Dimensionality-Varying Diffusion Process

Han Zhang[1,4†] Ruili Feng[2,4] Zhantao Yang[1,4†] Lianghua Huang[4] Yu Liu[4]
Yifei Zhang[1,4†] Yujun Shen[3] Deli Zhao[4] Jingren Zhou[4] Fan Cheng[1*]
[1]Shanghai Jiao Tong University [2]University of Science and Technology of China
[3]Ant Group [4]Alibaba Group
{hzhang9617, ruilifengustc, ztyang196, shenyujun0302, zhaodeli}@gmail.com
{xuangen.hlh, ly103369, jingren.zhou}@alibaba-inc.com qidouxiong619@sjtu.edu.cn chengfan@sjtu.edu.cn

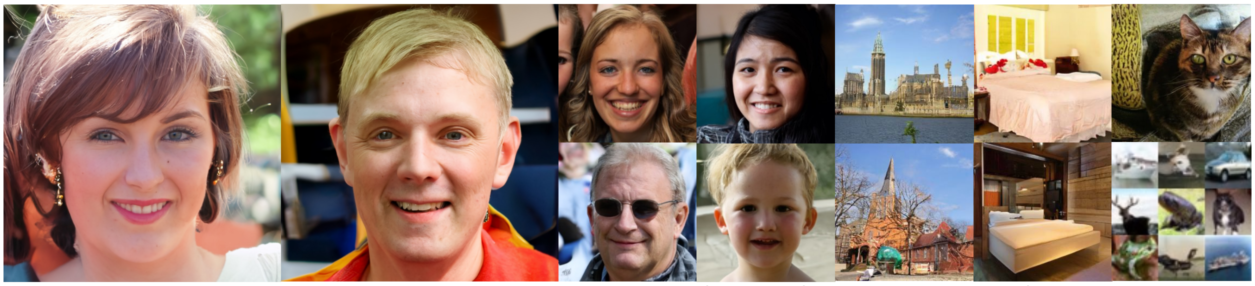

Figure 1. Synthesized samples on various datasets, including FFHQ (1024^2 and 256^2), LSUN Church (256^2), LSUN Bedroom (256^2), LSUN Cat (256^2) and CIFAR10 (32^2). All these samples are generated from a 64^2 noise except CIFAR10 from 16^2, while conventional diffusion models can only start from a noise with the same dimension as the final sample.

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 methods. 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 10.46. Code is available at https://github.com/damo-vilab/dvdp.

1. Introduction

Diffusion models [2, 6, 9, 15, 21, 24, 28] 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 completely destroyed, resulting in a pure noise distribution that 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 [6,26,28]. 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 [10, 22]. However, images present a measure of spatial redundancy [4] 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.

[*] corresponding author.
[†] Work performed at Alibaba DAMO Academy.

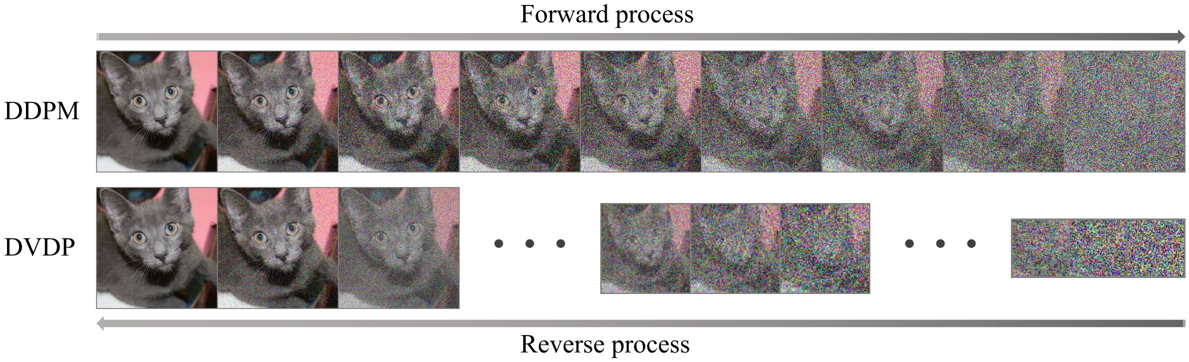

Figure 2. **Conceptual comparison** between DDPM [6] and our proposed DVDP, where our approach allows using a varying dimension in the diffusion process.

In this work, we propose dimensionality-varying diffusion process (DVDP), which allows dynamically adjusting the signal dimension when constructing the forward path. The varying dimensionality concept is shown in Fig. 2. 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 [5] as the evaluation metric, our 1024×1024 model trained on FFHQ improves the baseline [28] from 52.40 to 10.46. All these advantages benefit from using a lower-dimensional signal, which reduces the computational cost and mitigates the optimization difficulty.

2. Related Work

Diffusion models. [26] proposes diffusion models for the first time that generate samples from a target distribution by reversing a diffusion process in which target distribution is gradually disturbed to an easily sampled standard Gaussian. [6] further proposes DDPM to reverse the diffusion process by learning a noise prediction network. [28] considers diffusion models as stochastic differential equations with continuous timesteps and proposes a unified framework.

Accelerating diffusion models. Diffusion models significantly suffer from the low 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 [1, 14, 17, 19, 25, 27, 30, 31]. Besides improvements only on inference speed, there are other works aiming at speeding up both training and inference. [18] 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 [33] 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 DDPM on both training and inference from a different perspective, *i.e.*, reducing the dimensionality of the early diffusion process, and thus improving the efficiency while obtaining on-par or even better synthesis performance.

Varying dimensionality of diffusion models. Due to the redundancy in image signals, it is possible to improve the efficiency of diffusion models by varying dimensionality during the generation process. The most relevant work to our proposed model is subspace diffusion [10], which can also vary dimensionality in the diffusion process. However, subspace diffusion suffers from a trade-off between sampling acceleration and sample quality, while our DVDP can relieve this dilemma (see theoretical analysis in Sec. 4.4 and experimental results in Sec. 5.3). Instead of varying dimensionality in one diffusion process, there are works cascading several diffusion processes with growing dimensionality [7, 21, 23, 24], where the subsequent process is conditioned on the previous samples.

