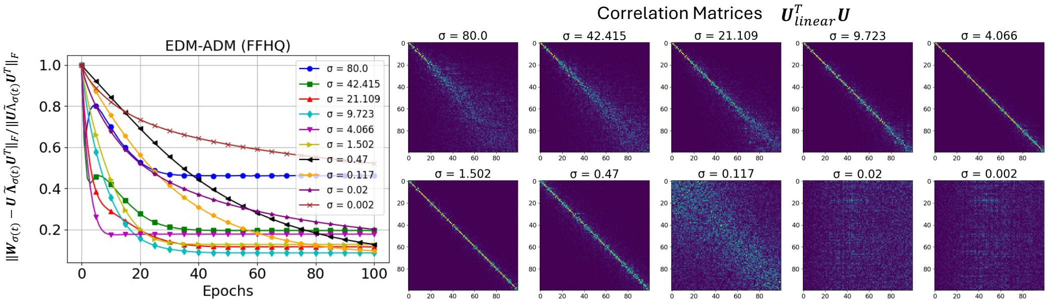


Figure 4: **Linear model shares similar function mapping with Gaussian model.** The left figure shows the difference between the linear weights and the Gaussian weights w.r.t. 100 training epochs of the linear distillation process for the 10 discrete noise levels. The right figure shows the correlation matrices between the first 100 singular vectors of the linear weights and Gaussian weights.

similar function mappings across various noise levels, leading us to hypothesize that the intrinsic linear structure underlying diffusion models corresponds to the Gaussian structure of the training data—specifically, its empirical mean and covariance. We validate this hypothesis by empirically showing that $\mathcal{D}_{\mathrm{L}}$ is close to $\mathcal{D}_{\mathrm{G}}$ through the following three complementary experiments:

- *Similarity in weight matrices.* As illustrated in Figure 4(left), $\boldsymbol{W}_{\sigma(t)}$ progressively converge towards $\boldsymbol{U}\tilde{\Lambda}_{\sigma(t)}\boldsymbol{U}^T$ throughout the linear distillation process, achieving small normalized MSE (less than 0.2) for most of the noise levels. The less satisfactory convergence behavior at $\sigma(t) = 80.0$ is due to inadequate training of the diffusion models at this particular noise level, which is minimally sampled during the training of actual diffusion models (see Appendix G.2 for more details).
- *Similarity in Score functions.* Furthermore, Figure 2(left, gray line) demonstrates that $\mathcal{D}_{\mathrm{L}}$ and $\mathcal{D}_{\mathrm{G}}$ maintain small score differences (RMSE less than 0.05) across all noise levels, indicating that these denoisers exhibit similar function mappings throughout the diffusion process.
- *Similarity in principal components.* As shown in Figure 4(right), for a wide noise range ($\sigma(t) \in [0.116, 80.0]$), the leading singular vectors of the linear weights $\boldsymbol{W}_{\sigma(t)}$ (denoted $\boldsymbol{U}_{Linear}$) align well with $\boldsymbol{U}$, the singular vectors of the Gaussian weights.[4] This implies that $\boldsymbol{U}$, representing the principal components of the training data, is effectively captured by the diffusion models. In the low-noise regime ($\sigma(t) \in [0.002, 0.116]$), however, $\mathcal{D}_{\boldsymbol{\theta}}$ approximates the identity mapping, leading to ambiguous singular vectors with minimal impact on image generation. Further analysis of $\mathcal{D}_{\boldsymbol{\theta}}$'s behavior in the low-noise regime is provided in Appendices D and F.1.

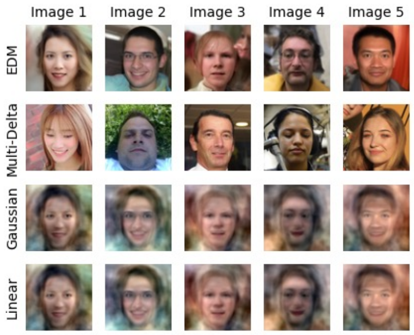


Figure 3: **Images sampled from various Models.** The figure shows the samples generated using different models starting from the same initial noises.

Since the optimization problem (9) is convex w.r.t. $\boldsymbol{W}_{\sigma(t)}$ and $\boldsymbol{b}_{\sigma(t)}$, the optimal solution $\mathcal{D}_{\mathrm{L}}$ represents the unique optimal linear approximation of $\mathcal{D}_{\boldsymbol{\theta}}$. Our analyses demonstrate that this optimal linear approximation closely aligns with $\mathcal{D}_{\mathrm{G}}$, leading to our central finding: diffusion models in the generalization regime exhibit an inductive bias (which we term as the Gaussian inductive bias) towards learning the Gaussian structure of training data. This manifests in two main ways: (*i*) In the high-noise variance regime, well-trained diffusion models learn $\mathcal{D}_{\boldsymbol{\theta}}$ that closely approximate the linear Gaussian denoisers $\mathcal{D}_{\mathrm{G}}$; (*ii*) As noise variance decreases, although $\mathcal{D}_{\boldsymbol{\theta}}$ diverges from $\mathcal{D}_{\mathrm{G}}$, $\mathcal{D}_{\mathrm{G}}$ remains nearly identical to the optimal linear approximation $\mathcal{D}_{\mathrm{L}}$, and images generated by $\mathcal{D}_{\mathrm{G}}$ retain structural similarity to those generated by $\mathcal{D}_{\boldsymbol{\theta}}$.

Finally, we emphasize that the Gaussian inductive bias only emerges in the generalization regime. By contrast, in the memorization regime, Figure 5 shows that $\mathcal{D}_{\mathrm{L}}$ significantly diverges from $\mathcal{D}_{\mathrm{G}}$, and both $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\mathrm{L}}$ provide considerably poorer approximations of $\mathcal{D}_{\boldsymbol{\theta}}$ compared to the generalization regime.

[4] For $\sigma(t) \in [0.116, 80.0]$, the less well recovered singular vectors have singular values close to 0, whereas those corresponding to high singular values are well recovered.

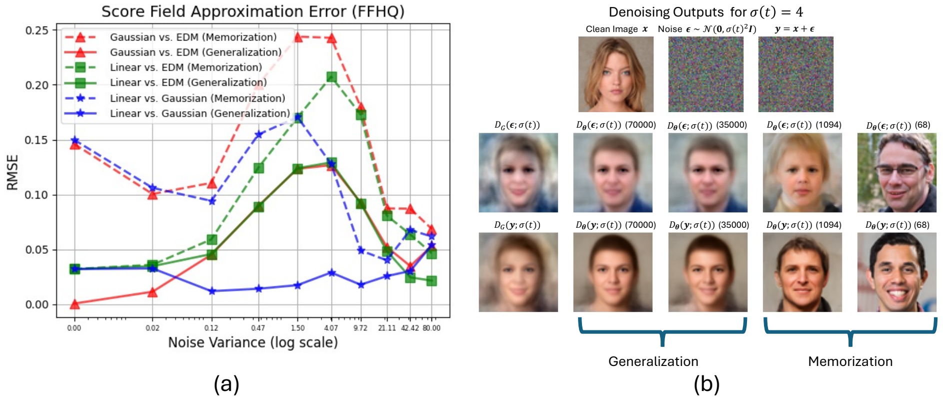


Figure 5: **Comparison between the diffusion denoisers in memorization and generalization regimes**. Figure(a) demonstrates that in the memorization regime (trained on small datasets of size 1094 and 68), $\mathcal{D}_{\mathrm{L}}$ significantly diverges from $\mathcal{D}_{\mathrm{G}}$, and both provide substantially poorer approximations of $\mathcal{D}_{\boldsymbol{\theta}}$ compared to the generalization regime (trained on larger datasets of size 35000 and 1094). Figure(b) qualitatively shows that the denoising outputs of $\mathcal{D}_{\boldsymbol{\theta}}$ closely match those of $\mathcal{D}_{\mathrm{G}}$ only in the generalization regime—a similarity that persists even when the denoisers process pure noise inputs.

3.3 Theoretical Analysis

In this section, we demonstrate that imposing linear constraints on diffusion models while minimizing the denoising score matching objective (3) leads to the emergence of Gaussian structure.

Theorem 1. *Consider a diffusion denoiser parameterized as a single-layer linear network, defined as $\mathcal{D}(\boldsymbol{x}_t; \sigma(t)) = \boldsymbol{W}_{\sigma(t)}\boldsymbol{x}_t + \mathbf{b}_{\sigma(t)}$, where $\boldsymbol{W}_{\sigma(t)} \in \mathbb{R}^{d\times d}$ is a linear weight matrix and $\mathbf{b}_{\sigma(t)} \in \mathbb{R}^d$ is the bias vector. When the data distribution $p_{data}(\boldsymbol{x})$ has finite mean $\boldsymbol{\mu}$ and bounded positive semidefinite covariance $\boldsymbol{\Sigma}$, the optimal solution to the score matching objective (3) is exactly the Gaussian denoiser defined in (6):*

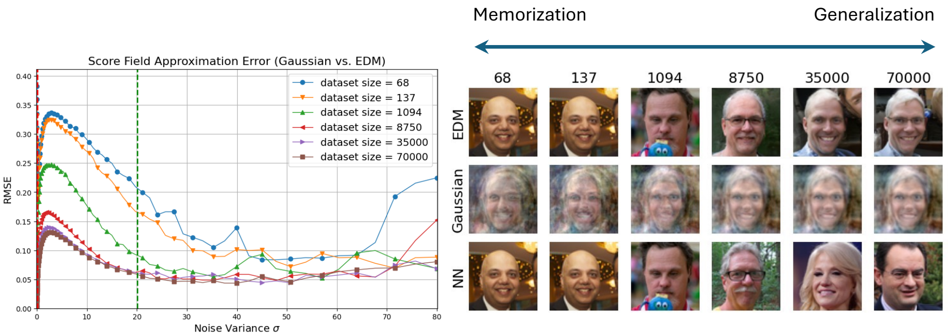
$$\mathcal{D}_{\mathrm{G}}(\boldsymbol{x}_t; \sigma(t)) \;=\; \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T(\boldsymbol{x}_t - \boldsymbol{\mu}) + \boldsymbol{\mu},$$

with $\boldsymbol{W}_{\sigma(t)} = \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T$ and $\mathbf{b}_{\sigma(t)} = \left(\boldsymbol{I} - \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T\right)\boldsymbol{\mu}$.

The detailed proof is postponed to Appendix E. This optimal solution corresponds to the classical Wiener filter [31], revealing that diffusion models naturally learn the Gaussian denoisers when constrained to linear architectures. To understand why highly nonlinear diffusion models operate near this linear regime, it is helpful to model the training data distribution as the multi-delta distribution $p(\boldsymbol{x}) = \frac{1}{N}\sum_{i=1}^{N}\delta(\boldsymbol{x} - \boldsymbol{y}_i)$, where $\{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$ is the finite training images. Notice that this formulation better reflects practical scenarios where only a finite number of training samples are available rather than the ground truth data distribution. Importantly, it is proved in [25] that the optimal denoisers $\mathcal{D}_{\mathrm{M}}$ in this case is approximately equivalent to $\mathcal{D}_{\mathrm{G}}$ for high noise variance $\sigma(t)$ and query points far from the finite training data. This equivalence explains the strong similarity between $\mathcal{D}_{\mathrm{G}}$ and D_{M} in the high-noise variance regime, and consequently, why $\mathcal{D}_{\boldsymbol{\theta}}$ and $\mathcal{D}_{\mathrm{G}}$ exhibit high similarity in this regime—deep networks converge to the optimal denoisers for finite training datasets.

However, this equivalence between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\mathrm{M}}$ breaks down at lower $\sigma(t)$ values. The denoising outputs of $\mathcal{D}_{\mathrm{M}}$ are convex combinations of training data points, weighted by a softmax function with temperature $\sigma(t)^2$. As $\sigma(t)^2$ decreases, this softmax function increasingly approximates an argmax function, effectively retrieving the training point $\boldsymbol{y}_i$ closest to the input $\boldsymbol{x}$. Learning this optimal solution requires not only sufficient model capacity to memorize the entire training dataset but also, as shown in [32], an exponentially large number of training samples. Due to these learning challenges, deep networks instead converge to local minima $\mathcal{D}_{\boldsymbol{\theta}}$ that, while differing from $\mathcal{D}_{\mathrm{M}}$, exhibit better generalization property. Our experiments reveal that these learned $\mathcal{D}_{\boldsymbol{\theta}}$ share similar function mappings with $\mathcal{D}_{\mathrm{G}}$. The precise mechanism driving diffusion models trained with gradient descent towards this particular solution remains an open question for future research.

Notably, modeling $p_{\text{data}}(\boldsymbol{x})$ as a multi-delta distribution reveals a key insight: while unconstrained optimal denoisers (5) perfectly capture the scores of the empirical distribution, they have no gen-



(a)

(b)

Figure 6: **Diffusion models learn the Gaussian structure when training dataset is large.** Models with a fixed scale (channel size 128) are trained across various dataset sizes. The left and right figures show the score difference and the generated images respectively. ”NN” denotes the nearest neighbor in the training dataset to the images generated by the diffusion models.

eralizability. In contrast, Gaussian denoisers, despite having higher score approximation errors due to the linear constraint, can generate novel images that closely match those produced by the actual diffusion models. This suggests that the generative power of diffusion models stems from the imperfect learning of the score functions of the empirical distribution.

4 Conditions for the Emergence of Gaussian Structures and Generalizability

In Section 3, we demonstrate that diffusion models exhibit an inductive bias towards learning denoisers that are close to the Gaussian denoisers. In this section, we investigate the conditions under which this bias manifests. Our findings reveal that this inductive bias is linked to model generalization and is governed by *(i)* the model capacity relative to the dataset size and *(ii)* the training duration. For additional results, including experiments on CIFAR-10 dataset, see Appendix F.

4.1 Gaussian Structures Emerge when Model Capacity is Relatively Small

First, we find that the Gaussian inductive bias and the generalization of diffusion models are heavily influenced by the relative size of the model capacity compared to the training dataset. In particular, we demonstrate that:

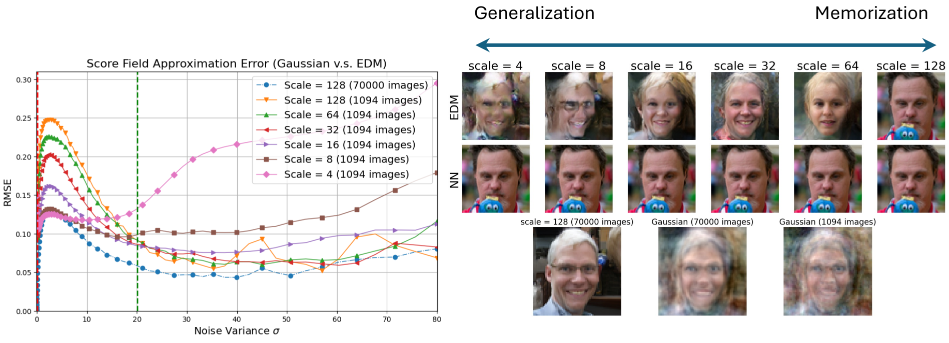
Diffusion models learn the Gaussian structures when the model capacity is relatively small compared to the size of training dataset.

This argument is supported by the following two key observations:

- ***Increasing dataset size prompts the emergence of Gaussian structure at fixed model scale.*** We train diffusion models using the EDM configuration [4] with a fixed channel size of 128 on datasets of varying sizes $[68, 137, 1094, 8750, 35000, 70000]$ until FID convergence. Figure 6(left) demonstrates that the score approximation error between diffusion denoisers $\mathcal{D}_{\boldsymbol{\theta}}$ and Gaussian denoisers $\mathcal{D}_{\mathrm{G}}$ decreases as the training dataset size grows, particularly in the crucial intermediate noise variance regime ($\sigma(t) \in [0.116, 20]$). This increasing similarity between $\mathcal{D}_{\boldsymbol{\theta}}$ and $\mathcal{D}_{\mathrm{G}}$ correlates with a transition in the models' behavior: from a memorization regime, where generated images are replicas of training samples, to a generalization regime, where novel images exhibiting Gaussian structure[5] are produced, as shown in Figure 6(b). This correlation underscores the critical role of Gaussian structure in the generalization capabilities of diffusion models.

