DIMENSIONALITY-VARYING DIFFUSION PROCESS

Anonymous authors
Paper under double-blind review

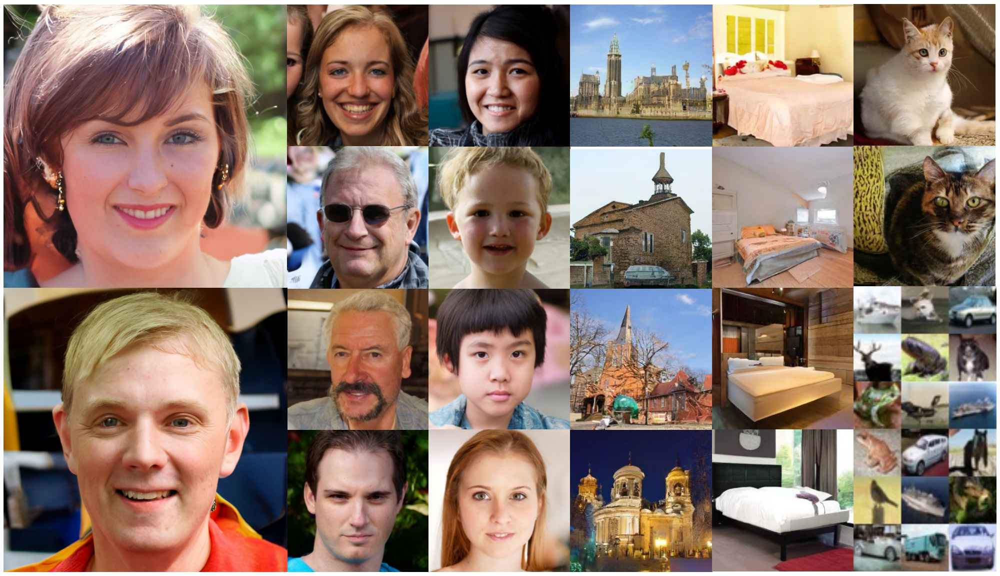

Figure 1: Synthesized samples on various datasets, including FFHQ (1024^2 and 256^2), LSUN Church (256^2), LSUN Bedroom (256^2), LSUN Cat 256 (256^2) and CIFAR10 (32^2).

ABSTRACT

Diffusion models, which learn to reverse a signal destruction process to generate new data, typically require the signal at each step to have the same dimension. We argue that, considering the spatial redundancy in image signals, there is no need to maintain a high dimensionality in the evolution process, especially in the early generation phase. To this end, we make a theoretical generalization of the forward diffusion process via signal decomposition. Concretely, we manage to decompose an image into multiple orthogonal components and control the attenuation of each component when perturbing the image. That way, along with the noise strength increasing, we are able to diminish those inconsequential components and thus use a lower-dimensional signal to represent the source, barely losing information. Such a reformulation allows to vary dimensions in both training and inference of diffusion models. Extensive experiments on a range of datasets suggest that our approach substantially reduces the computational cost and achieves on-par or even better synthesis performance compared to baseline method. We also show that our strategy facilitates high-resolution image synthesis and improves FID of diffusion model trained on FFHQ at 1024×1024 resolution from 52.40 to 15.07. Code and models will be made publicly available.

1 INTRODUCTION

Diffusion models (Ho et al., 2020; Song et al., 2021; Dhariwal & Nichol, 2021; Ramesh et al., 2022; Saharia et al., 2022) have recently shown great potential in image synthesis. Instead of directly learning the observed distribution, it constructs a multi-step forward process through gradually adding noise onto the real data (*i.e.*, diffusion). After a sufficiently large number of steps, the source signal could be considered as completely destroyed, resulting in a pure noise distribution that

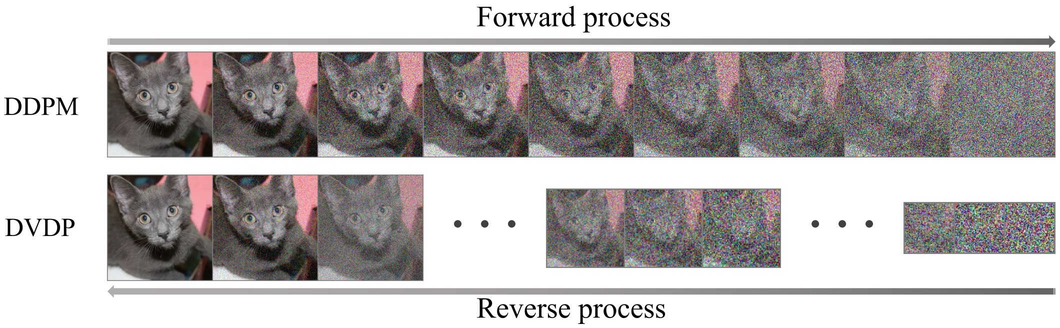

Figure 2: **Concept comparison** between DDPM (Ho et al., 2020) and our proposed DVDP, where our approach allows using a varying dimension in the diffusion process.

naturally supports sampling. In this way, starting from sampled noises, we can expect new instances after reversing the diffusion process step by step.

As it can be seen, the above pipeline does not change the dimension of the source signal throughout the entire diffusion process (Sohl-Dickstein et al., 2015; Ho et al., 2020; Song et al., 2021). It thus requires the reverse process to map a high-dimensional input to a high-dimensional output at every single step, causing heavy computation overheads (Rombach et al., 2022; Jing et al., 2022). However, images present a measure of spatial redundancy (He et al., 2022) from the semantic perspective (*e.g.*, an image pixel could usually be easily predicted according to its neighbours). Given such a fact, when the source signal is attenuated to some extent along with the noise strength growing, it should be possible to get replaced by a lower-dimensional signal. We therefore argue that there is no need to follow the source signal dimension along the entire distribution evolution process, especially at early steps (*i.e.*, steps close to the pure noise distribution) for coarse generation.

In this work, we propose dimensionality-varying diffusion process (DVDP), which allows dynamically adjusting the signal dimension when constructing the forward path. For this purpose, we first decompose an image into multiple orthogonal components, each of which owns dimension lower than the original data. Then, based on such a decomposition, we theoretically generalize the conventional diffusion process such that we can control the attenuation of each component when adding noise. Thanks to this reformulation, we manage to drop those inconsequential components after the noise strength reaches a certain level, and thus represent the source image using a lower-dimensional signal with little information lost. The remaining diffusion process could inherit this dimension and apply the same technique to further reduce the dimension.

We evaluate our approach on various datasets, including objects, human faces, animals, indoor scenes, and outdoor scenes. Experimental results suggest that DVDP achieves on-par or even better synthesis performance than baseline models on all datasets. More importantly, DVDP relies on much fewer computations, and hence speeds up both training and inference of diffusion models. We also demonstrate the effectiveness of our approach in learning from high-resolution data. For example, we are able to start from a 64×64 noise to produce an image under 1024×1024 resolution. With FID (Heusel et al., 2017) as the evaluation metric, our 1024×1024 model trained on FFHQ improves the baseline Song et al. (2021) from 52.40 to 15.07. All these advantages benefit from using a lower-dimensional signal, which reduces the computational cost and mitigates the optimization difficulty.

2 Background

We first introduce the background of Denoising Diffusion Probabilistic Models (DDPM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) and some of their extensions which are closely related to our work. DDPM constructs a forward process to perturb the distribution of data $q(\mathbf{x}_0)$ into a standard Gaussian $\mathcal{N}(\mathbf{0}; \mathbf{I})$. Considering an increasing variance schedule of noises $\beta_1, \ldots, \beta_T$, DDPMs define the forward process as

$$\mathbf{x}_t = \sqrt{\bar{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \mathbf{x}_0 \sim q(\mathbf{x}_0), \tag{1}$$

where $\alpha_t := 1 - \beta_t, t \in [1, T]$ and $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$. In order to generate high-fidelity images, DDPM (Ho et al., 2020) denoises the samples from a standard Gaussian iteratively utilizing the

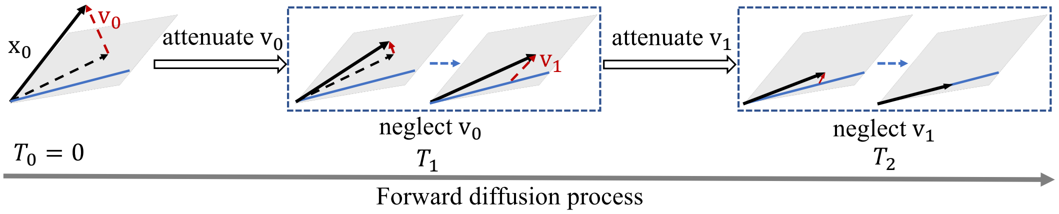

Figure 3: **Framework illustration.** We first decompose the source signal, $\mathbf{x}_0$, and then control the forward diffusion process to attenuate the inconsequential components, such that these components can be neglected at some particular steps.