Discussion with latent diffusion. Besides varying dimensionality in image space, there are other methods, which we generally call latent diffusion, that directly apply diffusion models in a low dimensional latent space, obtained by an autoencoder [3, 8, 20, 22, 29]. Although latent diffusion can also speed up the training and sampling of diffusion models, it decreases dimensionality by an additional model and keeps the diffusion process unchanged. In this paper, however, we focus on the improvement on the diffusion process itself to accelerate training and sampling, which is totally a different route. Besides training and sampling efficiency, another important contribution of this work is to prove that it it unnecessary for diffusion process to keep a fixed dimension along time. By controlling the attenuation of each data component, it is possible to change dimensionality while keeping the process reversible. Thus, we will not further compare our DVDP with latent diffusion.

3. Background

We first introduce the background of Denoising Diffusion Probabilistic Model (DDPM) [6, 26] and some of its 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 $\mathcal{N}(\mathbf{0}; \boldsymbol{I})$. Considering an increasing variance schedule of noises $\beta_1, \ldots, \beta_T$, DDPM defines the forward process as a Markov chain

$$\mathbf{x}_t = \sqrt{1-\beta_t}\mathbf{x}_{t-1} + \sqrt{\beta_t}\epsilon,\ t = 1, 2, \cdots, T, \tag{1}$$

where ϵ is a standard Gaussian noise. In order to generate high-fidelity images, DDPM [6] denoises samples from a standard Gaussian iteratively utilizing the reverse process parameterized as

$$\mathbf{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{x}_t, t)\right) + \sqrt{\beta_t}\epsilon, \tag{2}$$

where $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, and ϵ_θ is a neural network used to predict ϵ from $\mathbf{x}_t$. The parameters θ are learned by minimizing the following loss function

$$L(\theta) = \mathbb{E}_{t,\mathbf{x}_0,\epsilon}\left[\|\epsilon - \epsilon_\theta(\mathbf{x}_t(\mathbf{x}_0, \epsilon), t)\|^2\right]. \tag{3}$$

While the standard diffusion model is implemented directly in the image space, [13] takes the relative importance of different frequency components into consideration, and generalizes the forward diffusion process as

$$\mathbf{x}_t = \boldsymbol{U}(\mathbf{I} - \mathbf{B}_t)^{\frac{1}{2}}\boldsymbol{U}^T\mathbf{x}_{t-1} + \boldsymbol{U}\mathbf{B}_t^{\frac{1}{2}}\boldsymbol{U}^T\epsilon, \tag{4}$$

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 $\mathbf{B}_t$. As Eq. (4) is too general to be directly instantiated as an implementation, [13] only discusses one special case called *blur diffusion*, which owns a clearly different focus from our work. In this work, we extend the generalized framework further and make it possible to vary dimensionality during the diffusion process.

4. Dimensionality-Varying Diffusion Process

We formulate the dimensionality-varying diffusion process in this section, which progressively decreases the dimension of $\mathbf{x}_t$ in the forward process, and can be effectively reversed 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 decrease the dimensionality of $\mathbf{x}_t$ at dimensionality turning points by downsampling operator (Sec. 4.1), which is approximately reversible (Sec. 4.2) with controllable small error caused by the loss of attenuated $\mathbf{x}_0$ component (Sec. 4.3).

4.1. Forward Process of DVDP

In this section, we will construct the forward process of our DVDP, which decreases the dimensionality as time evolves and can be effectively reversed. To this end, we concatenate multiple diffusion processes with different dimensions into an entire Markov chain by downsampling operations, while we elaborately design each process so that the information loss induced by downsampling is negligible. Fig. 3 illustrates this overall framework. The concatenation of different processes enables us to decrease the dimensionality, and the control on information loss ensures that downsampling operations are approximately reversible, such that the entire process can be reversed (discussed later in Sec. 4.2). To limit the information loss, we decompose the data into orthogonal components and control the attenuation of each component in the forward process of each concatenated diffusion, which we call *Attenuated Diffusion Process* (ADP). Once the lost data component induced by downsampling is attenuated to be small enough, the information loss will be negligible.

Notation list. As a prerequisite for constructing each ADP, we first define a sequence of subspaces and other necessary notations as follows:

- $\mathbb{S}_0 \supsetneqq \mathbb{S}_1 \supsetneqq \cdots \supsetneqq \mathbb{S}_K$ is a sequence of subspaces 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, $K \in \mathbb{N}_+$. For simplicity, $\mathbb{S}_{K+1} \triangleq \{\mathbf{0}\}$ and $\bar{d}_{K+1} \triangleq 0$.
- $d_i = \dim(\mathbb{S}_i/\mathbb{S}_{i+1})$, $i = 0, 1, \cdots, K$. Note that $d_K = \dim(\mathbb{S}_K/\mathbb{S}_{K+1}) = \dim(\mathbb{S}_K)$.
- $\boldsymbol{U}_0 = [\hat{\boldsymbol{U}}_0, \cdots, \hat{\boldsymbol{U}}_K] \in \mathbb{R}^{d\times d}$ is an orthogonal matrix, and column vectors of its sub-matrix $\hat{\boldsymbol{U}}_i \in \mathbb{R}^{d\times d_i}$ span subspace $\mathbb{S}_i/\mathbb{S}_{i+1}$ for $i = 0, 1, \cdots, K$.
- $\boldsymbol{U}_k \in \mathbb{R}^{\bar{d}_k\times\bar{d}_k}$ is an orthogonal matrix for $k = 0, 1, \cdots, K$, and can be split as $\boldsymbol{U}_k = [\boldsymbol{N}_k, \boldsymbol{P}_k]$ when $k < K$ where $\boldsymbol{N}_k \in \mathbb{R}^{\bar{d}_k\times d_k}$, $\boldsymbol{P}_k \in \mathbb{R}^{\bar{d}_k\times\bar{d}_{k+1}}$.
- $\boldsymbol{I}_n \in \mathbb{R}^{n\times n}$ is an identity matrix.
- $\boldsymbol{O}_n \in \mathbb{R}^{n\times n}$ is a zero matrix.

