

Non-Uniform Diffusion Models

Georgios Batzolis
 DAMTP, University of Cambridge
 gbs511@cam.ac.uk

Carola-Bibiane Schönlieb
 DAMTP, University of Cambridge
 cbs31@cam.ac.uk

Jan Stanczuk
 DAMTP, University of Cambridge
 js2164@cam.ac.uk

Christian Etmann
 Deep Render
 christian.etmann@deeprender.ai

Abstract

Diffusion models have emerged as one of the most promising frameworks for deep generative modeling. In this work, we explore the potential of non-uniform diffusion models. We show that non-uniform diffusion leads to multi-scale diffusion models which have similar structure to this of multi-scale normalizing flows. We experimentally find that in the same or less training time, the multi-scale diffusion model achieves better FID score than the standard uniform diffusion model. More importantly, it generates samples 4.4 times faster in 128×128 resolution. The speed-up is expected to be higher in higher resolutions where more scales are used. Moreover, we show that non-uniform diffusion leads to a novel estimator for the conditional score function which achieves on par performance with the state-of-the-art conditional denoising estimator. Our theoretical and experimental findings are accompanied by an open source library `MSDiff` which can facilitate further research of non-uniform diffusion models.

1. Introduction

The goal of generative modelling is to learn a probability distribution from a finite set of samples. This classical problem in statistics has been studied for many decades, but until recently efficient learning of high-dimensional distributions remained impossible in practice. For images, the strong inductive biases of convolutional neural networks have recently enabled the modelling of such distributions, giving rise to the field of deep generative modelling.

Deep generative modelling became one of the central areas of deep learning with many successful applications. In recent years much progress has been made in unconditional and conditional image generation. The most prominent approaches are auto-regressive models [3], variational auto-

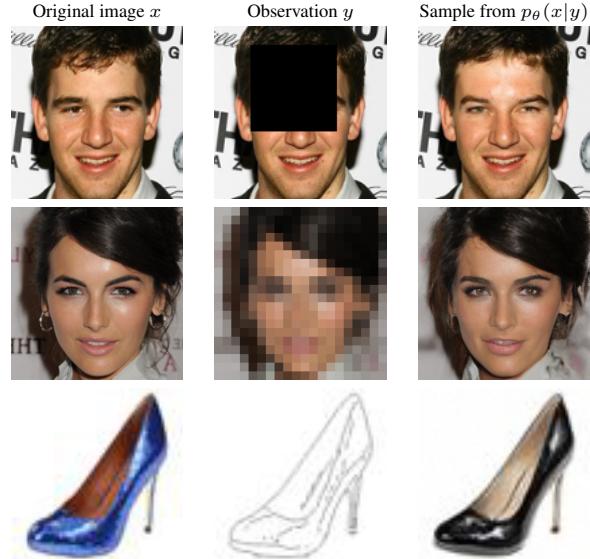


Figure 1. Results from our conditional multi-speed diffusive estimator.

encoders (VAEs) [11], normalizing flows [18] and generative adversarial networks (GANs) [5].

Despite their success, each of the above methods suffers from important limitations. Auto-regressive models allow for likelihood estimation and high-fidelity image generation, but suffer from poor time complexity in high resolutions. VAEs and normalizing flows are less computationally expensive and allow for likelihood estimation, but tend to produce samples of lower visual quality. Moreover, normalizing flows put restrictions on the possible model architectures (requiring invertibility of the network and a Jacobian log-determinant that is computationally tractable), thus limiting their expressivity. While GANs produce state-of-the art quality samples, they don't allow for likelihood

estimation and are notoriously hard to train due to training instabilities and mode collapse.

Recently, score-based [8] and diffusion-based [20] generative models have been revived and improved in [22] and [7]. The connection between the two frameworks in discrete-time formulation has been discovered in [25]. Recently in [23], both frameworks have been unified into a single continuous-time approach based on stochastic differential equations [23] and are called score-based diffusion models. These approaches have recently received a lot of attention, achieving state-of-the-art performance in likelihood estimation [23] and unconditional image generation [4], surpassing even the celebrated success of GANs.

In addition to achieving state-of-the art performance in both image generation and likelihood estimation, score-based diffusion models don't suffer from training instabilities or mode collapse [4,23]. However, although their time complexity in high resolutions is better than that of autoregressive models [4], it is still notably worse to that of GANs, normalizing flows and VAEs. Despite the recent efforts to close the sampling time gap between diffusion models and the faster frameworks, diffusion models still require significantly more time to achieve equal performance.

