# Subspace Diffusion Generative Models

Bowen Jing*, Gabriele Corso*, Renato Berlinghieri, and Tommi Jaakkola

Massachusetts Institute of Technology
{bjing, gcorso, renb}@mit.edu, tommi@csail.mit.edu

**Abstract.** Score-based models generate samples by mapping noise to data (and vice versa) via a high-dimensional diffusion process. We question whether it is necessary to run this entire process at high dimensionality and incur all the inconveniences thereof. Instead, we restrict the diffusion via projections onto *subspaces* as the data distribution evolves toward noise. When applied to state-of-the-art models, our framework simultaneously *improves* sample quality—reaching an FID of 2.17 on unconditional CIFAR-10—and *reduces* the computational cost of inference for the same number of denoising steps. Our framework is fully compatible with continuous-time diffusion and retains its flexible capabilities, including exact log-likelihoods and controllable generation. Code is available at https://github.com/bjing2016/subspace-diffusion.

**Keywords:** Generative models, diffusion models, score matching

## 1 Introduction

Score-based models are a class of generative models that learn the score of the data distribution as it evolves under a diffusion process in order to generate data via the reverse process [20, 6]. These models—also known as diffusion models—can generate high-quality and diverse samples, evaluate exact log-likelihoods, and are easily adapted to conditional and controlled generation tasks [20]. On the CIFAR-10 image dataset, they have recently achieved state-of-the-art performance in sample generation and likelihood evaluation [21, 9].

Despite these strengths, in this work we focus on and aim to address a drawback in the current formulation of score-based models: the forward diffusion occurs in the full ambient space of the data distribution, destroying its structure but retaining its high dimensionality. However, it does not seem parsimonious to represent increasingly noisy latent variables—which approach zero mutual information with the original data—in a space with such high dimensionality. The practical implications of this high latent dimensionality are twofold:

*High-dimensional extrapolation.* The network must learn the score function over the entire support of the high-dimensional latent variable, even in areas very far (relative to the scale of the data) from the data manifold. Due to the curse of dimensionality, much of this support may never be visited during training, and the accuracy of the score model in these regions is called into question by

[*] Equal contribution

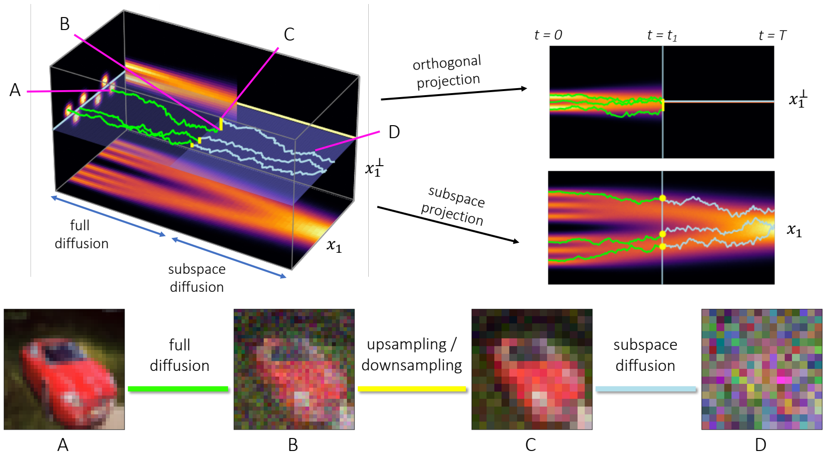

Fig. 1: Visual schematic of subspace diffusion with one projection step. *Top left*: The starting data distribution $\mathbf{x}_0(0)$ lies near a subspace (light blue line). As the data evolves, the distribution of the orthogonal component $\mathbf{x}_1^\perp(t)$ approaches a Gaussian faster than the subspace component $\mathbf{x}_1(t)$. At time $t_1$ we project onto the subspace and restrict the remaining diffusion to the subspace. To generate data, we use the full and subspace score models to reverse the full and subspace diffusion steps, and sample $\mathbf{x}_1^\perp(t_1)$ from a Gaussian to reverse the projection step. *Top right*: The diffusion of the subspace component $\mathbf{x}_1(t)$ is unaffected by the projection step and restriction to the subspace; while the orthogonal component is diffused until $t_1$ and discarded afterwards. *Bottom*: CIFAR-10 images corresponding to points along the trajectory, where the subspaces correspond to lower-resolution images and projection is equivalent to downsampling.

the uncertain extrapolation abilities of neural networks [23]. Learning to match a lower-dimensional score function may lead to refined training coverage and further improved performance.

*Computational cost.* Hundreds or even thousands of evaluations of the high-dimensional score model are required to generate an image, making inference with score-based models much slower than GANs and VAEs [6, 20]. A number of recent works aim to address this challenge by reducing the number of steps required for inference [18, 16, 8, 13, 3, 10, 22, 17, 11, 2]. However, these methods trade-off inference runtime with sample quality. Moreover, the dimensionality of the score function—and thereby the computational cost of a single score evaluation—is an independent and equally important factor to the overall runtime, but this factor has received less attention in existing works.

***Subspace diffusion models*** aim to address these challenges. In many real-world domains, target data lie near a linear subspace, such that under isotropic forward diffusion, the components of the data orthogonal to the subspace become Gaussian significantly before the components in the subspace. We propose to use a full-dimensional network to model the score only at lower noise levels, when all components are sufficiently non-Gaussian. At higher noise levels, we use smaller networks to model in the subspace only those components of the score which remain non-Gaussian. As this reduces both the number and domain of queries to the full-dimensional network, subspace diffusion addresses both of our motivating concerns. Moreover, in contrast to many prior works, subspace diffusion remains fully compatible with the underlying continuous diffusion framework [20], and therefore preserves all the capabilities available to continuous score-based models, such as likelihood evaluation, probability flow sampling, and controllable generation.

While subspace diffusion can be formulated in fully general terms, in this work we focus on generative modeling of natural images. Because the global structure of images is dominated by low-frequency visual components—i.e., adjacent pixels values are highly correlated—images lie close to subspaces corresponding to lower-resolution versions of the same image. Learning score models over these subspaces has the advantage of remaining compatible with the translation equivariance of convolutional neural networks, and therefore requires no architectural modifications to the score model.