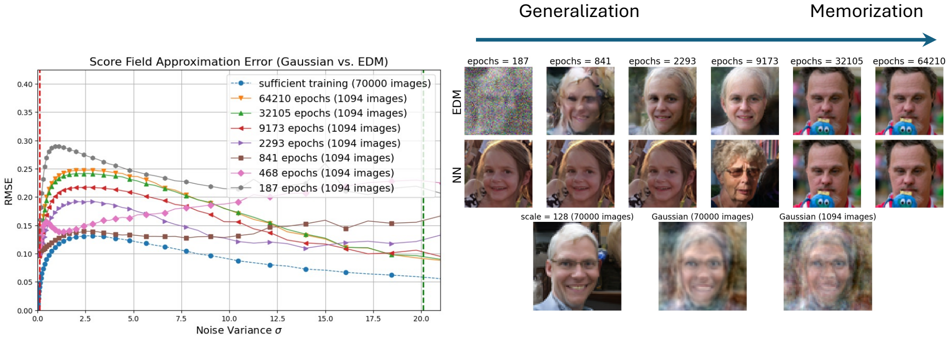
- ***Decreasing model capacity promotes the emergence of Gaussian structure at fixed dataset sizes.*** Next, we investigate the impact of model scale by training diffusion models with varying channel sizes $[4, 8, 16, 32, 64, 128]$, corresponding to $[64\text{k}, 251\text{k}, 992\text{k}, 4\text{M}, 16\text{M}, 64\text{M}]$ parameters, on a fixed training dataset of 1094 images. Figure 7(left) shows that in the intermediate noise variance regime ($\sigma(t) \in [0.116, 20]$), the discrepancy between $\mathcal{D}_{\boldsymbol{\theta}}$ and $\mathcal{D}_{\mathrm{G}}$ decreases with decreasing model scale, indicating that Gaussian structure emerges in low-capacity models. Figure 7(right) demonstrates that

[5] We use the term ”exhibiting Gaussian structure” to describe images that resemble those generated by Gaussian denoisers.



(a) (b)

Figure 7: **Diffusion model learns the Gaussian structure when model scale is small.** Models with different scales are trained on a fixed training dataset of 1094 images. The left and right figures show the score difference and the generated images respectively.



(a) (b)

Figure 8: **Diffusion model learns the Gaussian structure in early training epochs.** Diffusion model with same scale (channel size 128) is trained using 1094 images. The left and right figures shows the score difference and the generated images respectively.

this trend corresponds to a transition from data memorization to the generation of images exhibiting Gaussian structure. Here we note that smaller models lead to larger discrepancy between $\mathcal{D}_{\boldsymbol{\theta}}$ and $\mathcal{D}_{\mathrm{G}}$ in the high-noise regime. This phenomenon arises because diffusion models employ a bell-shaped noise sampling distribution that prioritizes intermediate noise levels, resulting in insufficient training at high noise variances, especially when model capacity is limited (see more details in Appendix F.2).

These two experiments collectively suggest that the inductive bias of diffusion models is governed by the relative capacity of the model compared to the training dataset size.

4.2 Overparameterized Models Learn Gaussian Structures before Memorization

In the overparameterized regime, where model capacity significantly exceeds training dataset size, diffusion models eventually memorize the training data when trained to convergence. However, examining the learning progression reveals a key insight:

Diffusion models learn the Gaussian structures with generalizability before they memorize.

Figure 8(a) demonstrates that during early training epochs (0-841), $\mathcal{D}_{\boldsymbol{\theta}}$ progressively converge to $\mathcal{D}_{\mathrm{G}}$ in the intermediate noise variance regime, indicating that the diffusion model is progressively learning the Gaussian structure in the initial stages of training. Notably. By epoch 841, the diffusion model generates images strongly resembling those produced by the Gaussian model, as shown in Figure 8(b). However, continued training beyond this point increases the difference between $\mathcal{D}_{\boldsymbol{\theta}}$ and $\mathcal{D}_{\mathrm{G}}$ as the model transitions toward memorization. This observation suggests that early stopping could be an effective strategy for promoting generalization in overparameterized diffusion models.

5 Connection between Strong Generalizability and Gaussian Structure

A recent study [20] reveals an intriguing ”strong generalization” phenomenon: diffusion models trained on large, non-overlapping image datasets generate nearly identical images from the same initial

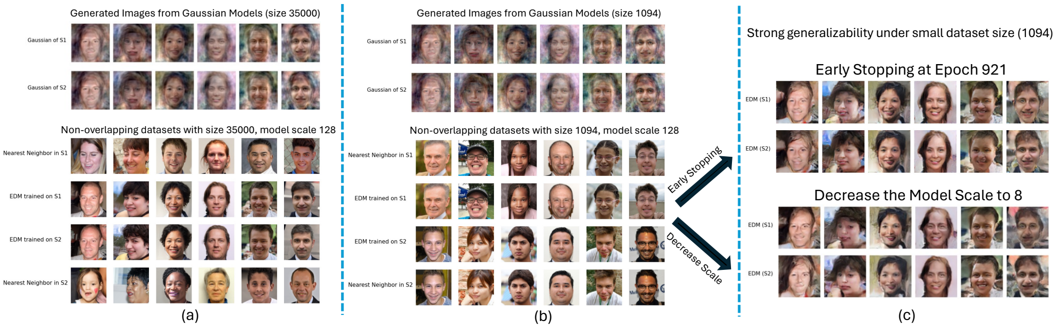

Generated Images from Gaussian Models (size 35000)
Gaussian of S1
Gaussian of S2
Non-overlapping datasets with size 35000, model scale 128
Nearest Neighbor in S1
EDM trained on S1
EDM trained on S2
Nearest Neighbor in S2
(a)
Generated Images from Gaussian Models (size 1094)
Gaussian of S1
Gaussian of S2
Non-overlapping datasets with size 1094, model scale 128
Nearest Neighbor in S1
EDM trained on S1
EDM trained on S2
Nearest Neighbor in S2
(b)
Early Stopping
Decrease Scale
Strong generalizability under small dataset size (1094)
Early Stopping at Epoch 921
EDM (S1)
EDM (S2)
Decrease the Model Scale to 8
EDM (S1)
EDM (S2)
(c)

Figure 9: **Diffusion models in the strong generalization regime generate similar images as the Gaussian models.** Figure(a) Top: Generated images of Gausisan models; Bottom: Generated images of diffusion models, with model scale 128; S1 and S2 each has 35000 non-overlapping images. Figure(b) Top: Generated images of Gausisan model; Bottom: Generated images of diffusion models in the memorization regime, with model scale 128; S1 and S2 each has 1094 non-overlapping images. Figure(c): Early stopping and reducing model capacity help transition diffusion models from memorization to generalization.

noise. While this phenomenon might be attributed to deep networks' inductive bias towards learning the "true" continuous distribution of photographic images, we propose an alternative explanation: rather than learning the complete distribution, deep networks may capture certain low-dimensional common structural features shared across these datasets and these features can be partially explained by the Gaussian structure.

To validate this hypothesis, we examine two diffusion models with channel size 128, trained on non-overlapping datasets S1 and S2 (35000 images each). Figure 9(a) shows that images generated by these models (bottom) closely match those from their corresponding Gaussian models (top), highlighting the Gaussian structure's role in strong generalization.

Comparing Figure 9(a)(top) and (b)(top), we observe that $\mathcal{D}_{\mathrm{G}}$ generates nearly identical images whether the Gaussian structure is calculated on a small dataset (1094 images) or a much larger one (35000 images). This similarity emerges because datasets of the same class can exhibit similar Gaussian structure (empirical covariance) with relatively few samples—just hundreds for FFHQ. Given the Gaussian structure's critical role in generalization, small datasets may already contain much of the information needed for generalization, contrasting previous assertions in [20] that strong generalization requires training on datasets of substantial size (more than 10^5 images). However, smaller datasets increase memorization risk, as shown in Figure 9(b). To mitigate this, as discussed in Section 4, we can either reduce model capacity or implement early stopping (Figure 9(c)). Indeed, models trained on 1094 and 35000 images generate remarkably similar images, though the smaller dataset yields lower perceptual quality. This similarity further demonstrates that small datasets contain substantial generalization-relevant information closely tied to Gaussian structure. Further discussion on the connections and differences between our work and [20] are detailed in Appendix H.

6 Discussion

In this study, we empirically demonstrate that diffusion models in the generalization regime have the inductive bias towards learning diffusion denoisers that are close to the corresponding linear Gaussian denoisers. Although real-world image distributions are significantly different from Gaussian, our findings imply that diffusion models have the bias towards learning and utilizing low-dimensional data structures, such as the data covariance, for image generation. However, the underlying mechanism by which the nonlinear diffusion models, trained with gradient descent, exhibit such linearity remains unclear and warrants further investigation.

Moreover, the Gaussian structure only partially explains diffusion models' generalizability. While models exhibit increasing linearity as they transition from memorization to generalization, a substantial gap persists between the linear Gaussian denoisers and the actual nonlinear diffusion models, especially in the intermediate noise regime. As a result, images generated by Gaussian denoisers fall short in perceptual quality compared to those generated by the actual diffusion models especially for complex dataset such as CIFAR-10. This disparity highlights the critical role of nonlinearity in high-quality image generation, a topic we aim to investigate further in future research.

Data Availability Statement

The code and instructions for reproducing the experiment results will be made available in the following link: https://github.com/Morefre/Understanding-Generalizability-of-Diffusion-Models-Requires-Rethinking-the-Hidden-Gaussian-Structure.

Acknowledgment

We acknowledge funding support from NSF CAREER CCF-2143904, NSF CCF-2212066, NSF CCF- 2212326, NSF IIS 2312842, NSF IIS 2402950, ONR N00014-22-1-2529, a gift grant from KLA, an Amazon AWS AI Award, and MICDE Catalyst Grant. We also acknowledge the computing support from NCSA Delta GPU [33]. We thank Prof. Rongrong Wang (MSU) for fruitful discussions and valuable feedbacks.

References

[1] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[3] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*.

[4] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems*, 35:26565–26577, 2022.

[5] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[6] Sitan Chen, Sinho Chewi, Jerry Li, Yuanzhi Li, Adil Salim, and Anru R Zhang. Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions. In *International Conference on Learning Representations*, 2023.

[7] Valentin De Bortoli. Convergence of denoising diffusion models under the manifold hypothesis. *Transactions on Machine Learning Research*, 2022.

[8] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence for score-based generative modeling with polynomial complexity. *Advances in Neural Information Processing Systems*, 35:22870–22882, 2022.

[9] Holden Lee, Jianfeng Lu, and Yixin Tan. Convergence of score-based generative modeling for general data distributions. In *International Conference on Algorithmic Learning Theory*, pages 946–985. PMLR, 2023.

[10] Yuchen Wu, Yuxin Chen, and Yuting Wei. Stochastic runge-kutta methods: Provable acceleration of diffusion models. *arXiv preprint arXiv:2410.04760*, 2024.

[11] Gen Li, Yuting Wei, Yuejie Chi, and Yuxin Chen. A sharp convergence theory for the probability flow odes of diffusion models. *arXiv preprint arXiv:2408.02320*, 2024.

[12] Zhihan Huang, Yuting Wei, and Yuxin Chen. Denoising diffusion probabilistic models are optimally adaptive to unknown low dimensionality. *arXiv preprint arXiv:2410.18784*, 2024.

[13] Minshuo Chen, Kaixuan Huang, Tuo Zhao, and Mengdi Wang. Score approximation, estimation and distribution recovery of diffusion models on low-dimensional data. In *International Conference on Machine Learning*, pages 4672–4712. PMLR, 2023.

[14] Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

[15] Kulin Shah, Sitan Chen, and Adam Klivans. Learning mixtures of gaussians using the ddpm objective. *Advances in Neural Information Processing Systems*, 36:19636–19649, 2023.

[16] Hugo Cui, Eric Vanden-Eijnden, Florent Krzakala, and Lenka Zdeborova. Analysis of learning a flow-based generative model from limited sample complexity. In *The Twelfth International Conference on Learning Representations*, 2023.

[17] Huijie Zhang, Jinfan Zhou, Yifu Lu, Minzhe Guo, Liyue Shen, and Qing Qu. The emergence of reproducibility and consistency in diffusion models. In *Forty-first International Conference on Machine Learning*, 2024.

[18] Peng Wang, Huijie Zhang, Zekai Zhang, Siyi Chen, Yi Ma, and Qing Qu. Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint arXiv:2409.02426*, 2024.

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

[20] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representation. In *The Twelfth International Conference on Learning Representations*, 2023.

[21] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[22] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 36:47783–47803, 2023.

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

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

[25] Binxu Wang and John J Vastola. The hidden linear structure in score-based models and its application. *arXiv preprint arXiv:2311.10892*, 2023.

[26] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

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

[28] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8188–8197, 2020.

[29] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

[30] Hyojun Go, Yunsung Lee, Seunghyun Lee, Shinhyeok Oh, Hyeongdon Moon, and Seungtaek Choi. Addressing negative transfer in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

[31] Mallat Stéphane. Chapter 11 - denoising. In Mallat Stéphane, editor, *A Wavelet Tour of Signal Processing (Third Edition)*, pages 535–610. Academic Press, Boston, third edition edition, 2009.

[32] Chen Zeno, Greg Ongie, Yaniv Blumenfeld, Nir Weinberger, and Daniel Soudry. How do minimum-norm shallow denoisers look in function space? *Advances in Neural Information Processing Systems*, 36, 2024.

[33] Timothy J Boerner, Stephen Deems, Thomas R Furlani, Shelley L Knuth, and John Towns. Access: Advancing innovation: Nsf's advanced cyberinfrastructure coordination ecosystem: Services & support. In *Practice and Experience in Advanced Research Computing*, pages 173–176. 2023.

[34] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[35] DP Kingma. Adam: a method for stochastic optimization. In *Int Conf Learn Represent*, 2014.

[36] Alfred O. Hero. Statistical methods for signal processing. 2005.

[37] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, 2008.

[38] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[40] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[41] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Atlanta, GA, 2013.

[42] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[43] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[44] Sreyas Mohan, Zahra Kadkhodaie, Eero P Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. In *International Conference on Learning Representations*.

[45] Hila Manor and Tomer Michaeli. On the posterior distribution in denoising: Application to uncertainty quantification. In *The Twelfth International Conference on Learning Representations*.

[46] Siyi Chen, Huijie Zhang, Minzhe Guo, Yifu Lu, Peng Wang, and Qing Qu. Exploring low-dimensional subspaces in diffusion models for controllable image editing. *arXiv preprint arXiv:2409.02374*, 2024.

Appendices

Contents

A Measuring the Linearity of Diffusion Denoisers

In this section, we provide a detailed discussion on how to measure the linearity of diffusion model. For a diffusion denoiser, $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$, to be considered approximately linear, it must fulfill the following conditions:

- *Additivity:* The function should satisfy $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1+\boldsymbol{x}_2;\sigma(t)) \approx \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t)) + \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_2;\sigma(t))$.
- *Homogeneity:* It should also adhere to $\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x};\sigma(t)) \approx \alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$.