reverse process parameterized as

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right), \beta_t\right), \tag{2}$$

where ϵ_θ is a neural network used to predict ϵ from $\mathbf{x}_t$. The parameters θ are learned by maximizing the following loss function

$$L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2\right]. \tag{3}$$

The standard diffusion model is implemented directly in the image space, which is probably not the optimal choice according to Lee et al. (2022), which takes the relative importance of different frequency components into consideration. Lee et al. (2022) implements the diffusion models in a designed space by generalizing diffusion process with the forward process formulated as

$$q(\boldsymbol{U}^T\mathbf{x}_t|\boldsymbol{U}^T\mathbf{x}_{t-1}) = \mathcal{N}(\boldsymbol{U}^T\mathbf{x}_t; (\mathbf{I}-\mathbf{B}_t)^{\frac{1}{2}}\boldsymbol{U}^T\mathbf{x}_{t-1}, \mathbf{B}_t\mathbf{I}), \tag{4}$$

where $\boldsymbol{U}$ is an orthogonal matrix to impose a rotation on $\mathbf{x}_t$, and the noise schedule is defined by the diagonal matrix $\mathbf{B}_t$. In this work, we extend the aforementioned generalized framework further and make it possible to vary dimensionality during the diffusion process.

3 Dimensionality-Varying Diffusion Process

This section proposes the dimensionality-varying diffusion process (DVDP), which progressively decreases the dimension of $\mathbf{x}_t$ in forward process, thus allowing us to generate high-dimensional data from a low-dimensional noise. To establish DVDP, we gradually attenuate components of $\mathbf{x}_0$ in different subspaces and effectively approximate $\mathbf{x}_t$ in a sequence of subspaces with decreasing dimensionality as timestep t progresses. To this end, we firstly decompose $\mathbf{x}_0$ into multiple orthogonal components, and progressively attenuate each component as timestep evolves, which can be approximated by DVDP (Sec. 3.1). Then, we construct a approximate reverse process with controllable small error caused by the loss of attenuated $\mathbf{x}_0$ component (Sec. 3.2). Finally, we analyze the approximation error in both forward and reverse processes (Sec. 3.3).

3.1 Dimensionality-Varying Diffusion Process

To construct a diffusion process with decreasing dimensionality, we decompose the data into multiple subspaces and attenuate these components with different rates, enabling data to be gradually approximated in decreasing dimensional subspaces.

To decompose the data point $\mathbf{x}_0$, we define a sequence of subspaces $\mathbb{S}_0 \supsetneqq \mathbb{S}_1 \supsetneqq \cdots \supsetneqq \mathbb{S}_K$ with decreasing dimensionality $d = \bar{d}_0 > \bar{d}_1 > \cdots > \bar{d}_K$, where $\mathbb{S}_0 = \mathbb{R}^d$ is the original space, $K \in \mathbb{N}_+$. Then, $\mathbf{x}_0$ can be decomposed as

$$\mathbf{x}_0 = \mathbf{v}_0 + \mathbf{v}_1 + \cdots + \mathbf{v}_K, \tag{5}$$

where $\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_K$ are $K+1$ orthogonal vectors satisfying $\mathbf{v}_i \in \mathbb{S}_i/\mathbb{S}_{i+1},\ i = 0, 1, \cdots, K-1$ and $\mathbf{v}_K \in \mathbb{S}_K$. With this definition of subspaces and decomposition of $\mathbf{x}_0$, we can attenuate $\mathbf{v}_0, \mathbf{v}_1, \cdots, \mathbf{v}_{K-1}$ progressively with time and gradually change the main component of $\mathbf{x}_t$ from $\mathbb{S}_0$ to the low-dimensional subspace $\mathbb{S}_K$, which will be shown later.

Because of the fact that adjacent pixels of an image tend to be similar in color, the main components of images are in a low-dimensional subspace, encouraging us to define $\mathbb{S}_i$ as the subspace after some 2×2 average-pooling operations on $\mathbb{S}_0$. As $\mathbb{S}_i$ is a $\bar{d}_i$-dimensional subspace, we use function $\mathcal{D}_i : \mathbb{S}_i \to \mathbb{R}^{\bar{d}_i}$ to denote the homomorphism between it and $\mathbb{R}^{\bar{d}_i}$. In practice, this function will downsample the attenuated image signal to low-resolution scale without lossing information in $\mathbb{S}_k$, thus we call it *homomorphism downsampling operator* similar with Jing et al. (2022).

With the decomposition defined in Eq. (5), it is easy to control the attenuation of each component $\mathbf{v}_i,\ i < K$ in the forward diffusion process. Such a generalized diffusion can be expressed as

$$\mathbf{x}_t = \sum_{i=0}^{K} \left(\bar{\lambda}_{i,t} \mathbf{v}_i + \bar{\sigma}_{i,t} \mathbf{z}_i \right), \tag{6}$$

where $\bar{\lambda}_{i,t}$ controls the attenuation of $\mathbf{v}_i$, $\mathbf{z}_i$ is the component of a standard Gaussian noise in subspace $\mathbb{S}_i$, and $\bar{\sigma}_{i,t}$ is the standard deviation of $\mathbf{z}_i$. Here the decomposition of noise is a prerequisite for approximating $\mathbf{x}_t$ in low-dimensional subspace, since we also need to discard noise components besides attenuating signal components.

With Eq. (6), we can progressively approximate $\mathbf{x}_t$ by a sequence of subspace components $\mathbf{x}_{1,t} \in \mathbb{S}_1 \to \mathbf{x}_{2,t} \in \mathbb{S}_2 \to \cdots \mathbf{x}_{K,t} \in \mathbb{S}_K$ in the following manner: consider a strictly increasing time sequence $T_1, T_2, \cdots, T_K$, if for each $k \leq K$, $\bar{\lambda}_{0,t}, \bar{\lambda}_{1,t}, \cdots, \bar{\lambda}_{k-1,t}$ become small enough after time T_k ($T_0 \triangleq 0$ is the start time of forward diffusion), then the diffusion process can be approximated in subspace $\mathbb{S}_k$ (dropping the small high-dimensional term $\sum_{i=0}^{k-1} \bar{\lambda}_{i,t} \mathrm{v}_i$) when time t satisfies $T_k < t \leq T_{k+1}$ ($T_{K+1} \triangleq T$) as

$$\mathbf{x}_t \approx \mathbf{x}_{k,t} + \sum_{i=0}^{k-1} \bar{\sigma}_{i,t} \mathbf{z}_i = \sum_{i=k}^{K} \left(\bar{\lambda}_{i,t} \mathbf{v}_i + \bar{\sigma}_{i,t} \mathbf{z}_i \right) + \sum_{i=0}^{k-1} \bar{\sigma}_{i,t} \mathbf{z}_i, T_k \leq t \leq T_{k+1}. \tag{7}$$

where $\mathbf{x}_{k,t} = \sum_{i=k}^{K} \left(\bar{\lambda}_{i,t} \mathbf{v}_i + \bar{\sigma}_{i,t} \mathbf{z}_i \right)$ is an approximation of $\mathbf{x}_t$ in subspace $\mathbb{S}_k$ when the time variable t is inside $[T_k, T_{k+1}]$. Note that there is a turning point of dimensionality at $\mathbf{x}_{k,T_k} \leftrightarrow \mathbf{x}_{k-1,T_k}$ in both forward and reverse process. For better understanding, the attenuation process of signal part in $\mathbf{x}_{k,t}$ (i.e., $\sum_{i=k}^{K} \bar{\lambda}_{i,t} \mathbf{v}_i$) is illustrated in Fig. 3. This approximation includes two terms: 1) low-dimensional component $\mathbf{x}_{k,t}$ that can be stored and processed subsequently in a low-resolution scale through the homomorphism downsampling operator $\mathcal{D}_k$; and 2) high-dimensional noise $\sum_{i=0}^{k-1} \bar{\sigma}_{i,t} \mathbf{z}_i$ that can be discarded but equivalently compensated at the turning point of resolution in the reverse process, i.e., $\mathbf{x}_{k,T_k} \to \mathbf{x}_{k-1,T_k}$, as we will discuss in Sec. 3.2. The error of this approximation will be analyzed in Sec. 3.3. Thus we can approximate $\mathbf{x}_t$ by $\mathbf{x}_{k,t}$ in $\mathbb{R}^{\bar{d}_k}$ during the attenuation process of $\bar{\lambda}_{k,t}$ as

$$\mathbf{y}_{k,t} = \mathcal{D}_k \mathbf{x}_{k,t} = \sum_{i=k}^{K} \left(\bar{\lambda}_{i,t} \mathcal{D}_k \mathbf{v}_i + \bar{\sigma}_{i,t} \mathcal{D}_k \mathbf{z}_i \right), T_k \leq t \leq T_{k+1}, \tag{8}$$

where $\mathbf{y}_{k,t} \in \mathbb{R}^{\bar{d}_k}$ has a decreasing dimensionality as time evolves. The noise compensation in the turning point of the reverse process, i.e., $\mathbf{y}_{k,T_k} \to \mathbf{y}_{k-1,T_k}$ will be discussed later in Eq. (14). They constitute the trajectory of our DVDP.