**Contributions** We formulate the diffusion process, training procedure, and sampling procedure in subspaces; to our knowledge, this represents the first investigation of dimensionality reduction in a score-based model framework. We develop a method, the *orthogonal Fisher divergence*, for choosing among candidate subspaces and the parameters of the subspace diffusion. Experimentally, we train and evaluate lower-dimensional subspace models in conjunction with state-of-the-art pretrained full-dimensional models from [20]. We improve over those models in sample quality and runtime, achieving an FID of 2.17 and a IS of 9.99 on CIFAR-10 generation. Finally, we demonstrate probability flow sampling and likelihood evaluation with subspace models.

## 2 Background and Related Work

**Score-based models** In score-based models, one considers the data distribution $\mathbf{x}(0) \in \mathbb{R}^d$ to be the starting distribution for a continuous diffusion process, defined by an Ito stochastic differential equation (SDE)

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\ dt + \mathbf{G}(\mathbf{x}, t)\ d\mathbf{w} \quad t \in (0, T) \tag{1}$$

known as the *forward process*, which transforms $\mathbf{x}(0)$ into (approximately) a simple Gaussian $\mathbf{x}(T)$. By convention, we typically set $T = 1$. A neural network is then trained to model the score $\nabla_\mathbf{x} \log p_t(\mathbf{x})$ conditioned on $t$. Solving the reverse stochastic differential equation

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\ dt - \mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^T \nabla_\mathbf{x} \log p(\mathbf{x}, t)\ dt + \mathbf{G}(t)\ d\bar{\mathbf{w}} \tag{2}$$

starting with samples from the simple Gaussian distribution $\mathbf{x}(T)$ yields samples from the data distribution $\mathbf{x}(0)$ [20, 1]. Score-based models were originally formulated separately in terms of denoising score matching at multiple noise scales

[19]; and of reversing a discrete-time Markov chain of diffusion steps [6]. Due to the latter formulation (associated with the term *diffusion model*), $\mathbf{x}(t)$ for $t > 0$ are often referred to as *latents* of $\mathbf{x}(0)$, and the simple Gaussian $\mathbf{x}$ as the *prior*. The two views are unified by the observation that the variational approximation to the reverse Markov chain matches the score of the diffused data [20].

The score model $s_\theta(\mathbf{x}, t)$ can be trained via denoising score matching [19] using the perturbation kernels $p(\mathbf{x}(t) \mid \mathbf{x}(0))$, which are analytically determined by $\mathbf{f}(\mathbf{x}, t), \mathbf{G}(t)$ at each time $t$. The learned score can be readily adjusted with fixed terms for controlled generation tasks in the same manner as energy-based models [5]. Finally, the reverse stochastic differential equation produces the same marginals $\mathbf{x}$ as the *ordinary* differential equation (ODE)

$$d\mathbf{x} = \mathbf{f}(\mathbf{x}, t)\ dt - \frac{1}{2}\mathbf{G}(\mathbf{x}, t)\mathbf{G}(\mathbf{x}, t)^T \nabla_{\mathbf{x}} \log p(\mathbf{x}, t)\ dt \tag{3}$$

which enables evaluation of exact log-likelihoods, but empirically results in degraded quality when used for sampling [20].

**Accelerating score-based models** Due to the fine discretization required to solve (2) to high accuracy, score-based models suffer from slow inference. Several recent works aim to address this. Denoising diffusion implicit models (DDIM) [18] can be viewed as solving the equivalent ODE with a reduced number of steps. Progressive distillation [16] proposes a student-teacher framework for learning sampling networks requiring logarithmically fewer steps. [8] derives an adaptive step-size solver for the reverse SDE. Other works [13, 3, 10, 22, 17, 11, 2] focus on reducing the number of steps in the discrete-time Markov chain formulation. However, these approaches generally result in degraded sample quality compared to the best continuous-time models.

Taking a different approach, latent score-based generative models (LSGM) [21] use a score-based model as the prior of a deep VAE, resulting in more Gaussian scores, improved sample quality, and fewer model evaluations. In a similar vein, critically-damped Langevin diffusion (CLD-SGM) [4] augments the data dimensions with momentum dimensions and diffuses only in momentum space, resulting in more Gaussian scores and fewer evaluations for comparable quality. However, both these methods significantly modify the original formulation of score-based models, such that exact likelihood evaluation and controllable generation become considerably more difficult.[1]

Unlike these previous works, subspace diffusion simultaneously improves sample quality and inference runtime while also preserving all the capabilities of the original formulation. Compared with LSGM and CLD-SGM, subspace diffusion also has the advantage of being compatible with existing trained score models, incurring only the overhead required to train the smaller subspace score models.

[1] In CLD-SGM, one must marginalize over the momentum variables; and in LSGM one must marginalize over the latent variable of VAE.

**Cascading generative models** Subspace diffusion bears some similarity to cascading generative models consisting of one low-dimensional model followed by one or more super-resolution models [12, 14]. Cascading score-based models have yielded strong results on high-resolution class-conditional ImageNet generation [3, 15, 7]. These models formulate each super-resolution step as a full diffusion process conditioned on the lower-resolution image. Subspace diffusion, on the other hand, models a single diffusion process punctuated by projection steps. This leads to a more general theoretical framework that is useful even in domains where the concept of super-resolution does not apply (see for example the synthetic experiments in the supplementary material). Chaining conditional diffusion processes also complicates the application of other capabilities of score-based models—for example, evaluating log-likelihoods would require marginalizing over the intermediate lower-resolution images. Our subspace diffusion framework is a modification of a single diffusion and does not incur these difficulties.