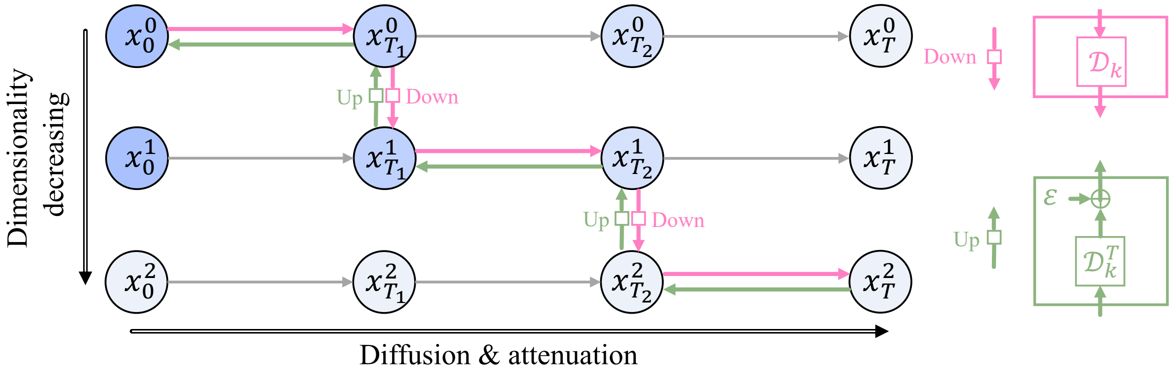

Figure 3. **Framework illustration** of DVDP. Each row represents an Attenuated Diffusion Process (ADP), which controls the attenuation of each data component while adding noise. All $K+1$ ADPs ($K=2$ here) have different dimensions decreasing from top to bottom, and are concatenated by some simple opeartions to obtain our DVDP. In the forward process, the concatenation is achieved by *downsampling operation*, and in reverse process, it is the *upsampling operation* followed by adding a Gaussian noise.

With the above definitions, we can first construct an ADP $\mathbf{x}_t^0 \in \mathbb{S}_0$ for $t = 0, 1, \cdots, T$ as

$$\begin{aligned}\mathbf{x}_t^0 =& \sum_{i=0}^{K}(\bar\lambda_{i,t}\mathbf{v}_i^0 + \bar\sigma_{i,t}\mathbf{z}_i^0)\\ =& \boldsymbol{U}_0\bar{\boldsymbol{\Lambda}}_{0,t}\boldsymbol{U}_0^T\mathbf{x}_0^0 + \boldsymbol{U}_0\bar{\boldsymbol{L}}_{0,t}\boldsymbol{U}_0^T\epsilon^0,\end{aligned} \tag{5}$$

where $\mathbf{v}_i^0 \in \mathbb{S}_i/\mathbb{S}_{i+1}$ is the component of original data point $\mathbf{x}_0^0$ in subspace $\mathbb{S}_i/\mathbb{S}_{i+1}$, $\bar\lambda_{i,t}$ controls the attenuation of $\mathbf{v}_i^0$ along timestep t, $\mathbf{z}_i^0$ is the component of a standard Gaussian noise $\epsilon^0 \in \mathbb{R}^{\bar d_0}$ in the same subspace as $\mathbf{v}_i^0$ (*i.e.*, $\mathbb{S}_i/\mathbb{S}_{i+1}$), $\bar\sigma_{i,t}$ is the standard deviation of $\mathbf{z}_i^0$, and $\bar{\boldsymbol{\Lambda}}_{0,t}$, $\bar{\boldsymbol{L}}_{0,t} \in \mathbb{R}^{\bar d_0 \times \bar d_0}$ are two diagonal matrices defined as $\bar{\boldsymbol{\Lambda}}_{0,t} = \mathrm{diag}(\bar\lambda_{0,t}\boldsymbol{I}_{d_0}, \ \bar\lambda_{1,t}\boldsymbol{I}_{d_1}, \ \cdots, \ \bar\lambda_{K,t}\boldsymbol{I}_{d_K})$ and $\bar{\boldsymbol{L}}_{0,t} = \mathrm{diag}(\bar\sigma_{0,t}\boldsymbol{I}_{d_0}, \ \bar\sigma_{1,t}\boldsymbol{I}_{d_1}, \ \cdots, \ \bar\sigma_{K,t}\boldsymbol{I}_{d_K})$. To control the attenuation of each data component $\mathbf{v}_i^0$, we require $\bar\lambda_{i,t}$ to gradually decrease from 1 to approximate 0 for $i = 0, 1, \cdots, K-1$ as timestep t evolves. As for $\bar\lambda_{K,t}$, it is not required to decrease (explained later after Eq. (8)).

Starting from $\mathbf{x}_t^0$, we can recursively construct a dimensionality-decreasing sequence of ADPs

$$\begin{aligned}\mathbf{x}_t^k =& \mathcal{D}_k\mathbf{x}_t^{k-1} = \sum_{i=k}^{K}(\bar\lambda_{i,t}\mathbf{v}_i^k + \bar\sigma_{i,t}\mathbf{z}_i^k)\\ =& \boldsymbol{U}_k\bar{\boldsymbol{\Lambda}}_{k,t}\boldsymbol{U}_k^T\mathbf{x}_0^k + \boldsymbol{U}_k\bar{\boldsymbol{L}}_{k,t}\boldsymbol{U}_k^T\epsilon^k, \ 1 \le k \le K,\end{aligned} \tag{6}$$