To jointly assess these properties, we propose to measure the difference between $\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t))$ and $\alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t)) + \beta D(\boldsymbol{x}_2;\sigma(t))$. While the linearity score is introduced as the cosine similarity between $\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t))$ and $\alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t)) + \beta D(\boldsymbol{x}_2;\sigma(t))$ in the main text:

$$\texttt{LS}(t) = \mathbb{E}_{\boldsymbol{x}_1,\boldsymbol{x}_2\sim p(\boldsymbol{x};\sigma(t))}\left[\left|\left\langle \frac{\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t))}{\|\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t))\|_2}, \frac{\alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t))+\beta\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t))}{\|\alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t))+\beta\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t))\|_2}\right\rangle\right|\right], \tag{11}$$

it can also be defined with the normalized mean square difference (NMSE):

$$\mathbb{E}_{\boldsymbol{x}_1,\boldsymbol{x}_2\sim p(\boldsymbol{x};\sigma(t))}\frac{||\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t)) - (\alpha\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t))+\beta\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_1;\sigma(t)))||_2}{||\mathcal{D}_{\boldsymbol{\theta}}(\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2;\sigma(t))||_2}, \tag{12}$$

where the expectation is approximated with its empirical mean over 100 randomly sampled pairs of $(\boldsymbol{x}_1,\boldsymbol{x}_2)$. In the next section, we will demonstrate the linearity score with both metrics.

Since the diffusion denoisers are trained solely on inputs $\boldsymbol{x}\sim p(\boldsymbol{x};\sigma(t))$, their behaviors on out-of-distribution inputs can be quite irregular. To produce a denoised output with meaningful image structure, it is critical that the noise component in the input $\boldsymbol{x}$ matches the correct variance $\sigma(t)^2$. Therefore, our analysis of linearity is restricted to in-distribution inputs $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, which are randomly sampled images with additive Gaussian noises calibrated to noise variance $\sigma(t)^2$. We also need to ensure that the values of α and β are chosen such that $\alpha^2+\beta^2=1$, maintaining the correct variance for the noise term in the combined input $\alpha\boldsymbol{x}_1+\beta\boldsymbol{x}_2$. We present the linearity scores, calculated with varying values of α and β, for diffusion models trained on diverse datasets in Figure 10. These models are trained with the EDM-VE configuration proposed in [4], which ensures the resulting models are in the generalization regime. Typically, setting $\alpha=\beta=1/\sqrt{2}$ yields the lowest linearity score; however, even in this scenario, the cosine similarity remains impressively high, exceeding 0.96. This high value underscores the presence of significant linearity within diffusion denoisers.

We would like to emphasize that for linearity to manifest in diffusion denoisers, it is crucial that they are well-trained, achieving a low denoising score matching loss as indicated in (3). As shown in Figure 11, the linearity notably reduces in a less well trained diffusion model (Baseline-VE) comapred to its well-trained counterpart (EDM-VE). Although both models utilize the same 'VE' network architecture $\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$ [2], they differ in how the diffusion denoisers are parameterized:

$$\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t)) := c_{\text{skip}}(\sigma(t))\boldsymbol{x} + c_{\text{out}}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))), \tag{13}$$

where c_{skip} is the skip connection and c_{out} modulate the scale of the network output. With carefully tailored c_{skip} and c_{out}, the EDM-VE configuration achieves a lower score matching loss compared to Baseline-VE, resulting in samples with higher quality as illustrated in Figure 11(right).

B Emerging Linearity of Diffusion Models

In this section we provide a detailed discussion on the observation that diffusion models exhibit increasing linearity as they transition from memorization to generalization, which is briefly described in Section 3.1.

B.1 Generalization and Memorization Regimes of Diffusion Models

As shown in Figure 12, as the training dataset size increases, diffusion models transition from the memorization regime—where they can only replicate its training images—to the generalization regime, where the they produce high-quality, novel images. To measure the generalization capabilities of diffusion models, it is crucial to assess their ability to generate images that are not mere replications of the training dataset. This can be quantitatively evaluated by generating a large set of images from the diffusion model and measuring the average difference between these generated images and their nearest neighbors in the training set. Specifically, let $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_k\}$ represent k randomly sampled images from the diffusion models (we choose $k = 100$ in our experiments), and let $Y := \{\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N\}$ denote the training dataset consisting of N images. We define the generalization score as follows:

$$\text{GL Score} := \frac{1}{k}\sum_{i=1}^{k}\frac{||\boldsymbol{x}_i - \text{NN}_Y(\boldsymbol{x}_i)||_2}{||\boldsymbol{x}_i||_2} \tag{14}$$

where $\text{NN}_Y(\boldsymbol{x}_i)$ represents the nearest neighbor of the sample $\boldsymbol{x}_k$ in the training dataset Y, determined by the Euclidean distance on a per-pixel basis. Empirically, a GL score exceeding 0.6 indicates that the diffusion models are effectively generalizing beyond the training dataset.

B.2 Diffusion Models Exhibit Linearity in the Generalization Regime

As demonstrated in Figure 13(a) and (d), diffusion models transition from the memorization regime to the generalization regime as the training dataset size increases. Concurrently, as depicted in Figure 13(b), (c), (e) and (f), the corresponding diffusion denoisers exhibit increasingly linearity. This phenomenon persists across diverse datasets datasets including FFHQ [26], AFHQ [28] and LSUN-Churches [29], as well as various model architectures including EDM-VE [3], EDM-VP [2] and EDM-ADM [34]. This emerging linearity implies that the hidden linear structure plays an important role in the generalizability of diffusion model.

C Linear Distillation

As discussed in Section 3.1, we propose to study the hidden linearity observed in diffusion denosiers with linear distillation. Specifically, for a given diffusion denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$, we aim to approxi-

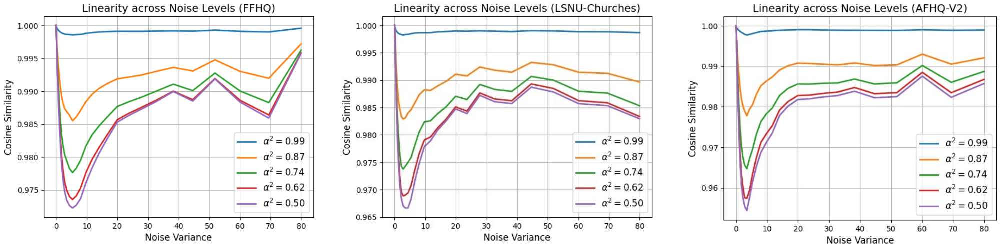


Figure 10: **Linearity scores for varying α and β.** The diffusion models are trained with the edm-ve configuration [4], which ensures the models are in the generalization regime.

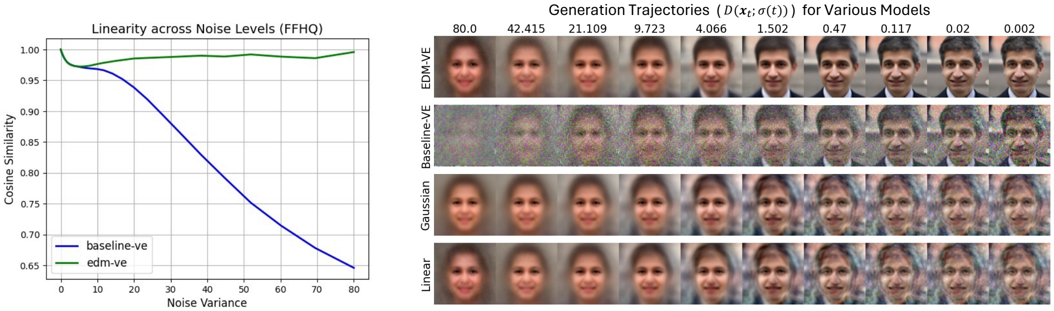


Figure 11: **Linearity scores and sampling trajectory.** The left and right figures demonstrate the linearity scores and the sampling trajectories $\mathcal{D}(\boldsymbol{x}_t; \sigma(t)$ of actual diffusion model (EDM-VE and Baseline-VE), Multi Delta model, linear model, and Gaussian model respectively.

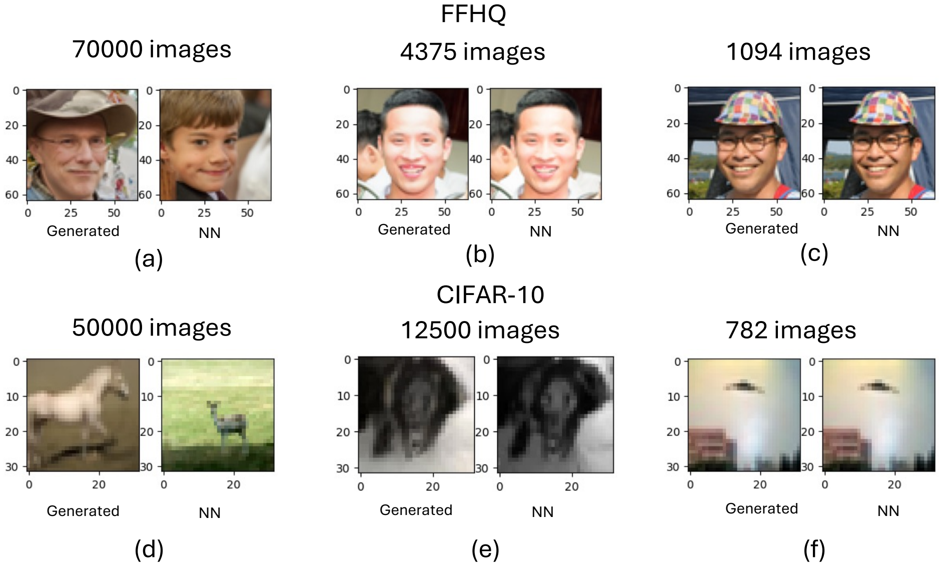


Figure 12: **Memorization and generalization regimes of diffusion models.** Figures(a) to (c) show the images generated by diffusion models trained on 70000, 4375, 1094 FFHQ images and their corresponding nearest neighbors in the training dataset respectively. Figures(d) to (f) show the images generated by diffusion models trained on 50000, 12500, 782 CIFAR-10 images and their corresponding nearest neighbors in the training dataset respectively. Notice that when the training dataset size is small, diffusion model can only generate images in the training dataset.

mate it with a linear function (with a bias term for more expressibility):

$$\mathcal{D}_{\mathrm{L}}(\boldsymbol{x}; \sigma(t)) := \boldsymbol{W}_{\sigma(t)}\boldsymbol{x} + \boldsymbol{b}_{\sigma(t)} \;\approx\; \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t)),$$

for $\boldsymbol{x} \sim p(\boldsymbol{x}; \sigma(t))$. Notice that for three dimensional images with size (c, h, w), $\boldsymbol{x} \in \mathbb{R}^d$ represents their vectorized version, where $d = c \times w \times h$. Let

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}) = \frac{1}{n}\sum_{i=1}^{n} \left\| \boldsymbol{W}_{\sigma(t)}\{k-1\}(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i) + \boldsymbol{b}_{\sigma(t)}\{k-1\} - \mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i; \sigma(t)) \right\|_2^2$$

We train 10 independent linear models for each of the selected noise variance level $\sigma(t)$ with the procedure summarized in Algorithm 1:

In practice, the gradients on $\boldsymbol{W}_{\sigma(t)}$ and $\boldsymbol{b}_{\sigma(t)}$ are obtained through automatic differentiation. Additionally, we employ the Adam optimizer [35] for updates. Additional linear distillation results are provided in Figure 14.

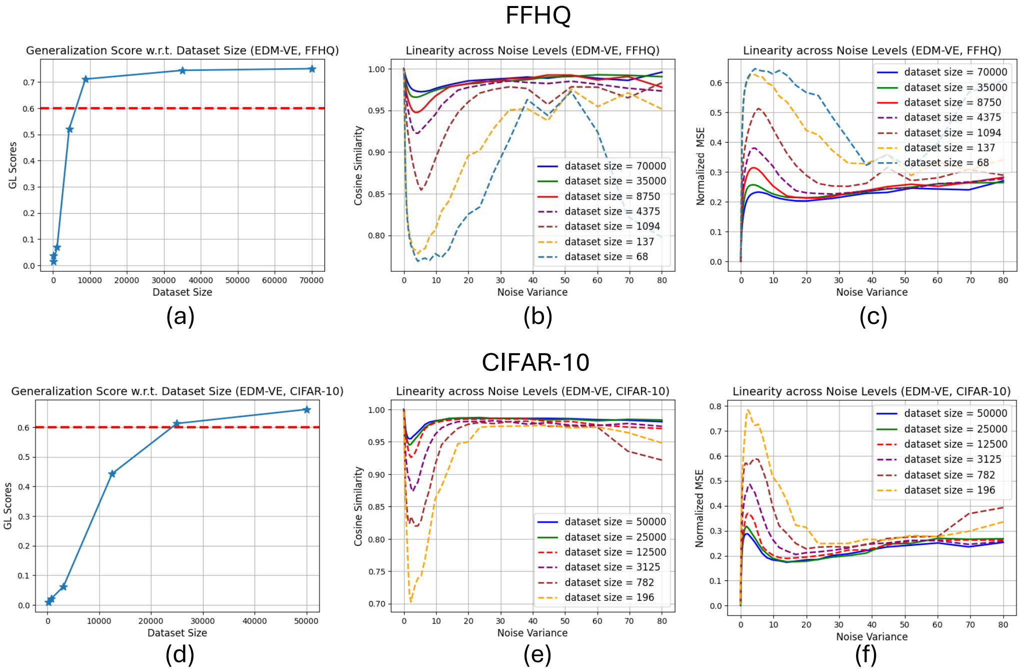


Figure 13: **Diffusion model exhibit increasing linearity as they transition from memorization to generalization.** Figure(a) and (d) demonstrate that for both FFHQ and CIFAR-10 datasets, the generalization score increases with the training dataset size, indicating progressive model generalization. Figure(b), (c), (e), and (f) show that this transition towards generalization is accompanied by increasing denoiser linearity. Specifically, Figure(b) and (e) display linearity scores calculated using cosine similarity (11), while Figure(c) and (f) show scores computed using NMSE (12). Both metrics reveal consistent trends.

D Diffusion Models in Low-noise Regime are Approximately Linear Mapping

It should be noted that the low score difference between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ within the low-noise regime ($\sigma(t) \in [0.002, 0.116]$) does not imply the diffusion denoisers capture the Gaussian structure, instead, the similarity arises since both of them are converging to the identity mapping as $\sigma(t)$ decreases. As shown in Figure 15, within this regime, the differences between the noisy input $\boldsymbol{x}$ and their corresponding denoised outputs $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))$ quickly approach 0. This indicates that the learned denoisers $\mathcal{D}_{\boldsymbol{\theta}}$ progressively converge to the identity function. Additionally, from (6), it is evident that the difference between the Gaussian weights and the identity matrix diminishes as $\sigma(t)$ decreases, which explains why $\mathcal{D}_{\mathrm{G}}$ can well approximate $\mathcal{D}_{\boldsymbol{\theta}}$ in the low noise variance regime.

We hypothesize that $\mathcal{D}_{\boldsymbol{\theta}}$ learns the identity function because of the following two reasons:

(i) within the low-noise regime, since the added noise is negligible compared to the clean image, the identity function already achieves a small denoising error, thus serving as a shortcut which is exploited by the deep network.

(ii) As discussed in Appendix A, diffusion models are typically parameterized as follows:

$$\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t)) := c_{\text{skip}}(\sigma(t))\boldsymbol{x} + c_{\text{out}}(\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t))),$$

where $\mathcal{F}_{\boldsymbol{\theta}}$ represents the deep network, and $c_{\text{skip}}(\sigma(t))$ and $c_{\text{out}}(\sigma(t))$ are adaptive parameters for the skip connection and output scaling, respectively, which adjust according to the noise variance levels. For canonical works on diffusion models [2–4, 34], as $\sigma(t)$ approaches zero, c_{skip} and c_{out} converge to 1 and 0 respectively. Consequently, at low variance levels, the function forms of diffusion denoisers are approximatly identity mapping: $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}; \sigma(t)) \approx \boldsymbol{x}$.