In this work, we explore non-uniform diffusion models. In non-uniform diffusion models, different parts of the input tensor diffuse with different diffusion speeds or more generally according to different stochastic differential equations. We find that the generalization of the original uniform diffusion framework can lead to multi-scale diffusion models which achieve improved sampling performance at a significantly faster sampling speed.

Moreover, we find that non-uniform diffusion can be used for conditional generation, because it leads to a novel estimator of the conditional score. We conduct a review and classification of existing approaches and perform a systematic comparison to find the best way of estimating the conditional score. We provide a proof of validity for the *conditional denoising estimator* (which has been used in [19,24] without justification), and we thereby provide a firm theoretical foundation for using it in future research.

The contributions of this paper are as follows:

1. We introduce a principled objective for training non-uniform diffusion models.
2. We show that non-uniform diffusion leads to the multi-scale diffusion models which are more efficient than uniform diffusion models. In less training time, the multi-scale models reach improved FID scores with significantly faster sampling speed. The speed up factor is expected to increase as we increase the number of scales.
3. We show that non-uniform diffusion leads to *conditional multi-speed diffusive estimator* (CMDE), a

novel estimator of conditional score, which unifies previous methods of conditional score estimation.

4. We provide a proof of consistency for the *conditional denoising estimator* - one of the most successful approaches to estimating the conditional score.
5. We review and empirically compare score-based diffusion approaches to modelling conditional distributions of image data. The models are evaluated on the tasks of super-resolution, inpainting and edge to image translation.
6. We provide an open-source library `MSDiff`, to facilitate further research on conditional and non-uniform diffusion models.¹

2. Notation

In this work we will use the following notation:

- **Functions of time**

$$f_t := f(t)$$

- **Indexing vectors**

Let $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ and let $1 \leq i < j < n$. Then:

$$v[:j] := (v_1, v_2, \dots, v_j) \in \mathbb{R}^j,$$

cf. Section 3.4.2.

- **Probability distributions**

We denote the probability distribution of a random variable solely via the name of its density's argument, e.g.

$$p(x_t) := p_{X_t}(x_t),$$

where x_t is a realisation of the random variable X_t .

- **Iterated Expectations**

$$\begin{aligned} & \mathbb{E}_{\substack{z_1 \sim p(z_1) \\ \vdots \\ z_n \sim p(z_n)}} [f(z_1, \dots, z_n)] \\ &:= \mathbb{E}_{z_1 \sim p(z_1)} \cdots \mathbb{E}_{z_n \sim p(z_n)} [f(z_1, \dots, z_n)] \end{aligned}$$

3. Methods

In the following, we will provide details about the framework and estimators discussed in this paper.

¹The code will be released in the near future.

3.1. Background: Score matching through Stochastic Differential Equations

3.1.1 Score-Based Diffusion

In a recent work [23] score-based [8, 22] and diffusion-based [7, 20] generative models have been unified into a single continuous-time score-based framework where the diffusion is driven by a stochastic differential equation. This framework relies on Anderson’s Theorem [1], which states that under certain Lipschitz conditions on $f : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_x}$ and $G : \mathbb{R}^{n_x} \times \mathbb{R} \rightarrow \mathbb{R}^{n_x} \times \mathbb{R}^{n_x}$ and an integrability condition on the target distribution $p(\mathbf{x}_0)$ a forward diffusion process governed by the following SDE:

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + G(\mathbf{x}_t, t)d\mathbf{w}_t \quad (1)$$

has a reverse diffusion process governed by the following SDE:

$$\begin{aligned} d\mathbf{x}_t = & [f(\mathbf{x}_t, t) - G(\mathbf{x}_t, t)G(\mathbf{x}_t, t)^T \nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t)]dt \\ & + G(\mathbf{x}_t, t)d\bar{\mathbf{w}}_t, \end{aligned} \quad (2)$$

where $\bar{\mathbf{w}}_t$ is a standard Wiener process in reverse time.

