The Diffusion Process as a Correlation Machine: Linear Denoising Insights

Anonymous authors
Paper under double-blind review

Abstract

Recently, diffusion models have gained popularity due to their impressive generative abilities. These models learn the implicit distribution given by a training dataset, and sample new data by transforming random noise through the reverse process, which can be thought of as gradual denoising. In this work, to shed more light on the evolution of denoisers in the reverse process, we examine the generation process as a "correlation machine", where random noise is repeatedly enhanced in correlation with the implicit given distribution. To this end, we explore the linear case, where the optimal denoiser in the MSE sense is known to be the PCA projection. This enables us to connect the theory of diffusion models to the spiked covariance model, where the dependence of the denoiser on the noise level and the amount of training data can be expressed analytically, in the rank-1 case. In a series of numerical experiments, we extend this result to general low rank data, and show that low frequencies emerge earlier in the generation process, where the denoising basis vectors are more aligned to the true data with a rate depending on their eigenvalues. This model allows us to show that the linear reverse process is a generalization of the prevalent power iteration method, where the generated distribution is composed of several estimations of the given covariance, in varying stages of convergence. Finally, we empirically demonstrate the applicability of our findings beyond the linear case, in the Jacobians of a deep, non-linear denoiser, used in general image generation tasks.

1 Introduction

Recently, diffusion models have gained much popularity as very successful generative models, showcasing impressive performance in image generation tasks (18; 19; 43; 45). These models learn the implicit distribution given by a training dataset and sample new data by transforming random noise inputs through a reverse diffusion process, which can be thought of as gradual denoising. More formally, it has been shown in (23) that learning the underlying distribution is equivalent to optimal denoising at all noise levels.

In order to shed more light onto the mechanism behind the success of diffusion models, in this work we analyze the behavior of denoisers in the context of image generation, where pure noise is gradually processed into a sample from a given (implicit) distribution by gradual denoising. Unlike other works, e.g. (23), we focus on the denoisers throughout the generation process, and not only on the final generated data.

To this end, we suggest the following simple model to illustrate our point. Consider the class of linear denoisers, where the optimal denoiser in MSE sense has a closed-form solution. We explore two linear denoising trajectories, corresponding to the DDPM (19) and DDIM (41) approaches of sampling. To simulate the diffusion generation process, we learn a series of projections onto noisy data at different noise levels, and use them to transform pure noise into samples from the underlying distribution. Given this simple model we can inspect the evolution of eigenvectors spanning gradual projections with decreasing noise levels, as well as the distribution of the generated data samples.

We show that the correlation of the noisy basis eigenvectors with their clean version decays as the noise level increases, with a rate determined by the eigenvalues and the size of the training dataset. In other words, we show that low frequencies, corresponding to large eigenvalues, emerge earlier in the reverse process, as empirically observed in (19), and analyze how more training data contribute to generalization (23). Analytically, this corresponds to the spiked covariance model (22), in which we bound this decay to the leading eigenvector (corresponding to the largest eigenvalue).

Next, we demonstrate the applicability of our findings to more general, non-linear deep denoisers. Although the network is not linear, its application can be written as a linear operation of the Jacobian calculated on the input image. We empirically show that the aforementioned decay of eigenvector correlations is prevalent also in the Jacobians of a deep denoiser, in the final stages of image generation, thus showing the relevance of our analysis in a broader context, and not just in a simplified linear case.

2 Background and Related Work

Since their introduction in (40), diffusion models have been vastly used in image generation tasks (18; 19; 43; 45), more general computer vision tasks (2; 5; 10; 11), and in other domains such as natural language processing (4; 20; 29; 39; 51) and temporal data modeling (1; 14; 25; 38; 48). On top of their practical success, different flavors of training and sampling have risen based on interesting theoretical reasoning, e.g., considering the statistical properties of the intermediate data (41; 40), or by framing the problem in the form of stochastic differential equations (SDEs) (24; 42; 45; 12) or score based generative models (43; 44). In this work, we look at diffusion models in the context of iterative denoising, and focus on the properties of the learned denoiser (32).

Recently, the work in (23) showed that the learned denoising functions are equivalent to a shrinkage operation in a basis adapted to the underlying image. In this sense, the diffusion denoiser is an adaptive filter (31; 46; 47). While they focus on the analysis of the nonlinear denoiser at the point of the final generated data, we are interested in the evolution (adaptation) of the denoiser throughout the generation process, and its dependence on the noise level. To this end, we suggest a simple linear denoising model, presented in Section 3. In this case, the (optimal) denoiser does depend on the underlying image, and its dependence on the noise level can be traced analytically, as we show hereafter.

Due to their phenomenal empirical success, some attempts have been devoted towards providing theory supporting the sample and iteration complexity of diffusion models. The current body of work can be generally parted to attaining iteration complexity bounds assuming approximately accurate scores (27; 26; 15; 21; 6), and to assessing the sample complexity to learn the score functions (13; 8; 7). Among these works, many assume a low dimensional data distribution (9; 28; 36; 13; 50), which is a reasonable assumption in practice (see e.g., (37)). Yet, it might particularly explain the gap between the current iteration bounds and the much lower complexity apparent in practice (28). In our work, we consider linear models and deduce a linear sample complexity bound associated with learning the score function in Sec. 4 and discuss the trade-offs of the synthesis conversion rate in Sec. 4.1. The previous works mentioned above mainly develop bounds assuming specific samplers and scaling details, which differ from our setting. In addition, they generally bound the Total Variation distance (under varying assumptions on the target distributions), which is not trivial to translate to the generated covariance matrix that we focus on even in the linear Gaussian case (17). The difference in our setting enables us to connect the theory of diffusion models to a broad body of work concerning the spiked covariance model (22), and supports the analysis of denoising diffusion as a correlation machine, which is the main purpose of this paper.

In the setting of Statistical Mechanics, the work in (7) analyses diffusion models in very large dimensions, focusing on the Curie-Weiss model of ferromagnetism. As an introduction to their work, they also discuss a simple linear score model, in the context of the sample complexity of learning the score function. They focus their discussion on the case of Gaussian data, where the eigenvalues of the covariance matrices can be typically characterized. Relatedly, the work in (49) recently showed that the learned neural score is dominated by its Gaussian approximation for moderate to high noise scales, and supply both theoretical and empirical arguments to support this claim. Compared to these works, we consider data that reside in a low dimensional subspace, with no specific distribution, described in Sec. 4. We limit the denoiser to be linear and focus on two stochastic sampling trajectories, which give rise to the spiked covariance model.

Power iteration is a fundamental algorithm for approximating the dominant eigenvalue and eigenvector of a matrix. It relies on iteratively multiplying an initial vector by the matrix, where its convergence rate is proportional to the ratio of the largest and second-largest eigenvalues. The method's simplicity and scalability have made it a cornerstone in various fields, including numerical linear algebra, machine learning, and graph theory. For the ease of reading, we include a formal presentation of the method and discuss its convergence in Appendix A. In this work, we shall show how a linear denoising chain converges in mean to the celebrated power iteration method.

3 Linear Diffusion - Problem Setup

For our analysis, we define the following simple iterative linear generation model. First, define the standard diffusion model. Let q denote the natural data distribution and let $x_0 \sim q$ be a sample from the natural data ($x \in \mathbb{R}^d$). The forward (diffusion) process is defined (19) by

$$q(x_t|x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{1}$$

for some fixed noise schedule $\{\beta_t\}_{t=1}^T$ and $x_0 \sim q$. It can be shown that

$$q(x_t|x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I}), \tag{2}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \Pi_{s=1}^t \alpha_s$. For our simplified model, consider the process (without scaling),

$$q(x_t|x_{t-1}) = \mathcal{N}(x_{t-1}, \sigma_t^2\mathbf{I}). \tag{3}$$

This implies that $x_t = x_{t-1} + \epsilon_{\sigma_t}$, where $\epsilon_{\sigma_t} \sim \mathcal{N}(0, \sigma_t^2\mathbf{I})$ for some fixed noise schedule $\{\sigma_t\}_{t=1}^T$. We discard the scaling to comply with previous analysis of the spiked covariance model (34) (more details in Section 4). This corresponds to the "Exploding Variance" formulation, used with Langevin dynamics to sample data as a variant of score based diffusion models (43; 45; 44). We choose to present the "standard" diffusion models in the setting of denoising diffusion (19) and not using the score-based approach entirely, as we focus our discussion on the qualities of the denoiser.