## 3 Subspace Diffusion

A concrete formulation of a score-based model requires a choice of forward diffusion process, specified by $\mathbf{f}(\mathbf{x}, t)$, $\mathbf{G}(\mathbf{x}, t)$. Almost always, these are chosen to be *isotropic*, i.e., of the form

$$\mathbf{f}(\mathbf{x}, t) = f(t)\,\mathbf{x} \quad \mathbf{G}(\mathbf{x}, t) = g(t)\,\mathbf{I}_d \tag{4}$$

where $d$ is the data dimensionality. For example, the variance exploding (VE) SDE has $f(t) = 0$ and $g(t) = \sqrt{d\sigma^2/dt}$ where $\sigma^2(t)$ is the variance of the perturbation kernel at time $t$ [20]. The sole exception is the Langevin diffusion in CLD-SGM [4], but this required new forms of score-matching and specialised SDE solvers for numerical stability. We aim to keep the simplicity and convenience of form (4) while addressing its limitations discussed in Section 1. We thus propose that at every point in time, the diffusion is restricted *to some subspace*, but is otherwise isotropic *in that subspace*. Specifically, the forward diffusion begins in the full space, but is projected and restricted to increasingly smaller subspaces as time goes on. Any isotropic forward diffusion can therefore be converted into a subspace diffusion.

For any diffusion with the form (4), define the corresponding subspace diffusion as follows. Divide $(0, T)$ into $K + 1$ subintervals, $(t_0, t_1), \ldots, (t_K, t_{K+1})$ where for notational convenience $t_0 = 0, t_{K+1} = T$. Then define:

$$\mathbf{G}(\mathbf{x}, t) = g(t)\mathbf{U}_k\mathbf{U}_k^T \tag{5}$$

for each interval $t_k < t < t_{k+1}$, where $\mathbf{U}_k \in \mathbb{R}^{d \times n_k}$ is the matrix whose $n_k \leq d$ orthonormal columns span a subspace of $\mathbb{R}^d$. We refer to this subspace as the $k$th subspace and to the columns of $\mathbf{U}_k$ as its basis. For notational convenience, $\mathbf{U}_0 = \mathbf{I}_d$. We choose $n_k$ such that $d = n_0 > n_1 > \ldots > n_K$. We also require the $k$th subspace to be a subspace of the $j$th subspace for any $j < k$, which can be written as $\mathbf{U}_j\mathbf{U}_j^T\mathbf{U}_k = \mathbf{U}_k$. Together, these definitions state that diffusion is coupled or constrained to occur in progressively smaller subspaces defined by $\mathbf{U}_k$ in the interval $(t_k, t_{k+1})$.

Turning to $\mathbf{f}(\mathbf{x}, t)$, define

$$\mathbf{f}(\mathbf{x}, t) = f(t)\,\mathbf{x} + \sum_{k=1}^{K} \delta(t - t_k)(\mathbf{U}_k \mathbf{U}_k^T - \mathbf{I}_d)\,\mathbf{x} \tag{6}$$

where $\delta$ is the Dirac delta. This states that at time $t_k$, $\mathbf{x}$ is projected onto the $k$th subspace. Figure 1 illustrates the high-level idea of subspace diffusion, along with some of its properties discussed in more detail below.

**Notation** For the rest of the exposition, we define:

- $\mathbf{U}_{k|j} = \mathbf{U}_j^T \mathbf{U}_k \in \mathbb{R}^{n_j \times n_k}$ for $j \leq k$ defines the $k$th subspace written in the basis of the $j$th subspace. In particular, $\mathbf{U}_{k|0} = \mathbf{U}_k$ and $\mathbf{U}_{k|k} = \mathbf{I}_{n_k}$.
- $\mathbf{P}_{k|j} = \mathbf{U}_{k|j} \mathbf{U}_{k|j}^T \in \mathbb{R}^{n_j \times n_j}$ for $j \leq k$ is the projection operator onto the $k$th subspace, written in the basis of the $j$th subspace.
- $\mathbf{P}_{k|j}^{\perp} = \mathbf{I}_{n_j} - \mathbf{P}_{k|j} \in \mathbb{R}^{n_j \times n_j}$ for $j < k$ is the projection operator onto the complement of the $k$th subspace, written in the basis of the $j$th subspace.
- $\mathbf{x}_k = \mathbf{U}_k^T \mathbf{x} \in \mathbb{R}^{n_k}$ is the component of $\mathbf{x}$ in the $k$th subspace, written in that basis. In particular, $\mathbf{x}_0 = \mathbf{x}$.
- $\mathbf{x}_{k|j}^{\perp} = \mathbf{P}_{k|j}^{\perp} \mathbf{x}_j \in \mathbb{R}^{n_j}$ for $j < k$ is the component of $\mathbf{x}_j$ orthogonal to the $k$th subspace, written in the basis of the $j$th subspace.

### 3.1 Score matching

To generate data, we need to learn the score $\nabla_{\mathbf{x}} \log p(\mathbf{x}, t)$ as usual. However, for times $t_k < t < t_{k+1}$, the support of $p(\mathbf{x}, t)$ is only in the $k$th subspace. This means that if we learn a separate score model $\mathbf{s}_k(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p(\mathbf{x}, t)$ for each interval $t \in (t_k, t_{k+1})$, then the model $\mathbf{s}_k$ *only needs to have dimensionality* $n_k$. In particular, we use models smaller than $n_0 = d$ for all times $t > t_1$.

To learn these lower-dimensional models, we leverage the fact that the subspace components $\mathbf{x}_k$ of the data diffuse under an SDE with the same $f(t), g(t)$ as the full data, independent of the orthogonal components. This is due to the fact that the original diffusion is isotropic. To see this, consider (for simplicity) the case $K = 1$, i.e., we only use one proper subspace. Then since $d\mathbf{x}_1 = \mathbf{U}_1^T d\mathbf{x}$,

$$\begin{aligned} d\mathbf{x}_1 = f(t)\mathbf{U}_1^T \mathbf{x}\; dt + \delta(t - t_1)\mathbf{U}_1^T(\mathbf{U}_1\mathbf{U}_1^T - \mathbf{I}_d)\mathbf{x}\; dt \\ + g(t)\left(\mathbf{U}_1^T \left(\mathbb{1}_{t<t_1}\mathbf{I}_d + \mathbb{1}_{t>t_1}\mathbf{U}_1\mathbf{U}_1^T\right)\right)\; d\mathbf{w} \end{aligned} \tag{7}$$