The forward diffusion process transforms the *target distribution* $p(\mathbf{x}_0)$ to a *diffused distribution* $p(\mathbf{x}_T)$ after diffusion time T . By appropriately selecting the drift and the diffusion coefficients of the forward SDE, we can make sure that after sufficiently long time T , the diffused distribution $p(\mathbf{x}_T)$ approximates a simple distribution, such as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. We refer to this simple distribution as the *prior distribution*, denoted by π . The reverse diffusion process transforms the diffused distribution $p(\mathbf{x}_T)$ to the data distribution $p(\mathbf{x}_0)$ and the prior distribution π to a distribution p^{SDE} . p^{SDE} is close to $p(\mathbf{x}_0)$ if the diffused distribution $p(\mathbf{x}_T)$ is close to the prior distribution π . We get samples from p^{SDE} by sampling from π and simulating the reverse sde from time T to time 0.

To get samples by simulating the reverse SDE, we need access to the time-dependent score function $\nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t)$ for all \mathbf{x}_t and t . In practice, we approximate the time-dependent score function with a neural network $s_\theta(\mathbf{x}_t, t) \approx \nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t)$ and simulate the reverse SDE in equation 3 to map the prior distribution π to p_θ^{SDE} .

$$\begin{aligned} d\mathbf{x}_t = & [f(\mathbf{x}_t, t) - G(\mathbf{x}_t, t)G(\mathbf{x}_t, t)^T s_\theta(\mathbf{x}_t, t)]dt \\ & + G(\mathbf{x}_t, t)d\bar{\mathbf{w}}_t, \end{aligned} \quad (3)$$

If the prior distribution is close to the diffused distribution and the approximated score function is close to the ground truth score function, the modeled distribution p_θ^{SDE} is probably close to the target distribution $p(\mathbf{x}_0)$. This statement is formalised in the language of distributional distances in the next subsection.

3.1.2 Uniform Diffusion Models

Previous works [4, 7, 22] used the same forward SDE for the diffusion of all the pixels. For this reason, we classify them as uniform diffusion models. In uniform diffusion models, the sde in equation 4 describes the forward diffusion for all pixels in an image:

$$dx_t = f(x_t, t)dt + g(t)d\bar{w}_t, \quad (4)$$

We used unbold notation for the random variables to show that this equation describes diffusion in one dimension. For uniform diffusion models, the neural network $s_\theta(\mathbf{x}_t, t)$ can be trained to approximate the score function $\nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t)$ by minimizing the weighted score matching objective

$$\mathcal{L}_{SM}(\theta, \lambda(\cdot)) := \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ \mathbf{x}_t \sim p(\mathbf{x}_t)}} [\lambda(t) \|\nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t)\|_2^2] \quad (5)$$

where $\lambda : [0, T] \rightarrow \mathbb{R}_+$ is a positive weighting function.

However, the above quantity cannot be optimized directly since we don’t have access to the ground truth score $\nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t)$. Therefore in practice, a different objective has to be used [8, 22, 23]. In [23], the weighted denoising score-matching objective is used, which is defined as

$$\begin{aligned} \mathcal{L}_{DSM}(\theta, \lambda(\cdot)) := & \\ & \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ \mathbf{x}_0 \sim p(\mathbf{x}_0) \\ \mathbf{x}_t \sim p(\mathbf{x}_t | \mathbf{x}_0)}} [\lambda(t) \|\nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t | \mathbf{x}_0) - s_\theta(\mathbf{x}_t, t)\|_2^2] \end{aligned} \quad (6)$$

The difference between DSM and SM is the replacement of the ground truth score which we do not know by the score of the perturbation kernel which we know analytically for many choices of forward SDEs. The choice of the weighted DSM objective is justified because the weighted DSM objective is equal to the SM objective up to a constant that does not depend on the parameters of the model θ . The reader can refer to [25] for the proof.

The choice of the weighting function is also important, because it determines the quality of score-matching in different diffusion scales. A principled choice for the weighting function is $\lambda(t) = g(t)^2$, where $g(\cdot)$ is the diffusion coefficient of the forward SDE. This weighting function is called the likelihood weighting function [21], because it ensures that we minimize an upper bound on the Kullback–Leibler divergence from the target distribution to the model distribution by minimizing the weighted DSM objective with this weighting. The previous statement is implied by the combination of inequality 7 which is proven in [21] and the relationship between the DSM and SM objectives.

$$D_{KL}(p(\mathbf{x}_0) \parallel p_\theta^{SDE}) \leq L_{SM}(\theta, g(\cdot)^2) + D_{KL}(p(\mathbf{x}_T) \parallel \pi) \quad (7)$$

Other weighting functions have also yielded very good results with particular choices of forward sdes. However, we do not have theoretical guarantees that alternative weightings would yield good results with arbitrary choices of forward sdes.