This convergence to identity mapping has several implications. First, the weights $\boldsymbol{W}_{\sigma(t)}$ of the distilled linear models $\mathcal{D}_{\mathrm{L}}$ approach the identity matrix at low variances, leading to ambiguous

Algorithm 1 Linear Distillation

```
Require:
  (i) the targeted diffusion denoiser $\mathcal{D}_{\boldsymbol{\theta}}(\cdot;\sigma(t))$,
  (ii) weights $\boldsymbol{W}_{\sigma(t)}$ and biases $\boldsymbol{b}_{\sigma(t)}$, both initialized to zero,
  (iii) gradient step size $\eta$,
  (iv) number of training iterations $K$,
  (v) training batch size $n$,
  (vi) image dataset $S$.
  for $k = 1$ to $K$ do
      Randomly sample a batch of training images $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\}$ from $S$.
      Randomly sample a batch of noises $\{\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2, \ldots, \boldsymbol{\epsilon}_n\}$ from $\mathcal{N}(\boldsymbol{0}, \sigma(t)\boldsymbol{I})$.
      Update $\boldsymbol{W}_{\sigma(t)}$ and $\boldsymbol{b}_{\sigma(t)}$ with gradient descent:
  $\boldsymbol{W}_{\sigma(t)}\{k\} = \boldsymbol{W}_{\sigma(t)}\{k-1\} - \eta\nabla_{\boldsymbol{W}_{\sigma(t)}\{k-1\}}\mathcal{L}(\boldsymbol{W}, \boldsymbol{b})$
  $\boldsymbol{b}_{\sigma(t)}\{k\} = \boldsymbol{b}_{\sigma(t)}\{k-1\} - \eta\nabla_{\boldsymbol{b}_{\sigma(t)}\{k-1\}}\mathcal{L}(\boldsymbol{W}, \boldsymbol{b})$
  end for
  Return $\boldsymbol{W}_{\sigma(t)}\{K\}, \boldsymbol{b}_{\sigma(t)}\{K\}$
```

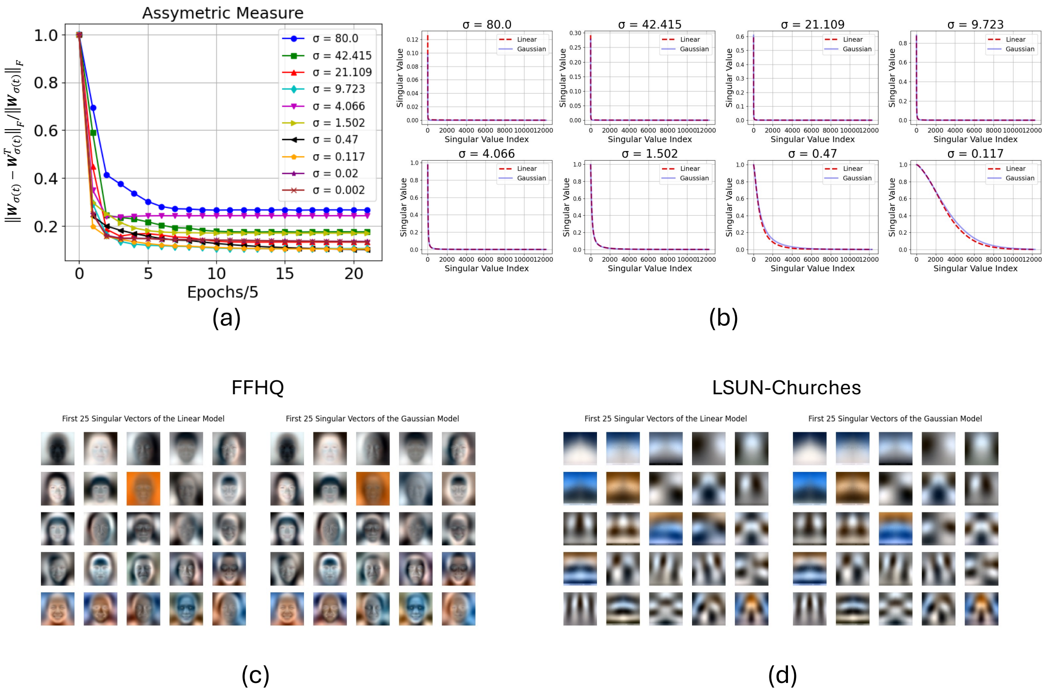


Figure 14: **Additional linear distillation results.** Figure(a) demonstrates the gradual symmetrization of linear weights during the distillation process. Figure(b) shows that at convergence, the singular values of the linear weights closely match those of the Gaussian weights. Figure(c) and Figure(d) display the leading singular vectors of both linear and Gaussian weights at $\sigma(t) = 4$ for FFHQ and LSUN-Churches datasets, respectively, revealing a strong correlation.

singular vectors. This explains the poor recovery of singular vectors for $\sigma(t) \in [0.002, 0.116]$ shown in Figure 4. Second, the presence of the bias term in (8) makes it challenging for our linear model to learn the identity function, resulting in large errors at $\sigma(t) = 0.002$ as shown in Figure 4(a).

Finally, from (4), we observe that when $\mathcal{D}_{\boldsymbol{\theta}}$ acts as an identity mapping, $\boldsymbol{x}_{i+1}$ remains unchanged from $\boldsymbol{x}_i$. This implies that sampling steps in low-variance regions minimally affect the generated image content, as confirmed in Figure 2, where image content shows negligible variation during these steps.

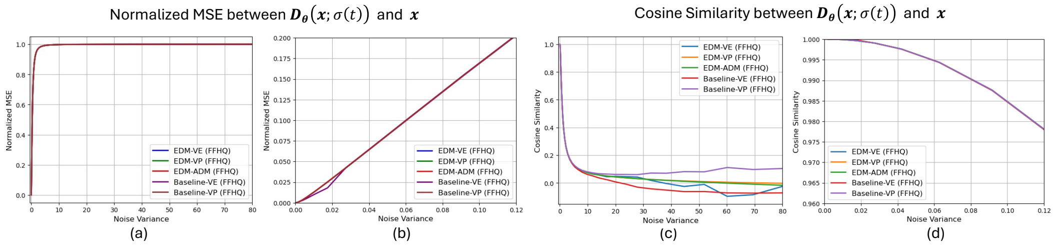


Figure 15: **Difference between $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$ and $\boldsymbol{x}$ for various noise variance levels.** Figures(a) and (c) show the differences between $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$ and $\boldsymbol{x}$ across $\sigma(t)\in[0.002,80]$, measured by normalized MSE and cosine similarity, respectively. Figures(b) and (d) provide zoomed-in views of (a) and (c). The diffusion models were trained on the FFHQ dataset. Notice that the difference between $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x};\sigma(t))$ and $\boldsymbol{x}$ quickly converges to near zero in the low noise variance regime. The trend is consistent for various model architectures.

E Theoretical Analysis

E.1 Proof of Theorem 1

In this section, we give the proof of Theorem 1 (Section 3.3). Our theorem is based on the following two assumptions:

Assumption 1. *Suppose that the diffusion denoisers are parameterized as single-layer linear networks, defined as $\mathcal{D}(\boldsymbol{x};\sigma(t))=\boldsymbol{W}_{\sigma(t)}\boldsymbol{x}+\boldsymbol{b}_{\sigma(t)}$, where $\boldsymbol{W}_{\sigma(t)}\in\mathbb{R}^{d\times d}$ is the linear weight and $\boldsymbol{b}_{\sigma(t)}\in\mathbb{R}^{d}$ is the bias.*

Assumption 2. *The data distribution $p_{data}(\boldsymbol{x})$ has finite mean $\boldsymbol{\mu}$ and bounded positive semidefinite covariance $\boldsymbol{\Sigma}$*

Theorem 1. *Under Assumption 1 and Assumption 2, the optimal solution to the denoising score matching objective (3) is exactly the Gaussian denoiser: $\mathcal{D}_{\mathrm{G}}(\boldsymbol{x},\sigma(t))=\boldsymbol{\mu}+\boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^{T}(\boldsymbol{x}-\boldsymbol{\mu})$, where $\boldsymbol{\Sigma}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^{T}$ represents the SVD of the covariance matrix, with singular values $\lambda_{\{k=1,\dots,d\}}$ and $\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}=\mathrm{diag}[\frac{\lambda_k}{\lambda_k+\sigma(t)^2}]$. Furthermore, this optimal solution can be obtained via gradient descent with a proper learning rate.*

To prove **Theorem 1**, we first show that the Gaussian denoiser is the optimal solution to the denoising score matching objective under the linear network constraint. Then we will show that such optimal solution can be obtained via gradient descent with a proper learning rate.

The Global Optimal Solution. Under the constraint that the diffusion denoiser is restricted to a single-layer linear network with bias:

$$\mathcal{D}(\boldsymbol{x};\sigma(t))=\boldsymbol{W}_{\sigma(t)}\boldsymbol{x}+\boldsymbol{b}_{\sigma(t)}, \tag{15}$$

We get the following optimizaiton problem from Equation (3):

$$\boldsymbol{W}^{\star},\boldsymbol{b}^{\star}=\arg\min_{\boldsymbol{W},\boldsymbol{b}}\mathcal{L}(\boldsymbol{W},\boldsymbol{b};\sigma(t)):=\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma(t)^2\boldsymbol{I})}||\boldsymbol{W}(\boldsymbol{x}+\boldsymbol{\epsilon})+\boldsymbol{b}-\boldsymbol{x}||_2^2, \tag{16}$$

where we omit the footnote $\sigma(t)$ in $\boldsymbol{W}_{\sigma(t)}$ and $\boldsymbol{b}_{\sigma(t)}$ for simplicity. Since expectation preserves convexity, the optimization problem Equation (16) is a convex optimization problem. To find the global optimum, we first eliminate $\boldsymbol{b}$ by requiring the partial derivative $\nabla_{\boldsymbol{b}}\mathcal{L}(\boldsymbol{W},\boldsymbol{b};\sigma(t))$ to be $\mathbf{0}$. Since

$$\begin{aligned}\nabla_{\boldsymbol{b}}\mathcal{L}(\boldsymbol{W},\boldsymbol{b};\sigma(t))&=2*\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\sigma(t)^2\boldsymbol{I})}((\boldsymbol{W}-\boldsymbol{I})\boldsymbol{x}+\boldsymbol{W}\boldsymbol{\epsilon}+\boldsymbol{b}) &(17)\\&=2*\mathbb{E}_{\boldsymbol{x}\sim p_{\mathrm{data}}}((\boldsymbol{W}-\boldsymbol{I})\boldsymbol{x}+\boldsymbol{b}) &(18)\\&=2*((\boldsymbol{W}-\boldsymbol{I})\boldsymbol{\mu}+\boldsymbol{b}), &(19)\end{aligned}$$

we have

$$\boldsymbol{b}^{\star}=(\boldsymbol{I}-\boldsymbol{W}^{*})\boldsymbol{\mu}. \tag{20}$$

Utilizing the expression for $\boldsymbol{b}$, we get the following equivalent form of the optimization problem:

$$\boldsymbol{W}^\star = \arg\min_{\boldsymbol{W}} \mathcal{L}(\boldsymbol{W};\sigma(t)) := 2 * \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\sigma(t)^2\boldsymbol{I})}||\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{\mu}+\boldsymbol{\epsilon})-(\boldsymbol{x}-\boldsymbol{\mu})||_2^2. \tag{21}$$

The derivative $\nabla_{\boldsymbol{W}}\mathcal{L}(\boldsymbol{W};\sigma(t))$ is:

$$\begin{aligned}\nabla_{\boldsymbol{W}}\mathcal{L}(\boldsymbol{W};\sigma(t)) &= 2 * \mathbb{E}_{\boldsymbol{x}}\mathbb{E}_{\boldsymbol{\epsilon}}(\boldsymbol{W}(\boldsymbol{x}-\boldsymbol{\mu}+\boldsymbol{\epsilon})(\boldsymbol{x}-\boldsymbol{\mu}+\boldsymbol{\epsilon})^T - (\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu}+\boldsymbol{\epsilon})^T) & (22)\\ &= 2 * \mathbb{E}_{\boldsymbol{x}}((\boldsymbol{W}-\boldsymbol{I})(\boldsymbol{x}-\boldsymbol{\mu})(\boldsymbol{x}-\boldsymbol{\mu})^T + \sigma(t)^2\boldsymbol{W}) & (23)\\ &= 2 * \boldsymbol{W}(\boldsymbol{\Sigma}+\sigma(t)^2\boldsymbol{I}) - 2 * \boldsymbol{\Sigma}. & (24)\end{aligned}$$

Suppose $\boldsymbol{\Sigma} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$ is the SVD of the empirical covariance matrix, with singular values $\lambda_{\{k=1,\dots,n\}}$, by setting $\nabla_{\boldsymbol{W}}\mathcal{L}(\boldsymbol{W};\sigma(t))$ to $\boldsymbol{0}$, we get the optimal solution:

$$\begin{aligned}\boldsymbol{W}^\star &= \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\boldsymbol{U}(\boldsymbol{\Lambda}+\sigma(t)^2\boldsymbol{I})^{-1}\boldsymbol{U}^T & (25)\\ &= \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T, & (26)\end{aligned}$$

where $\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}[i,i] = \frac{\lambda_i}{\lambda_i+\sigma(t)^2}$ and $\lambda_i = \boldsymbol{\Lambda}[i,i]$. Substitute $\boldsymbol{W}^\star$ back to Equation (20), we have:

$$\boldsymbol{b}^\star = (\boldsymbol{I} - \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T)\boldsymbol{\mu}. \tag{27}$$

Notice that the expression for $\boldsymbol{W}^\star$ and $\boldsymbol{b}^\star$ is exactly the Gaussian denoiser. Next, we will show this optimal solution can be achieved with gradient descent.