The reverse (generation) process is defined using a parameterized distribution model p_θ, generally defined by the Markov process

$$p_\theta(x_{0:T}) = p(x_T)\Pi_{t=1}^T p_\theta(x_{t-1}|x_t), \tag{4}$$

$$p_\theta(x_{t-1}|x_t) \triangleq \mathcal{N}(\mu_\theta(x_t,t), \Sigma_\theta(x_t,t)), \tag{5}$$

where $p(x_T) = \mathcal{N}(0, \mathbf{I})$. By choices of parametrization and loss manipulations (see (19)), one generally learns to estimate the error $\epsilon_\theta(x_t,t)$, where

$$\mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\Big), \tag{6}$$

$\Sigma_\theta(x_t,t) = e_t^2\mathbf{I}$, and e_t is a designed schedule (usually chosen to be equal to σ_t). Thus, the reverse process can be expressed as a denoising chain

$$D_t(x_t) = \frac{1}{\sqrt{\alpha_t}}\Big(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\Big) + e_t z, \tag{7}$$

where $z \sim \mathcal{N}(0, \mathbf{I})$ and $z_1 = 0$. This is a stochastic denoiser which preserves the Markovian property of the forward process. Later versions suggested similar (non-Markovian) deterministic denoisers, e.g., DDIM (41), or more general stochastic denoiser chains, for a continuous forward model (InDI (16)).

In our case, we restrict the denoisers to be a linear function of x_t. For the reverse process, we shall now define two linear denoising trajectories, corresponding to different approaches of diffusion models. Considering an intermediate denoising step from $t+1 \to t$, the optimal linear denoiser D_t in the ℓ_2 sense is the minimizer of the loss

$$\ell_{t+1\to t} = \mathbb{E}_{x_t,w}\left\|D_t(x_t + \sigma_t w) - x_t\right\|_2^2, \tag{8}$$

where $w \sim \mathcal{N}(0, \mathbb{I})$, given by

$$D_t = (\Sigma_t + \sigma_t^2 \mathbb{I})^{-1} \Sigma_t \tag{9}$$

(full derivation in Appendix B). Notice, that in the limit of diminishing σ_t,

$$D_t = U_t \begin{pmatrix} \frac{\lambda_0}{\lambda_0 + \sigma_t^2} & & \\ & \ddots & \\ & & \frac{\lambda_{r-1}}{\lambda_{r-1} + \sigma_t^2} \end{pmatrix} U_t^\dagger \underset{\sigma_t \to 0}{\rightarrow} U_t U_t^\dagger = D_{\text{PCA}}^t, \tag{10}$$

where U_t is the diagonalizing basis of Σ_t (more on that in Sec. 4). Thus, for the reverse process, we learn a simple PCA denoiser (projection) based on X_{t-1}, which is the cleaner version of the training set $X = \{x_1, ..., x_n\}$ at time $t-1$. At each time step we learn

$$D_{PCA}^t(x_t) = D_{PCA}^t(x_t; X_{t-1}) = P_t(x_t; X_0 + E_{\bar{\sigma}_t}), \tag{11}$$

where each column in $E_{\bar{\sigma}_t}$ is distributed by $\mathcal{N}(0, \bar{\sigma}_t^2 \mathbf{I})$ and $\bar{\sigma}_t$ is a function of $\{\sigma_s\}_{s=1}^t$. Our sampling process is based on the sequential application of $P_t \in \mathbb{R}^{d \times d}$, which is the projection on perturbed principal components with respect to the clean data distribution q, followed by the addition of noise with variance σ_t. This generation path is in the spirit of (19), with its gradual temporal denoising. However, D_{PCA}^t is a deterministic denoiser given the sampling of training data and noise, which does not depend either on x_t nor on x_0. In Section 4, we analyze the change in D_{PCA}^t over time, to study its evolution along the generation trajectory.

Given a similar ℓ_2 loss (to Equation 8), an alternative denoising chain can use multiple estimations of x_0, in the essence of (41). The corresponding loss is thus

$$\ell_{t \to 0} = \mathbb{E}_{x_t, w} \left\| D_t(x_0 + \bar{\sigma}_t w) - x_0 \right\|_2^2, \tag{12}$$

where $\bar{\sigma}_t$ is the overall added noise (see Section 4). The adequate denoising chain in this case is the application of D_t to estimate x_0, followed by the addition of noise with the appropriate variance $\bar{\sigma}_{t-1}^2$, before the iterative application of D_{t-1}. In this case, the optimal denoiser is given by

$$D_t = (\Sigma_0 + \bar{\sigma}_t^2 \mathbb{I})^{-1} \Sigma_0. \tag{13}$$

Despite the different approaches the two paths represent, their resulting denoising chains exhibit similar properties - in both cases, the appearance of frequencies in the generated images is gradual, where low frequencies are first to emerge.

Empirical Demonstration of a Linear Diffusion Model. To illustrate the forward and backward processes in the linear case, we perform a numerical simulation using the MNIST dataset, which is simple enough to be estimated via a linear model. We start here with the training and generation procedures, and use the same setting and trained denoisers to demonstrate our findings throughout the paper.

In the following experiment we simulate the process described above using the MNIST dataset (we use the default train / test splits). In the class conditioned case, we learn a PCA denoiser with 30 components for each time step where $x_t = x_{t-1} + \epsilon_{\sigma_t}$, $\sigma_t \propto t$, and $T = 65$ iterations. Figure 1 shows a (decimated) example of digit generation from pure noise, where we apply the sequence of denoisers $D_{PCA}^t = P_t$, which will be more accurately defined in Section 4. In order to understand the reverse process, we now turn to analyze the gradual change of P_t, that might be expressed by the angle between the clean and noisy components over time.

Notations. We use A_t to denote the matrix A at time t, and a_i^t to denote the ith column of A_t.

4 Linear Diffusion as Basis Perturbation

We now turn to analyze the linear model presented above and show how the generation process can be seen as a kernel “correlation machine”. Specifically, we are interested in the temporal (i.e., noise level) dependence

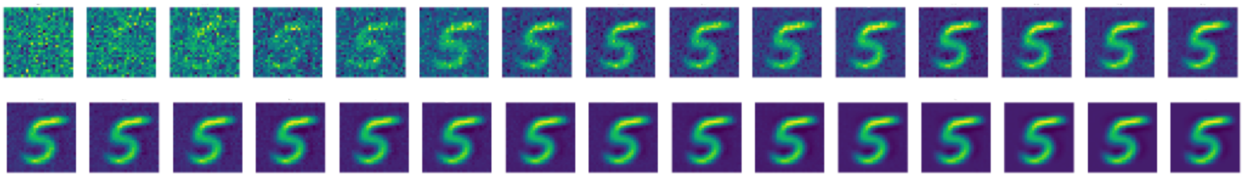

Figure 1: Digit generation from pure noise (class conditioned). The reverse process runs from left to right.

of the denoiser Equation 11 throughout the generation process. Recall that at each time step $x_t = x_{t-1} + \epsilon_{\sigma_t}$, where $\epsilon_{\sigma_t} \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I})$ (Equation 3). Since the noise is assumed to be Gaussian, we can write $x_t = x_0 + \epsilon_{\bar{\sigma}_t}$, where $\bar{\sigma}_t = \sqrt{\sum_{i=0}^{t} \sigma_i^2}$. Assume that the data distribution is such that its population covariance is given by

$$\Sigma_t = \mathbb{E} x_0 x_0^\dagger + \bar{\sigma}_t^2 \mathbf{I} = \sum_{i=0}^{r-1} \lambda_i^2 u_i u_i^\dagger + \bar{\sigma}_t^2 \mathbf{I} \triangleq \Sigma_0 + \bar{\sigma}_t^2 \mathbf{I}, \tag{14}$$

where $r - 1 < d$, i.e., the data reside in a low dimensional subspace (which is generally true for natural data). This is known as the “spiked model” (22), with a vast body of work covering the distribution and identifiability of the spikes spectrum (e.g., (34)). Throughout the paper, we use the term "index" to refer to to the index i in 14, where the eigenvalues λ_i are ordered largest to smallest.