where $\mathcal{D}_k : \mathbb{R}^{\bar d_{k-1}} \to \mathbb{R}^{\bar d_k}$ is a linear surjection, which we call the k-th *downsampling operator* as it reduces the dimensionality of the operand (without ambiguity, we also use $\mathcal{D}_k$ to denote the corresponding matrix in $\mathbb{R}^{\bar d_k \times \bar d_{k-1}}$), $\mathbf{v}_i^k = \overline{\mathcal{D}}_k\mathbf{v}_i^0 \in \overline{\mathcal{D}}_k(\mathbb{S}_i/\mathbb{S}_{i+1})$ is the component of $\mathbf{x}_0^k = \overline{\mathcal{D}}_k\mathbf{x}_0^0 \in \mathbb{R}^{\bar d_k}$ ($\overline{\mathcal{D}}_k \triangleq \prod_{i=1}^k \mathcal{D}_i$), $\mathbf{z}_i^k = \overline{\mathcal{D}}_k\mathbf{z}_i^0 \in \overline{\mathcal{D}}_k(\mathbb{S}_i/\mathbb{S}_{i+1})$ is the component of a standard Gaussian noise $\epsilon^k = \overline{\mathcal{D}}_k\epsilon^0 \in \mathbb{R}^{\bar d_k}$, $\bar{\boldsymbol{\Lambda}}_{k,t} = \mathrm{diag}(\bar\lambda_{k,t}\boldsymbol{I}_{d_k}, \ \bar\lambda_{k+1,t}\boldsymbol{I}_{d_{k+1}}, \ \cdots, \ \bar\lambda_{K,t}\boldsymbol{I}_{d_K}) \in \mathbb{R}^{\bar d_k \times \bar d_k}$, $\bar{\boldsymbol{L}}_{k,t} = \mathrm{diag}(\bar\sigma_{k,t}\boldsymbol{I}_{d_k}, \ \bar\sigma_{k+1,t}\boldsymbol{I}_{d_{k+1}}, \ \cdots, \ \bar\sigma_{K,t}\boldsymbol{I}_{d_K}) \in \mathbb{R}^{\bar d_k \times \bar d_k}$, and orthogonal matrix $\boldsymbol{U}_k \in \mathbb{R}^{\bar d_k \times \bar d_k}$ consists of the last $\bar d_k$ columns of $\mathcal{D}_k\boldsymbol{U}_{k-1}$, *i.e.*, $\boldsymbol{U}_k = \mathcal{D}_k\boldsymbol{P}_{k-1}$ (note that this condition requires further restriction on $\mathcal{D}_k$).

Both Eqs. (5) and (6) can be derived from Markov chains with Gaussian kernels as (see *Supplementary Material* for the proof)

$$\mathbf{x}_t^k = \boldsymbol{U}_k\boldsymbol{\Lambda}_{k,t}\boldsymbol{U}_k^T\mathbf{x}_{t-1}^k + \boldsymbol{U}_k\boldsymbol{L}_{k,t}\boldsymbol{U}_k^T\epsilon^k, \ 0 \le k \le K, \tag{7}$$

where $\boldsymbol{\Lambda}_{k,t} = \bar{\boldsymbol{\Lambda}}_{k,t-1}^{-1}\bar{\boldsymbol{\Lambda}}_{k,t}$, $\boldsymbol{L}_{k,t} = (\bar{\boldsymbol{L}}_{k,t}^2 - \boldsymbol{\Lambda}_{k,t}^2\bar{\boldsymbol{L}}_{k,t-1}^2)^{1/2}$.

Now with the ADPs $\{\mathbf{x}_t^k\}_{k=0}^K$ given by Eq. (7), we can construct the forward process of DVDP by merging different parts of $\{\mathbf{x}_t^k\}_{k=0}^K$ in the following manner: consider a strictly increasing time sequence $T_1, \cdots, T_K, T_{K+1} = T$, if for each k satisfying $1 \le k \le K$, $\bar\lambda_{k-1,T_k}$ becomes small enough, then $\mathbf{x}_{T_k}^{k-1}$ is downsampled by $\mathcal{D}_k$ to obtain $\mathbf{x}_{T_k}^k$ with lower dimensionality, and each T_k is a *dimensionality turning point*. The entire process can be expressed as

$$\begin{array}{lllll} & \mathbf{x}_0^0 & \longrightarrow \mathbf{x}_1^0 & \longrightarrow \cdots & \longrightarrow \mathbf{x}_{T_1}^0 \\ \xrightarrow{\mathcal{D}_1} & \mathbf{x}_{T_1}^1 & \longrightarrow \mathbf{x}_{T_1+1}^1 & \longrightarrow \cdots & \longrightarrow \mathbf{x}_{T_2}^1 \\ \vdots & & & & \\ \xrightarrow{\mathcal{D}_K} & \mathbf{x}_{T_K}^K & \longrightarrow \mathbf{x}_{T_K+1}^K & \longrightarrow \cdots & \longrightarrow \mathbf{x}_T^K. \end{array} \tag{8}$$

Between two adjacent dimensionality turning points T_{k-1} and T_k, $\mathbf{x}_t^{k-1}$ diffuses and attenuates data components $\mathbf{v}_i^{k-1}$, $i \ge k-1$, which keeps the dimensionality $\bar d_{k-1}$. When it comes to T_k, $\mathbf{x}_{T_k}^{k-1} \xrightarrow{\mathcal{D}_k} \mathbf{x}_{T_k}^k$ decreases the dimension from $\bar d_{k-1}$ to $\bar d_k$. After the last dimensionality turning point T_k, $\mathbf{x}_t^K$ can just evolve as conventional diffusion without data component attenuation. Thus, the entire process in Eq. (8) decreases the dimensionality by K times from $\bar d_0 = d$ to $\bar d_K$. It should be noted that process in Eq. (8) is also Markovian since each diffusion subprocess $\mathbf{x}_{T_{k-1}}^{k-1} \to \mathbf{x}_{T_k}^{k-1}$ is Markovian, and the result of each downsampling operation $\mathbf{x}_{T_k}^{k}$ is uniquely determined by the previous state $\mathbf{x}_{T_k}^{k-1}$. Also note that each downsampling operation $\mathcal{D}_k$ loses little information because of small $\bar{\lambda}_{k-1,T_k}$ which is a controllable hyperparameter. For better understanding, consider the relationship between $\mathbf{x}_{T_k}^{k-1}$ and $\mathbf{x}_{T_k}^{k}$ derived from Eq. (6)

$$\mathbf{x}_{T_k}^{k-1} = \mathcal{D}_k^T \mathbf{x}_{T_k}^{k} + \underbrace{\bar{\lambda}_{k-1,T_k} \mathbf{v}_{k-1}^{k-1}}_{\text{data component}} + \underbrace{\bar{\sigma}_{k-1,T_k} \mathbf{z}_{k-1}^{k-1}}_{\text{noise component}}, \tag{9}$$

where $\mathcal{D}_k^T$ is the transpose of matrix $\mathcal{D}_k$, named as the k-th *upsampling operator*. From Eq. (9), it is clear that $\mathbf{x}_{T_k}^{k}$ actually loses two terms compared with $\mathbf{x}_{T_k}^{k-1}$: 1) data component $\bar{\lambda}_{k-1,T_k} \mathbf{v}_{k-1}^{k-1}$ which is informative but negligible as $\bar{\lambda}_{k-1,T_k}$ is set to be small enough, and 2) noise component $\bar{\sigma}_{k-1,T_k} \mathbf{z}_{k-1}^{k-1}$ that is non-informative and can be compensated in the reverse process, as we will discuss in Sec. 4.2.