Gradient Descent Recovers the Optimal Solution. Consider minimizing the population loss:

$$\mathcal{L}(\boldsymbol{W},\boldsymbol{b};\sigma(t)) := \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\sigma(t)^2\boldsymbol{I})}||\boldsymbol{W}(\boldsymbol{x}+\boldsymbol{\epsilon})+\boldsymbol{b}-\boldsymbol{x}||_2^2. \tag{28}$$

Define $\tilde{\boldsymbol{W}} := [\boldsymbol{W} \quad \boldsymbol{b}]$, $\tilde{\boldsymbol{x}} := \begin{bmatrix}\boldsymbol{x}\\1\end{bmatrix}$ and $\tilde{\boldsymbol{\epsilon}} = \begin{bmatrix}\boldsymbol{\epsilon}\\0\end{bmatrix}$, then we can rewrite Equation (28) as:

$$\mathcal{L}(\tilde{\boldsymbol{W}};\sigma(t)) := \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0},\sigma(t)^2\boldsymbol{I})}||\tilde{\boldsymbol{W}}(\tilde{\boldsymbol{x}}+\tilde{\boldsymbol{\epsilon}})-\boldsymbol{x}||_2^2. \tag{29}$$

We can compute the gradient in terms of $\tilde{\boldsymbol{W}}$ as:

$$\begin{aligned}\nabla\mathcal{L}(\tilde{\boldsymbol{W}}) &= 2 * \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}}(\tilde{\boldsymbol{W}}(\tilde{\boldsymbol{x}}+\tilde{\boldsymbol{\epsilon}})(\tilde{\boldsymbol{x}}+\tilde{\boldsymbol{\epsilon}})^T - \boldsymbol{x}(\tilde{\boldsymbol{x}}+\tilde{\boldsymbol{\epsilon}})^T) & (30)\\ &= 2 * \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon}}(\tilde{\boldsymbol{W}}(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T + \tilde{\boldsymbol{x}}\tilde{\boldsymbol{\epsilon}}^T + \tilde{\boldsymbol{\epsilon}}\tilde{\boldsymbol{x}}^T + \tilde{\boldsymbol{\epsilon}}\tilde{\boldsymbol{\epsilon}}^T) - \boldsymbol{x}\tilde{\boldsymbol{x}}^T - \boldsymbol{x}\tilde{\boldsymbol{\epsilon}}^T). & (31)\end{aligned}$$

Since $\mathbb{E}_{\boldsymbol{\epsilon}}(\tilde{\boldsymbol{\epsilon}}) = \boldsymbol{0}$ and $\mathbb{E}_{\boldsymbol{\epsilon}}(\tilde{\boldsymbol{\epsilon}}\tilde{\boldsymbol{\epsilon}}^T) = \begin{bmatrix}\sigma(t)^2\boldsymbol{I}_{d\times d} & \boldsymbol{0}_{d\times 1}\\ \boldsymbol{0}_{1\times d} & 0\end{bmatrix}$, we have:

$$\nabla\mathcal{L}(\tilde{\boldsymbol{W}}) = 2 * \mathbb{E}_{\boldsymbol{x}}(\tilde{\boldsymbol{W}}(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T + \begin{bmatrix}\sigma(t)^2\boldsymbol{I}_{d\times d} & \boldsymbol{0}_{d\times 1}\\ \boldsymbol{0}_{1\times d} & 0\end{bmatrix}) - \boldsymbol{x}\tilde{\boldsymbol{x}}^T). \tag{32}$$

Since $\mathbb{E}(\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^T) = \begin{bmatrix}\mathbb{E}(\boldsymbol{x}\boldsymbol{x}^T) & \mathbb{E}(\boldsymbol{x})\\ \mathbb{E}(\boldsymbol{x}^T) & 1\end{bmatrix}$, we have:

$$\nabla\mathcal{L}(\tilde{\boldsymbol{W}}) = 2\tilde{\boldsymbol{W}}\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)+\sigma(t)^2\boldsymbol{I} & \boldsymbol{\mu}\\ \boldsymbol{\mu}^T & 1\end{bmatrix} - 2\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu}\end{bmatrix}. \tag{33}$$

With learning rate η, we can write the update rule as:

$$\begin{aligned}\tilde{\boldsymbol{W}}(t+1) &= \tilde{\boldsymbol{W}}(t)(1-2\eta\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)+\sigma(t)^2\boldsymbol{I} & \boldsymbol{\mu}\\ \boldsymbol{\mu}^T & 1\end{bmatrix}) + 2\eta\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu}\end{bmatrix} & (34)\\ &= \tilde{\boldsymbol{W}}(t)(1-2\eta\boldsymbol{A}) + 2\eta\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu}\end{bmatrix}, & (35)\end{aligned}$$

where we define $\boldsymbol{A} := \boldsymbol{I} - 2\eta\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T)+\sigma(t)^2\boldsymbol{I} & \boldsymbol{\mu}\\ \boldsymbol{\mu}^T & 1\end{bmatrix}$ for simplicity. By recursively expanding the expression for $\tilde{\boldsymbol{W}}$, we have:

$$\tilde{\boldsymbol{W}}(t+1) = \tilde{\boldsymbol{W}}(0)\boldsymbol{A}^{t+1} + 2\eta\begin{bmatrix}\mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu}\end{bmatrix}\sum_{i=0}^{t}\boldsymbol{A}^i. \tag{36}$$

Notice that there exists a η, such that every eigen value of $\boldsymbol{A}$ is smaller than 1 and greater than 0. In this case, $\boldsymbol{A}^{t+1} \to \boldsymbol{0}$ as $t \to \infty$. Similarly, by the property of matrix geometric series, we have $\sum_{i=0}^{t} \boldsymbol{A}^i \to (\boldsymbol{I} - \boldsymbol{A})^{-1}$. Therefore we have:

$$\tilde{\boldsymbol{W}} \to \begin{bmatrix} \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu} \end{bmatrix} \begin{bmatrix} \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T) + \sigma(t)^2\boldsymbol{I} & \boldsymbol{\mu} \\ \boldsymbol{u}^T & 1 \end{bmatrix}^{-1} \tag{37}$$

$$= \begin{bmatrix} \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}^T\boldsymbol{x}) & \boldsymbol{\mu} \end{bmatrix} \begin{bmatrix} \boldsymbol{B} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}^{-1}, \tag{38}$$

where we define $\boldsymbol{B} := \mathbb{E}_{\boldsymbol{x}}(\boldsymbol{x}\boldsymbol{x}^T) + \sigma(t)^2\boldsymbol{I}$ for simplicity. By the Sherman–Morrison–Woodbury formula, we have:

$$\begin{bmatrix} \boldsymbol{B} & \boldsymbol{\mu} \\ \boldsymbol{\mu}^T & 1 \end{bmatrix}^{-1} = \begin{bmatrix} (\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} & -(\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1}\boldsymbol{\mu} \\ -(1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu})^{-1}\boldsymbol{\mu}^T\boldsymbol{B}^{-1} & (1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu})^{-1} \end{bmatrix}. \tag{39}$$

Therefore, we have:

$$\tilde{\boldsymbol{W}} \to \begin{bmatrix} \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T](\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1-\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} & -\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T](\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1}\boldsymbol{\mu} + \frac{\boldsymbol{\mu}}{1-\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} \end{bmatrix}, \tag{40}$$

from which we have

$$\boldsymbol{W} \to \mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T](\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} \tag{41}$$

$$\boldsymbol{b} \to -\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T](\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1}\boldsymbol{\mu} + \frac{\boldsymbol{\mu}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} \tag{42}$$

Since $\mathbb{E}_{\boldsymbol{x}}[\boldsymbol{x}\boldsymbol{x}^T] = \mathbb{E}_{\boldsymbol{x}}[(\boldsymbol{x} - \boldsymbol{\mu})((\boldsymbol{x} - \boldsymbol{\mu})^T] + \boldsymbol{\mu}\boldsymbol{\mu}^T$, we have:

$$\boldsymbol{W} = \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \sigma(t)^2\boldsymbol{I})^{-1} + \boldsymbol{\mu}\boldsymbol{\mu}^T(\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}}. \tag{43}$$

Applying Sherman-Morrison Formula, we have:

$$(\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} = \boldsymbol{B}^{-1} + \frac{\boldsymbol{B}^{-1}\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}}, \tag{44}$$

therefore

$$\boldsymbol{\mu}\boldsymbol{\mu}^T(\boldsymbol{B} - \boldsymbol{\mu}\boldsymbol{\mu}^T)^{-1} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} = \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} - \frac{\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}} \tag{45}$$

$$= \frac{\boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}}{1 - \boldsymbol{\mu}^T\boldsymbol{B}^{-1}\boldsymbol{\mu}}(\boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1} - \boldsymbol{\mu}\boldsymbol{\mu}^T\boldsymbol{B}^{-1}) \tag{46}$$

$$= \boldsymbol{0} \tag{47}$$

, which implies

$$\boldsymbol{W} \to \boldsymbol{\Sigma}(\boldsymbol{\Sigma} + \sigma(t)^2\boldsymbol{I})^{-1} \tag{48}$$

$$= \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T. \tag{49}$$

Similarly, we have:

$$\boldsymbol{b} \to (\boldsymbol{I} - \boldsymbol{U}\tilde{\boldsymbol{\Lambda}}_{\sigma(t)}\boldsymbol{U}^T)\boldsymbol{\mu}. \tag{50}$$

Therefore, gradient descent with a properly chosen learning rate η recovers the Gaussian Denoisers when time goes to infinity.

E.2 Two Extreme Cases

Our empirical results indicate that the best linear approximation of $\mathcal{D}_{\boldsymbol{\theta}}$ is approximately equivalent to $\mathcal{D}_{\mathrm{G}}$. According to the orthogonality principle [36], this requires $\mathcal{D}_{\boldsymbol{\theta}}$ to satisfy:

$$\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}\mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(\boldsymbol{0};\sigma(t)^2\boldsymbol{I})}\{(\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x} + \boldsymbol{\epsilon}; \sigma(t)) - (\boldsymbol{x} - \boldsymbol{\mu}))(\boldsymbol{x} + \boldsymbol{\epsilon} - \boldsymbol{\mu})^T\} \approx \boldsymbol{0}. \tag{51}$$

Notice that (51) does not hold for general denoisers. Two extreme cases for this to hold are:

- Case 1: $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t)) \approx \boldsymbol{x}$ for $\forall \boldsymbol{x} \sim p_{\text{data}}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \boldsymbol{I})$.
- Case 2: $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t)) \approx \mathcal{D}_{\mathrm{G}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t))$ for $\forall \boldsymbol{x} \sim p_{\text{data}}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \boldsymbol{I})$.

Case 1 requires $\mathcal{D}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\epsilon};\sigma(t))$ to be the oracle denoiser that perfectly recover the ground truth clean image, which never happens in practice except when $\sigma(t)$ becomes extremely small. Instead, our empirical results suggest diffusion models in the generalization regime bias towards Case 2, where deep networks learn $\mathcal{D}_{\boldsymbol{\theta}}$ that approximate (not equal) to $\mathcal{D}_{\mathrm{G}}$. This is evidenced in Figure 5(b), where diffusion models trained on larger datasets (35000 and 7000 images) produce denoising outputs similar to $\mathcal{D}_{\mathrm{G}}$. Notice that this similarity holds even when the denoisers take pure Gaussian noise as input. The exact mechanism driving diffusion models trained with gradient descent towards this particular solution remains an open question and we leave it as future work.

F More Discussion on Section 4

While in Section 4 we mainly focus on the discussion of the behavior of diffusion denoisers in the intermediate-noise regime, in this section we study the denoiser dynamics in both low and high-noise regime. We also provide additional experiment results on CIFAR-10 dataset.

F.1 Behaviors in Low-noise Regime

We visualize the score differences between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ in low-noise regime in Figure 16. The left figure demonstrates that when the dataset size becomes smaller than a certain threshold, the score difference at $\sigma = 0$ remains persistently non-zero. Moreover, the right figure shows that this difference depends solely on dataset size rather than model capacity. This phenomenon arises from two key factors: *(i)* $\mathcal{D}_{\boldsymbol{\theta}}$ converges to the identity mapping at low noise levels, independent of training dataset size and model capacity, and *(ii)* $\mathcal{D}_{\mathrm{G}}$ approximates the identity mapping at low noise levels only when the empirical covariance matrix is full-rank, as can be seen from (6).

Since the rank of the covariance matrix is upper-bounded by the training dataset size, $\mathcal{D}_{\mathrm{G}}$ differs from the identity mapping when the dataset size is smaller than the data dimension. This creates a persistent gap between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$, with smaller datasets leading to lower rank and consequently larger score differences. These observations align with our discussion in Appendix D.

F.2 Behaviors in High-noise Regime

As shown in Figure 7(a), while a decreased model scale pushes $\mathcal{D}_{\boldsymbol{\theta}}$ in the intermediate noise region towards $\mathcal{D}_{\mathrm{G}}$, their differences enlarges in the high noise variance regime. This phenomenon arises because diffusion models employ a bell-shaped noise sampling distribution that prioritizes intermediate noise levels, resulting in insufficient training at high noise variances. A shown in Figure 17, for high $\sigma(t)$, $\mathcal{D}_{\boldsymbol{\theta}}$ converge to $\mathcal{D}_{\mathrm{G}}$ when trained with sufficient model capacity (Figure 17(b)) and training time (Figure 17(c)). This behavior is consistent irrespective of the training dataset sizes (Figure 17(a)). Convergence in the high-noise variance regime is less crucial in practice, since diffusion steps in

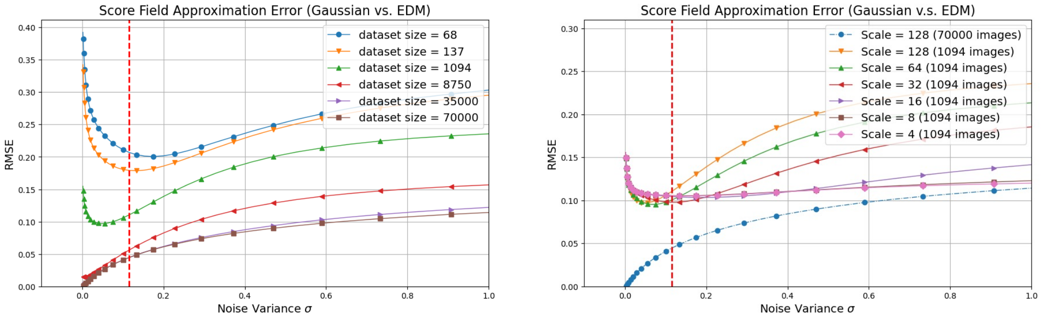


Figure 16: **Score differences for low-noise variances.** The left and right figures are the zoomed-in views of Figure 6(a) and Figure 7(a) respectively. Notice that when the dataset size is smaller than the dimension of the image, the score differences are always non-zero at $\sigma = 0$.

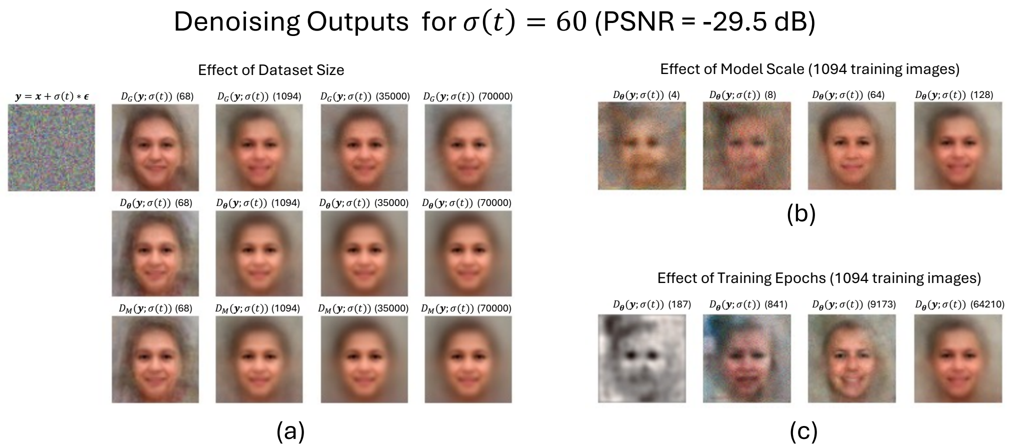

Denoising Outputs for $\sigma(t) = 60$ (PSNR = -29.5 dB)
Effect of Dataset Size
$y = x + \sigma(t) * \epsilon$
$D_G(y;\sigma(t))$ (68)
$D_G(y;\sigma(t))$ (1094)
$D_G(y;\sigma(t))$ (35000)
$D_G(y;\sigma(t))$ (70000)
$D_\theta(y;\sigma(t))$ (68)
$D_\theta(y;\sigma(t))$ (1094)
$D_\theta(y;\sigma(t))$ (35000)
$D_\theta(y;\sigma(t))$ (70000)
$D_M(y;\sigma(t))$ (68)
$D_M(y;\sigma(t))$ (1094)
$D_M(y;\sigma(t))$ (35000)
$D_M(y;\sigma(t))$ (70000)
(a)
Effect of Model Scale (1094 training images)
$D_\theta(y;\sigma(t))$ (4)
$D_\theta(y;\sigma(t))$ (8)
$D_\theta(y;\sigma(t))$ (64)
$D_\theta(y;\sigma(t))$ (128)
(b)
Effect of Training Epochs (1094 training images)
$D_\theta(y;\sigma(t))$ (187)
$D_\theta(y;\sigma(t))$ (841)
$D_\theta(y;\sigma(t))$ (9173)
$D_\theta(y;\sigma(t))$ (64210)
(c)

Figure 17: **$\mathcal{D}_{\boldsymbol{\theta}}$ converge to $\mathcal{D}_{\mathrm{G}}$ with no overfitting for high noise variances.** Figure(a) shows the denoising outputs of $\mathcal{D}_{\mathrm{M}}$, $\mathcal{D}_{\mathrm{G}}$ and well-trained (trained with sufficient model capacity till convergence) $\mathcal{D}_{\boldsymbol{\theta}}$. Notice that at high noise variance, the three different denoisers are approximately equivalent despite the training dataset size. Figure(b) shows the denoising outputs of $\mathcal{D}_{\boldsymbol{\theta}}$ with different model scales trained until convergence. Notice that $\mathcal{D}_{\boldsymbol{\theta}}$ converges to $\mathcal{D}_{\mathrm{G}}$ only when the model capacity is large enough. Figure(c) shows the denoising outputs of $\mathcal{D}_{\boldsymbol{\theta}}$ with sufficient large model capacity at different training epochs. Notice that $\mathcal{D}_{\boldsymbol{\theta}}$ converges to $\mathcal{D}_{\mathrm{G}}$ only when the training duration is long enough.