Given n samples concatenated as columns in the matrix X_0, at each time step we learn the PCA basis associated with $X_t = X_0 + E_{\bar{\sigma}_t}$, by the diagonalization of the sample covariate matrix

$$\hat{\Sigma}_t = \frac{1}{n} X_t X_t^\dagger = \frac{1}{n} (X_0 X_0^\dagger + X_0 E_{\bar{\sigma}_t}^\dagger + E_{\bar{\sigma}_t} X_0^\dagger + E_{\bar{\sigma}_t} E_{\bar{\sigma}_t}^\dagger) \triangleq U_t S_t U_t^\dagger. \tag{15}$$

Thus, during the reverse process, at each time step we apply the projection

$$D_{PCA}^t = P_t = U_t U_t^\dagger. \tag{16}$$

In order to understand the diffusion generation process, we analyze the decay of the product $\langle u_i^t, u_i \rangle$ over time, where u_i^t is the ith column of U_t. Note, that there are two drivers of change in the perturbation of u_i to u_i^t. The first being the added noise, i.e., $\|\Sigma_t - \Sigma_0\|$. This is the key in the diffusion process and our main focus. The second, is in the finite sample approximation $\left\|\hat{\Sigma}_t - \Sigma_t\right\|$. This source of error is interesting in the context of sample complexity, as it encompasses the approximation of the denoiser learned from a finite dataset, the equivalent of the sample complexity of learning the score function in (13; 8; 7). For the rank-1 case, (34) presented a finite sample theorem which holds with high probability for the closeness between the leading eigenvalue and eigenvector of sample and population PCA under a spiked covariance model similar to Equation 14. They bound the angle between the leading empirical eigenvector and its population counterpart with approximately $\mathcal{O}(d)$ sample complexity, and a linear dependence on the noise level:

$$\mathbb{E} \sin \theta_{\text{PCA}} = \mathbb{E} \sqrt{1 - \langle u^t, u \rangle^2} \approx \frac{\bar{\sigma}_t}{\lambda} \sqrt{\frac{d}{n}}, \tag{17}$$

where $\bar{\sigma}_t$ is assumed to be small and $d \gg 1$. This result shows that the leading eigenvector rotates in a rate proportional to the noise level. Our numerical experiments on the MNIST dataset (detailed in Section 4.1) show that this is a good approximation in practice, also for the rank-r case (Fig. 2).

Notice that in Equation 17 the angle is inversely linked to the eigenvalue, inferring a slower change with higher eigenvalues. In the reverse process, we gradually move from pure noise or high noise levels to smaller noise variance. Given the lower slope of the components corresponding to larger eigenvalues, we interpret the result in Fig. 2 as the earlier emergence of low frequencies in the generation process. The first component to be visible in the generated image is the one with the largest eigenvector, as it is the first one that shows a correlation in high noise levels. Throughout the generation process, when the noise level decreases, the

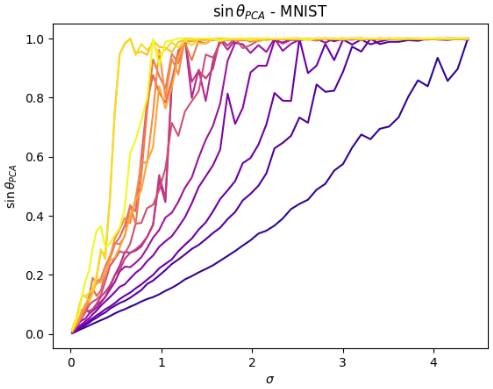

Figure 2: The sine of the angle between the clean principal components and their noisy versions, colored by the order of the eigenvalues (the darkest being largest eigenvalue). Low frequencies emerge earlier in the generation process (at higher noise levels). This motivates Assumption 4.1, that extends Equation 17 to higher ranks.

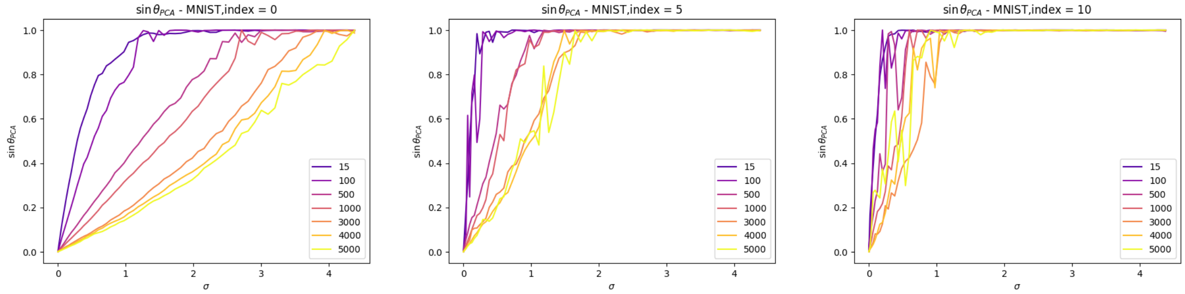

Figure 3: Effect of dataset size. The plots show $\sin\theta_{\text{PCA}}$ at different noise levels when trained on datasets with increasing size (lighter color). Each plot is of a different component index, for indices $0, 5, 10$ (left to right; index 0 corresponds to the largest eigenvalue). Increasing the amount of training data improves the robustness to noise, and allows the appearance of high frequencies at higher noise levels, hence capturing more data nuances in the generated data and better generalization.

next components take presence, by the order of their associated eigenvalue - from the larger to the smaller. Finally, the components with the smallest eigenvalues appear when the noise level is low.

In the linear case, Equation 17 shows that the diffusion model's sample complexity is determined by the sample complexity of PCA, with a linear dependence on the dimension of the data. To further enhance our understanding of the relationship between the amount of training data and the generalization of the diffusion model, we repeat the experiment with varying datasets sizes. Figure 3 shows the angle to noise profile for selected principal components, with the indices $0, 5, 10$ (left to right; index 0 corresponds to the largest in a list of ordered eigenvalues). Increasing the amount of training data improves robustness to noise and enables the emergence of higher frequency components at higher noise levels, thereby capturing more nuances in the generated data.

4.1 The Generated Distribution

We now turn to discuss the distribution of the generated output, and how it relates to the natural data distribution. We shall start from the first sampling path, considering the PCA based denoising (Equation 39), and then describe the second generation trajectory, using estimations of x_0 (Equation 13).

Given our linear model, the generated output is given by

$$\hat{x} = \Sigma_{t=0}^{T} \Pi_{\tau=0}^{t} P_\tau \xi_t = P_0 \cdots P_T \xi_T + \cdots + P_0 \xi_0 \tag{18}$$

for $\xi_t \sim \mathcal{N}(0, \sigma_t)$. For the ease of writing, define $\mathcal{P}_t = \Pi_{\tau=0}^{t} P_\tau$, and so $\hat{x} = \Sigma_{t=0}^{T} \mathcal{P}_t \xi_t$.