Finally, to complete the forward process of DVDP, we need to induce the transition kernel of our diffusion process. As a prerequisite, we write Eq. (8) in a matrix form, representing $\mathbf{y}_{k,t}$ by $\mathbf{y}_{k,0}$ as

$$\mathbf{y}_{k,t} = \boldsymbol{U}_k^T \bar{\boldsymbol{\Lambda}}_{k,t} \boldsymbol{U}_k \mathbf{y}_{k,0} + \boldsymbol{U}_k^T \bar{\boldsymbol{L}}_{k,t} \boldsymbol{U}_k \epsilon_k,\ T_k < t \leq T_{k+1}, \tag{9}$$

where $\mathbf{y}_{k,0,=} \sum_{i=k}^{K} \mathcal{D}_k \mathbf{v}_i \in \mathbb{R}^{\bar{d}_k}$, $\epsilon_k \in \mathbb{R}^{\bar{d}_k}$ is a standard Gaussian noise, $\boldsymbol{U}_k \in \mathbb{R}^{\bar{d}_k \times \bar{d}_k}$, $\bar{\boldsymbol{\Lambda}}_{k,t}$ and $\bar{\boldsymbol{L}}_{k,t}$ are given by

$$\boldsymbol{U}_k = \mathcal{D}_k \hat{\boldsymbol{U}}_k = \mathcal{D}_k [\boldsymbol{u}_{k,1}, \cdots, \boldsymbol{u}_{k,d_k}, \boldsymbol{u}_{k+1,1}, \cdots, \boldsymbol{u}_{k+1,d_{k+1}}, \cdots, \boldsymbol{u}_{K,1}, \cdots, \boldsymbol{u}_{K,d_K}] \tag{10}$$

$$\bar{\boldsymbol{\Lambda}}_{k,t} = \mathrm{diag}(\underbrace{\bar{\lambda}_{k,t}, \cdots, \bar{\lambda}_{k,t}}_{d_k}, \underbrace{\bar{\lambda}_{k+1,t}, \cdots, \bar{\lambda}_{k+1,t}}_{d_{k+1}}, \cdots, \underbrace{\bar{\lambda}_{K,t}, \cdots, \bar{\lambda}_{K,t}}_{d_K}), \tag{11}$$

$$\bar{\boldsymbol{L}}_{k,t} = \mathrm{diag}(\underbrace{\bar{\sigma}_{k,t}, \cdots, \bar{\sigma}_{k,t}}_{d_k}, \underbrace{\bar{\sigma}_{k+1,t}, \cdots, \bar{\sigma}_{k+1,t}}_{d_{k+1}}, \cdots, \underbrace{\bar{\sigma}_{K,t}, \cdots, \bar{\sigma}_{K,t}}_{d_K}), \tag{12}$$

with $d_i = \bar{d}_i - \bar{d}_{i+1} = \dim(\mathbb{S}_i/\mathbb{S}_{i+1}),\ i = 0, 1, \cdots, K-1$ and $d_K = \dim(\mathbb{S}_K)$, $\boldsymbol{u}_{i,1}, \cdots, \boldsymbol{u}_{i,d_i}$ is a set of orthonormal basis in $\mathbb{S}_i/\mathbb{S}_{i+1}$ for $i = 0, 1, \cdots, K-1$ and $\boldsymbol{u}_{K,1}, \cdots, \boldsymbol{u}_{K,d_K}$ is a set of orthonormal basis in $\mathbb{S}_K$, these orthonormal vectors are columm vectors of $\hat{\boldsymbol{U}}_k \in \mathbb{R}^{d\times\bar{d}_k}$.

Under the Markov assumption, the transition kernel can be expressed using the reparameterization trick as (see Appendix A for proof)

$$\mathbf{y}_{k,t} = \boldsymbol{U}_k^T \boldsymbol{\Lambda}_{k,t} \boldsymbol{U}_k \mathbf{y}_{k,t-1} + \boldsymbol{U}_k^T \boldsymbol{L}_{k,t} \boldsymbol{U}_k \epsilon_k,\ T_k < t \le T_{k+1}, \tag{13}$$

where $\boldsymbol{\Lambda}_{k,t} = \bar{\boldsymbol{\Lambda}}_{k,t-1}^{-1}\bar{\boldsymbol{\Lambda}}_{k,t}$, $\boldsymbol{L}_{k,t} = \bar{\boldsymbol{L}}_{k,t-1}^{-1}\bar{\boldsymbol{L}}_{k,t}$.

3.2 Reverse Process Approximating DVDP

The forward process of DVDP has already been given by Eqs. (9) and (13) in Sec. 3.1. In this section, we will derive an approximate reverse process, which induces a data generation process with progressively growing dimensionality. In Sec. 3.3, the approximation error will be discussed, and we can find that it actually converges to zero.

The reverse process is also defined as a Markov chain $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$ with learned Gaussian transitions like DDPM when $T_k < t \le T_{k+1}$ (see Appendix B). However, we cannot step over T_k (i.e., predict $\mathbf{y}_{k-1,T_k}$ from $\mathbf{y}_{k,T_k}$ at the turning point of dimensionality) using $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$. As $\mathbf{x}_{k-1,T_k} \in \mathbb{S}_{k-1}$ and $\mathbf{x}_{k,T_k} \in \mathbb{S}_k \subsetneq \mathbb{S}_{k-1}$, it means that we need to predict the component of $\mathbf{x}_{k-1,T_k}$ in subspace $\mathbb{S}_{k-1}/\mathbb{S}_k$ from another orthogonal signal, which is impossible. Thus it is also impossible to predict $\mathbf{y}_{k-1,T_k}$ from $\mathbf{y}_{k,T_k}$. Fortunately, the component of $\mathbf{x}_{k-1,0}$ in subspace $\mathbb{S}_{k-1}/\mathbb{S}_k$ has already attenuated to almost zero, thus we can approximately sample $\mathbf{x}_{k-1,T_k}$ from $\mathbf{x}_{k,T_k}$ by simply adding a Gaussian noise in $\mathbb{S}_{k-1}/\mathbb{S}_k$ to compensate the high-dimensional noise in $\mathbf{x}_{k-1,T_k}$. Formally, this can be represented by $\mathbf{y}_{k-1,T_k}$ and $\mathbf{y}_{k,T_k}$ as (see Appendix C.3):

$$\mathbf{y}_{k-1,T_k} = \hat{\boldsymbol{U}}_{k-1}^T \hat{\boldsymbol{U}}_k \mathbf{y}_{k,T_k} + \boldsymbol{U}_{k-1}^T \Delta\bar{\boldsymbol{L}}_{k-1,T_k} \boldsymbol{U}_{k-1}\epsilon_{k-1}, \tag{14}$$

where $\Delta\bar{\boldsymbol{L}}_{k-1,T_k} = \operatorname{diag}(\underbrace{\bar{\sigma}_{k-1,T_k}, \cdots, \bar{\sigma}_{k-1,T_k}}_{d_{k-1}}, \underbrace{0, \cdots, 0}_{\bar{d}_k})$, represents the standard deviation of added noise, and ϵ_{k-1} is a standard Gaussian noise in $\mathbb{R}^{\bar{d}_{k-1}}$.

As the mean of $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$ is parameterized to predict noise, the loss function to train the model $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$ has a similar form with that in DDPM:

$$L(\theta) = \mathbb{E}_{k,\mathbf{x}_0,\epsilon}\mathbb{E}_{t\sim\mathcal{U}(T_k,T_{k+1})}\left[\|\epsilon_k - \epsilon_\theta(\mathbf{y}_{k,t}(\mathbf{x}_0,\epsilon_k),t)\|_2\right], \tag{15}$$

where $\mathcal{U}(T_k, T_{k+1})$ is a uniform distribution between T_k and T_{k+1}.