However, because $\mathbf{U}_1^T \mathbf{U}_1 = \mathbf{I}_d$, the above simplifies as

$$d\mathbf{x}_1 = f(t)\mathbf{x}_1\; dt + g(t)\; d\mathbf{w}_1 \tag{8}$$

where, because the columns of $\mathbf{U}_1$ are orthonormal, $d\mathbf{w}_1 := \mathbf{U}_1^T\; d\mathbf{w}$ is a Brownian diffusion in $\mathbb{R}^{n_1}$. As a result, the perturbation kernels $p(\mathbf{x}_1(t) \mid \mathbf{x}_1(0))$ in the subspace have the same form as in the full space. This allows us to train a model to match the scores $\nabla_{\mathbf{x}_1} \log p(\mathbf{x}_1, t)$ via precisely the same procedure as in [20], except we treat $\mathbf{x}_1(0)$ as the original undiffused data. These scores are related to the full-dimensional scores $\nabla_{\mathbf{x}} \log p(\mathbf{x}, t)$ via $\mathbf{U}_1$, but since $\mathbf{x} = \mathbf{U}_1\mathbf{x}_1$ for times $t > t_1$, we can directly work with data points $\mathbf{x}_1$ and score models $\nabla_{\mathbf{x}_1} \log p(\mathbf{x}_1, t)$ with no loss of information for times $t > t_1$. Thus, in the general case, we train $K + 1$ different score models $\mathbf{s}_k(\mathbf{x}_k, t) \approx \nabla_{\mathbf{x}_k} \log p(\mathbf{x}_k, t)$, where we consider $\mathbf{x}_k$ to have diffused under the original $f(t), g(t)$ for the full time scale $(0, T)$.

### 3.2 Sampling

To generate a sample, we use each score model $\mathbf{s}_k(\mathbf{x}_k, t)$ in the corresponding interval $(t_k, t_{k+1})$ to solve the reverse diffusion of $\mathbf{x}_k$. However, we cannot use the score to reverse the projection steps at the boundaries times $t_k$. Thus, to impute $\mathbf{x}_{k-1}(t_k)$ from $\mathbf{x}_k(t_k)$, we sample $\mathbf{x}^{\perp}_{k|k-1}(t_k)$ by injecting isotropic Gaussian noise orthogonal to the $k$th subspace. The variance $\Sigma^{\perp}_{k|k-1}$ of the injected noise is chosen to match the marginal variance of $\mathbf{x}^{\perp}_{k|k-1}$ at time $t_k$, which is the sum of the original variance of $\mathbf{x}^{\perp}_{k|k-1}$ in the data and the variance of the perturbation kernel:

$$\Sigma^{\perp}_{k|k-1}(t_k) := \frac{\alpha^2(t_k)}{n_{k-1} - n_k} \mathbb{E}\left[ \|\mathbf{x}^{\perp}_{k|k-1}(0)\|_2^2 \right] + \sigma^2(t_k) \tag{9}$$

where $\alpha(t)$ and $\sigma^2(t)$ are the scale and variance of the perturbation kernels.

Sampling $\mathbf{x}^{\perp}_{k|k-1}$ in this manner assumes that (at time $t_k$) it is independent of $\mathbf{x}_k$ and roughly an isotropic Gaussian. The final sample quality will depend on the validity of this assumption. Intuitively, however, we specifically choose subspaces and times such that the original magnitude of $\mathbf{x}^{\perp}_{k|k-1}$ (the first term in (9)) is very small compared to the diffusion noise (the second term), which is indeed isotropic and independent of the data. We also find that a few conditional Langevin dynamics steps with $s_k(\mathbf{x}_k, t_{k+1})$ to correct for the approximations of noise injection help sampling quality. The complete sampling procedure is outlined in Algorithm 1.

So far we have presented subspace diffusion as an explicit modification to the forward diffusion involving projection and confined diffusion, which best matches how we implement unconditional sample generation. However, an alternate view is more suitable for controlled generation, where a full-dimensional score model is required; or in ODE-based likelihood evaluation or probability flow sampling, where the adaptive, non-monotonic evaluations make working with discrete projection steps inconvenient. In these settings, we regard subspace diffusion at time $t \in (t_k, t_{k+1})$ as *explicitly* modeling the score component in $k$th subspace with $\mathbf{s}_k(\mathbf{x}_k, t)$, and *implicitly* modeling all orthogonal components with Gaussians. Specifically, for $t \in (t_k, t_{k+1})$ we decompose $\mathbf{x}$ as

$$\mathbf{x} = \sum_{j=0}^{k-1} \mathbf{U}_j \mathbf{x}^{\perp}_{j+1|j} + \mathbf{U}_k \mathbf{x}_k \tag{10}$$

**Algorithm 1:** Unconditional sampling with subspace diffusion

```
Input: subspaces $\mathbf{U}_k$, projection times $t_k$, score models $\mathbf{s}_k(\mathbf{x}_k, t)$, $k = 0 \dots K$
Output: approximate sample $\mathbf{x}_0$ from $p(\mathbf{x}_0, 0) = p_{\text{data}}(\mathbf{x})$
$\mathbf{x}_K \leftarrow$ sample from prior $p(\mathbf{x}_K, T) \in \mathbb{R}^{n_K}$;
for $k \leftarrow K$ to $0$ do
    $\mathbf{x}_k \leftarrow$ solve reverse SDE with $\mathbf{s}_k(\mathbf{x}_k, t)$ from $t_{k+1}$ to $t_k$ starting from $\mathbf{x}_k$;
    if $k > 0$ then
        $\mathbf{x}^{\perp}_{k|k-1} \leftarrow$ sample from $\mathcal{N}(\mathbf{0}, \varSigma^{\perp}_{k|k-1}(t_k)\ \mathbf{I}) \in \mathbb{R}^{n_{k-1}}$ ;
        $\mathbf{x}^{\perp}_{k|k-1} \leftarrow \mathbf{P}^{\perp}_{k|k-1}\mathbf{x}^{\perp}_{k|k-1}$ ;
        $\mathbf{x}_{k-1} \leftarrow \mathbf{U}_{k|k-1}\mathbf{x}_k + \mathbf{x}^{\perp}_{k|k-1}$ ;
        for $i \leftarrow 1$ to $n$ do                          // n is a hyperpameter
            $\mathbf{x}_{k-1} \leftarrow$ LangevinStep$(\mathbf{x}_{k-1}, t_k)$ ;
```