Other than the visual aesthetic of the generated images, we are interested in their distribution, and how well it represents the natural distribution of training images. Thus, we would like to compare the generated covariance $\mathbb{E}\hat{x}\hat{x}^\dagger$ to the natural covariance Σ_0. We start our analysis by focusing on the first summand comprising $\hat{x}$,

$$\hat{x}_T = \mathcal{P}_T \xi_T. \tag{19}$$

In this context, a natural comparison is the power iteration (PI) method, which may be used to estimate the leading eigenvector of a matrix. This can be seen as another iterative form of generating data from random vectors. Unlike our projection, in PI we "project" a random vector onto the entire matrix, i.e. including the eigenvalues. In this case the denoiser would be $D_{PI}^t = \Sigma_0 \forall t$, where we ignore the normalization and focus on the direction of the final vector, since there is no normalization constraint for generated data in diffusion models.

We now turn to show how the reverse process performed by a repeated denoising as in Equation 19 converges in mean to PI. To this end, we make the following assumptions.

Assumption 4.1. *Assume that Equation 17 holds for all eigenvectors, i.e.,*

$$\mathbb{E}\sqrt{1 - \langle u_i^t, u_i \rangle^2} \approx \frac{\bar{\sigma}_t}{\lambda_i} \sqrt{\frac{d}{n}}, \tag{20}$$

for $i = 0, \ldots, r-1$.

This assumption is the extension of Equation 17 to higher ranks, and is motivated by our simulations (Fig. 2). In addition, we make the following assumption regarding the cross products of components of different indices, at consecutive time steps.

Assumption 4.2. *For each index i there exists a time τ_i, where for $t \leq \tau_i$ and $j \leq i$,*

$$\mathbb{E}\langle u_i^t, u_j^{t+1} \rangle = 0. \tag{21}$$

In addition, $\tau_i > \tau_j$ for $i < j$.

This assumption is supported by our simulations in Fig. 5, and will be further discussed hereafter. Assumptions 4.2, 4.1 are an extension of (34) to higher ranks. We leave their explicit derivation to future work, and focus on their implications to linear diffusion.

We are now ready to state our main result.

Theorem 4.3 (Convergence to Power Iteration)**.** *Let $\sigma_t = \frac{1}{T}$, $t = 0, \ldots, T$. Assuming 4.2, 4.1, in the limit $T \to \infty$,*

$$\mathbb{E}\hat{x}_T \hat{x}_T^\dagger \propto u_0 u_0^\dagger. \tag{22}$$

Proof. Let us analyze the product in Equation 19 to show how it relates to the power method. The linear operator representing the reverse process can be written as

$$\mathcal{P}_T = U_0 \Pi_{t=0}^{T-1} (U_t^\dagger U_{t+1}) U_T^\dagger. \tag{23}$$

The matrix product $U_t^\dagger U_{t+1}$ can be analyzed using the extension of Equation 17 to higher ranks. Given 4.1, the expected inner product with the natural data component $u_i = u_i^{t=0}$ is given by

$$\mathbb{E}\langle u_i^t, u_i \rangle \approx 1 - \frac{\bar{\sigma}_t^2}{\lambda_i^2} \frac{d}{n}. \tag{24}$$

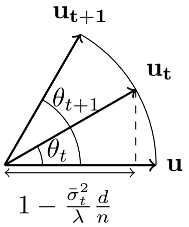

Figure 4: Schematic illustration of the basis perturbation, per index.

The evolution of this product over time is depicted in Figure 4. We are interested in the projection of u_{t+1} onto u_t, which is the cosine of the angle $\Delta\theta = \theta_{t+1} - \theta_t$. This angle is tractable for small noise levels, so we divide our analysis to two parts: $0 \leq t \leq \tau$ and $\tau \leq t \leq T$, where the choice of τ will soon be motivated.

First, we inspect the limit of $t \to 0$ ($0 \leq t \leq \tau$). For small angles, we can write

$$\Delta\theta = \arccos\Big(1 - \frac{\bar{\sigma}_{t+1}^2}{\lambda^2}\frac{d}{n}\Big) - \arccos\Big(1 - \frac{\bar{\sigma}_t^2}{\lambda^2}\frac{d}{n}\Big) \approx \frac{d}{\lambda^2 n}(\bar{\sigma}_{t+1}^2 - \bar{\sigma}_t^2) = \frac{\sigma_{t+1}^2 d}{\lambda^2 n}, \tag{25}$$

since $\arccos\theta \approx \frac{\pi}{2} - \theta$ and $\bar{\sigma}_t^2 = \sum_{\tau=0}^t \sigma_\tau^2$. The diagonal elements in $U_t^\dagger U_{t+1}$ are then given by

$$\mathbb{E}\langle u_i^t, u_i^{t+1}\rangle \approx \cos\frac{\sigma_{t+1}^2 d}{\lambda_i^2 n}, \tag{26}$$

where the off-diagonal elements are negligible, since

$$\mathbb{E}\langle u_i^t, u_j^{t+1}\rangle \approx \mathbb{E}\langle u_i^t, u_j^t\rangle = 0, \tag{27}$$

which holds for $t \leq \tau_{r-1}$ by Assumption 4.2. Notice, that in small angles, $\langle u_i^t - u_i^{t+1}, u\rangle = (\sigma_{t+1}^2 d)/(\lambda^2 n) \to 0$, so the vectors u_i^t are co planar, as depicted in Figure 4. Thus, the time point basis correlations $U_t^\dagger U_{t+1}$ form an approximately diagonal matrix with the fraction $c_i \triangleq \cos\frac{\sigma_{t+1}^2 d}{\lambda_i^2 n}$ on the diagonal, where $c_i > c_j$ for $i < j$. We eliminate the dependence of c_i on t by choosing the constant schedule $\sigma_t = 1/T\ \forall t$, to simplify the proof. However, many schedules can be used, as long as $c_{i,t} > c_{j,t}$ remains correct. Define the partial linear diffusion operator until time τ by $\mathbb{E}\mathcal{P}_\tau = \Pi_{t=0}^\tau P_t$. Then

$$\mathbb{E}\mathcal{P}_\tau = U_0 \begin{pmatrix} c_0^\tau & & \\ & \ddots & \\ & & c_{r-1}^\tau \end{pmatrix} U_\tau^\dagger = U_0 c_0^\tau \begin{pmatrix} 1 & & \\ & \left(\frac{c_1}{c_0}\right)^\tau & \\ & & \ddots \end{pmatrix} U_\tau^\dagger \underset{\tau\to T}{\to} U_0 \begin{pmatrix} c_0^\tau & & \\ & 0 & \\ & & \ddots \end{pmatrix} U_\tau^\dagger, \tag{28}$$

where the diagonal elements decay as τ grows larger, as $c_i > c_j$ for $i < j$. Similarly to power iteration, the convergence rate depends on the ratio c_1/c_0. The convergence rate might not be fast enough for the process to converge while the small angles approximation still holds. Thus, we continue with the second phase of our analysis, showing the convergence of the full reverse process.

We now turn to analyse the phase where $\tau \leq t \leq T$. In high noise levels, the correlation with the natural basis is low, and the products $U_t^\dagger U_{t+1}$ are not exactly diagonal. However, the correlation "leaks" to a close neighborhood of the original component and the temporal products are still somewhat concentrated around their diagonal. This process happens in accordance with Equation 4.1, where the spreading of the diagonal elements happens for high indices in lower values of t (less noise is needed to spread the correlation). This leads us to Assumption 4.2, claiming that for each index i there exists a time τ_i after which the small angle approximation does not hold; $\tau_i > \tau_j$ for $i < j$. This is apparent in practice, and depicted in 5 (left image per duo). However, given the decaying diagonal structure of the partial operator $\mathcal{P}_\tau$, we will now show that 4.2 is sufficient for the total operator to converge as desired.