3.3 Error Analysis

DVDP proposed in Sec. 3.1 is an approximation of the generalized diffusion in the whole space $\mathcal{S}$ because of discarding components of $\mathbf{x}_0$. In Sec. 3.2 we also indicate that the reverse process is just an approximation of DVDP. Both of these two approximation errors can be derived from Proposition 1 (see Appendix C.1):

Proposition 1 *Assume $p_1(\boldsymbol{x}|\boldsymbol{x}_0)$, $p_2(\boldsymbol{x}|\boldsymbol{x}_0)$ are two Gaussians that $p_1(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}_1\boldsymbol{x}_0, \boldsymbol{\Sigma})$ and $p_2(\boldsymbol{x}|\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{A}_2\boldsymbol{x}_0, \boldsymbol{\Sigma})$, where positive semi-definite matrices $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ satisfies $\boldsymbol{A}_1 \succeq \boldsymbol{A}_2 \succeq 0$, covariance matrix $\boldsymbol{\Sigma}$ is positive definite, and the support of distribution $p(\boldsymbol{x}_0)$ is bounded, then Jensen-Shannon Divergence* (JSD) *of the two marginal distributions $p_1(\boldsymbol{x})$ and $p_2(\boldsymbol{x})$ satisfies*

$$\mathrm{JSD}(p_1\|p_2) \le \frac{\sqrt{2}}{2}e^{-\frac{1}{2}}B\left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}}\right)\|\boldsymbol{\Sigma}^{-\frac{1}{2}}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\|_2 \tag{16}$$

where B is the upper bound of $\|\boldsymbol{x}_0\|_2$, $V_d(\cdot)$ is the volume of d-dimensional sphere with respect to the radius, and $r = 2B\|\boldsymbol{\Sigma}^{-\frac{1}{2}}A_1\|_2$.

In the forward process, $\mathbf{x}_t$ is approximated by $\mathbf{x}_{k,t}$ where $T_k < t \le T_{k+1}$. If we compensate $\boldsymbol{x}_{k,t}$ with a non-informative Gaussian noise and denote the result as $\tilde{\boldsymbol{x}}_{k,t}$, then the upper bound of JSD between $\tilde{\boldsymbol{x}}_{k,t}$ and $\boldsymbol{x}_t$ is claimed in Theorem 1 (see Appendix C.2 for proof):

Theorem 1 (Forward Process Error) *Assume $0 < k \leq K, t > T_k$, $\mathbf{x}_t$ and $\mathbf{x}_{k,t}$ follow distributions defined by Eqs. (6) and (7) respectively, $\tilde{\mathbf{x}}_{k,t} = \mathbf{x}_{k,t} + \sum_{i=0}^{k-1} \bar{\sigma}_{i,t}\mathbf{z}_i$ where $\mathbf{z}_i$ is component of standard Gaussian ϵ in $\mathbb{S}_i/\mathbb{S}_{i+1}$ for $i = 0, 1, \cdots, k-1$, and $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$, then*

$$\xi_1 \leq \frac{\sqrt{2}}{2} e^{-\frac{1}{2}} \sqrt{d} \left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}} \right) \max_{0 \leq i < k} \frac{\bar{\lambda}_{i,t}}{\bar{\sigma}_{i,t}} = o(\max_{0 \leq i < k} \bar{\lambda}_{i,t}) \tag{17}$$

where $r = 2\sqrt{d}\max_{0 \leq i \leq K} \frac{\bar{\lambda}_{i,t}}{\bar{\sigma}_{i,t}}$ and $\xi_1 \triangleq \mathrm{JSD}(q(\tilde{\mathbf{x}}_{k,t})||q(\mathbf{x}_{k,t}))$.

Note that the assumption $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$ is satisfied when $\mathbf{x}_0$ represents an image, since pixel values of images can be normalized in $[-1, 1]$. Theorem 1 indicates that as $\bar{\lambda}_{0,T_k}, \cdots, \bar{\lambda}_{k-1,T_k}$ all tend to zeros, the difference between the real distribution of $\mathbf{x}_t$ and our approximated distribution of $\mathbf{x}_{k,t}$ converges to zero.

In the reverse process, approximation occurs when stepping over time T_k. The upper bound of JSD between $p(\mathbf{y}_{k-1,T_k})$ obtained by the reverse process and $q(\mathbf{y}_{k-1,T_k})$ defined by the forward process is claimed in Theorem 2 (see Appendix C.3 for proof)

Theorem 2 (Reverse Process Error) *Assume $0 < k \leq K$, $q(\mathbf{y}_{k-1,T_k})$ and $q(\mathbf{y}_{k,T_k})$ are defined by Eq. (9), $p(\mathbf{y}_{k-1,T_k})$ is the marginal distribution of $q(\mathbf{y}_{k,T_k})p(\mathbf{y}_{k-1,T_k}|\mathbf{y}_{k,T_k})$ where $p(\mathbf{y}_{k-1,T_k}|\mathbf{y}_{k,T_k})$ is defined by Eq. (14), and $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$, then*

$$\xi_2 \leq \frac{\sqrt{2}}{2} e^{-\frac{1}{2}} \sqrt{d} \left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}} \right) \frac{\bar{\lambda}_{k-1,T_k}}{\bar{\sigma}_{k-1,T_k}} = o(\bar{\lambda}_{k-1,T_k}) \tag{18}$$

where $r = 2\sqrt{d}\max_{k-1 \leq i \leq K} \frac{\bar{\lambda}_{i,T_k}}{\bar{\sigma}_{i,T_k}}$ and $\xi_2 \triangleq \mathrm{JSD}(q(\mathbf{y}_{k-1,T_k})||p(\mathbf{y}_{k-1,T_k}))$.

Similar to ξ_1, ξ_2 can be arbitrarily small as $\bar{\lambda}_{k-1,T_k} \to 0$. It means that if we get an exact $q(\mathbf{y}_{k,T_k})$ by reverse process, then the approximation error caused by stepping over T_k can be small enough.

4 Experiments

In this section, we show that our DVDP can speed up both training and inference of diffusion models while achieving competitive performance. Besides, thanks to the varying dimension, DVDP is able to generate high-quality and high-resolution images from a low-dimensional subspace and exceeds all of the existing methods containing advanced Cascaded Diffusion Models (CDM) (Ho et al., 2022) on FFHQ 1024 × 1024. Specifically, we first introduce our experimental setup in Sec. 4.1. Then we compare our DVDP with existing alternatives on several widely evaluated datasets in terms of visual quality and modeling efficiency in Sec. 4.2. After that, we meticulously compare our DVDP with the recently proposed Subspace Diffusion (Jing et al., 2022) which is closely related to us in Sec. 4.3. Finally, we implement the necessary ablation studies in the last Sec. 4.4.

4.1 Experimental Setup

Implementation details. We adopt the UNet of Nichol & Dhariwal (2021); Dhariwal & Nichol (2021) which achieves better performance than the traditional version (Ho et al., 2020) in all of the experiments. Considering that the Subspace Diffusion in Sec. 4.3 is the combination of two diffusion processes of different resolutions, we use two networks for this comparison but only a single network in all of the rest experiments. In principle, we use the same network structures for DVDP and corresponding baselines. However, when the resolution comes to 1024, since DVDP has timesteps equipped with low dimensionality, the structure of score SDE is too deep to be directly applied in DVDP, thus we only maintain a similar number of parameters but use a different structure.

We set the number of timesteps to $T = 1000$ for all of our experiments. For DVDP, we reduce the dimensionality by one-fourth, namely $h \times h \to \frac{h}{2} \times \frac{h}{2}$, when the timestep t reaches one of the pre-designed values, which are denoted as a set $\hat{T}$. As for the timestep to vary dimensionality, we set $\hat{T} = \{600\}$ for DVDP of 32×32, indicating the resolution is decreased from 32×32 to 16×16 when $t = 600$. Similarly, we set $\hat{T} = \{300, 600\}$ for DVDP of 256×256 and $\hat{T} = \{200, 400, 600, 800\}$ for DVDP of 1024×1024.

Table 1: **Quantitative comparison** between DDPM (Ho et al., 2020) and our DVDP on various datasets regarding image quality and model efficiency. $*$ indicates our reproduced DDPM, using the same network architecture as ours.