4.2. Reverse Process Approximating DVDP

In this section, we will derive an approximate reverse process, which induces a data generation process with progressively growing dimensionality. The approximation error will be discussed in Sec. 4.3, and we can find that it actually converges to zero. Loss function will also be given at the end of this section. Implementation details of training and sampling can be found in *Supplementary Material*.

Reverse transition. Since DVDP is a sequence of fixed-dimensionality diffsion processes connected by downsampling operations at dimensionality turning points, we consider reverse transition kernels *between* and *at* dimensionality turning points separately.

For reverse transition *between* two adjacent dimensionality turning points, *i.e.*, $p_\theta(\mathbf{x}_{t-1}^{k} | \mathbf{x}_t^{k})$ with $T_{k-1} \leq t \leq T_k$ for $k \in [1, K+1]$, it can be defined as a Gaussian kernel $p_\theta(\mathbf{x}_{t-1}^{k} | \mathbf{x}_t^{k}) = \mathcal{N}(\mathbf{x}_{t-1}^{k}; \boldsymbol{\mu}_\theta(\mathbf{x}_t^{k}, t), \boldsymbol{\Sigma}_t)$. As in DDPM [6], the covariance matrices $\boldsymbol{\Sigma}_t$ are set to untrained time-dependent constants, and the mean term $\boldsymbol{\mu}_\theta(\mathbf{x}_t^{k}, t)$ is defined as (see *Supplementary Material* for details)

$$\begin{aligned} \boldsymbol{\mu}_\theta =& \tilde{\boldsymbol{\mu}}_{k,t}\big(\mathbf{x}_t^{k}, \boldsymbol{U}_k \bar{\boldsymbol{\Lambda}}_{k,t}^{-1} \boldsymbol{U}_k^T \mathbf{x}_t^{k} \\ & - \boldsymbol{U}_k \bar{\boldsymbol{\Lambda}}_{k,t}^{-1} \bar{\boldsymbol{L}}_{k,t} \boldsymbol{U}_k^T \boldsymbol{\epsilon}_\theta(\mathbf{x}_t^{k}, t)\big), \end{aligned} \tag{10}$$

where $\tilde{\boldsymbol{\mu}}_{k,t}(\cdot,\cdot)$ is the mean function of forward process posterior $q(\mathbf{x}_{t-1}^{k} | \mathbf{x}_t^{k}, \mathbf{x}_0^{k}) = \mathcal{N}(\mathbf{x}_{t-1}^{k}; \tilde{\boldsymbol{\mu}}_{k,t}(\mathbf{x}_t^{k}, \mathbf{x}_0^{k}), \tilde{\boldsymbol{\Sigma}}_{k,t})$, and $\boldsymbol{\epsilon}_\theta$ represents a trainable network.

For reverse transition *at* dimensionality turning points, *i.e.*, $p_\theta(\mathbf{x}_{T_k}^{k-1} | \mathbf{x}_{T_k}^{k})$ for $k \in [1, K]$, the corresponding forward transitions barely lose information as illustrated by Eq. (9) in Sec. 4.1, thus $\mathbf{x}_{T_k}^{k} \to \mathbf{x}_{T_k}^{k-1}$ can be approximately achieved without any trainable network as

$$\mathbf{x}_{T_k}^{k-1} = \mathcal{D}_k^T \mathbf{x}_{T_k}^{k} + \boldsymbol{U}_{k-1} \Delta \boldsymbol{L}_{k-1} \boldsymbol{U}_{k-1}^T \epsilon^{k-1}, \tag{11}$$

where $\mathcal{D}_k^T \in \mathbb{R}^{\bar{d}_k \times \bar{d}_{k-1}}$ is the upsampling operator, and $\Delta \boldsymbol{L}_{k-1} = \operatorname{diag}(\bar{\sigma}_{k-1,T_k} \boldsymbol{I}_{d_{k-1}}, \boldsymbol{O}_{\bar{d}_k})$ represents the standard deviation of the added Gaussian noise $\epsilon^{k-1} \in \mathbb{R}^{\bar{d}_{k-1}}$. Eq. (11) can be understood as: we first upsample $\mathbf{x}_{T_k}^{k}$, then add a Gaussian noise in compensation for the lost noise component in the forward downsampling operation, *i.e.*, $\bar{\sigma}_{k-1,T_k} \mathbf{z}_{k-1}^{k-1}$ in Eq. (9). The approximation error comes from neglecting data component $\bar{\lambda}_{k-1,T_k} \mathbf{v}_{k-1}^{k-1}$, and will be analyzed later in Sec. 4.3.

Loss function. Similar with DDPM [6], a loss function can be derived from a weighted variational bound as (see *Supplementary Material* for details)

$$L(\theta) = \mathbb{E}_k \mathbb{E}_{\mathbf{x}_0^k, \epsilon^k, t \sim \mathcal{U}_k} \left[\left\| \epsilon^k - \epsilon_\theta(\mathbf{x}_t^k(\mathbf{x}_0^k, \epsilon^k), t) \right\|^2 \right], \tag{12}$$

where $\mathcal{U}_k = \mathcal{U}\big((T_k, T_{k+1}]\big)$ is a discrete uniform distribution between T_k (exclusive) and T_{k+1} (inclusive), and $\mathbf{x}_t^k(\mathbf{x}_0^k, \epsilon^k)$ represents the forward $\mathbf{x}_t^k$ determined by $\mathbf{x}_0^k$ and ϵ^k given in Eq. (6).