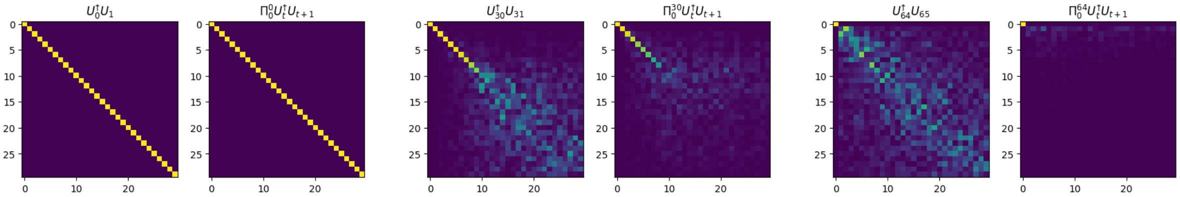

Figure 5: The time point basis correlation matrices $U_\tau^T U_{\tau+1}$ (left per pair), together with the partial product $\Pi_{t=0}^{\tau}(U_t^\dagger U_{t+1})$ (right per pair) at different time points. This justifies Assumption 4.2, and shows that the total projection (bottom right image, for $\tau = T$) converges to the first eigenvector, similarly to the power method.

Suppose we added one more matrix multiplication to our former analysis, i.e. observe

$$\mathbb{E}\mathcal{P}_\tau U_{\tau+1} = U_0 c_0^\tau \begin{pmatrix} 1 & & \\ & \left(\frac{c_1}{c_0}\right)^\tau & \\ & & \ddots \end{pmatrix} U_\tau^\dagger U_{\tau+1}. \tag{29}$$

Assumption 4.2 guarantees $U_\tau^\dagger U_{\tau+1}$ is diagonal just enough not to spoil the diagonality of the next partial operator $\mathbb{E}\mathcal{P}_{\tau+1}$. To see this, let us inspect some intermediate index i, where the entries in $j > i$ are already practically zero. Thus, we have

$$\mathbb{E}\mathcal{P}_{\tau_i} U_{\tau_i+1} = U_0 c_0^{\tau_i+1} \underbrace{\begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \left(\frac{c_i}{c_0}\right)^{\tau_i} & \\ & & & \mathbb{O} \end{pmatrix}}_{\triangleq C_{\tau_i}} \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \frac{c_i}{c_0} & \\ & & & \mathbb{A} \end{pmatrix} = U_0 c_0^{\tau_i+1} \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & \left(\frac{c_i}{c_0}\right)^{\tau_i+1} & \\ & & & \mathbb{O} \end{pmatrix}$$

where $\mathbb{O}$ is a block of zeros and $\mathbb{A}$ is a block matrix the same size as $\mathbb{O}$, that can have nonzero entries, by Assumption 4.2. Since the elements of the partial product C_{τ_i} decay faster with i than any single product $U_{\tau_i}^\dagger U_{\tau_i+1}$, C_{τ_i+1} is also diagonal. Overall, the final product is a diagonal matrix with a spectrum that converges to be concentrated around the first eigenvalue, where we can control the distribution of the generated data by the choice of the diffusion parameters. Figure 5 shows our simulation of the process, supporting both assumption 4.2 and the result stated by this theorem. □

Thus, the generated output is a combination of a (purely) noisy image that was repeatedly correlated to converge to v_0 (as shown above), with generally lower noise levels that are "lightly" correlated, although to the cleaner projection operators. The generated output can thus be seen as a combination of three conceptual parts, with a different balance of the noise level and the portrayed components.

The first eigenvector The first part of the sum in Equation 18 is $P_0 \cdots P_T \xi_T$, the estimation of the eigenvector with the largest eigenvalue, as shown above theoretically in Equation 28 and empirically in the right matrix of the bottom right duo in Fig. 5. The "strongest" noise is repeatedly correlated to be concentrated around the first eigenvector.

The entire (clean) spectrum The last part in Equation 18 is $P_0\xi_0$, a weak noise level that is spread across all components. This noise is very lightly and not repeatedly correlated, although to a clean version of the natural data basis.

In between The third part consists of all the intermediate products $\Pi_{\tau=0}^{t} P_\tau \xi_t$. The product operators $\Pi_{\tau=0}^{t} P_\tau$ preserve varying parts of the natural spectrum, according to t - as t grows, the total projection tends to retain only the components associated with larger eigenvalues. This can bee seen in Fig. 5. The right matrix in each pair shows the product $\Pi_{\tau=0}^{t} P_\tau$ for varying values of t. The total projections range from the entire spectrum (top left) to only the leading eigenvalue (bottom right). In between, the products are diagonal matrices where the entries in the indices of the smaller eigenvalues have already diminished, in a similar way to the convergence described in Equation 28.

Thus, we get a combination of a solid estimation of the leading eigenvector, together with a more uniform and week sampling of the components with low eigenvalues in the natural data basis. In between, the intermediate projections are at different levels of convergence to the leading eigenvector, hence tend to be more concentrated on components with large eigenvalues as $t \to T$. The freedom in choice of schedule $\{\xi_t\}_{t=0}^{T}$, allows control of the spread of the final distribution on the natural data components.

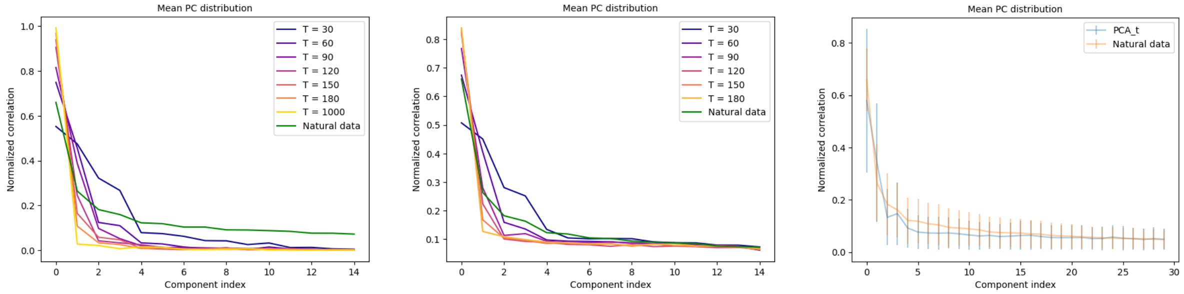

Figure 6: The empirical distribution of generated images over the natural principal components, with (middle) and without (left) injected noise. On the right - the best configuration with the generated standard deviation (see Sec. 4.1).

To inspect this, we plot the empirical distribution of generated images over the clean PCs, given by

$$p_i = \frac{1}{n} \sum_{j=1}^{n} \frac{|\langle u_i, x_j \rangle|}{\|x_j\|_2}, \tag{30}$$

where u_i is the clean principal component with index i (defined in Equation 14) and x_j is a generated sample, out of n examples. Figure 6 shows the empirical distribution of generated images over the clean principal components. On the left, we plot the distribution without injected noise (i.e., $\hat{x} = \mathcal{P}_T \xi$), for various values of T. As we show above, the distribution tends to be concentrated on the first eigenvector as T increases. The center plot shows the distribution of the process including the injected noise in the intermediate denoising steps. While in the low indices the dominant behavior is similar to the former case, the higher orders do not converge to zero and maintain their presence in the generated distribution. We note, that more sophisticated nonlinear deterministic samplers might not require the injection of noise in order to converge to the natural data distribution (e.g. (30)). However, given a linear model, it is natural to accept added stochasticity in the lack of nonlinearity (more on that in Section 5). On the right, we picked the best configuration ($T = 65$ in this case) to approximate the natural distribution. Notice, that the final generated distribution depends on the choice of parameters, where one can control the mean of the generated spectrum (this might be a feature for some applications, such as segmentation via diffusion, etc.). It might be interesting to derive the optimal parametrization for the convergence of the linear model - we leave this for future work. In addition to the convergence in mean, we included the standard deviation of the natural and generated samples, resulting in a decent fit to the target distribution.