			Training	Training	Sampling	Sampling
Dataset	Method	FID↓	Speed	Speed Up	Speed	Speed Up
			(sec/iter)		(sec/sample)	
CIFAR10 64×64	DDPM	3.17	–	–	–	–
	DDPM*	**3.16**	0.18	–	0.34	–
	DVDP	3.24	**0.15**	1.2×	**0.26**	1.3×
LSUN Bedroom 256×256	DDPM	6.36	–	–	–	–
	DDPM*	6.75	0.99	–	12.2	–
	DVDP	**5.34**	**0.45**	2.2×	**5.01**	2.4×
LSUN Church 256×256	DDPM	7.89	–	–		
	DDPM*	7.54	0.99	–	12.2	–
	DVDP	**7.03**	**0.45**	2.2×	**5.01**	2.4×
LSUN Cat 256×256	DDPM	19.75	–	–		
	DDPM*	18.11	0.99	–	12.2	–
	DVDP	**16.50**	**0.45**	2.2×	**5.01**	2.4×
FFHQ 256×256	DDPM*	8.33	0.99	–	12.2	–
	DVDP	**8.03**	**0.45**	2.2×	**5.01**	2.4×

The noise schedule is similar to the linear schedule (Ho et al., 2020), with adaptation to keep a comparable signal-to-noise ratio (SNR), as the ratio of lost noise is much larger than the ratio of lost image signal when downsampling, and causing an increased SNR (see Appendix D for details).

Datasets. In order to verify that DVDP is widely applicable, we use six datasets covering a wide range of resolutions from 32 to 1024 and various classes. To be specific, we implement DVDP on CIFAR10 (Krizhevsky et al., 2009), FFHQ 256×256 (Karras et al., 2019), LSUN Bedroom 256×256 (Yu et al., 2015), LSUN Church 256×256, LSUN Cat 256×256, and FFHQ 1024×1024.

Evaluation metrics. For all of our experiments, we calculate the FID score (Heusel et al., 2017) of 50k samples to evaluate the visual quality of samples, except for FFHQ 1024×1024 with 10k samples due to a much slower sampling. As for training and sampling speed, both of them are evaluated on a single NVIDIA A100 GPU. Training speed is the average number of iterations per second estimated over 4,000 iterations, and sampling speed is the average number of samples per second estimated on the generation of 100 batches. The training batch size and sampling batch size are 128, 256 respectively for CIFAR10, and 24, 64 respectively for other 256×256 datasets.

4.2 Improving Visual Quality and Modeling Efficiency.

Comparison with existing alternatives. We compare DVDP with other alternatives here to show that DVDP has the capability of acceleration while maintaining a reasonable or even better visual quality. For the sake of fairness, we reproduce DDPM using the same network structure as DVDP with the same hyperparameters, represented as DDPM*. Tab. 1 demonstrates our experimental results on CIFAR10, FFHQ 256×256, and three LSUN datasets. The results show that our proposed DVDP achieves better FID scores on all of the datasets of 256×256, illustrating an improved visual quality. Meanwhile, DVDP enjoys improved training and sampling speeds. Specifically, DDPM and DDPM* spend 2.2 times more time than DVDP when training the same epochs, while they spend 2.4 times more time generating one image. Although the superiority of DVDP is obvious on the datasets of 256×256, it becomes indistinct when it comes to CIFAR10, which is reasonable considering the negligible redundancy of images of CIFAR10 due to the low resolution.

Towards high-resolution image synthesis. Since the high computation cost, it is hard for diffusion models to generate high-resolution images. Score SDE (Song et al., 2021) tries this task by directly training a large diffusion model but the sample quality is far from reasonable. Recently, CDM aims to synthesize high-resolution images and obtain impressive results (Ramesh et al., 2022; Saharia

Table 2: **Synthesis performance** of different models trained on FFHQ 1024×1024.

Model	#Params (M)	FID
DVDP	105	**15.07**
Score-SDE	100	52.40
CDM	98 165 286	24.7 17.35 17.24

Table 3: **Ablation study** on the number of downsampling times on CelebA 128×128.

Downsampling times	0	1	2	3
FID	6.14	5.99	6.10	6.37
Training Speed Up	–	1.98×	2.24×	2.25×
Sampling Speed Up	–	2.12×	2.36×	2.43×

et al., 2022). We compare DVDP with score SDE and CDM on FFHQ 1024×1024 in Tab. 2, where CDM is implemented using three levels of scale following Ho et al. (2022). The results prove that DVDP beats the strongest CDM and achieves state-of-the-art performance in terms of high-resolution image synthesis.

4.3 Comparison with Subspace Diffusion

Subspace Diffusion (Jing et al., 2022) can also vary dimensionality during the forward process. However, as they lack a proper method to replenish the loss of high-frequency information introduced by dimensionality variation, their timestep to vary dimensionality has to be large enough to make sure the variance of the noise is large, and thus the loss mentioned above can be ignored. As a result, there is not enough space left for additional variation in dimensionality as the timestep of the first variation is too large.

Considering that the dimensionality decreases only once, we compare DVDP with Subspace Diffusion when the downsampling is carried out at different timestep t. Besides, since the Subspace Diffusion is only implemented on continuous timesteps before, we reproduce it on discrete timesteps similar as DDPM and use the reproduced version as a baseline. Fig. 4 illustrates that DVDP is consistently better with regard to sample quality on CelebA 64×64 (Liu et al., 2015) when downsampling at different timesteps, where the advantage gets larger when the downsampling timestep gets smaller. In addition, some samples of DVDP and Subspace Diffusion are shown in Fig. 5, where the sample quality of Subspace Diffusion is apparently worse than that of DVDP especially when downsampling timestep is small. In conclusion, DVDP is much more insensitive to the downsampling timestep than Subspace Diffusion, resulting in the possibility of more variation in dimensionality.

4.4 Ablation Study

We implement ablation study in this section to show that DVDP is able to keep effective when the number of downsampling growing. Specifically, we verify that on four different settings of downsamling times, which are $n = 0, 1, 2, 3$. When $n = 1$, $\hat{T}$ is set to $\{250\}$. Similarly, when $n = 2$ and $n = 3$, $\hat{T}$ is set to $\{250, 500\}$ and $\{250, 500, 750\}$, respectively. Furthermore, we use the same noise schedule for those four different settings. Tab. 3 shows that when the number of downsampling grows, the sampling quality preserves a reasonable level, indicating that DVDP can vary dimensionality for multiple times.

5 Related Work

Diffusion models. Sohl-Dickstein et al. (2015) proposes diffusion models for the first time that generate samples of target distribution by reversing a diffusion process in which target distribution is gradually disturbed to an easily sampled standard Gaussian. Ho et al. (2020) further proposes DDPM to reverse the diffusion process by learning a noise prediction network. Song et al. (2021) considers diffusion models as stochastic differential equations with continuous timesteps and proposes a unified framework.

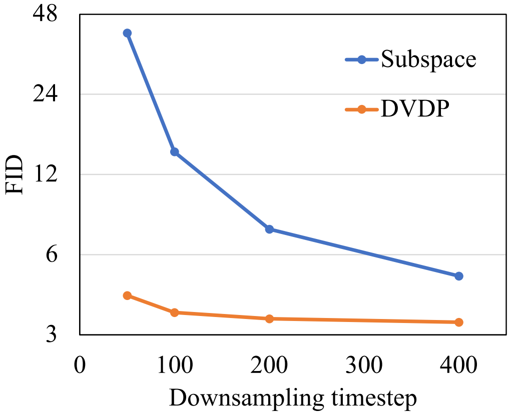

Figure 4: **Quantitative comparison** between subspace diffusion (Jing et al., 2022) and our DVDP on CelebA 64×64 regarding different timesteps to perform downsampling.

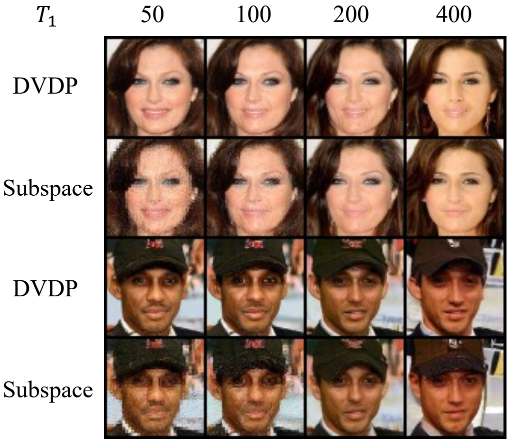

Figure 5: **Qualitative comparison** between subspace diffusion (Jing et al., 2022) and our DVDP on CelebA 64×64. T_1 denotes the timestep to perform downsampling.