where the sum corresponds to the components that are ”Gaussianized” out by each projection step. We thus model each $\mathbf{x}^{\perp}_{j+1|j}$ implicitly as isotropic Gaussian with variance $\varSigma^{\perp}_{j+1|j}$, and model $\mathbf{x}_k$ explicitly with score model $\mathbf{s}_k(\mathbf{x}_k, t)$, giving the full score:

$$\nabla_{\mathbf{x}} \log p_t(\mathbf{x}, t) \approx \mathbf{U}_k \mathbf{s}_k(\mathbf{U}_k^T \mathbf{x}, t) - \sum_{j=0}^{k-1} \left(\mathbf{P}_{j|0} - \mathbf{P}_{j+1|0}\right) \frac{\mathbf{x}}{\varSigma^{\perp}_{j+1|j}(t)} \tag{11}$$

where, for clarity, we write all components in terms of $\mathbf{x}$ in the original basis.

### 3.3 Image subspaces

We now restrict our attention to generative modeling of natural images.[2] Motivated by the observation that adjacent pixels tend to be *similar* in color, we choose subspaces that correspond to images where adjacent groups of pixels are *equal* in color—i.e., downsampled versions of the image. Henceforth, we refer to such *downsampling subspaces* in terms of their resolution (e.g., the 16 × 16 subspace), refer to projection onto subspaces at times $t_k$ as *downsampling*, and to the reverse action as *upsampling*.[3]

To more precisely formulate these subspaces, suppose we have a full-resolution image $\mathbf{X} \in \mathbb{R}^{(n \times n \times 3)}$. In particular, we will work with $n$ that are integer powers of 2. Then we define a downsampling operator $\mathcal{D} : \mathbb{R}^{(n \times n \times 3)} \to \mathbb{R}^{(n/2 \times n/2 \times 3)}$ such that if $\mathbf{X}_{k+1} = \mathcal{D}\mathbf{X}_k$, then

$$\mathbf{X}_{k+1}[a, b, c] = \frac{1}{2} \sum_{(i,j) \in \{0,1\}^2} \mathbf{X}_k[2a + i, 2b + j, c] \tag{12}$$

[2] See the supplementary material for experiments on more generic synthetic data.

[3] It is via this choice of subspace that subspace diffusion superficially resembles the cascading models discussed in Section 2.

which states that $\mathbf{X}_{k+1}^{(t)}$ is simply $\mathbf{X}_k^{(t)}$ after mean-pooling $2 \times 2$ patches, multiplied by 2. We can use $\mathcal{D}$ to implicitly define $\mathbf{U}_k$:

$$\mathbf{U}_k^T \mathbf{x} = \mathcal{D}^k \mathbf{x} \quad \text{or} \quad \mathbf{U}_k^T \mathbf{x} = \mathcal{D} \mathbf{U}_{k-1}^T \mathbf{x} \tag{13}$$

where here we consider $\mathbf{x}$ to be the column vector representation of the array $\mathbf{X}$. The choice of $\mathcal{D}$ corresponds to orthonormal $\mathbf{U}_k$, as each column of $\mathbf{U}_k$ has $2^{2k}$ nonzero entries, each with magnitude $1/2^k$. Thus, all of the general results from the preceding section apply. In particular, we can consider the same forward diffusion process defined by $f(t), g(t)$ to be occurring for each downsampled image $\mathcal{D}^k \mathbf{x}$, such that the subspace score models $\mathbf{s}_k(\mathbf{x}_k, t)$ correspond to the same score model trained over a lower-resolution version of the same dataset.

It is natural to consider whether there may exist more optimal subspaces for natural images. In Table 1 we compare the downsampling subspaces to the optimal subspaces of equivalent dimensionality[4] found by principle components analysis (PCA) in terms of root mean square distance (RMSD) of the data from the subspace. Generally, the downsampling subspaces can be seen to be suboptimal. However, if we were to use the optimal PCA subspaces, the coordinates would not take the form of an image-structured latent with translation equivariance, and thus would be incompatible with the convolutional neural networks in the score model. Therefore, a more appropriate comparison is with the subspace found via PCA of the distribution of all *patches* of pixels of the appropriate size, which we call Patch-PCA (see supplementary information for details). These subspaces offer only minor improvements over the downsampling subspaces, so we did not explore them further.

It is also possible that for any given dimensionality $n < d$, the $n$-dimensional substructure that best approximates the data distribution is a nonlinear manifold rather than a subspace. However, leveraging such manifolds to reduce the dimensionality of diffusion would require substantial modifications to the present framework. While potentially promising, we leave such extensions to nonlinear manifolds to future work.

### 3.4 Orthogonal Fisher divergence

We now propose a principled manner to choose among the candidate subspaces for a given image dataset, as well as the downsampling times $t_k$.

For any fixed choice of proper subspaces $\mathbf{U}_1 \dots \mathbf{U}_k$, the optimal values of each $t_k$ must balance two factors: smaller $t_k$ reduces the number of reverse diffusion steps occurring at higher dimensionality $n_{k-1}$, whereas larger $t_k$ makes the Gaussian approximation of the orthogonal components $\mathbf{x}_{k|k-1}^{\perp}$ more accurate when we sample at time $t_k$. This suggests that we should choose the minimum times that keep the error of the Gaussian approximation below some tolerance threshold. However, we cannot quantify the true error as we do not have access to the underlying distribution of $\mathbf{x}_{k|k-1}^{\perp}$. Thus, we instead examine how much

[4] That is, an $N \times N$ subspace has dimensionality $3N^2$.