4.3. Error Analysis

In Sec. 4.2, we mention that the reverse process is just an approximation of the forward DVDP at each dimensionality turning point T_k. In this section, we will measure this approximation error in probability sense, *i.e.*, the difference between the real forward distribution $q(\mathbf{x}_{T_k}^{k-1})$ and the reverse distribution $p(\mathbf{x}_{T_k}^{k-1})$ under proper assumptions, and will find that this difference converges to zero.

To measure the difference between two distributions, we use *Jensen-Shannon Divergence* (JSD) as a metric. Under this metric, upper bound of the approximation error can be derived from Proposition 1 (see *Supplementary Material* for proof):

Proposition 1 *Assume $p_1(\boldsymbol{x}|\boldsymbol{x}_0)$, $p_2(\boldsymbol{x}|\boldsymbol{x}_0)$ are two Gaussians such 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 $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ satisfy $\boldsymbol{A}_1 \succeq \boldsymbol{A}_2 \succeq 0$, 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*

$$\begin{aligned} \mathrm{JSD}(p_1 || p_2) \leq & \frac{\sqrt{2}}{2} e^{-\frac{1}{2}} B \left(2\sqrt{2} + \frac{V_d(r)}{(2\pi)^{\frac{d}{2}}} \right) \\ & \cdot \|\boldsymbol{\Sigma}^{-\frac{1}{2}} (\boldsymbol{A}_1 - \boldsymbol{A}_2)\|_2, \end{aligned} \tag{13}$$

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

With Proposition 1, the upper bound of JSD between the forward distribution $q(\mathbf{x}_{T_k}^{k-1})$ and the reverse distribution $p(\mathbf{x}_{T_k}^{k-1})$ can be obtained by Theorem 1 (see *Supplementary Material* for proof)

Theorem 1 (Reverse Process Error) *Assume $0 < k \leq K$, $q(\mathbf{x}_{T_k}^{k-1})$ and $q(\mathbf{x}_{T_k}^{k})$ are defined by Eqs. (5) and (6), $p(\mathbf{x}_{T_k}^{k-1})$ is the marginal distribution of $q(\mathbf{x}_{T_k}^{k})p(\mathbf{x}_{T_k}^{k-1}|\mathbf{x}_{T_k}^{k})$ where $p(\mathbf{x}_{T_k}^{k-1}|\mathbf{x}_{T_k}^{k})$ is defined by Eq. (11), and $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$, then*

$$\begin{aligned}\xi_1 \leq &\frac{\sqrt{2}}{2}e^{-\frac{1}{2}}\sqrt{d}\left(2\sqrt{2}+\frac{V_d(r)}{(2\pi)^{\frac{d}{2}}}\right)\frac{\bar{\lambda}_{k-1,T_k}}{\bar{\sigma}_{k-1,T_k}} \\ =&o(\bar{\lambda}_{k-1,T_k}),\end{aligned} \tag{14}$$

where $\xi_1 \triangleq \mathrm{JSD}(q(\mathbf{x}_{T_k}^{k-1})||p(\mathbf{x}_{T_k}^{k-1}))$, and $r = 2\sqrt{d}\max_{k-1\leq i\leq K}\frac{\bar{\lambda}_{i,T_k}}{\bar{\sigma}_{i,T_k}}$.

Note that the assumption $\|\mathbf{x}_0\|_2 \leq \sqrt{d}$ can be satisfied for image data, since pixel values can be normalized in $[-1, 1]$. Thus, Theorem 1 claims that ξ_1 can be arbitrarily small as $\bar{\lambda}_{k-1,T_k} \to 0$ if we can get an exact $q(\mathbf{x}_{T_k}^{k-1})$ by reverse process. It means that the approximation error caused by stepping over T_k can be small enough.

4.4. Comparison with Subspace Diffusion

To reduce the dimensionality of latent space in diffusion models, subspace diffusion is proposed to model in a low-dimensional subspace at high noise levels, and keep the original full-dimensional network at low noise levels [10]. This can also be seen as a concatenation of multiple diffusion processes with different dimensionality like our DVDP, but without controllable attenuation on each data component. Each concatenated processes is just conventional isotropic diffusion.

Thus, subspace diffusion can be seen as a special case of our DVDP with $\bar{\lambda}_t \triangleq \bar{\lambda}_{0,t} = \bar{\lambda}_{1,t} = \cdots = \bar{\lambda}_{K,t}$ and $\bar{\sigma}_t \triangleq \bar{\sigma}_{0,t} = \bar{\sigma}_{1,t} = \cdots = \bar{\sigma}_{K,t}$, which limits the choice of dimensionality turning points. This limitation can be further explained by Eq. (9): at dimensionality turning point T_k in the forward process, $\mathbf{x}_{T_k}^{k-1}$ will lose an informative data component $\bar{\lambda}_{k-1,T_k}\mathbf{v}_{k-1}^{k-1}$ and a non-informative noise component $\bar{\sigma}_{k-1,T_k}\mathbf{z}_{k-1}^{k-1}$. To safely neglect the data component in the reverse transition, it requires that $\bar{\lambda}_{k-1,T_k}/\bar{\sigma}_{k-1,T_k} \ll \|\mathbf{z}_{k-1}^{k-1}\|/\|\mathbf{v}_{k-1}^{k-1}\|$. For subspace diffusion, it means that the consistent $\bar{\lambda}_t/\bar{\sigma}_t$ for components in all subspaces should be small enough, which usually indicates a large T_k, *i.e.*, a large number of diffusion steps in high dimensional space.

Therefore, as claimed in [10], the choice of T_k should balance two factors: 1) smaller T_k reduces the number of reverse diffusion steps occurring at higher dimensionality, whereas 2) larger T_k makes the reverse transition at T_k more accurate. Although [10] additionally proposes to compensate the loss of data component by adding an extra Gaussian noise besides compensation for the noise component, this trade-off still exists.

However, our DVDP can set much smaller T_k with little loss in accuracy, which benefits from the controllable attenuation for each data component. Theorem 1 supports this advantage theoretically, and experimental results in Sec. 5.3 further demonstrate it.