Accelerating diffusion models. Diffusion models significantly suffer from the slow training and inference speed. There are many methods that speed up sampling from thousands of steps to tens of steps while keeping an acceptable sample quality (Bao et al., 2022; Lu et al., 2022; Song et al., 2020; Nichol & Dhariwal, 2021; Watson et al., 2021a;b; San-Roman et al., 2021; Liu et al., 2022). Besides improvements only on inference speed, there are other works aiming at speeding up both training and inference. Luhman & Luhman (2022) proposes a patch operation to decrease the dimensionality of each channel while accordingly increasing the number of channels, which greatly reduces the complexity of computation. Besides, a trainable forward process (Zhang & Chen, 2021) is also proven to benefit a faster training and inference speed. However, the price of their acceleration is a poor sampling quality evaluated by FID score. In this work, we accelerate DPM on both training and inference from a different perspective by heavily reducing the dimensionality of the early diffusion process and thus improving the efficiency while obtaining on-par or even better quality of generation.

Varying dimensionality generative process using diffusion models The typical requirement of diffusion models that maintain a high dimensionality may lead to poor efficiency and difficulty to generate high-resolution images. Due to the spatial redundancy in image signals, it is possible to improve the efficiency of diffusion models by varying dimensionality during the diffusion process. The most relevant work to our proposed model is Jing et al. (2022), which can also vary dimensionality in the diffusion process. However, simply varying the dimensionality as in Jing et al. (2022) may result in an inferior performance (see experimental results in Sec. 4.3), as it lacks a proper method to supplement the loss of high-frequency information at the moment of variation, which is our main contribution in this paper. Consequently, the number of dimensionality variations is limited to one time in Jing et al. (2022). Furthermore, Ryu & Ye (2022) takes advantage of position coding to train a diffusion model adaptable to multiple resolutions, and inference by repeating a series of growing dimensionality reverse processes, undergoing a lower efficiency.

6 Conclusion

This paper generalizes the traditional diffusion process to a dimensionality-varying diffusion process (DVDP). The proposed DVDP has both theoretical and experimental contributions. Theoretically, we carefully decompose the signal in the diffusion process into multiple orthogonal dynamic attenuation components. With a rigorously deduced approximation strategy, this then leads to a novel reverse process that generates images from much lower dimensional noises compared with the image resolutions. Experimentally this design allows much faster training and sampling speed of the diffusion models with on-par or even better synthesis performance, and superiority performance in synthesizing large images of 1024 × 1024 resolution compared with classic methods. The results in this work can promote the understanding and applications of diffusion models in broader scenarios.

REFERENCES

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. *Int. Conf. Learn. Represent.*, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Adv. Neural Inform. Process. Syst.*, pp. 8780–8794, 2021.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 16000–16009, 2022.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Process. Syst.*, 2017.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, pp. 6840–6851, 2020.

Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, pp. 47–1, 2022.

Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. *arXiv preprint arXiv:2205.01490*, 2022.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 4401–4410, 2019.

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

Sangyun Lee, Hyungjin Chung, Jaehyeon Kim, and Jong Chul Ye. Progressive deblurring of diffusion models for coarse-to-fine image synthesis. *arXiv preprint arXiv:2207.11192*, 2022.

Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778*, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Int. Conf. Comput. Vis.*, pp. 3730–3738, 2015.

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. *arXiv preprint arXiv:2206.00927*, 2022.

Troy Luhman and Eric Luhman. Improving diffusion model efficiency through patching. *arXiv preprint arXiv:2207.04316*, 2022.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Int. Conf. Mach. Learn.*, pp. 8162–8171, 2021.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pp. 10684–10695, 2022.

Dohoon Ryu and Jong Chul Ye. Pyramidal denoising diffusion probabilistic models. *arXiv preprint arXiv:2208.01864*, 2022.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*, 2021.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Int. Conf. Mach. Learn.*, pp. 2256–2265, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *Int. Conf. Learn. Represent.*, 2021.

Daniel Watson, William Chan, Jonathan Ho, and Mohammad Norouzi. Learning fast samplers for diffusion models by differentiating through sample quality. In *Int. Conf. Learn. Represent.*, 2021a.

Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *arXiv preprint arXiv:2106.03802*, 2021b.

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.

Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. *Adv. Neural Inform. Process. Syst.*, pp. 16280–16291, 2021.

Appendix

A Proof of the Forward Transition Kernel

The Gaussian transition kernel can be written as

$$q(\mathbf{y}_{k,t}|\mathbf{y}_{k,t-1}) = \mathcal{N}(\mathbf{y}_{k,t}; \boldsymbol{U}_k{}^T \boldsymbol{\Lambda}_{k,t} \boldsymbol{U}_k \mathbf{y}_{k,t-1}, \boldsymbol{U}_k{}^T \boldsymbol{L}^2_{k,t} \boldsymbol{U}_k), \tag{A1}$$

where $\boldsymbol{\Lambda}_{k,t}$ and $\boldsymbol{L}_{k,t}$ are to be determined.

$q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,0})$ is given by Eq. (9) as

$$q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,0}) = \mathcal{N}(\mathbf{y}_{k,t-1}; \boldsymbol{U}_k{}^T \bar{\boldsymbol{\Lambda}}_{k,t-1} \boldsymbol{U}_k \mathbf{y}_{k,0}, \boldsymbol{U}_k{}^T \bar{\boldsymbol{L}}^2_{k,t-1} \boldsymbol{U}_k) \tag{A2}$$

Thus, $q(\mathbf{y}_{k,t}|\mathbf{y}_{k,0})$ can be derived from Eqs. (A1) and (A2) and is also Gaussian with mean $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}_t$ expressed as

$$\boldsymbol{\mu}_t = \boldsymbol{U}_k{}^T \boldsymbol{\Lambda}_{k,t} \bar{\boldsymbol{\Lambda}}_{k,t-1} \boldsymbol{U}_k \mathbf{y}_{k,0} \tag{A3}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{U}_k{}^T \left(\boldsymbol{\Lambda}_{k,t} \bar{\boldsymbol{L}}^2_{k,t-1} \boldsymbol{\Lambda}^T_{k,t} + \boldsymbol{L}^2_{k,t} \right) \boldsymbol{U}_k \tag{A4}$$

According to the expression of $q(\mathbf{y}_{k,t}|\mathbf{y}_{k,0})$ in Eq. (9), $\boldsymbol{\mu}_t$ and $\boldsymbol{\Sigma}$ also satisfies

$$\boldsymbol{\mu}_t = \boldsymbol{U}_k{}^T \bar{\boldsymbol{\Lambda}}_{k,t} \boldsymbol{U}_k \mathbf{y}_{k,0} \tag{A5}$$

$$\boldsymbol{\Sigma}_t = \boldsymbol{U}_k{}^T \bar{\boldsymbol{L}}^2_{k,t} \boldsymbol{U}_k \tag{A6}$$

Combining Eqs. (A3) to (A6), we can obtain $\boldsymbol{\Lambda}_{k,t},\ \boldsymbol{L}_{k,t}$, which is just the definition in Eq. (13).

B Proofs of Reverse Transition Kernel

As in DDPM (Ho et al., 2020), the covariance matrix of $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$ can be set as the covariance of $q(\mathbf{y}_{k,t}|\mathbf{y}_{k,0})$ (i.e., $\boldsymbol{U}_k^T \bar{\boldsymbol{L}}^2_{k,t} \boldsymbol{U}_k$) or the covariance of $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t}, \mathbf{y}_{k,0})$, denoted as $\tilde{\boldsymbol{\Sigma}}_{k,t}$. The mean of $p_\theta(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t})$ is set to learn the target $\tilde{\boldsymbol{\mu}}_{k,t}(\mathbf{y}_{k,t}(\mathbf{y}_{k,0}, \epsilon_k), \mathbf{y}_{k,0})$, which is the mean of $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t}, \mathbf{y}_{k,0})$. Here we derive expressions of $\tilde{\boldsymbol{\Sigma}}_{k,t}$ and $\tilde{\boldsymbol{\mu}}_{k,t}$.