| | | RMSD per dim. | | |
|---|---|---|---|---|
| Dataset | Subspace dim. | PCA | Patch-PCA | Downsampling |
| CIFAR-10<br>($32 \times 32$) | $16 \times 16$ | 0.024 | 0.064 | 0.075 |
| | $8 \times 8$ | 0.061 | 0.093 | 0.110 |
| CelebA-HQ<br>($256 \times 256$) | $128 \times 128$ | — | 0.034 | 0.034 |
| | $32 \times 32$ | 0.041 | 0.063 | 0.073 |
| | $8 \times 8$ | 0.083 | 0.117 | 0.141 |
| LSUN Church<br>($256 \times 256$) | $128 \times 128$ | — | 0.058 | 0.070 |
| | $32 \times 32$ | 0.082 | 0.099 | 0.109 |
| | $8 \times 8$ | 0.126 | 0.146 | 0.158 |

Table 1: Comparison of downsampling subspaces with optimal subspaces of equivalent dimensionality found by PCA and Patch-PCA. RMSD per dim refers to RMSD$/\sqrt{d-n}$, where $d, n$ are the original and subspace dimensionalities. PCA and Patch-PCA were run on a subset of CelebA and LSUN.

the *learned* full-dimensional score model $\mathbf{s}_0(\mathbf{x}, t)$ diverges from the Gaussian approximation on $\mathbf{x}_{k|k-1}^{\perp}$ as $t$ is varied. To quantify this divergence, for any $j < k$ we introduce the *orthogonal Fisher divergence* of $\mathbf{U}_{k|j}$ as:

$$D_F(\mathbf{U}_{k|j}; t) = \frac{\Sigma_{k|j}^{\perp}(t)}{n_j - n_k} \mathbb{E}_{\mathbf{x}(t)} \left[ \left\| \mathbf{P}_{k|j}^{\perp} \mathbf{U}_j^T \mathbf{s}_0(\mathbf{x}, t) + \frac{\mathbf{x}_{k|j}^{\perp}}{\Sigma_{k|j}^{\perp}(t)} \right\|^2 \right] \tag{14}$$

The first term is the component of the score orthogonal to $\mathbf{U}_{k|j}$, and the second term is the score of the Gaussian approximation of $\mathbf{x}_{k|j}^{\perp}$. The divergence is normalized by the (approximate) expected norm of the Gaussian score, which enables values for different $t, j, k$ to be compared. The expectation over $\mathbf{x}(t)$ can then be approximated using the training data. The divergence $D(\mathbf{U}_{k|k-1}; t)$ then corresponds to the error that would be introduced by the upsampling step at time $t$.

Given a sequence of subspaces, the divergence threshold becomes the sole hyperparameter of the sampling process, as we can compute (14) to determine the upsampling times $t_k$ for any threshold. Once the $t_k$ are known, we can estimate the runtime improvement over the full-dimensional score model. Thus, we can choose the subspaces sequence to minimize the estimated runtime. Additionally, it is more convenient to consider $D_F(\mathbf{U}_{k|0}; t)$ as opposed to $D(\mathbf{U}_{k|k-1}; t)$, which corresponds to assuming that at time $t_k$, $\mathbf{x}_{k|k-1}^{\perp}$ is sampled with variance $\Sigma_{k|0}^{\perp}$ rather than $\Sigma_{k|k-1}^{\perp}$.[5] The benefit of this approximation is that we can speak of the divergence purely as a property of the subspace, independent of the preceding subspace (if any). Thus, we can simultaneously plot the orthogonal

[5] The difference is minimal as the variance of the perturbation kernel dominates either term for reasonable divergence thresholds.

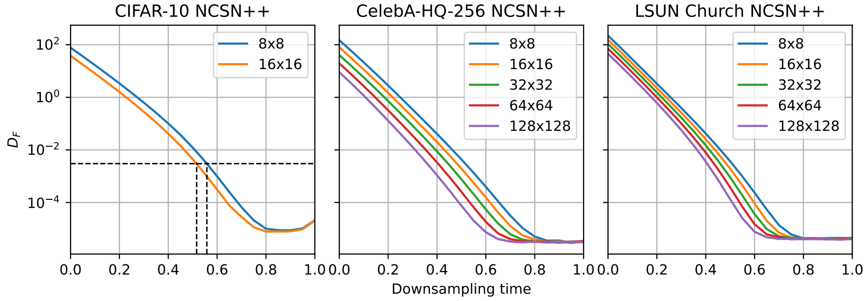

Fig. 2: Orthogonal Fisher divergence plots computed with respect to the pretrained NCSN++ full-dimensional score models from [20]. Similar plots can be generated for other models. Given a divergence threshold, the optimal downsampling times $t_k$ for any subspace sequence are the times at which the corresponding divergences attain that threshold. For example, on CIFAR-10 with a target $D_F = 3 \times 10^{-3}$ and the sequence $32 \to 16 \to 8$, the downsampling times are $t_1 = 0.516, t_2 = 0.558$. In this case, the intermediate $16 \times 16$ subspace would be used for only 4.2% of the diffusion. As the plot shows, this imbalance would characterise any sequence of more than one proper subspace.

Fisher divergence for each downsampling subspace, as illustrated in Figure 2. The choice of intervals for any subspace sequence and divergence threshold can then be directly read off the plot.

As Figure 2 shows, for standard image datasets there appears to be little utility to using more than one proper subspace, as the diffusion in intermediate dimensions would be very brief. On the other hand, training additional models is computationally expensive and adds to the sum of the model sizes required for inference. Thus, our experiments focus on subspace diffusions consisting of only one proper downsampling subspace. In particular, for CIFAR-10, we consider the $8 \times 8$ and $16 \times 16$ subspaces separately, while for CelebA-HQ and LSUN Church we consider only the $64 \times 64$ subspace, which offers the best potential runtime improvement.