5. 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 varied dimension, DVDP is able to generate high-quality and high-resolution images from a low-dimensional subspace and exceeds existing methods including score-SDE [28] and Cascaded Diffusion Models (CDM) [7] on FFHQ 1024^2. Specifically, we first introduce our experimental setup in Sec. 5.1. Then we compare our DVDP with existing alternatives on several widely evaluated datasets in terms of visual quality and modeling efficiency in Sec. 5.2. After that, we compare our DVDP with Subspace Diffusion [10], a closely related work proposed recently, in Sec. 5.3. Finally, we implement necessary ablation studies in the last Sec. 5.4.

5.1. Experimental Setup

Datasets. In order to verify that DVDP is widely applicable, we use six image datasets covering various classes and a wide range of resolutions from 32 to 1024. To be specific, we implement DVDP on CIFAR10 32^2 [12], LSUN Bedroom 256^2 [32], LSUN Church 256^2, LSUN Cat 256^2, FFHQ 256^2, and FFHQ 1024^2 [11].

Implementation details. We adopt the UNet improved by [2] which achieves better performance than the traditional version [6]. Since most baseline methods adopt a single UNet network for all timesteps in the whole diffusion process, our DVDP also keeps this setting, except the comparison with Subspace Diffusion in Sec. 5.3, which takes two networks for different generation stages [10]. In principle, the network structure of our DVDP is kept the same as the corresponding baseline. However, when resolution comes to 1024×1024, the UNet in conventional diffusion models should be deep enough to contain sufficient downsampling blocks, thus to obtain embeddings with proper size (usually 4×4 or 8×8) in the bottleneck layer. On the contrary, our DVDP does not need such a deep network since the generation starts from a low resolution noise (64×64 in our case). Thus we only maintain a similar amount of parameters but use a different hyperparameter setting on FFHQ 1024^2.

We set the number of timesteps $T = 1,000$ in all of our experiments. For DVDP, we reduce the dimensionality by $\frac{1}{4}$ (*i.e.*, $h \times w \to \frac{h}{2} \times \frac{w}{2}$ for image resolution) when the timestep t reaches a dimensionality turning point. For simplicity, we adopt $\mathbb{T} \triangleq \{T_1, \cdots, T_K\}$. For CIFAR10

Table 1. **Quantitative comparison** between DDPM [6] and our DVDP on various datasets regarding image quality and model efficiency. $*$ indicates our reproduced DDPM. Both DDMP* and our DVDP adopt the improved UNet [2] for a fair comparison.

Dataset	Method	FID (50k)↓	Training Speed (sec/iter)	Training Speed Up	Sampling Speed (sec/sample)	Sampling Speed Up
CIFAR10 32 × 32	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*	5.74	0.99	–	12.2	–
	DVDP	**4.88**	**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×

32^2, we set $\mathbb{T} = \{600\}$, indicating that the resolution is decreased from 32×32 to 16×16 when $t = 600$. Similarly, we set $\mathbb{T} = \{300, 600\}$ for all 256^2 datasets and $\mathbb{T} = \{200, 400, 600, 800\}$ for FFHQ 1024^2.

In all of our experiments, the noise schedule of DVDP is an adapted version of linear schedule [6], which is suitable for DVDP and keeps a comparable signal-to-noise ratio (SNR) with the original version (see *Supplementary Material* for details).

Evaluation metrics. For sample quality, we calculate the FID score [5] on 50k samples, except for FFHQ 1024^2 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. Training speed is measured by the mean time of each iteration (estimated over 4,000 iterations), and sampling speed is measured by the mean time of each sample (estimated over 100 batches). The training batch size and sampling batch size are 128, 256 respectively for CIFAR10 32^2, and 24, 64 respectively for other 256^2 datasets.

5.2. 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*. Some samples generated by our DVDP are shown in Fig. 1, and Tab. 1 summarizes the quantitative results on CIFAR10, FFHQ 256^2, and three LSUN datasets. The results show that our proposed DVDP achieves better FID scores on all of the 256^2 datasets, demonstrating an improved sample quality. Meanwhile, DVDP enjoys improved training and sampling speeds. Specifically, DDPM and DDPM* spend 2.2× time as DVDP when training the same epochs, and they spend 2.4× time in sampling. Although the superiority of DVDP is obvious on all 256^2 datasets, it becomes indistinct when it comes to CIFAR10 32^2, which is reasonable considering the small pixel redundancy and low resolution of images in CIFAR10.

Towards high-resolution image synthesis. It is hard for diffusion models to generate high-resolution images directly. This can attribute to the curse of dimensionality: the support of high-dimensional data with large noise can only be visited a small part during training, leading to inaccurate prediction on unseen points [10]. Score-SDE [28] tries this task by directly training a single diffusion model, but the sample quality is far from reasonable. CDM performs better in high-resolution image synthesis and obtains impressive results [21, 24]. It first generates low-resolution images, followed by several conditional diffusion models as super-resolution modules. We compare DVDP with score-SDE and CDM on FFHQ 1024^2 in Tab. 2. The FID of score-SDE is evaluated on samples generated from their official code and model weight without acceleration, and CDM is implemented by three cascaded diffusions as in [21]. Our DVDP is sampled by both DDPM method with 1000 steps and DDIM method with 675 steps. The results show that DVDP beats both score-SDE and CDM.

5.3. Comparison with Subspace Diffusion

Subspace diffusion [10] can also vary dimensionality during the diffusion process. As mentioned in [10] and also

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

Model	#Params (M)	NFE	FID (10k)↓
Score-SDE	100	2000	52.40
CDM	98	1525	24.7
	165	1525	17.35
	286	1525	17.24
DVDP	105	675	12.43
		1000	**10.46**

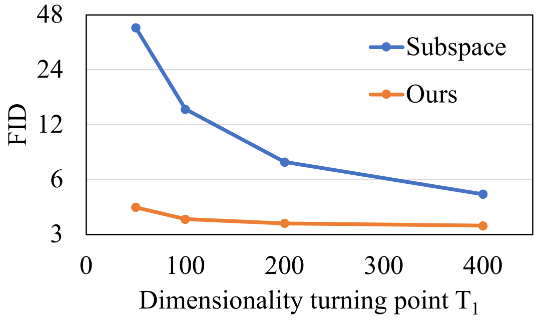

Figure 4. **Quantitative comparison** between subspace diffusion [10] and our DVDP on CelebA 64×64 regarding different dimensionality turning point T_1.