3.2. Non-Uniform Diffusion Models

In this section, we describe non-uniform diffusion models. We call them non-uniform to indicate that the forward diffusion of each pixel is potentially governed by a different SDE. Considering a vectorised form $x = \text{vec}(X) = [x^1, x^2, \dots, x^{mnc}]$ of an image $X \in [0, 1]^m \times [0, 1]^n \times [0, 1]^c$, we assume that the diffusion of the i^{th} pixel is governed by the following SDE:

$$dx_t^i = f_i(x_t^i, t)dt + g_i(t)dw_t^i \quad (8)$$

Equation 8 is a special case of the general Itô SDE described in equation 1, but provides more flexibility compared to uniform diffusion where all pixels diffuse independently according to the same SDE. The diffusion of the entire image vector is summarised by the following SDE:

$$d\mathbf{x}_t = f(\mathbf{x}_t, t)dt + G(t)d\mathbf{w}_t, \quad (9)$$

where $f(\mathbf{x}_t) = [f_1(x^1, t), \dots, f_{mnc}(x_t^{mnc}, t)]$ and $G(t) = \text{diag}([g_1(t), \dots, g_{mnc}(t)])$.

In this more general setup, the DSM objective as described in equation 6 must also be generalised. The positive weighting function $\lambda(\cdot)$ is replaced by a positive definite matrix $\Lambda(\cdot)$ which gives the form of the DSM objective for non-uniform diffusion models:

$$\begin{aligned} \mathcal{L}_{DSM}(\theta, \Lambda(\cdot)) := \\ \frac{1}{2} \mathbb{E}_{\substack{\mathbf{x}_t \sim U(0, T) \\ \mathbf{x}_0 \sim p(\mathbf{x}_0) \\ \mathbf{x}_t \sim p(\mathbf{x}_t | \mathbf{x}_0)}} [\mathbf{v}_\theta(\mathbf{x}_0, \mathbf{x}_t, t)^T \Lambda(t) \mathbf{v}_\theta(\mathbf{x}_0, \mathbf{x}_t, t)], \end{aligned} \quad (10)$$

where $\mathbf{v}_\theta(\mathbf{x}_0, \mathbf{x}_t, t) = \nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t | \mathbf{x}_0) - s_\theta(\mathbf{x}_t, t)$

We prove that a principled choice for the positive weighting matrix is $\Lambda_{MLE}(t) = G(t)G(t)^T$. We call it the likelihood weighting matrix for non-uniform diffusion because it ensures minimization of an upper bound to the KL divergence from the target distribution to the model distribution. The previous statement is summarised in Theorem 1 which is proved in section A.3 of the Appendix.

Theorem 1. *Let $p(\mathbf{x}_t)$ denote the distribution implied by the forward SDE at time t and $p_\theta^{SDE}(\mathbf{x}_t)$ denote the distribution implied by the parametrized reverse SDE at time t . Then*

under regularity assumptions of [21, Theorem 1] we have that

$$\begin{aligned} KL(p(\mathbf{x}_0) | p_\theta^{SDE}(\mathbf{x}_0)) &\leq KL(p(\mathbf{x}_T) | \pi(\mathbf{x}_T)) \\ &+ \frac{1}{2} \mathbb{E}_{\substack{\mathbf{x}_t \sim U(0, T) \\ \mathbf{x}_t \sim p(\mathbf{x}_t)}} [\mathbf{v}^T G(t) G(t)^T \mathbf{v}], \end{aligned}$$

where $\mathbf{v} = \nabla_{\mathbf{x}_t} \ln p(\mathbf{x}_t) - s_\theta(\mathbf{x}_t, t)$.

3.3. Application of Non-Uniform Diffusion in multi-scale diffusion

We design the forward process so that different groups of pixels diffuse with different speeds which creates a multi-scale diffusion structure. The intuition stems from multi-scale normalising flows. Multi-scale normalising flows invertibly transform the input tensor to latent encodings of different scales by splitting the input tensor into two parts after transformation in each scale. The multi-scale structure in normalising flows is shown to lead to faster training and sampling without compromise in generated image quality.