We now turn to analyze the second sampling procedure, considering the loss defined in Equation 12 and repeated estimations of x_0. The generation starts from the denoising of ξ_T by D_T (defined in Equation 13), to obtain the first estimate of x_0, $D_T \xi_T$. The next denoiser is optimal considering the noise level $\bar{\sigma}_{T-1}$, so prior to its application, we add the next noise instance, w_{T-1}. Thus, the iteration in this denoising chain is given by

$$x_{t-1} = D_t x_t + \xi_{t-1}, \tag{31}$$

where again $\xi_t \sim \mathcal{N}(0, \bar{\sigma}_t^2 \mathbb{I})$. similarly to the former case, the final generated output $\hat{x}$ can be expressed as

$$\hat{x} = \Sigma_{t=0}^{T} \Pi_{\tau=0}^{t} D_\tau w_t = D_0 \cdots D_T \xi_T + \cdots + D_0 \xi_0. \tag{32}$$

The difference between the generation path in Equation 32 and the one described in Equation 18 is in the applied denoisers, where the former utilizes the denoiser defined in Equation 13, and the latter employs the PCA denoiser (defined in Equation 16 and described in Equation 39). In addition, the accompanying noise schedules should match the denoiser: $\{\sigma_t\}$ for the PCA denoiser and $\{\bar{\sigma}_t\}$ considering Equation 13.

Notice, that in this case as well, if we inspect the first element in Equation 32, i.e., $D_0 \cdots D_T \xi_T$, the dominant direction is concentrated in the first eigenvector of Σ_0. This can be seen by looking at the diagonalization of D_t,

$$\begin{aligned} D_t &= (\Sigma_0 + \bar{\sigma}_t^2 \mathbb{I})^{-1} \Sigma_0 \\ &= U_0 \begin{pmatrix} \frac{\lambda_0}{\lambda_0 + \bar{\sigma}_t^2} & & \\ & \ddots & \\ & & \frac{\lambda_{r-1}}{\lambda_{r-1} + \bar{\sigma}_t^2} \end{pmatrix} U_0^\dagger, \end{aligned} \tag{33}$$

since $\frac{x}{x+a}$ is monotonically increasing for $x, a \geq 0$. Thus, similarly to the case described in Equation 18, the generated output can be interpreted as a sum of high noise levels that were repeatedly correlated to estimate the leading data eigenvector, and lower noise levels that sample the entire data spectrum, in accordance with our discussion in Section 4.1.

5 Empirical Extension to Deep Denoisers

In the linear case described above, the optimal denoiser is given by the PCA projection onto the clean(er) data. These denoisers are computed with the training data, and their principal components do not depend on the input in the reverse process. When the denoiser is nonlinear, and might be implemented using a deep neural network, its input-output mapping can be locally expressed via the network Jacobian, by

$$D(x_t) = \nabla D(x_t) x_t = V_t \Lambda_t V_t^\dagger x_t, \tag{34}$$

where $V_t \Lambda_t V_t^\dagger$ denotes the eigen decomposition of the Jacobian calculated at x_t. For simplicity, we assume that the Jacobian is symmetric and non-negative (which is approximately true (33)). Note that in this case, the denoising base depends on the input image (and noise level). While the network is non linear, we can follow the generation path in the sampling process and inspect the basis of the network Jacobians calculated at the intermediate sampled points x_t. We can then trace $\sin \theta_J = \sqrt{1 - \langle v_i^t, v_i^{t=0} \rangle^2}$ where the subscript "J" stands for Jacobian, v_i^t is the i^{th} column in V_t defined in Equation 34, in a similar way to our simulations of the linear case (Figure 2). This can be calculated per generation path, where x_0 is the final generated image, and V_0 is the basis of the Jacobian calculated at this final point.

Figure 7 shows $\sin \theta_J$ calculated using the Jacobians of a UNet based diffusion model, described in (35). This model was simply chosen as the [1]state-of-the-art in the task of image generation considering the CelebA dataset at the time of writing this paper. We used the default settings and calculated the Jacobians at the final iterations. We plot the results for the leading 300 Jacobian eigenvectors, where the color is assigned by the index - darker colors for lower indices i. We repeated the experiment sampling images from the CelebA dataset (left) and CIFAR 10 (right). Even though the denoising model is far from linear, the decay of the angle between the denoising basis in high noise levels and the natural denoising basis is similar to the decay in the linear case (compare to Figure 2). In this case as well, the correlation of the low indices (and hence low frequencies) withstands higher noise levels, thus appearing first in the generation process. As this is the basis of our analysis comparing the reverse diffusion process to power iteration, this experiment shows that our analysis is relevant in a broader context and not just in the simplified linear case.

---

[1] `https://paperswithcode.com/sota/image-generation-on-celeba-64x64`

This analysis focuses on the local behavior of the nonlinear denoiser at the end of the generation process, demonstrating its similarity to a linear denoising chain. Each plot represents a single generation path, not the overall distribution of generated outputs.

While linear diffusion models are easy to analyze, they may struggle to generate complex datasets. Nonlinear models, on the other hand, can navigate a diverse set of linearized regions during the generation process (as illustrated in Figure 7). This allows them to generate diverse data even without added noise, unlike linear models which ultimately converge in mean to a single point (Theorem 4.3) and therefore require noise injection for diverse outputs. This contrasts with some deterministic nonlinear samplers (e.g., (30)) that do not rely on added noise.

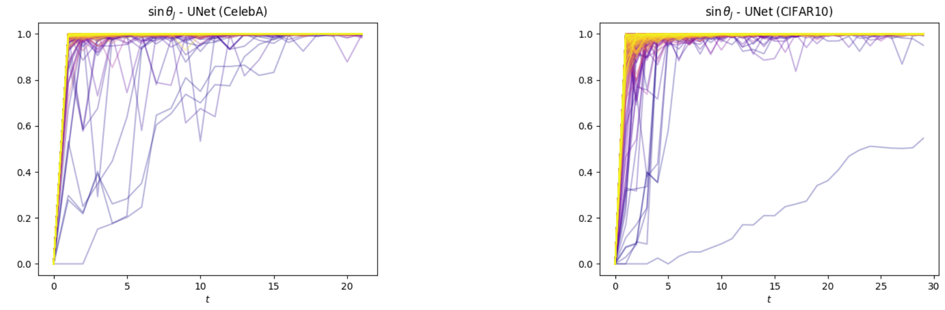

Figure 7: Image generation - the sine of the angle between Jacobian eigenvectors at the final generated image ($t = 0$) and intermediate iterations ($t > 0$). The diffusion model includes a UNet-based denoiser trained on CelebA (left) or CIFAR10 (rignt). Color by index (the darker the color the lower the index, referring to columns of the Jacobian basis V_t). The Jacobians of the nonlinear denoiser conform to the behavior of the linear model.

6 Conclusion

In this paper, we discuss a simple diffusion model with a linear denoiser and normalization free sampler, that allows us to cast the diffusion problem as noisy PCA, and make the connection to the spiked covariance model assuming that the natural data distribution reside in a low dimensional subspace. This enables us to show that in the linear case, the generation process acts as a "correlation machine", where initial random noise is repeatedly correlated to noisy estimations of the natural data basis, to finally embody the true distribution, in a manner similar to the power iteration method. We show that in this process, low frequencies emerge earlier, and more data contributes to a richer representation per the same diffusion configuration. Finally, we demonstrate the relevance of our analysis also in a deep, non-linear diffusion denoiser.

We acknowledge the limitation of admitting a linear model, with its lack of ability to represent the complex data often expected of diffusion models. While our theoretical setting is modest, we empirically demonstrate how our observations deduced from a simple linear model and classic theory (22; 34) are relevant to more general models and datasets. This enables us to shed light on the internal mechanism powering this technology, and connect it to a rich pool of theory and prevalent methods such as power iteration.