## 4 Experiments

We demonstrate the utility and versatility of our method by improving upon and accelerating state-of-the-art continuous score-based models. Specifically, we take the pretrained models on CIFAR-10, CelebA-256-HQ, and LSUN Church from [20] as full-dimensional score models, train additional subspace score models of the same architecture, and use them together in the subspace diffusion framework. All lower-dimensional models are trained with the same hyperparameters and training procedure as the original model. During inference, we use the unmodified reverse SDE solvers and the same number and spacing of denoising steps. We investigate results for a range of divergence thresholds, corresponding

| Model | FID ↓ |
|---|---|
| DDIM [18] | 4.04 |
| FastDPM [10] | 2.86 |
| Bilateral DPM [11] | 2.38 |
| Analytic DPM [2] | 3.04 |
| Prog. Distillation [16] | 2.57 |
| CLD-SGM [4] | 2.23 |
| LSGM [21] | 2.10 |
| Adaptive solver [8] | 2.44 |

| Model | | FID ↓ | IS ↑ | Thresh. | $t_1$ | Run. |
|---|---|---|---|---|---|---|
| NCSN$^{++}$ | full | 2.38 | 9.83 | | | |
| | subspace | 2.29 | **9.99** | 3e-3 | 0.52 | 0.66 |
| NSCN$^{++}$ (deep) | full | 2.20 | 9.89 | | | |
| | subspace | **2.17** | 9.94 | 1e-3 | 0.56 | 0.69 |
| DDPM$^{++}$ | full | 2.61 | 9.56 | | | |
| | subspace | 2.60 | 9.54 | 3e-5 | 0.62 | 0.73 |
| DDPM$^{++}$ (deep) | full | 2.41 | 9.57 | | | |
| | subspace | 2.40 | 9.66 | 1e-4 | 0.56 | 0.69 |

Table 2: CIFAR-10 sample quality for 50k images. *Left*: the best performance of previous methods to accelerate score-based models. *Right*: the original full diffusion from [20] and the respective best subspace diffusion (all 16 × 16), with the corresponding divergence threshold, downsampling time $t_1$, and empirical runtime relative to the full model.

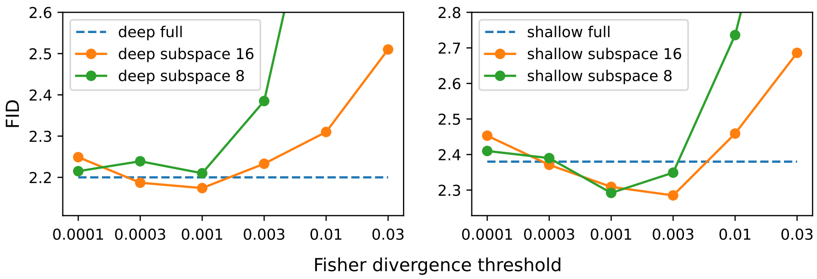

Fig. 3: CIFAR-10 sample quality from NCSN++ subspace diffusion (shallow and deep models) with different subspaces and divergence thresholds.

to different durations of diffusion in the subspace. For all experiments, further results and additional samples may be found in the supplementary material.

**Unconditional sampling** We evaluate subspace diffusion on unconditional CIFAR-10 generation with the Inception score (IS) and Frechet Inception distance (FID) as metrics. We examine both the NCSN++ and DDPM++ models from [20], which correspond to different forward diffusion processes, as well as the deep versions of these models, for a total of 4 full-dimensional models. For each model, we separately construct subspace diffusion with 8 × 8 and 16 × 16 subspaces. As in [20], we choose the best checkpoint by FID.

In Figure 3, we show the performance of the NCSN++ subspace diffusion models for different choices of the Fisher divergence threshold $D_F$. In all cases, the models display U-shaped performance curves as the threshold is varied. When the threshold is small, most of the diffusion is done at full dimensionality, and the performance is close to that of the full model alone. As the threshold increases and more diffusion is done in the subspace, the models *improve* over the full model until reaching the best performances between $D_F = 1 \times 10^{-3}$ and $D_F = 3 \times 10^{-3}$. This improvement offers support for the hypothesis, discussed in the introduction, that restricting the dimensionality (or support) of the score to be matched can help the subspace model learn and extrapolate more accurately than the full-dimensional model. Finally, for large thresholds the performance deteriorates as the Gaussian approximation of the orthogonal component becomes too inaccurate.

Table 2 compares the performance of the best subspace diffusion models with the original full-dimensional models from [20] and with prior methods for accelerating score-based models. Subspace diffusion and LSGM [21] are the only methods where the improved runtime does not come at the cost of decreased performance (relative to [20]). The runtime improvement over the full-dimensional baseline varies with the choice of divergence threshold; for those leading to the best sample qualities, the improvements are typically around 30%. Since the concept of subspace diffusion is orthogonal to the techniques used by most previous work (see Section 2), it can potentially be used in combination with them for further runtime improvement.

Next, we show the applicability of our method to larger resolution datasets by generating samples on CelebA-HQ-256 with NCSN++ subspace diffusion. As discussed in Section 3.4, we use only the 64 × 64 subspaces and perform no hyperparameter tuning or checkpoint selection. In Figure 4, we show random samples from CelebA-HQ for different amounts of diffusion in the subspace, along with the corresponding Fisher divergence. Qualitatively, we can restrict up to 60-70% of the diffusion to the subspace without significant loss of quality.

**ODE sampling and likelihood** Subspace diffusion retains the flexible capabilities of the continuous-time SDE framework. In particular, the corresponding probability flow ODE (3) can be used to evaluate exact log-likelihoods and generate samples, as described in [20]. In Table 3, we show results for these tasks on CIFAR-10 for subspace diffusion in combination with the DDPM++ (deep) model. We use the alternate subspace score formulation (11) with the original ODE solvers, and use the last checkpoint of each training run. Subspace diffusion has little to no impact on the log-likelihoods obtained and slightly hurts sample quality.

| Subspace | Thresh. | NLL ↓ | FID ↓ |
|---|---|---|---|
| None | — | 2.995 | 2.95 |
| 8 × 8 | $1 \times 10^{-4}$ | 2.998 | 3.02 |
| | $3 \times 10^{-4}$ | 2.999 | 3.12 |
| | $1 \times 10^{-3}$ | 2.998 | 3.53 |
| 16 × 16 | $1 \times 10^{-4}$ | 2.997 | 2.95 |
| | $3 \times 10^{-4}$ | 2.997 | 3.00 |
| | $1 \times 10^{-3}$ | 2.997 | 3.17 |

Table 3: ODE sampling and NLL evaluation on CIFAR-10 from DDPM++ (deep) with subspace diffusion.