We intend to transfer this idea to score-based modeling by diffusing some parts of the tensor faster. There are many ways to split the image into different parts which diffuse faster. It has been experimentally discovered that cascaded diffusion models [19] yield improved results compared to standard diffusion models. This gave us the intuition to use a multi-level haar transform to transform every image to n high frequency scales d_1, \dots, d_n (detail coefficients) and one low frequency scale a_n (approximation coefficient). The natural generation order of the haar coefficients (in line with cascaded diffusion) is $a_n, d_n, d_{n-1}, \dots, d_1$. For this reason, we choose to diffuse lower frequency coefficients slower than high frequency coefficients. More specifically, we design the forward process so that all coefficients reach the same signal-to-noise ratio at the end of their diffusion time. We set the diffusion time for a_n to $T_{a_n} = 1$ and for d_i to $T_{d_i} = \frac{i}{n+1}$ for each $i \in [1, \dots, n]$.

3.3.1 Training

We approximate the score of the distribution of $c_i(t) = [a_n(t), d_n(t), \dots, d_i(t)]$ in the time range $[(i-1)/(n+1), i/(n+1)]$ with a separate neural network $s_i(c_i(t), t)$. We also use a separate network $s_{n+1}(c_{n+1}(t), t)$ to approximate the score of the distribution of $c_{n+1}(t) = a_n(t)$ in the diffusion time range $[n/(n+1), 1]$. We use different networks in each scale to leverage the fact that we approximate the score of lower dimensional distributions. This enables faster score function evaluation and, therefore, faster training and sampling. We train each network separately using the likelihood weighting matrix for non-uniform diffusion (see section 3.2).

3.3.2 Sampling

The sampling process is summarised in the following steps:

1. Sample $a_n(1)$ from the stationary distribution (e.g. standard normal distribution) and integrate the reverse sde for a_n from time $t = 1$ to time $t = n/(n+1)$.
2. Sample $d_n(n/(n+1))$ from the stationary distribution and solve the reverse sde for $[a_n, d_n]$ from time $t = n/(n+1)$ to time $t = (n-1)/(n+1)$.
3. The process is continued as implied until we reach the final generation level, where we sample $d_1(1/(n+1))$ from the stationary distribution and solve the reverse sde for $[a_n, d_n, \dots, d_1]$ from time $t = 1/(n+1)$ to time $t = \epsilon$ (e.g., $\epsilon = 10^{-5}$).
4. We convert the generated haar coefficients $[a_n(\epsilon), d_n(\epsilon), \dots, d_1(\epsilon)]$ to the generated image using the multi-level inverse haar transform.

Our experimental results presented in section 4.1 show that multiscale diffusion is more efficient and effective than uniform diffusion.

3.4. Application of Non-Uniform Diffusion in Conditional generation

The continuous score-matching framework can be extended to conditional generation, as shown in [23]. Suppose we are interested in $p(x|y)$, where x is a *target image* and y is a *condition image*. Again, we use the forward diffusion process (Equation 1) to obtain a family of diffused distributions $p(x_t|y)$ and apply Anderson’s Theorem to derive the *conditional reverse-time SDE*

$$dx = [\mu(x, t) - \sigma(t)^2 \nabla_x \ln p_{X_t}(x|y)]dt + \sigma(t)d\tilde{w}. \quad (11)$$

Now we need to learn the score $\nabla_{x_t} \ln p(x_t|y)$ in order to be able to sample from $p(x|y)$ using reverse-time diffusion.

In this work, we discuss the following approaches to estimating the conditional score $\nabla_{x_t} \ln p(x_t|y)$:

1. Conditional denoising estimators
2. Conditional diffusive estimators
3. Multi-speed conditional diffusive estimators (our method)

We discuss each of them in a separate section.

In [23] an additional approach to conditional score estimation was suggested: This method proposes learning

$\nabla_{x_t} \ln p(x_t)$ with an unconditional score model, and learning $p(y|x_t)$ with an auxiliary model. Then, one can use

$$\nabla_{x_t} \ln p(x_t|y) = \nabla_{x_t} \ln p(x_t) + \nabla_{x_t} \ln p(y|x_t)$$

to obtain $\nabla_{x_t} \ln p(x_t|y)$. Unlike other approaches, this requires training a separate model for $p(y|x_t)$. Appropriate choices of such models for tasks discussed in this paper have not been explored yet. Therefore we exclude this approach from our study.