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

Downsampling times K	0	1	2	3
FID (50k)	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×

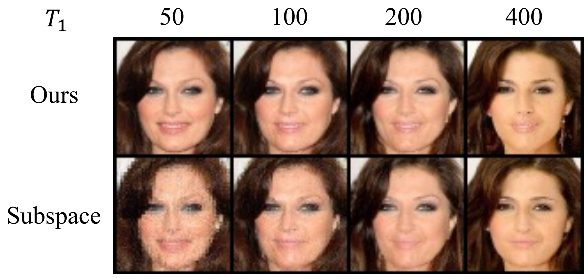

Figure 5. **Qualitative comparison** between subspace diffusion [10] and our DVDP on CelebA 64×64. T_1 denotes the dimensionality turning point.

discussed in Sec. 4.4, dimensionality turning point T_k in subspace diffusion should be large enough to maintain the sample quality. However, large T_k means more diffusion steps in high dimensional space, which will impair the advantage of such dimensisonality-varying method, *e.g.*, less acceleration in sampling. Thus, T_k is expected to be as small as possible while maintaining the sampling quality.

Considering that the dimensionality decreases only once, *i.e.*, $K = 1$, and only one dimensionality turning point T_1, we compare DVDP with subspace diffusion when the downsampling is carried out at different T_1. Since the subspace diffusion is only implemented on continuous timesteps in [10], 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 [16] when T_1 varies, where the advantage gets larger when T_1 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 T_1 is small. In conclusion, DVDP is much more insensitive to the dimensionality turning point than subspace diffusion.

5.4. Ablation Study

We implement ablation study in this section to show that DVDP is able to keep effective when the number of downsampling, *i.e.*, K, grows. Specifically, we train DVDP models on four different settings of dimensionality turning points $\mathbb{T}$ for $K = 0, 1, 2, 3$. When $K = 1$, $\mathbb{T}$ is set to $\{250\}$. Similarly, when $K = 2$ and $K = 3$, $\mathbb{T}$ is set to $\{250, 500\}$ and $\{250, 500, 750\}$, respectively. Furthermore, we use the same noise schedule for these four different settings. Tab. 3 shows that when the number of downsampling grows, the sample quality preserves a reasonable level, indicating that DVDP can vary dimensionality for multiple times.

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-attenuated 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. This design allows much higher training and sampling speed of the diffusion models with on-par or even better synthesis performance, and superior performance in synthesizing large-size 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.

This work was supported in part by the National Key R&D Program of China under Grant 2022YFA1005000, in part by the NSFC under Grant 61701304.

References

[1] 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. 2

[2] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. *Adv. Neural Inform. Process. Syst.*, pages 8780–8794, 2021. 1, 6, 7

[3] Patrick Esser, Robin Rombach, Andreas Blattmann, and Björn Ommer. Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. *Adv. Neural Inform. Process. Syst.*, pages 3518–3532, 2021. 3

[4] 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.*, pages 16000–16009, 2022. 1

[5] 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. 2, 7

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Adv. Neural Inform. Process. Syst.*, pages 6840–6851, 2020. 1, 2, 3, 5, 6, 7

[7] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *J. Mach. Learn. Res.*, pages 47–1, 2022. 2, 6

[8] Minghui Hu, Yujie Wang, Tat-Jen Cham, Jianfei Yang, and Ponnuthurai N Suganthan. Global context with discrete diffusion in vector quantised modelling for image generation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 11502–11511, 2022. 3

[9] Lianghua Huang, Di Chen, Yu Liu, Yujun Shen, Deli Zhao, and Jingren Zhou. Composer: Creative and controllable image synthesis with composable conditions. *ArXiv:2302.09778*, 2023. 1

[10] Bowen Jing, Gabriele Corso, Renato Berlinghieri, and Tommi Jaakkola. Subspace diffusion generative models. *ArXiv:2205.01490*, 2022. 1, 2, 6, 7, 8

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4401–4410, 2019. 6

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

[13] Sangyun Lee, Hyungjin Chung, Jaehyeon Kim, and Jong Chul Ye. Progressive deblurring of diffusion models for coarse-to-fine image synthesis. *ArXiv:2207.11192*, 2022. 3

[14] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. Pseudo numerical methods for diffusion models on manifolds. *ArXiv:2202.09778*, 2022. 2

[15] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept neurons in diffusion models for customized generation. *ArXiv:2303.05125*, 2023. 1

[16] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Int. Conf. Comput. Vis.*, pages 3730–3738, 2015. 8

[17] 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. *Adv. Neural Inform. Process. Syst.*, 2022. 2

[18] Troy Luhman and Eric Luhman. Improving diffusion model efficiency through patching. *ArXiv:2207.04316*, 2022. 2

[19] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Int. Conf. Mach. Learn.*, pages 8162–8171, 2021. 2

[20] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10619–10629, 2022. 3

[21] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *ArXiv:2204.06125*, 2022. 1, 2, 7

[22] 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.*, pages 10684–10695, 2022. 1, 3

[23] Dohoon Ryu and Jong Chul Ye. Pyramidal denoising diffusion probabilistic models. *ArXiv:2208.01864*, 2022. 2

[24] 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:2205.11487*, 2022. 1, 2, 7

[25] Robin San-Roman, Eliya Nachmani, and Lior Wolf. Noise estimation for generative diffusion models. *ArXiv:2104.02600*, 2021. 2

[26] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Int. Conf. Mach. Learn.*, pages 2256–2265, 2015. 1, 2, 3

[27] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *Int. Conf. Learn. Represent.*, 2021. 2

[28] 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. 1, 2, 6, 7

[29] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Adv. Neural Inform. Process. Syst.*, pages 11287–11302, 2021. 3

[30] 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.*, 2021. 2

[31] Daniel Watson, Jonathan Ho, Mohammad Norouzi, and William Chan. Learning to efficiently sample from diffusion probabilistic models. *ArXiv:2106.03802*, 2021. 2

[32] 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:1506.03365*, 2015. 6

[33] Qinsheng Zhang and Yongxin Chen. Diffusion normalizing flow. *Adv. Neural Inform. Process. Syst.*, pages 16280–16291, 2021. 2