References

[1] Juan Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023.

[2] Tomer Amit, Tal Shaharbany, Eliya Nachmani, and Lior Wolf. Segdiff: Image segmentation with diffusion probabilistic models. *arXiv preprint arXiv:2112.00390*, 2021.

[3] Stephen Andrilli and David Hecker. Chapter 9 - numerical techniques. In Stephen Andrilli and David Hecker, editors, *Elementary Linear Algebra (Sixth Edition)*, pages 401–439. Academic Press, sixth edition edition, 2023.

[4] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021.

[5] Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *International Conference on Learning Representations*, 2022.

[6] Joe Benton, Valentin De Bortoli, Arnaud Doucet, and George Deligiannidis. Nearly \$d\$-linear convergence bounds for diffusion models via stochastic localization. In *The Twelfth International Conference on Learning Representations*, 2024.

[7] Giulio Biroli and Marc Mézard. Generative diffusion in very large dimensions. *Journal of Statistical Mechanics: Theory and Experiment*, 2023(9):093402, 2023.

[8] Adam Block, Youssef Mroueh, and Alexander Rakhlin. Generative modeling with denoising auto-encoders and langevin sampling. *arXiv preprint arXiv:2002.00107*, 2020.

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

[10] Emmanuel Asiedu Brempong, Simon Kornblith, Ting Chen, Niki Parmar, Matthias Minderer, and Mohammad Norouzi. Denoising pretraining for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4175–4186, 2022.

[11] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 364–381. Springer, 2020.

[12] Defang Chen, Zhenyu Zhou, Can Wang, Chunhua Shen, and Siwei Lyu. On the trajectory regularity of ODE-based diffusion sampling. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 7905–7934. PMLR, 21–27 Jul 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] Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations*, 2021.

[15] Sitan Chen, Giannis Daras, and Alex Dimakis. Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers. In *International Conference on Machine Learning*, pages 4462–4484. PMLR, 2023.

[16] Mauricio Delbracio and Peyman Milanfar. Inversion by direct iteration: An alternative to denoising diffusion for image restoration. *Transactions on Machine Learning Research*, 2023. Featured Certification.

[17] Luc Devroye, Abbas Mehrabian, and Tommy Reddad. The total variation distance between high-dimensional gaussians with the same mean. *arXiv preprint arXiv:1810.08693*, 2018.

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

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

[20] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in Neural Information Processing Systems*, 34:12454–12465, 2021.

[21] Daniel Zhengyu Huang, Jiaoyang Huang, and Zhengjiang Lin. Convergence analysis of probability flow ode for score-based generative models. *arXiv preprint arXiv:2404.09730*, 2024.

[22] Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of statistics*, 29(2):295–327, 2001.

[23] Zahra Kadkhodaie, Florentin Guth, Eero P Simoncelli, and Stéphane Mallat. Generalization in diffusion models arises from geometry-adaptive harmonic representations. In *The Twelfth International Conference on Learning Representations*, 2024.

[24] 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.

[25] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[26] Gen Li, Yuting Wei, Yuxin Chen, and Yuejie Chi. Towards non-asymptotic convergence for diffusion-based generative models. In *The Twelfth International Conference on Learning Representations*, 2024.

[27] 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.

[28] Gen Li and Yuling Yan. Adapting to unknown low-dimensional structures in score-based diffusion models. *arXiv preprint arXiv:2405.14861*, 2024.

[29] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35:4328–4343, 2022.

[30] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022.

[31] Peyman Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. *IEEE Signal Processing Magazine*, 30(1):106–128, 2013.

[32] Peyman Milanfar and Mauricio Delbracio. Denoising: A powerful building-block for imaging, inverse problems, and machine learning. *arXiv preprint arXiv:2409.06219*, 2024.

[33] 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*, 2020.

[34] Boaz Nadler. Finite sample approximation results for principal component analysis: A matrix perturbation approach. *The Annals of Statistics*, 36(6):2791 – 2817, 2008.

[35] Mang Ning, Enver Sangineto, Angelo Porrello, Simone Calderara, and Rita Cucchiara. Input perturbation reduces exposure bias in diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[36] 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.

[37] Phil Pope, Chen Zhu, Ahmed Abdelkader, Micah Goldblum, and Tom Goldstein. The intrinsic dimension of images and its impact on learning. In *International Conference on Learning Representations*, 2021.

[38] Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *International Conference on Learning Representations*, 2021.

[39] Nikolay Savinov, Junyoung Chung, Mikolaj Binkowski, Erich Elsen, and Aaron van den Oord. Step-unrolled denoising autoencoders for text generation. In *International Conference on Learning Representations*, 2022.

[40] 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.

[41] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

[42] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.

[43] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

[44] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020.

[45] 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*, 2021.

[46] Hossein Talebi and Peyman Milanfar. Nonlocal image editing. *IEEE Transactions on Image Processing*, 23(10):4460–4473, 2014.

[47] Hossein Talebi and Peyman Milanfar. Fast multilayer laplacian enhancement. *IEEE Transactions on Computational Imaging*, 2(4):496–509, 2016.

[48] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

[49] Binxu Wang and John Vastola. The unreasonable effectiveness of gaussian score approximation for diffusion models and its applications. *Transactions on Machine Learning Research*, 2024.

[50] 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.

[51] Peiyu Yu, Sirui Xie, Xiaojian Ma, Baoxiong Jia, Bo Pang, Ruiqi Gao, Yixin Zhu, Song-Chun Zhu, and Ying Nian Wu. Latent diffusion energy-based model for interpretable text modeling. In *Proceedings of International Conference on Machine Learning (ICML)*, July 2022.

A Power Iteration and its Convergence

Power Iteration is a simple algorithm used to compute the dominant eigenvalue and its corresponding eigenvector of a matrix. It iteratively refines an initial random vector by multiplying it by the matrix, which gradually aligns with the eigenvector corresponding to the largest eigenvalue. For a thorough introduction to the method, see, e.g. (3). Given a square matrix $A \in \mathbb{R}^{n \times n}$, the goal is to compute the dominant eigenvalue λ_1 and its corresponding eigenvector v_1. The Power Iteration algorithm is defined as follows:

Algorithm 1 Power Iteration Algorithm

```
**Input:** Matrix $A \in \mathbb{R}^{n \times n}$, initial vector $v_0 \in \mathbb{R}^n$, number of iterations $k$
**Output:** Approximate dominant eigenvector $v_k$
Normalize the initial vector: $v_0 \leftarrow \frac{v_0}{\|v_0\|}$
for each iteration $i = 1, 2, \ldots, k$ do
    $v_i \leftarrow Av_{i-1}$
    Normalize $v_i \leftarrow \frac{v_i}{\|v_i\|}$
end for
return $v_k$
```

The algorithm starts with an arbitrary vector v_0, which is normalized to ensure numerical stability. In each iteration, the vector v_i is updated by multiplying it by the matrix A, followed by normalization. After k iterations, the vector v_k is expected to be close to the eigenvector corresponding to the largest eigenvalue of A.

A.1 Convergence Analysis

Let A be a square matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$, where the eigenvalues are ordered such that $|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Denote the corresponding eigenvectors by $v_1, v_2, \ldots, v_n$, where v_1 is the eigenvector corresponding to the dominant eigenvalue λ_1.

The key idea behind Power Iteration is that, after sufficient iterations, the sequence of vectors v_i converges to the eigenvector associated with λ_1, under certain conditions.