3.4.1 Conditional denoising estimator (CDE)

The conditional denoising estimator (CDE) is a way of estimating $p(x_t|y)$ using the denoising score matching approach [22, 25]. In order to approximate $p(x_t|y)$, the conditional denoising estimator minimizes

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t|x_0)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t, y, t)\|_2^2] \quad (12)$$

This estimator has been shown to be successful in previous works [19, 24], also confirmed in our experimental findings (cf. Section 4).

Despite the practical success, this estimator has previously been used without a theoretical justification of why training the above objective yields the desired conditional distribution. Since $p(x_t|y)$ does not appear in the training objective, it is not obvious that the minimizer approximates the correct quantity.

By extending the arguments of [25], we provide a formal proof that the minimizer of the above loss does indeed approximate the correct conditional score $p(x_t|y)$. This is expressed in the following theorem.

Theorem 2. *The minimizer (in θ) of*

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t|x_0)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t, y, t)\|_2^2]$$

is the same as the minimizer of

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_t, y \sim p(x_t, y)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t, y, t)\|_2^2]$$

The proof for this statement can be found in Appendix A.1. Using the above theorem, the consistency of the estimator can be established.

Corollary 1. *Let θ^* be a minimizer of a Monte Carlo approximation of (12), then (under technical assumptions, cf. Appendix A.2) the conditional denoising estimator*

$s_{\theta^*}(x, y, t)$ is a consistent estimator of the conditional score $\nabla_{x_t} \ln p(x_t|y)$, i.e.

$$s_{\theta^*}(x, y, t) \xrightarrow{P} \nabla_{x_t} \ln p(x_t|y)$$

as the number of Monte Carlo samples approaches infinity.

This follows from the previous theorem and the uniform law of large numbers. Proof in the Appendix A.2.

3.4.2 Conditional diffusive estimator (CDiffE)

Conditional diffusive estimators (CDiffE) have first been suggested in [23]. The core idea is that instead of learning $p(x_t|y)$ directly, we diffuse both x and y and approximate $p(x_t|y_t)$, using the denoising score matching. Just like learning diffused distribution $\nabla_{x_t} \ln p(x_t)$ improves upon direct estimation of $\nabla_x \ln p(x)$ [22, 23], diffusing both the input x and condition y , and then learning $\nabla_{x_t} \ln p(x_t|y_t)$ could make optimization easier and give better results than learning $\nabla_{x_t} \ln p(x_t|y)$ directly.

In order to learn $p(x_t|y_t)$, observe that

$$\nabla_{x_t} \ln p(x_t|y_t) = \nabla_{x_t} \ln p(x_t, y_t) = \nabla_{z_t} \ln p(z_t)[: n_x],$$

where $z_t := (x_t, y_t)$ and n_x is the dimensionality of x . Therefore we can learn the (unconditional) score of the joint distribution $p(x_t, y_t)$ using the denoising score matching objective just like as in the unconditional case, i.e

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_0 \sim p_0(z_0) \\ z_t \sim p(z_t|z_0)}} [\lambda(t) \|\nabla_{z_t} \ln p(z_t|z_0) - s_\theta(z_t, t)\|_2^2]. \quad (13)$$

We can then extract our approximation for the conditional score $\nabla_{x_t} \ln p(x_t|y_t)$ by simply taking the first n_x components of $s_\theta(x_t, y_t, t)$.

The aim is to approximate $\nabla_{x_t} \ln p(x_t|y)$ with $\nabla_{x_t} \ln p(x_t|\hat{y}_t)$, where \hat{y}_t is a sample from $p(y_t|y)$. Of course this approximation is imperfect and introduces an error, which we call the *approximation error*. CDiffE aims to achieve smaller optimization error by diffusing the condition y and making the optimization landscape easier, at a cost of making this approximation error.

Now in order to obtain samples from the conditional distribution, we sample a point $x_T \sim \pi$ and integrate

$$dx = [\mu(x, t) - \sigma(t)^2 \nabla_x \ln p_{x_t|Y_t}(x|\hat{y}_t)] dt + \sigma(t) d\tilde{w}$$

from T to 0, sampling $\hat{y}_t \sim p(y_t|y)$ at each time step.

3.4.3 Conditional multi-speed diffusive estimator (CMDE)

In this section we present a novel estimator for the conditional score $\nabla_{x_t} \ln p(x_t|y)$ which we call the *conditional multi-speed diffusive estimator* (CMDE).