**Inpainting** Subspace diffusion can also be used for controllable generation tasks, an example of which is image inpainting. Indeed, by using the alternate formulation (11), the subspace model appears as a full-dimensional model and integrates seamlessly with the existing inpainting procedures described in [20].

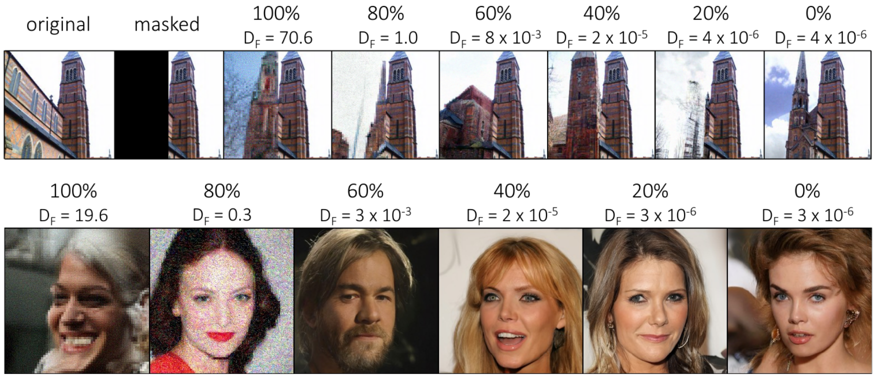

Fig. 4: Random high resolution samples with $64 \times 64$ subspace diffusion. *Top:* Inpainting on the 256x256 LSUN Church dataset. *Bottom:* Unconditional generation of samples for CelebA-HQ-256. From right to left, the fraction of the diffusion in the subspace increases in intervals of 20%, with the corresponding orthogonal Fisher divergence shown. As expected from the divergence analysis in Figure 2, there is little deterioration in quality for images generated with up to 60% of the diffusion in the subspace.

In Figure 4, we show inpainting results on LSUN Church with $64 \times 64$ subspace diffusion in conjunction with the pretrained NCSN++ model. As with the unconditional samples, quality does not significantly deteriorate with up to 60% of the diffusion occurring in the subspace.

## 5 Conclusion

We presented a novel method for more efficient generative modeling with score-based models. *Subspace diffusion models* restrict part of the diffusion to lower-dimensional subspaces such that the score of the projected distribution is faster to compute and easier to learn. Empirically on image datasets, our method provides inference speed-ups while preserving or improving the performance and capabilities of state-of-the-art models. Potential avenues of future work include applying subspace diffusion to other data domains and combining it with step-size based methods for accelerating inference. More generally, we hope that our work opens up further research on dimensionality reduction in diffusion processes, particularly to nonlinear manifolds and/or learned substructures, as a means of both simplifying and improving score-based generative models.

**Acknowledgements** We thank Yilun Du, Xiang Fu, Jason Yim, Shangyuan Tong, Yilun Xu, Felix Faltings, and Saro Passaro for helpful feedback and discussions. Bowen Jing acknowledges support from the Department of Energy Computational Science Graduate Fellowship.

## References

1. Anderson, B.D.: Reverse-time diffusion equation models. Stochastic Processes and their Applications (1982)
2. Bao, F., Li, C., Zhu, J., Zhang, B.: Analytic-dpm: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. ArXiv preprint (2022)
3. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. In: Advances in Neural Information Processing Systems (2021)
4. Dockhorn, T., Vahdat, A., Kreis, K.: Score-based generative modeling with critically-damped langevin diffusion. In: International Conference on Learning Representations (2022)
5. Du, Y., Mordatch, I.: Implicit generation and generalization in energy-based models. In: Advances in Neural Information Processing Systems (2019)
6. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (2020)
7. Ho, J., Saharia, C., Chan, W., Fleet, D.J., Norouzi, M., Salimans, T.: Cascaded diffusion models for high fidelity image generation. ArXiv preprint (2021)
8. Jolicoeur-Martineau, A., Li, K., Piché-Taillefer, R., Kachman, T., Mitliagkas, I.: Gotta go fast when generating data with score-based models. ArXiv preprint (2021)
9. Kingma, D.P., Salimans, T., Poole, B., Ho, J.: Variational diffusion models. In: Advances in Neural Information Processing Systems (2021)
10. Kong, Z., Ping, W.: On fast sampling of diffusion probabilistic models. In: ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models (2021)
11. Lam, M.W., Wang, J., Su, D., Yu, D.: Bddm: Bilateral denoising diffusion models for fast and high-quality speech synthesis. In: International Conference on Learning Representations (2021)
12. Menick, J., Kalchbrenner, N.: Generating high fidelity images with subscale pixel networks and multidimensional upscaling. In: International Conference on Learning Representations (2019)
13. Nichol, A., Dhariwal, P.: Improved denoising diffusion probabilistic models. In: International Conference on Machine Learning (2021)
14. Razavi, A., van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. In: Advances in Neural Information Processing Systems (2019)
15. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. ArXiv preprint (2021)
16. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: International Conference on Learning Representations (2022)
17. San-Roman, R., Nachmani, E., Wolf, L.: Noise estimation for generative diffusion models. ArXiv preprint (2021)
18. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (2021)
19. Song, Y., Ermon, S.: Generative modeling by estimating gradients of the data distribution. In: Advances in Neural Information Processing Systems (2019)
20. Song, Y., Sohl-Dickstein, J., Kingma, D.P., Kumar, A., Ermon, S., Poole, B.: Score-based generative modeling through stochastic differential equations. In: International Conference on Learning Representations (2021)
21. Vahdat, A., Kreis, K., Kautz, J.: Score-based generative modeling in latent space. In: Advances in Neural Information Processing Systems (2021)

22. Watson, D., Ho, J., Norouzi, M., Chan, W.: Learning to efficiently sample from diffusion probabilistic models. ArXiv preprint (2021)
23. Xu, K., Zhang, M., Li, J., Du, S.S., Kawarabayashi, K.i., Jegelka, S.: How neural networks extrapolate: From feedforward to graph neural networks. In: International Conference on Learning Representations (2021)