Let v_0 be the initial vector, which can be expressed as a linear combination of the eigenvectors of A:

$$v_0 = \sum_{i=1}^{n} \alpha_i v_i$$

where α_i are scalar coefficients. After applying the matrix A in each iteration, we obtain the sequence of vectors:

$$v_i = Av_{i-1} = A\left(\sum_{i=1}^{n} \alpha_i v_i\right) = \sum_{i=1}^{n} \alpha_i \lambda_i^i v_i$$

Thus, the i-th iteration amplifies the component of v_0 along the direction of the eigenvector corresponding to the eigenvalue λ_1, while the other components decay at a rate proportional to the magnitude of their respective eigenvalues. As the iterations proceed, the contribution of the eigenvectors associated with smaller eigenvalues diminishes, and the vector v_i becomes increasingly aligned with v_1, the eigenvector corresponding to λ_1.

Formally, we express the evolution of v_i as:

$$v_i = \lambda_1^i \alpha_1 v_1 + \lambda_2^i \alpha_2 v_2 + \cdots + \lambda_n^i \alpha_n v_n$$

The relative influence of the eigenvectors corresponding to $\lambda_2, \lambda_3, \ldots, \lambda_n$ decays exponentially as $i \to \infty$ because $\lambda_1 > |\lambda_2| \geq \cdots \geq |\lambda_n|$. Specifically, the error in approximating v_1 decreases at a rate proportional to $\frac{|\lambda_2|}{|\lambda_1|}$, leading to the following convergence result:

$$\frac{\|v_i - \lambda_1^i v_1\|}{\|v_1\|} \leq C \left(\frac{|\lambda_2|}{|\lambda_1|}\right)^i$$

for some constant C, where $\|\cdot\|$ is the vector norm (usually the Euclidean norm).

Therefore, the Power Iteration algorithm converges to the dominant eigenvector v_1 at a rate determined by the ratio of the magnitudes of the first and second largest eigenvalues, $\rho = \frac{|\lambda_2|}{|\lambda_1|}$. If λ_2 is much smaller than λ_1, convergence is fast. However, if λ_2 is close to λ_1, convergence can be slow, requiring more iterations to achieve a satisfactory approximation. The convergence is linear, with the error decaying exponentially as the number of iterations increases. For a matrix A with a well-separated dominant eigenvalue λ_1 (i.e., $|\lambda_1| \gg |\lambda_2|$), Power Iteration converges quickly, typically in $O(\log(\epsilon)/\log(\rho))$ iterations to achieve an error of size ϵ.

B PCA Optimality And Other Linear Denoising Chains

In the main text we discuss a gradual denoising chain, where noise is iteratively projected onto cleaner PCA bases (as defined in 16). In the following, we will clarify the sense in which PCA is optimal, and present another linear denoising scheme, which will help to frame the subject of this work.

The optimal linear denoiser at time t in the ℓ_2 sense is the minimizer of the loss

$$\ell_{t+1\to t} = \mathbb{E}_{x_t,w} \left\| D_t(x_t + \sigma_t w) - x_t \right\|_2^2, \tag{35}$$

where $w \sim \mathcal{N}(0, \mathbb{I})$. This can be minimized by deriving the expected loss

$$\begin{aligned} \mathbb{E}_{x_t,w} \left\| D_t(x_t + \sigma_t w) - x_t \right\|_2^2 &= \mathbb{E}_{x_t,w}[x_t^\dagger D_t^\dagger D_t x_t - 2x_t^\dagger D_t x_t + \sigma_t^2 w_t^\dagger D_t^\dagger D_t w_t + x_t^\dagger x_t] \\ &= \mathbb{E}_{x_t,w} \operatorname{Tr}[D_t^\dagger D_t x_t x_t^\dagger - 2D_t x_t x_t^\dagger + \sigma_t^2 D_t^\dagger D_t w_t w_t^\dagger + x_t x_t^\dagger] \\ &= \operatorname{Tr}[D_t^\dagger D_t \Sigma_t - 2D_t \Sigma_t + \sigma_t^2 D_t^\dagger D_t + \Sigma_t], \end{aligned} \tag{36}$$

where we have used the fact that w_t has zero mean. To derive the optimal linear denoiser, we have

$$\frac{d\ell}{dD_t} = 2D_t\Sigma_t - 2\Sigma_t + 2\sigma_t^2 D_t = 0, \tag{37}$$

and so

$$D_t = (\Sigma_t + \sigma_t^2 \mathbb{I})^{-1}\Sigma_t. \tag{38}$$

Notice, that in the limit of diminishing σ_t,

$$D^t = U_t \begin{pmatrix} \frac{\lambda_0}{\lambda_0+\sigma_t^2} & & \\ & \ddots & \\ & & \frac{\lambda_{r-1}}{\lambda_{r-1}+\sigma_t^2} \end{pmatrix} U_t^\dagger \underset{\sigma_t \to 0}{\to} U_t U_t^\dagger = D^t_{\text{PCA}}. \tag{39}$$

Alternatively, this can be seen as the minimizer when we average also on the input noise variance. In this work, we focus on the iterative application of D_t, and use the theory regarding noisy PCA (34) to analyze the convergence properties of this chain.

Given a similar ℓ_2 loss, one might suggest an alternative denoising chain, using multiple estimation of x_0. The corresponding loss is thus

$$\ell_{t\to 0} = \mathbb{E}_{x_t,w} \left\| D_t(x_0 + \bar{\sigma}_t w) - x_0 \right\|_2^2, \tag{40}$$

where $\bar{\sigma}_t$ is the overall added noise (see Section 4). The adequate denoising chain in this case is the application of D_t to estimate x_0, followed by the addition of noise with the appropriate variance $\bar{\sigma}_{t-1}^2$, before the iterative application of D_{t-1}. In this case, the optimal denoiser is given by

$$\begin{aligned} D_t &= (\Sigma_0 + \bar{\sigma}_t^2 \mathbb{I})^{-1} \Sigma_0 && (41) \\ &= U_0 \begin{pmatrix} \frac{\lambda_0}{\lambda_0 + \bar{\sigma}_t^2} & & \\ & \ddots & \\ & & \frac{\lambda_{r-1}}{\lambda_{r-1} + \bar{\sigma}_t^2} \end{pmatrix} U_0^\dagger . \end{aligned}$$

In order to generate data, this denoiser is applied on a series of noises w_t, where $w_t \sim \mathcal{N}(0, \bar{\sigma}_t^2 \mathbb{I})$ for some schedule $\{\bar{\sigma}_t\}_{t=0}^T$. The generation starts from the denoising of w_T by D_T, to obtain the first estimate of x_0, $D_T w_T$. The next denoiser is optimal considering the noise level $\bar{\sigma}_{T-1}$, so prior to its application, we add the next noise instance, w_{T-1}. Thus, the iteration in this denoising chain is given by

$$x_{t-1} = D_t x_t + w_{t-1}, \tag{42}$$

where again $w_t \sim \mathcal{N}(0, \bar{\sigma}_t^2 \mathbb{I})$. Due to the linearity of the denoisers, the final generated output x_g can be expressed as

$$x_g = \Sigma_{t=0}^T \Pi_{\tau=0}^t D_\tau w_t = D_0 \cdots D_T w_T + \cdots + D_0 w_0. \tag{43}$$

The difference between the generation path in Equation 32 and the one described in Equation 18 is in the applied denoisers, where the former utilizes the denoiser defined in Equation 13, and the latter employs the PCA denoiser (defined in Equation 16 and described in Equation 39). In addition, the accompanying noise schedules should match the denoiser: $\{\sigma_t\}$ for the PCA denoiser and $\{\bar{\sigma}_t\}$ considering Equation 13.

Notice, that in this case as well, if we inspect the first element in Equation 32, i.e., $D_0 \cdots D_T w_T$, the dominant direction is concentrated in the first eigenvector of Σ_0 (since $\frac{x}{x+a}$ is monotonically increasing for $x, a \geq 0$). Thus, similarly to the case described in Equation 18, the generated output can be interpreted as a sum of high noise levels that were repeatedly correlated to estimate the leading data eigenvector, and lower noise level that sample the entire data spectrum, in accordance with our discussion in Section 4.1.