Sources of error for different estimators

CDE

Optimization error:

$$s_\theta(x, y, t) \approx \nabla_{x_t} \ln p(x_t|y)$$

CDiffE and CMDE

Optimization error:

$$s_\theta(x, y, t) \approx \nabla_{x_t} \ln p(x_t|y_t)$$

Approximation error:

$$\nabla_{x_t} \ln p(x_t|\hat{y}_t) \approx \nabla_{x_t} \ln p(x_t|y)$$

CDiffE aims to achieve smaller optimization error at a cost of higher approximation error. By controlling the diffusion speed of y , CMDE tries to find an optimal balance between optimization error and approximation error.

Figure 2. Sources of error for different estimators

Our approach is based on two insights. Firstly, there is no reason why x_t and y_t in conditional diffusive estimation need to diffuse at the same rate. Secondly, by decreasing the diffusion rate of y_t while keeping the diffusion speed of x_t the same, we can bring $p(x_t|y_t)$ closer to $p(x_t|y)$, at the possible cost of making the optimization more difficult. This way we can *interpolate* between the conditional denoising estimator and the conditional diffusive estimator and find an optimal balance between optimization error and approximation error (cf. Figure 2). This can lead to a better performance, as indicated by our experimental findings (cf. Section 4).

In our conditional multi-speed diffusive estimator, x_t and y_t diffuse according to SDEs with the same drift but different diffusion rates,

$$\begin{aligned} dx &= \mu(x, t) dt + \sigma^x(t) dw \\ dy &= \mu(y, t) dt + \sigma^y(t) dw. \end{aligned}$$

Then, just like in the case of conditional diffusive estimator, we try to approximate the joint score $\nabla_{x_t, y_t} \ln p(x_t, y_t)$ with a neural network. Since x_t and y_t now diffuse according to different SDEs, we need to take this into account and replace the weighting function $\lambda(t) : \mathbb{R} \rightarrow \mathbb{R}_+$ with a positive definite weighting matrix $\Lambda(t) : \mathbb{R} \rightarrow \mathbb{R}^{(n_x+n_y) \times (n_x+n_y)}$. Hence, the new training objective becomes

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_0 \sim p_0(z_0) \\ z_t \sim p(z_t|z_0)}} [v^T \Lambda(t) v], \quad (14)$$

where $v = \nabla_{z_t} \ln p(z_t|z_0) - s_\theta(z_t, t)$, $z_t = (x_t, y_t)$.

In [21] authors derive a likelihood weighting function $\lambda^{\text{MLE}}(t)$, which ensures that the objective of the score-based model upper-bounds the negative log-likelihood of the data, thus enabling approximate maximum likelihood training of score-based diffusion models. We generalize this result to the multi-speed diffusion case by providing a likelihood weighting matrix $\Lambda^{\text{MLE}}(t)$ with the same properties.

Theorem 3. Let $\mathcal{L}(\theta)$ be the CMDE training objective (Equation 14) with the following weighting:

$$\Lambda_{i,j}^{\text{MLE}}(t) = \begin{cases} \sigma^x(t)^2, & \text{if } i = j, i \leq n_x \\ \sigma^y(t)^2, & \text{if } i = j, n_x < i \leq n_y \\ 0, & \text{otherwise} \end{cases}$$

Then the joint negative log-likelihood is upper bounded (up to a constant in θ) by the training objective of CMDE

$$-\mathbb{E}_{(x,y) \sim p(x,y)} [\ln p_\theta(x, y)] \leq \mathcal{L}(\theta) + C.$$

The proof can be found in Appendix A.3.

Moreover we show that the mean squared approximation error of a multi-speed diffusion model is upper bounded and the upper bound goes to zero as the diffusion speed of the condition $\sigma^y(t)$ approaches zero.

Theorem 4. Fix t , x_t and y . Under mild technical assumptions (cf. Appendix A.4) there exists a function $E : \mathbb{R} \rightarrow \mathbb{R}$ monotonically decreasing to 0, such that

$$\begin{aligned} \mathbb{E}_{y_t \sim p(y_t|y)} [\|\nabla_{x_t} \ln p(x_t|y_t) - \nabla_{x_t} \ln p(x_t|y)\|_2^2] \\ \leq E(1/\sigma^y(t)). \end{aligned}$$

The proof can be found in Appendix A.4.