Since $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t},\mathbf{y}_{k,0}) \propto q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,0})q(\mathbf{y}_{k,t}|\mathbf{y}_{k,t-1})$ where $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,0})$, $q(\mathbf{y}_{k,t}|\mathbf{y}_{k,t-1})$ are two Gaussians given by Eqs. (9) and (13) respectively, $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t},\mathbf{y}_{k,0})$ is also a Gaussian and its exponent, denoted as $\boldsymbol{h}$, can be written as

$$\begin{aligned}\boldsymbol{h} =& -\frac{1}{2}\big[(\mathbf{y}_{k,t-1} - \boldsymbol{U}_k{}^T\bar{\boldsymbol{\Lambda}}_{k,t-1}\boldsymbol{U}_k\mathbf{y}_{k,0})^T\boldsymbol{U}_k{}^T\bar{\boldsymbol{L}}_{k,t-1}^{-2}\boldsymbol{U}_k(\mathbf{y}_{k,t-1} - \boldsymbol{U}_k{}^T\bar{\boldsymbol{\Lambda}}_{k,t-1}\boldsymbol{U}_k\mathbf{y}_{k,0}) \\ & + (\mathbf{y}_{k,t} - \boldsymbol{U}_k{}^T\boldsymbol{\Lambda}_{k,t}\boldsymbol{U}_k\mathbf{y}_{k,t-1})^T\boldsymbol{U}_k{}^T\boldsymbol{L}_{k,t}^{-2}\boldsymbol{U}_k(\mathbf{y}_{k,t} - \boldsymbol{U}_k{}^T\boldsymbol{\Lambda}_{k,t}\boldsymbol{U}_k\mathbf{y}_{k,t-1})\big] \\ =& \mathbf{y}_{k,t-1}^T\boldsymbol{U}_k{}^T(\bar{\boldsymbol{L}}_{k,t-1}^2 + \boldsymbol{\Lambda}_{k,t}^2\boldsymbol{L}_{k,t}^{-2})\boldsymbol{U}_k\mathbf{y}_{k,t-1} \\ & - 2(\bar{\boldsymbol{\Lambda}}_{k,t-1}\bar{\boldsymbol{L}}_{k,t-1}^{-2}\mathbf{y}_{k,0} + \boldsymbol{\Lambda}_{k,t}\boldsymbol{L}_{k,t}^{-2}\mathbf{y}_{k,t})^T\mathbf{y}_{k,t-1} + C,\end{aligned} \tag{A7}$$

where C is a constant that irrelevant with $\mathbf{y}_{k,t-1}$.

Compare Eq. (A7) with the formulation of Gaussian distribution, we obtain the mean $\tilde{\boldsymbol{\mu}}_{k,t}$ and the covariance $\tilde{\boldsymbol{\Sigma}}_{k,t}$ of $q(\mathbf{y}_{k,t-1}|\mathbf{y}_{k,t},\mathbf{y}_{k,0})$

$$\tilde{\boldsymbol{\mu}}_{k,t} = \boldsymbol{U}_k{}^T\bar{\boldsymbol{L}}_{k,t}^{-2}(\boldsymbol{L}_{k,t}^2\bar{\boldsymbol{\Lambda}}_{k,t-1}\boldsymbol{U}_k\mathbf{y}_{k,0} + \bar{\boldsymbol{L}}_{k,t-1}^2\boldsymbol{\Lambda}_{k,t}\boldsymbol{U}_k\mathbf{y}_{k,t}) \tag{A8}$$

$$\tilde{\boldsymbol{\Sigma}}_{k,t} = \boldsymbol{U}_k{}^T(\bar{\boldsymbol{L}}_{k,t}^{-1}\bar{\boldsymbol{L}}_{k,t-1}\boldsymbol{L}_{k,t})^2\boldsymbol{U}_k \tag{A9}$$

C Proofs of Approximation Error

C.1 Proof of Proposition 1

According to the inequality between JSD and *total variation*, we have

$$\text{JSD}(p_1||p_2) \le \frac{1}{2}\int |p_1(x) - p_2(x)|d\boldsymbol{x} \tag{A10}$$

The right-hand side of Eq. (A10) satisfies

$$\begin{aligned}\frac{1}{2}\int |p_1(\boldsymbol{x}) - p_2(\boldsymbol{x})|d\boldsymbol{x} &= \frac{1}{2}\int \left|\mathbb{E}_{\mathbf{x}_0\sim p}[p_1(\boldsymbol{x}|\mathbf{x}_0) - p_2(\boldsymbol{x}|\mathbf{x}_0)]\right| d\boldsymbol{x} \\ &\le \frac{1}{2}\int \mathbb{E}_{\mathbf{x}_0\sim p}\left[|p_1(\boldsymbol{x}|\mathbf{x}_0) - p_2(\boldsymbol{x}|\mathbf{x}_0)|\right] d\boldsymbol{x}\end{aligned} \tag{A11}$$

By the assumption, $p_1(\mathbf{x}|\mathbf{x}_0)$, $p_2(\mathbf{x}|\mathbf{x}_0)$ are two Gaussians. Using the mean value theorem, there exists $\theta = \theta(\mathbf{x}_0, \boldsymbol{x}) \in [0,1]$ such that $\boldsymbol{\xi} = \theta(\boldsymbol{x} - \boldsymbol{A}_1\mathbf{x}_0) + (1-\theta)(\boldsymbol{x} - \boldsymbol{A}_2\mathbf{x}_0) = \boldsymbol{x} - [\theta\boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2]\mathbf{x}_0$ satisfies

$$\begin{aligned}p_1(\boldsymbol{x}|\mathbf{x}_0) - p_2(\boldsymbol{x}|\mathbf{x}_0) =& C_1\left[e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{A}_1\mathbf{x}_0)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{A}_1\mathbf{x}_0)} - e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{A}_2\mathbf{x}_0)^T\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{A}_2\mathbf{x}_0)}\right] \\ =& C_1e^{-\frac{1}{2}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\mathbf{x}_0 \\ =& C_1Fe^{-\frac{1}{4}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}},\end{aligned} \tag{A12}$$

where $C_1 = \frac{1}{(2\pi)^{1/2}\det(\boldsymbol{\Sigma})^{1/2}}$, $F = e^{-\frac{1}{4}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\mathbf{x}_0$.

Now, we consider F first.

$$\begin{aligned}|F| =& \|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\xi}\|_2 e^{-\frac{1}{4}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\xi}\|_2^2}\left|\frac{(\boldsymbol{\Sigma}^{1/2}\boldsymbol{\xi})^T}{\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{\xi}\|_2}\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\mathbf{x}_0\right| \\ \le& C_2\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\mathbf{x}_0\|_2 \\ \le& C_2B\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\|_2,\end{aligned} \tag{A13}$$

where $C_2 = \max_{a\ge 0} ae^{-\frac{1}{4}a^2} = \sqrt{2}e^{-\frac{1}{2}}$, and B is the upper bound of $\|\mathbf{x}_0\|_2$ as assumption.

Combining Eqs. (A11) to (A13), we have

$$\frac{1}{2}\int |p_1(\boldsymbol{x}) - p_2(\boldsymbol{x})|d\boldsymbol{x} \le \frac{1}{2}C_1C_2B\|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{A}_1 - \boldsymbol{A}_2)\|_2\int \mathbb{E}_{\mathbf{x}_0\sim p}[e^{-\frac{1}{4}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}}]d\boldsymbol{x}, \tag{A14}$$

where $\boldsymbol{\xi} = \boldsymbol{x} - [\theta \boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2]\mathbf{x}_0$.

Consider $\boldsymbol{\xi}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\xi} = \|\boldsymbol{\Sigma}^{-1/2}(\boldsymbol{x} - (\theta \boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2)\mathbf{x}_0)\|_2^2$, let $\mathbf{z} = \boldsymbol{\Sigma}^{-1/2}[\theta \boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2]\mathbf{x}_0$, then

$$\begin{aligned} \|\mathbf{z}\|_2 =& \|\boldsymbol{\Sigma}^{-1/2}[\theta \boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2]\mathbf{x}_0\|_2 \\ \leq& B\|\boldsymbol{\Sigma}^{-1/2}[\theta \boldsymbol{A}_1 + (1-\theta)\boldsymbol{A}_2]\|_2 \\ \leq& B\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{A}_1\|_2, \end{aligned} \tag{A15}$$

where the last inequality is derived from the assumption that $\boldsymbol{A}_1 \succeq \boldsymbol{A}_2 \succeq \mathbf{0}$.