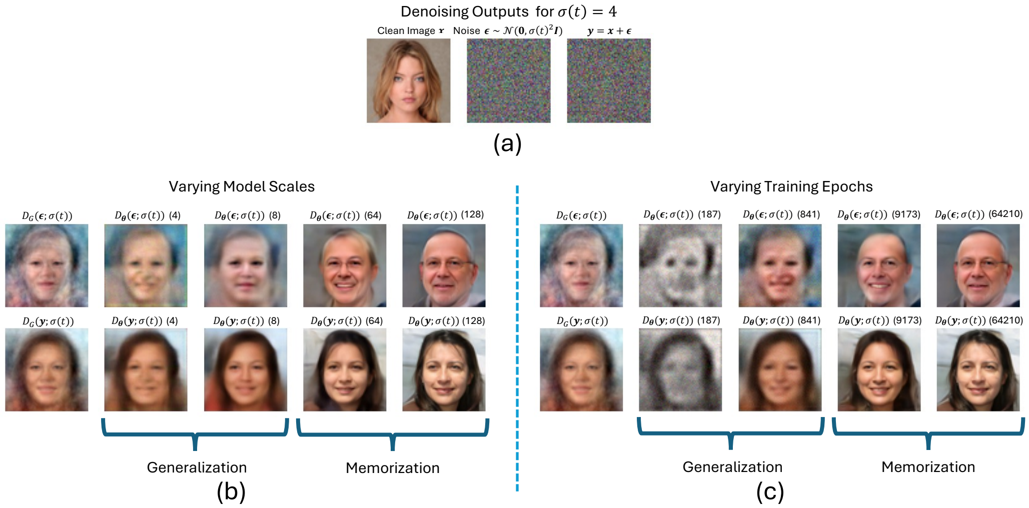
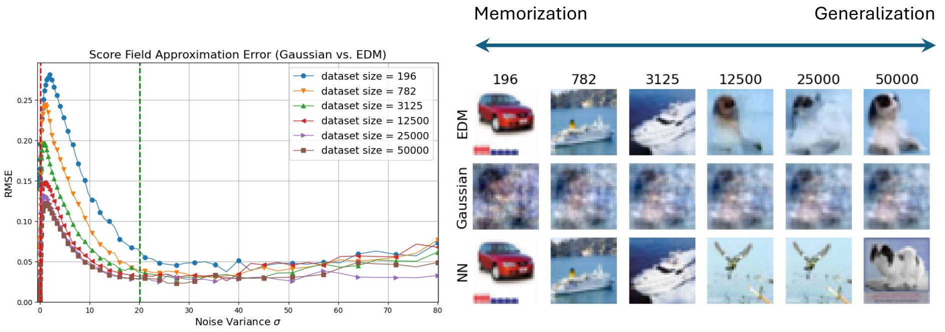

Denoising Outputs for $\sigma(t) = 4$
Clean Image x
Noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2 \boldsymbol{I})$
$y = x + \epsilon$
(a)
Varying Model Scales
$D_G(\epsilon;\sigma(t))$
$D_\theta(\epsilon;\sigma(t))$ (4)
$D_\theta(\epsilon;\sigma(t))$ (8)
$D_\theta(\epsilon;\sigma(t))$ (64)
$D_\theta(\epsilon;\sigma(t))$ (128)
$D_G(y;\sigma(t))$
$D_\theta(y;\sigma(t))$ (4)
$D_\theta(y;\sigma(t))$ (8)
$D_\theta(y;\sigma(t))$ (64)
$D_\theta(y;\sigma(t))$ (128)
Generalization
Memorization
(b)
Varying Training Epochs
$D_G(\epsilon;\sigma(t))$
$D_\theta(\epsilon;\sigma(t))$ (187)
$D_\theta(\epsilon;\sigma(t))$ (841)
$D_\theta(\epsilon;\sigma(t))$ (9173)
$D_\theta(\epsilon;\sigma(t))$ (64210)
$D_G(y;\sigma(t))$
$D_\theta(y;\sigma(t))$ (187)
$D_\theta(y;\sigma(t))$ (841)
$D_\theta(y;\sigma(t))$ (9173)
$D_\theta(y;\sigma(t))$ (64210)
Generalization
Memorization
(c)

Figure 18: **Denoising outputs of $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ at $\sigma = 4$.** Figure(a) shows the clean image $\boldsymbol{x}$ (from test set), random noise $\boldsymbol{\epsilon}$ and the resulting noisy image $\boldsymbol{y}$. Figure(b) compares denoising outputs of $\mathcal{D}_{\boldsymbol{\theta}}$ across different channel sizes [4, 8, 64, 128] with those of $\mathcal{D}_{\mathrm{G}}$. Figure(c) shows the evolution of $\mathcal{D}_{\boldsymbol{\theta}}$ outputs at training epochs [187, 841, 9173, 64210] alongside $\mathcal{D}_{\mathrm{G}}$ outputs. All models are trained on a fixed dataset of 1,094 images.

this regime contribute substantially less than those in the intermediate-noise variance regime—a phenomenon we analyze further in Appendix G.5.

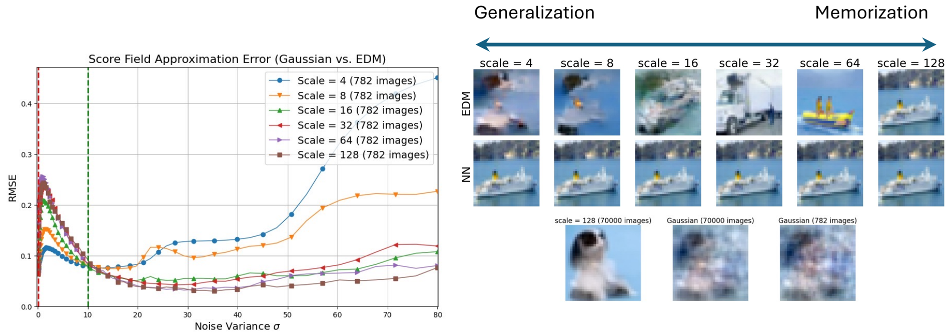
F.3 Similarity between Diffusion Denoiers and Gaussian Denoisers

In Section 4, we demonstrate that the Gaussian inductive bias is most prominent in models with limited capacity and during early training stages, a finding qualitatively validated in Figure 18. Specifically, Figure 18(b) shows that larger models (channel sizes 128 and 64) tend to memorize,



(a) (b)

Figure 19: **Large dataset size prompts the Gaussian structure.** Models with the same scale (channel size 64) are trained on CIFAR-10 datasets with varying sizes. Figure(a) shows that larger dataset size leads to increased similarity between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$, resulting in structurally similar generated images as shown in Figure(b).



(a) (b)

Figure 20: **Smaller model scale prompts the Gaussian structure.** Models with varying scales are trained on a fixed CIFAR-10 datasets with 782 images. Figure(a) shows that smaller model scale leads to increased similarity between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ in the intermediate noise regime ($\sigma \in [0.1, 10]$), resulting in structurally similar generated images as shown in figure(b). However, smaller scale leads to larger score differences in high-noise regime due to insufficient training from limited model capacity.

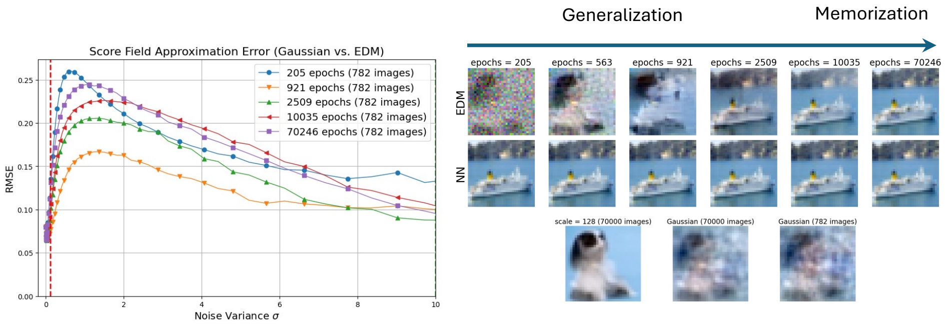
directly retrieving training data as denoising outputs. In contrast, smaller models (channel sizes 8 and 4) exhibit behavior similar to $\mathcal{D}_{\mathrm{G}}$, producing comparable denoising outputs. Similarly, Figure 18(c) reveals that during early training epochs (0-841), $\mathcal{D}_{\boldsymbol{\theta}}$ outputs progressively align with those of $\mathcal{D}_{\mathrm{G}}$. However, extended training beyond this point leads to memorization.

F.4 CIFAR-10 Results

The effects of model capacity and training duration on the Gaussian inductive bias, as demonstrated in Figures 19 to 21, extend to the CIFAR-10 dataset. These results confirm our findings from Section 4: the Gaussian inductive bias is most prominent when model scale and training duration are limited.

G Additional Experiment Results

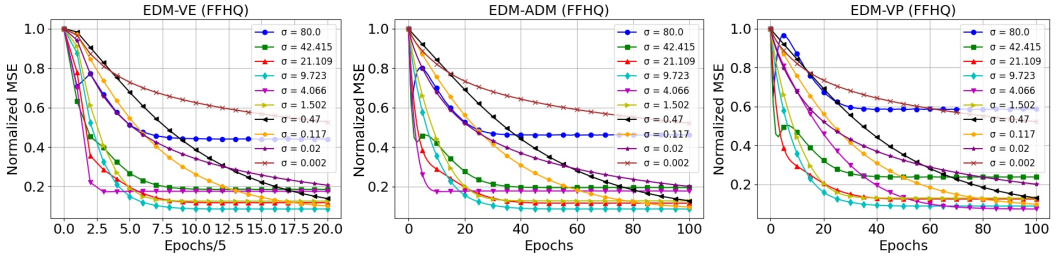
While in the main text we mainly demonstrate our findings using EDM-VE diffusion models trained on FFHQ, in this section we show our results are robust and extend to various model architectures and datasets. Furthermore, we demonstrate that the Gaussian inductive bias is not unique to diffusion models, but it is a fundamental property of denoising autoencoders [37]. Lastly, we verify that our



(a)

(b)

Figure 21: **Diffusion model learns the Gaussian structure in early training epochs.** Models with the same scale (channel size 128) are trained on a fixed CIFAR-10 datasets with 782 images. Figure(a) shows that the similarity between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ progressively increases during early training epochs (0-921) in the intermediate noise regime ($\sigma \in [0.1, 10]$), resulting in structurally similar generated images as shown in figure(b). However, continue training beyond this point results in diverged $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$, resulting in memorization.



(a)

(b)

(c)

Figure 22: **Linear model shares similar function mapping with Gaussian model.** The figures demonstrate the evolution of normalized MSE between the linear weights $\mathcal{D}_{\mathrm{L}}$ and the Gaussian weights $\mathcal{D}_{\mathrm{G}}$ w.r.t. linear distillation training epochs. Figures(a), (b) and (c) correspond to diffusion models trained on FFHQ, with EDM-VE, EDM-ADM and EDM-VP network architectures specified in [4] respectively.

conclusions remain consistent when using alternative metrics such as NMSE instead of the RMSE used in the main text.

G.1 Gaussian Structure Emerges across Various Network Architectures

We first demonstrate that diffusion models capture the Gaussian structure of the training dataset, irrespective of the deep network architectures used. As shown in Figure 22 (a), (b), and (c), although the actual diffusion models, $\mathcal{D}_{\boldsymbol{\theta}}$, are parameterized with different architectures, for all noise variances except $\sigma(t) \in \{0.002, 80.0\}$, their corresponding linear models, $\mathcal{D}_{\mathrm{L}}$, consistently converge towards the common Gaussian models, $\mathcal{D}_{\mathrm{G}}$, determined by the training dataset. Qualitatively, as depicted in Figure 23, despite variations in network architectures, diffusion models generate nearly identical images, matching those generated from the Gaussian models.

G.2 Gaussian Inductive Bias as a General Property of DAEs

In previous sections, we explored the properties of diffusion models by interpreting them as collections of deep denoisers, which are equivalent to the denoising autoencoders (DAEs) [37] trained on various noise variances by minimizing the denoising score matching objective (3). Although diffusion models and DAEs are equivalent in the sense that both of them are trying to learn the score function of the

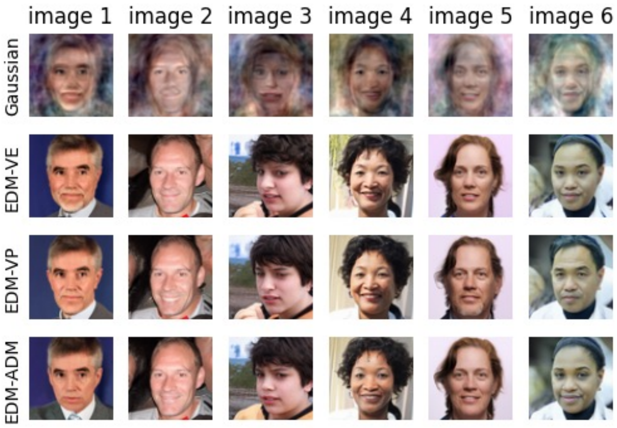


Figure 23: **Images sampled from various model.**The figure shows the sampled images from diffusion models with different network architectures and those from their corresponding Gaussian models.

noise-mollified data distribution [38], the training objective of diffusion models is more complex [4]:

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x},\boldsymbol{\epsilon},\sigma}[\lambda(\sigma)c_{\text{out}}(\sigma)^2||\mathcal{F}_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\epsilon},\sigma) - \underbrace{\frac{1}{c_{\text{out}}(\sigma)}(\boldsymbol{x} - c_{\text{skip}}(\sigma)(\boldsymbol{x}+\boldsymbol{\epsilon}))}_{\text{linear combination of } \boldsymbol{x} \text{ and } \boldsymbol{\epsilon}}||_2^2], \tag{52}$$

where $\boldsymbol{x} \sim p_{\text{data}}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma(t)^2\boldsymbol{I})$ and $\sigma \sim p_{\text{train}}$. Notice that the training objective of diffusion models has a few distinct characteristics:

- Diffusion models use a single deep network $\mathcal{F}_{\boldsymbol{\theta}}$ to perform denoising score matching across all noise variances while DAEs are typically trained separately for each noise level.
- Diffusion models are not trained uniformly across all noise variances. Instead, during training the probability of sampling a given noise level σ is controlled by a predefined distribution p_{train} and the loss is weighted by $\lambda(\sigma)$.
- Diffusion models often utilize special parameterizations (13). Therefore, the deep network $\mathcal{F}_{\boldsymbol{\theta}}$ is trained to predict a linear combination of the clean image $\boldsymbol{x}$ and the noise $\boldsymbol{\epsilon}$, whereas DAEs typically predict the clean image directly.

Given these differences, we investigate whether the Gaussian inductive bias is unique to diffusion models or a general characteristic of DAEs. To this end, we train separate DAEs (deep denoisers) using the vanilla denoising score matching objective (3) on each of the 10 discrete noise variances specified by the EDM schedule [80.0, 42.415, 21.108, 9.723, 4.06, 1.501, 0.469, 0.116, 0.020, 0.002], and compare the score differences between them and the corresponding Gaussian denoisers $\mathcal{D}_{\text{G}}$. We use no special parameterization so that $\mathcal{D}_{\boldsymbol{\theta}} = \mathcal{F}_{\boldsymbol{\theta}}$; that is, the deep network directly predicts the clean image. Furthermore, the DAEs for each noise variance are trained till convergence, ensuring all noise levels are trained sufficiently. We consider the following architectural choices:

- *DAE-NCSN*: In this setting, the network $\mathcal{F}_{\boldsymbol{\theta}}$ uses the NCSN architecture [3], the same as that used in the EDM-VE diffusion model.
- *DAE-Skip*: In this setting, $\mathcal{F}_{\boldsymbol{\theta}}$ is a U-Net [39] consisting of convolutional layers, batch normalization [40], leaky ReLU activation [41] and convolutional skip connections. We refer to this network as ”Skip-Net”. Compared to NCSN, which adapts the state of the art architecture designs, Skip-Net is deliberately constructed to be as simple as possible to test how architectural complexity affects the Gaussian inductive bias.
- *DAE-DiT*: In this setting, $\mathcal{F}_{\boldsymbol{\theta}}$ is a Diffusion Transformer (DiT) introduced in [42]. Vision Transformers are known to lack inductive biases such as locality and translation equivariance that are inherent to convolutional models [43]. Here we are interested in if this affects the Gaussian inductive bias.

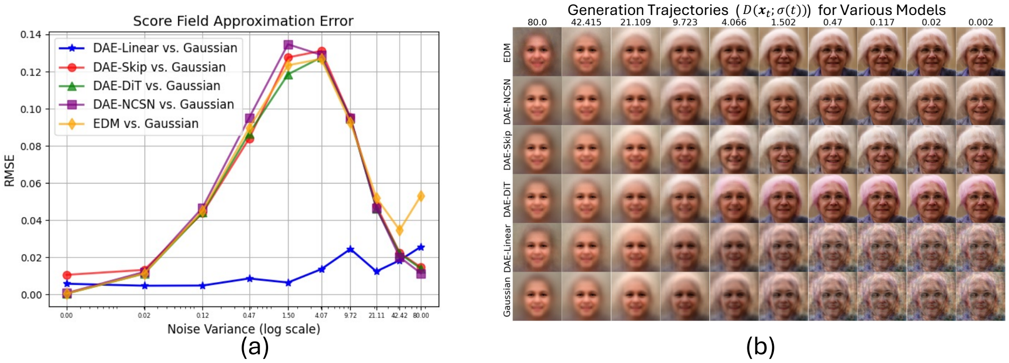


Figure 24: **Comparison between DAEs and diffusion models.** Figure(a) compares the score field approximation error between Gaussian models and both *(i)* diffusion models (EDM vs. Gaussian) and *(ii)* DAEs with varying architectures. Figure(b) illustrates the generation trajectories of different models initialized from the same noise input.

- *DAE-Linear*: In this setting we set $\mathcal{F}_{\boldsymbol{\theta}}$ to be a linear model with a bias term as in (8). According to Theorem 1, these models should converge to Gaussian denoisers.

The quantitative results are shown in Figure 24(a). First, the DAE-linear models well approximate $\mathcal{D}_{\mathrm{G}}$ across all 10 discrete steps (RMSE smaller than 0.04), consistent with Theorem 1. Second, despite the differences between diffusion models (EDM) and DAEs, they achieve similar score approximation errors relative to $\mathcal{D}_{\mathrm{G}}$ for most noise variances, meaning that they can be similarly approximated by $\mathcal{D}_{\mathrm{G}}$. However, diffusion models exhibit significantly larger deviations from $\mathcal{D}_{\mathrm{G}}$ at higher noise variances ($\sigma \in \{42.415, 80.0\}$) since they utilize a bell-shaped noise sampling distribution p_{train} that emphasizes training on intermediate noise levels, leading to under-training at high noise variances. Lastly, the DAEs with different architectures achieve comparable score approximation errors, and both DAEs and diffusion models generate images matching those from the Gaussian model, as shown in Figure 24(b). These findings demonstrate that the Gaussian inductive bias is not unique to diffusion models or specific architectures but is a fundamental property of DAEs.

G.3 Gaussian Structure Emerges across Various datasets

As illustrated in Figure 25, for diffusion models trained on the CIFAR-10, AFHQ and LSUN-Churches datasets that are in the generalization regime, their generated samples match those produced by the corresponding Gaussian models. Additionally, their linear approximations, $\mathcal{D}_{\mathrm{L}}$, obtained through linear distillation, align closely with the Gaussian models, $\mathcal{D}_{\mathrm{G}}$, resulting in nearly identical generated images. These findings confirm that the Gaussian structure is prevalent across various datasets.

G.4 Strong Generalization on CIFAR-10

Figure 26 demonstrates the strong generalization effect on CIFAR-10. Similar to the observations in Section 5, reducing model capacity or early stopping the training process prompts the Gaussian inductive bias, leading to generalization.

G.5 Measuring Score Approximation Error with NMSE

While in Section 3.1 we define the score field approximation error between denoisers $\mathcal{D}_1$ and $\mathcal{D}_2$ with RMSE ((10)), this error can also be quantified using NMSE:

$$\text{Score-Difference}(t) := \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x}), \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}; \sigma(t)^2 \boldsymbol{I})} \frac{||\mathcal{D}_1(\boldsymbol{x}+\boldsymbol{\epsilon}) - \mathcal{D}_2(\boldsymbol{x}+\boldsymbol{\epsilon})||_2}{||\mathcal{D}_1(\boldsymbol{x}+\boldsymbol{\epsilon})||_2}. \tag{53}$$

As shown in Figure 27, while the trend in intermediate-noise and low-noise regimes remains unchanged, NMSE amplifies differences in the high-noise variance regime compared to RMSE. This amplified score difference between $\mathcal{D}_{\mathrm{G}}$ and $\mathcal{D}_{\boldsymbol{\theta}}$ does not contradict our main finding that diffusion models in the generalization regime exhibit an inductive bias towards learning denoisers

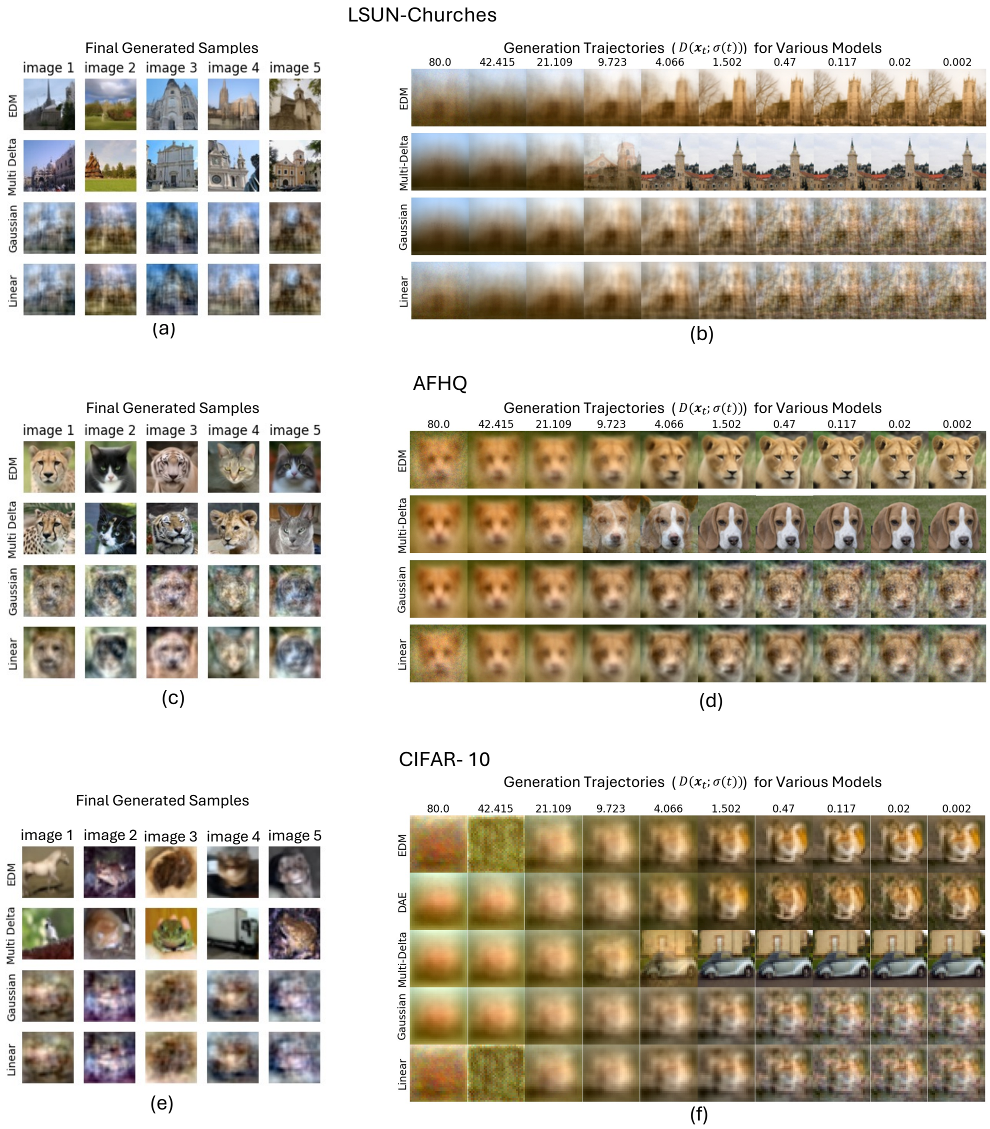


Figure 25: **Final generated images and sampling trajectories for various models.** Figures(a), (c) and (e) demonstrate the images generated using different models starting from the same noises for LSUN-Churches, AFHQ and CIFAR-10 respectively. Figures(b), (d) and (f) demonstrate the corresponding sampling trajectories.

approximately equivalent to $\mathcal{D}_{\mathrm{G}}$ in the high-noise variance regime. As discussed in Section 3.2 and appendices F.2 and G.2, this large score difference stems from inadequate training in this regime.

Figure 27 (Gaussian vs. DAE) demonstrates that when DAEs are sufficiently trained at specific noise variances, they still converge to $\mathcal{D}_{\mathrm{G}}$. Importantly, the insufficient training in the high-noise variance regime minimally affects final generation quality. Figure 25(f) shows that while the diffusion model (EDM) produces noisy trajectories at early timesteps ($\sigma \in \{80.0, 42.415\}$), these artifacts quickly disappear in later stages, indicating that the Gaussian inductive bias is most influential in the intermediate-noise variance regime.

Notably, even when $\mathcal{D}_{\boldsymbol{\theta}}$ are inadequately trained in the high-noise variance regime, they remain approximable by linear functions, though these functions no longer match $\mathcal{D}_{\mathrm{G}}$.

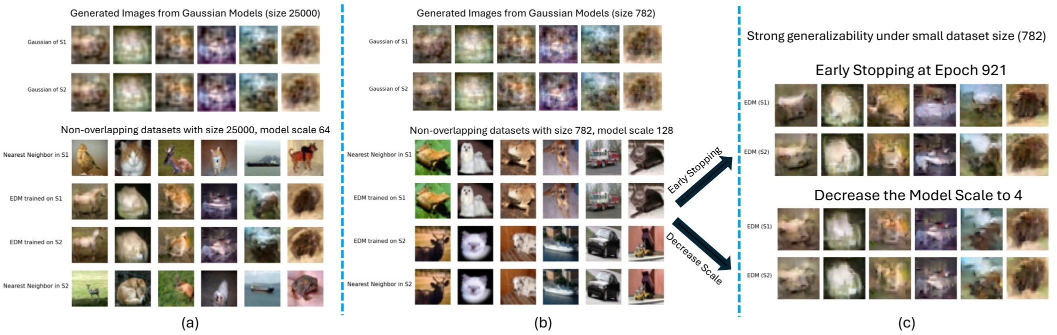

Generated Images from Gaussian Models (size 25000)
Gaussian of S1
Gaussian of S2
Non-overlapping datasets with size 25000, model scale 64
Nearest Neighbor in S1
EDM trained on S1
EDM trained on S2
Nearest Neighbor in S2
(a)
Generated Images from Gaussian Models (size 782)
Gaussian of S1
Gaussian of S2
Non-overlapping datasets with size 782, model scale 128
Nearest Neighbor in S1
EDM trained on S1
EDM trained on S2
Nearest Neighbor in S2
(b)
Early Stopping
Decrease Scale
Strong generalizability under small dataset size (782)
Early Stopping at Epoch 921
EDM (S1)
EDM (S2)
Decrease the Model Scale to 4
EDM (S1)
EDM (S2)
(c)

Figure 26: **Strong generalization on CIFAR-10 dataset.** Figure(a) Top: Generated images of Gausisan models; Bottom: Generated images of diffusion models, with model scale 64; S1 and S2 each has 25000 non-overlapping images. Figure(b) Top: Generated images of Gausisan model; Bottom: Generated images of diffusion models in the memorization regime, with model scale 128; S1 and S2 each has 782 non-overlapping images. Figure(c): Early stopping and reducing model capacity help transition diffusion models from memorization to generalization.

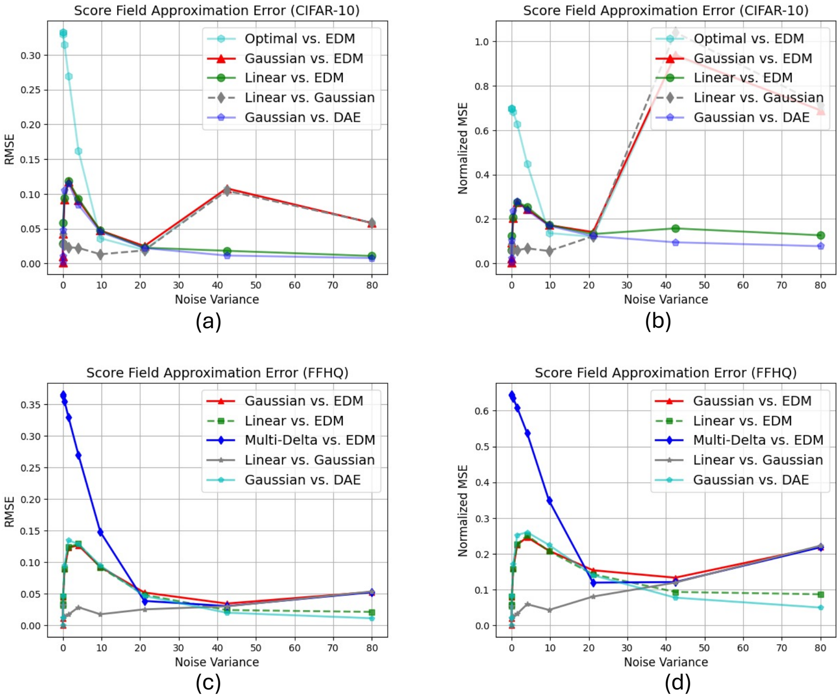


Figure 27: **Comparison between RMSE and NMSE score differences.** Figures(a) and (c) show the score field approximation errors measured with RMSE loss while figures(b) and (d) show these errors measured using NMSE loss. Compared to RMSE, the NMSE metric highlight the score differences in the high-noise regime, where diffusion models receive the least training.