Let $\mathbb{D} = \{\boldsymbol{x} : \|\boldsymbol{\Sigma}^{1/2}\boldsymbol{x}\|_2 \leq r\}$, where $r = 2B\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{A}_1\|_2$, then the integration in Eq. (A14) can be split into two regions as

$$\begin{aligned} \int \mathbb{E}_{\mathbf{x}_0\sim p}[e^{-\frac{1}{4}\boldsymbol{\xi}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\xi}}]d\boldsymbol{x} =& \int_{\mathbb{D}} \mathbb{E}_{\mathbf{x}_0\sim p}[e^{-\frac{1}{4}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}-\mathbf{z}\|_2^2}]d\boldsymbol{x} + \int_{\mathbb{D}^{\mathrm{C}}} \mathbb{E}_{\mathbf{x}_0\sim p}[e^{-\frac{1}{4}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}-\mathbf{z}\|_2^2}]d\boldsymbol{x} \\ \leq& \int_{\mathbb{D}} 1 d\boldsymbol{x} + \int e^{-\frac{1}{16}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{x}\|_2^2} d\boldsymbol{x} \\ \leq& V_d(r)\det(\Sigma)^{1/2} + 2\sqrt{2}(2\pi)^{d/2}\det(\Sigma)^{1/2}, \end{aligned} \tag{A16}$$

where $V_d(\cdot)$ is the volume of d-dimensional sphere with respect to the radius.

By Eqs. (A14) and (A16), we can get Proposition 1.

C.2 Proof of Theorem 1

Let $\boldsymbol{u}_{k,1}, \boldsymbol{u}_{k,2}, \boldsymbol{u}_{k,d_k}$ be a set of orthonormal basis of $\mathcal{S}_k$ for $k = 0, 1, \cdots, K$, and define $\boldsymbol{U} = [\boldsymbol{u}_{0,1}, \cdots, \boldsymbol{u}_{0,d_0}, \boldsymbol{u}_{1,1}, \cdots, \boldsymbol{u}_{1,d_1}, \cdots, \boldsymbol{u}_{K,1}, \cdots, \boldsymbol{u}_{K,d_K}]^T$, then Eq. (6) can be written as

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{U}^T\bar{\boldsymbol{\Lambda}}_t\boldsymbol{U}\mathbf{x}_0, \boldsymbol{U}^T\bar{\boldsymbol{L}}_t^2\boldsymbol{U}), \tag{A17}$$

where $\bar{\boldsymbol{\Lambda}}_t$ and $\bar{\boldsymbol{L}}_t$ are defined as

$$\bar{\boldsymbol{\Lambda}}_t = \mathrm{diag}(\underbrace{\bar{\lambda}_{0,t}, \cdots, \bar{\lambda}_{0,t}}_{d_0}, \underbrace{\bar{\lambda}_{1,t}, \cdots, \bar{\lambda}_{1,t}}_{d_1}, \cdots, \underbrace{\bar{\lambda}_{K,t}, \cdots, \bar{\lambda}_{K,t}}_{d_K}) \tag{A18}$$

$$\bar{\boldsymbol{L}}_t = \mathrm{diag}(\underbrace{\bar{\sigma}_{0,t}, \cdots, \bar{\sigma}_{0,t}}_{d_0}, \underbrace{\bar{\sigma}_{1,t}, \cdots, \bar{\sigma}_{1,t}}_{d_1}, \cdots, \underbrace{\bar{\sigma}_{K,t}, \cdots, \bar{\sigma}_{K,t}}_{d_K}) \tag{A19}$$

$q(\tilde{\mathbf{x}}_{k,t}|\mathbf{x}_0)$ can also be written as

$$q(\tilde{\mathbf{x}}_{k,t}|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \boldsymbol{U}^T\tilde{\boldsymbol{\Lambda}}_{k,t}\boldsymbol{U}\mathbf{x}_0, \boldsymbol{U}^T\bar{\boldsymbol{L}}_t^2\boldsymbol{U}), \tag{A20}$$

where

$$\tilde{\boldsymbol{\Lambda}}_{k,t} = \mathrm{diag}(\underbrace{0, \cdots, 0}_{d-\bar{d}_k}, \underbrace{\bar{\lambda}_{k,t}, \cdots, \bar{\lambda}_{k,t}}_{d_k}, \cdots, \underbrace{\bar{\lambda}_{K,t}, \cdots, \bar{\lambda}_{K,t}}_{d_K}) \tag{A21}$$

Thus, $\boldsymbol{U}^T\bar{\boldsymbol{\Lambda}}_t\boldsymbol{U} \succeq \boldsymbol{U}^T\tilde{\boldsymbol{\Lambda}}_{k,t}\boldsymbol{U} \succeq \mathbf{0}$, $\boldsymbol{U}^T\bar{\boldsymbol{L}}_t^2\boldsymbol{U} \succ \mathbf{0}$. According to the assumption, $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$. Then Theorem 1 can be easily derived from Proposition 1 as

$$\begin{aligned} \xi_1 \leq& \frac{\sqrt{2}}{2}e^{-\frac{1}{2}}\sqrt{d}\left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}}\right)\|\bar{\boldsymbol{L}}^{-1}(\bar{\boldsymbol{\Lambda}}_t - \tilde{\boldsymbol{\Lambda}}_{k,t})\|_2 \\ =& \frac{\sqrt{2}}{2}e^{-\frac{1}{2}}\sqrt{d}\left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}}\right)\max_{0\leq i<k}\frac{\bar{\lambda}_{i,t}}{\bar{\sigma}_{i,t}} \\ =& o(\max_{0\leq i<k}\bar{\lambda}_{i,t}), \end{aligned} \tag{A22}$$

where

$$r = 2\sqrt{d}\|\bar{\boldsymbol{L}}^{-1}\bar{\boldsymbol{\Lambda}}_t\|_2 = 2\sqrt{d}\max_{0\leq i\leq K}\frac{\bar{\lambda}_{i,t}}{\bar{\sigma}_{i,t}} \tag{A23}$$

C.3 Proof of Theorem 2

By Eq. (9), we have

$$q(\mathbf{y}_{k-1,T_k}|\mathbf{y}_{k-1,0}) = \mathcal{N}(\mathbf{y}_{k-1,T_k}; \boldsymbol{U}_{k-1}{}^T \bar{\boldsymbol{\Lambda}}_{k-1,T_k} \boldsymbol{U}_{k-1} \mathbf{y}_{k-1,0}, \boldsymbol{U}_{k-1}{}^T \bar{\boldsymbol{L}}^2_{k-1,T_k} \boldsymbol{U}_{k-1}). \quad \text{(A24)}$$

By the assumption and Eq. (9), we have

$$p(\mathbf{y}_{k-1,T_k}|\mathbf{y}_{k-1,0}) = \mathcal{N}(\mathbf{y}_{k-1,T_k}; \boldsymbol{U}_{k-1}{}^T \tilde{\boldsymbol{\Lambda}}_{k-1,T_k} \boldsymbol{U}_{k-1}, \boldsymbol{U}_{k-1}{}^T \bar{\boldsymbol{L}}^2_{k-1,T_k} \boldsymbol{U}_{k-1}), \quad \text{(A25)}$$

where $\tilde{\boldsymbol{\Lambda}}_{k-1,T_k} = \operatorname{diag}(\underbrace{0, \cdots, 0}_{d_{k-1}}, \underbrace{\bar{\lambda}_{k,t}, \cdots, \bar{\lambda}_{k,t}}_{d_k}, \cdots, \underbrace{\bar{\lambda}_{K,t}, \cdots, \bar{\lambda}_{K,t}}_{d_K})$.

Then by Proposition 1, we can get Theorem 2.

D Adaptation on Noise Schedule

Since we choose subspaces in which the main components of images stays, the image signal will not lose much components when getting close to the subspace and downsampled to a smaller size. However, Gaussian noise does not have this property and can lose large parts of components in the downsampling operation. Thus, the signal-to-noise (SNR) ratio at the last timestep T will be smaller than that in DDPM (Ho et al., 2020) if we just use the same noise schedule. Suppose at $T_k,\ k = 1, 2, \cdots, K$, $\mathbf{y}_{k-1,T_k} \in \mathbb{R}^{\bar{d}_{k-1}}$ is downsampled to $\mathbf{y}_{k,T_k} \in \mathbb{R}^{\bar{d}_k}$ with downsamling factor $f_k = \bar{d}_{k-1}/\bar{d}_k$, then the noise shedule is adapted as Algorithm 1, which can approximately keep the SNR at the last timestep meanwhile maintaining the continuity of $\bar{\sigma}$.

Algorithm 1 Adaptation on Noise Schedule

```
Initialize $\bar{\alpha}[0:T]$ as in DDPM
$\bar{\sigma} \leftarrow \sqrt{\frac{1}{\bar{\alpha}} - 1}$
for $k = 1, \cdots, K$ do
    $\bar{\sigma}[T_k:] \leftarrow \bar{\sigma}[T_k - 1] + f_k \cdot (\bar{\sigma}[T_k:] - \bar{\sigma}[T_k - 1])$
end for
```