H Discussion on Geometry-Adaptive Harmonic Bases

H.1 GAHB only Partially Explain the Strong Generalization

Recent work [20] observes that diffusion models trained on sufficiently large non-overlapping datasets (of the same class) generate nearly identical images. They explain this ”strong generalization” phenomenon by analyzing bias-free deep diffusion denoisers with piecewise linear input-output mappings:

$$\mathcal{D}(\boldsymbol{x}_t;\sigma(t)) = \nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))\boldsymbol{x} \tag{54}$$

$$= \sum_k \lambda_k(\boldsymbol{x}_t)\boldsymbol{u}_k(\boldsymbol{x}_t)\boldsymbol{v}_k^T(\boldsymbol{x}_t)\boldsymbol{x}_t, \tag{55}$$

where $\lambda_k(\boldsymbol{x}_t)$, $\boldsymbol{u}_k(\boldsymbol{x}_t)$, and $\boldsymbol{v}_k(\boldsymbol{x}_t)$ represent the input-dependent singular values, left and right singular vectors of the network Jacobian $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$. Under this framework, strong generalization occurs when two denoisers $\mathcal{D}_1$ and $\mathcal{D}_2$ have similar Jacobians: $\nabla\mathcal{D}_1(\boldsymbol{x}_t;\sigma(t)) \approx \nabla\mathcal{D}_2(\boldsymbol{x}_t;\sigma(t))$. The authors conjecture this similarity arises from networks' inductive bias towards learning certain optimal $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ that has sparse singular values and the singular vectors of which are the geometry-adaptive harmonic bases (GAHB)—near-optimal denoising bases that adapt to input $\boldsymbol{x}_t$.

While [20] provides valuable insights, their bias-free assumption does not reflect real-world diffusion models, which inherently contain bias terms. For feed forward ReLU networks, the denoisers are piecewise affine:

$$\mathcal{D}(\boldsymbol{x}_t;\sigma(t)) = \nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))\boldsymbol{x}_t + \boldsymbol{b}_{\boldsymbol{x}_t}, \tag{56}$$

where $\boldsymbol{b}_{\boldsymbol{x}_t}$ is the network bias that depends on both network parameterization and the noisy input $\boldsymbol{x}_t$ [44]. Here, similar Jacobians alone cannot explain strong generalization, as networks may differ significantly in $\boldsymbol{b}_{\boldsymbol{x}_t}$. For more complex network architectures where even piecewise affinity fails, we consider the local linear expansion of $\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$:

$$\mathcal{D}(\boldsymbol{x}_t + \Delta\boldsymbol{x};\sigma(t)) = \nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))\Delta\boldsymbol{x}_t + \mathcal{D}(\boldsymbol{x}_t;\sigma(t)), \tag{57}$$

which approximately holds for small perturbation $\Delta\boldsymbol{x}$. Thus, although $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ characterizes $\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$'s local behavior around $\boldsymbol{x}_t$, it does not provide sufficient information on the global properties.

Our work instead examines global behavior, demonstrating that $\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ is close to $\mathcal{D}_{\mathrm{G}}(\boldsymbol{x}_t;\sigma(t))$—the optimal linear denoiser under the Gaussian data assumption. This implies that strong generalization partially stems from networks learning similar Gaussian structures across non-overlapping datasets of the same class. Since our linear model captures global properties but not local characteristics, it complements the local analysis in [20].

H.2 GAHB Emerge only in Intermediate-Noise Regime

For completeness, we study the evolution of the Jacobian matrix $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ across various noise levels $\sigma(t)$. The results are presented in Figures 28 and 29, which reveal three distinct regimes:

- *High-noise regime [10,80].* In this regime, the leading singular vectors[6] of the Jacobian matrix $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ well align with those of the Gaussian weights (the leading principal components of the training dataset), consistent with our finding that diffusion denoisers approximate linear Gaussian denoisers in this regime. Notice that DAEs trained sufficiently on separate noise levels (Figure 29) show stronger alignment compared to vanilla diffusion models (Figure 28), which suffer from insufficient training at high noise levels.
- *Intermediate-noise regime [0.1,10]:* In this regime, GAHB emerge as singular vectors of $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ diverge from the principal components, becoming increasingly adaptive to the geometry of input image.
- *Low-noise regime [0.002,0.1].* In this regime, the leading singular vectors of $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ show no clear patterns, consistent with our observation that diffusion denoisers approach the identical mapping, which has unconstrained singular vectors.

Notice that the leading singular vectors of $\nabla\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ are the input directions that lead to the maximum variation in denoised outputs, thus revealing meaningful information on the local properties of $\mathcal{D}(\boldsymbol{x}_t;\sigma(t))$ at $\boldsymbol{x}_t$. As demonstrated in Figure 30, perturbing input $\boldsymbol{x}_t$ along these vectors at difference noise regimes leads to distinct effects on the final generated images: *(i)* in the high-noise regime where the leading singular vectors align with the principal components of the training dataset,

[6] We only care about leading singular vectors since the Jacobians in this regime are highly low-rank. The less well aligned singular vectors have singular values near 0.

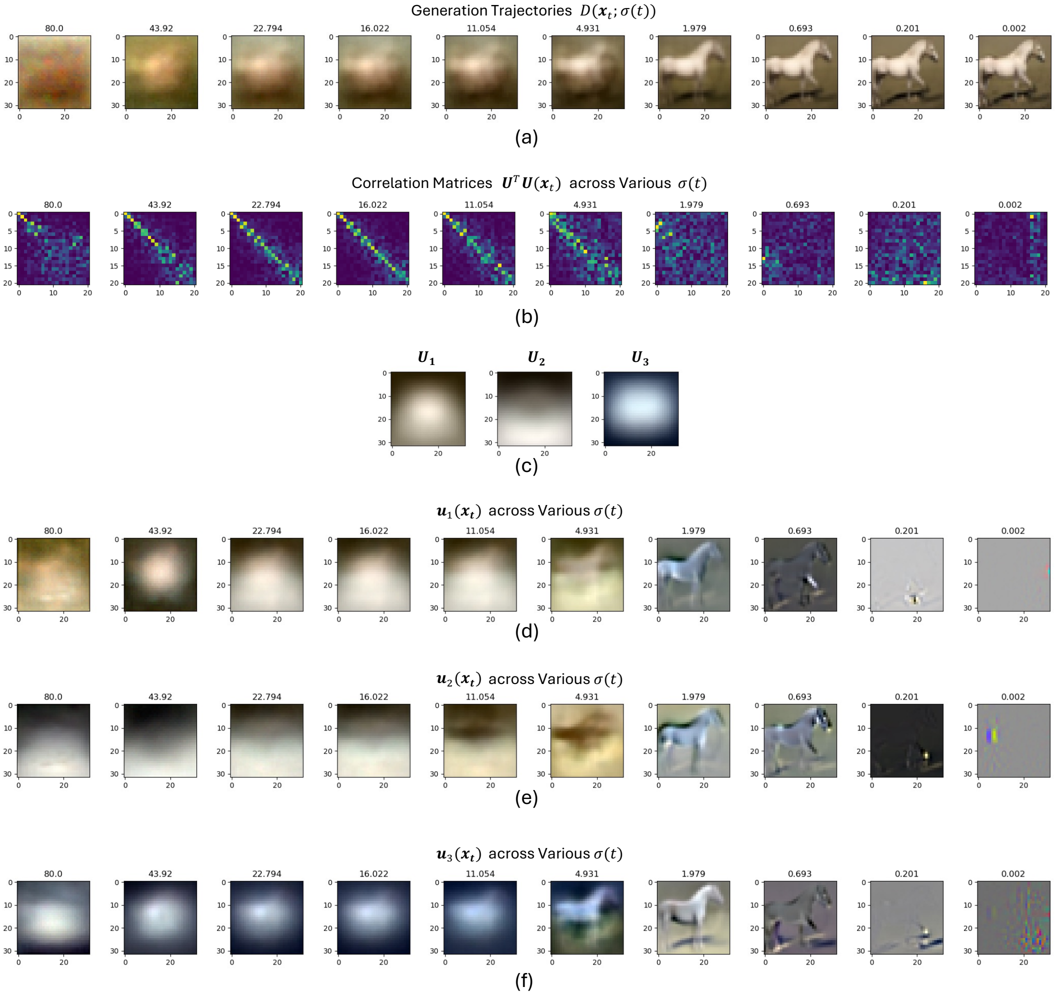


Figure 28: **Evolution of $\nabla\mathcal{D}(\boldsymbol{x}_t; \sigma(t))$ across varying noise levels.** Figure(a) shows the generation trajectory. Figure(b) shows the correlation matrix between Jacobian singular vectors $\boldsymbol{U}(\boldsymbol{x}_t)$ and training dataset principal components $\boldsymbol{U}$. Notice that the leading singular vectors of $\boldsymbol{U}(\boldsymbol{x}_t)$ and $\boldsymbol{U}$ well align in early timesteps but diverge in later timesteps. Figure(c) shows the first three principal components of the training dataset while figures(d-f) show the evolution of Jacobian's first three singular vectors across noise levels. These singular vectors initially match the principal components but progressively adapt to input image geometry, before losing distinct patterns at very low noise levels. While we present only left singular vectors, right singular vectors exhibit nearly identical behavior and yield equivalent results.

perturbing $\boldsymbol{x}_t$ along these directions leads to canonical changes such as image class, *(ii)* in the intermediate-noise regime where the GAHB emerge, perturbing $\boldsymbol{x}_t$ along the leading singular vectors modify image details such as colors while preserving overall image structure and *(iii)* in the low-noise regime where the leading singular vectors have no significant pattern, perturbing $\boldsymbol{x}_t$ along these directions yield no meaningful semantic changes.

These results collectively demonstrate that the singular vectors of the network Jacobian $\nabla\mathcal{D}(\boldsymbol{x}_t; \sigma(t))$ have distinct properties at different noise regimes, with GAHB emerging specifically in the intermediate regime. This characterization has significant implications for uncertainty quantification [45] and image editing [46].

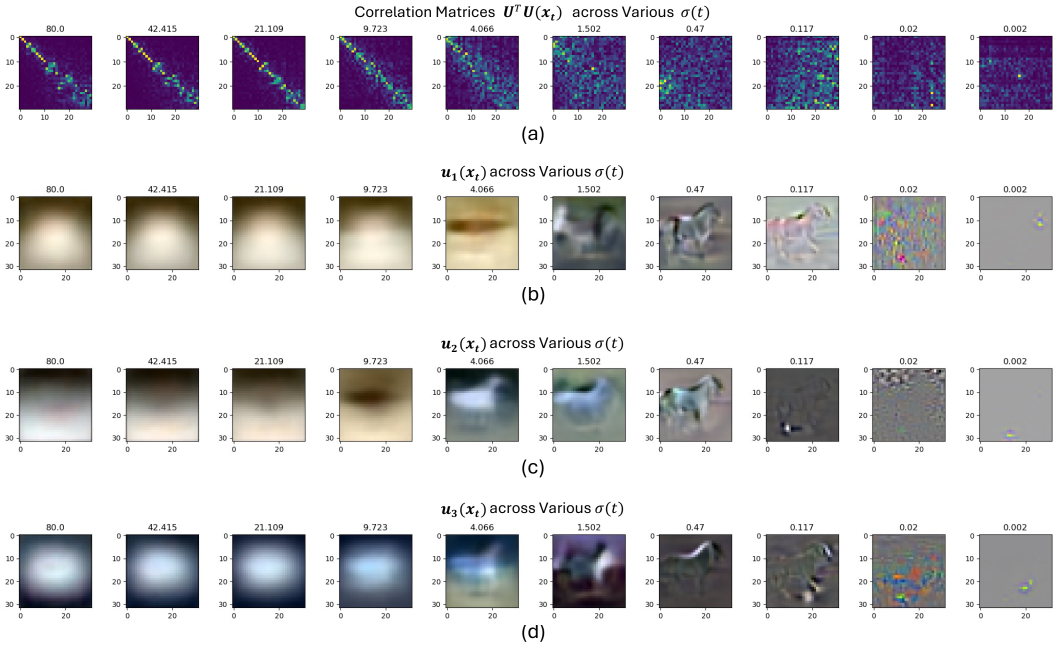


Figure 29: **Evolution of $\nabla\mathcal{D}(\boldsymbol{x}_t; \sigma(t))$ across varying noise levels for DAEs.** We repeat the experiments in Figure 28 on DAEs that are sufficiently trained on each discrete noise levels. Notice that with sufficient training, the Jacobian singular vectors $\boldsymbol{U}(\boldsymbol{x}_t)$ show a better alignment with principal components $\boldsymbol{U}$ in early timesteps.

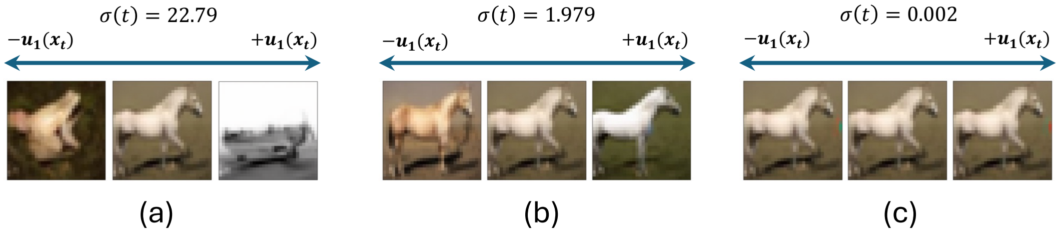

$\sigma(t) = 22.79$
$-\boldsymbol{u}_1(\boldsymbol{x}_t)$
$+\boldsymbol{u}_1(\boldsymbol{x}_t)$
$\sigma(t) = 1.979$
$\sigma(t) = 0.002$
(a)
(b)
(c)

Figure 30: **Effects of perturbing $\boldsymbol{x}_t$ along Jacobian singular vectors.** Figure(a)-(c) demonstrate the effects of perturbing input $\boldsymbol{x}_t$ along the first singular vector of the Jacobian matrix ($\boldsymbol{x}_t \pm \lambda\boldsymbol{u}_1(\boldsymbol{x}_t)$) on the final generated images. Perturbing $\boldsymbol{x}_t$ in high-noise regime (Figure (a)) leads to canonical image changes while perturbation in intermediate-noise regime (Figure (b)) leads to change in details but the overall image structure is preserved. At very low noise variances, perturbation has no significant effect (Figure (c)). Similar effects are observed in concurrent work [46].

I Computing Resources

All the diffusion models in the experiments are trained on A100 GPUs provided by NCSA Delta GPU [33].

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and precede the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While ”[Yes] ” is generally preferable to ”[No] ”, it is perfectly acceptable to answer ”[No] ” provided a proper justification is given (e.g., ”error bars are not reported because it would be too computationally expensive” or ”we were unable to find the license for the dataset we used”). In general, answering ”[No] ” or ”[NA] ” is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS paper checklist”**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope?

   Answer: [Yes]

   Justification: We mention the main contribution in both the abstract and introduction section, listed in the ”Contributions”.

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the ”Discussion” section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We provide the assumption and proof in Section 3.3 and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include the experiment details in Section 3, Section 4 and Section 5. We will also release our code upon publication.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).

- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We include the experiment details in Section 3, Section 4 and Section 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: For various plots we specify that the numbers are calculated over 100 random samples. However, since the standard deviations are quite small, the error bar is unnecessary.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer ”Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the computating resources in Appendix I.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our research doesn't have potential harm cause by research process and negative societal impact.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work is an empirical and theoretical work on the inductive bias of the diffusion models. It has more scientific contributions rather than societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our work is an empirical and theoretical work on how the diffusion model learns the data distribution. It doesn't have a high risk of misuse.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use the public released dataset and models, which properly credited the license.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: Our work is an empirical and theoretical work on how the diffusion model learns the data distribution. We don't release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.