Thus we see that the objective of CMDE approaches that of CDE as $\sigma^y(t) \rightarrow 0$, and CMDE coincides with CDiffE when $\sigma^y(t) = \sigma^x(t)$ (cf. Figure 2).

We experimented with different configurations of $\sigma^x(t)$ and $\sigma^y(t)$ and found configurations that lead to improvements upon CDiffE and CDE in certain tasks. The experimental results are discussed in detail in Section 4.

4. Experiments

4.1. Multiscale diffusion

In this part of the experimental section, we compare the performance of the multiscale model that depends on non-uniform pixel diffusion to the performance of the standard model that depends on uniform diffusion. We train and evaluate both models on Imagenet 128×128 and CelebA-HQ 128×128 .

For the standard diffusion model, we use the beta-linear VP SDE [7] and train the score model using the simple objective [4] because it is experimentally shown to favor generation quality. The architecture of the score model follows the architecture of [4].

For the multiscale model, we use 3-level haar transform to transform the original images, which means that we create a multiscale model with four scales. For this reason, we use four score models $s_{\theta_1}, s_{\theta_2}, s_{\theta_3}, s_{\theta_4}$ which approximate the score function in following diffusion ranges respectively $[\epsilon, 0.25], [0.25, 0.50], [0.50, 0.75], [0.75, 1]$. The reason we do not use s_{θ_1} to approximate the score function for the entire diffusion is that we stop the diffusion of the highest frequency detail coefficients d_1 at time 0.25, as they reach the target minimum SNR (by design of the forward SDE). The remaining diffusing tensor has a quarter of the dimensionality of the original tensor. Therefore, we need a less expressive neural network to approximate the score function in the next diffusion time range. The architecture of all models follows the architecture of [4]. We choose the number of base channels and the depth of the multiscale score models so that the total number of parameters of the multiscale model is approximately equal to the number of parameters of the standard diffusion model to ensure fair comparison. For the diffusion of each haar coefficient, we use a variance preserving process with log-linear SNR profile. We choose the maximum SNR (at $t = \epsilon$) and the minimum SNR (achieved at the terminal diffusion time for each coefficient) to match the maximum SNR and minimum SNR of the standard model respectively.

We evaluate both models using the FID score on 50K samples. We generate each sample by numerically integrating the reverse SDE with 256 total euler-maruyama steps. Our results (see tables 1, 2) show that for the same training time, the multiscale model achieves better FID score with significantly faster sampling speed (4.4 times faster). In fact, our results on ImageNet show that the multiscale model achieves improved FID score with faster sampling speed and less training time. The FID scores are higher than reported scores in prior work for both the multiscale and the standard model because we did not use Tweedie's formula for denoising the last step. We verified that by integrating the corresponding probability flow ODEs using the euler method. In that case, we got lower FID scores for both methods but the relative performance remained the same. Moreover, we used lighter neural networks than prior works to approximate the score function which led to generally worse performance. We opted for lighter models in this study because we wanted to conduct a fair comparison of the multiscale diffusion model and the standard uniform diffusion model. Improved techniques that led to state-of-the-art performance of the uniform diffusion model such as class conditioning and learning of the variance schedule [4] can also be readily employed in the multiscale model. Given the superiority of the multiscale model, we expect the employment of improved techniques to further improve the performance of the multiscale model and potentially redefine the state-of-the-art. We intend to explore this direction

in the future.

The training and sampling speed-up is attributed to the fact that we approximate the score of lower dimensional distributions for the majority of the diffusion. Therefore, we expect higher relative speed-ups in higher resolutions. We believe that the effectiveness of the multiscale model is attributed to the effectiveness of cascaded diffusion observed in previous works [4, 19]. The difference between our multiscale model and the previous works is that it does not suffer from the effect of the compounding error. Ho et al. [19] improve the performance of cascading models by using an expensive post-training tuning step which they call conditioning augmentation. Our multiscale model essentially employs a cascading modeling structure that does not require any post-training tuning for improved sample generation.

4.2. Conditional Generation

In this section we conduct a systematic comparison of different score-based diffusion approaches to modelling conditional distributions of image data. We evaluate these approaches on the tasks of super-resolution, inpainting and edge to image translation.

Datasets In our experiments, we use the CelebA [13] and Edges2shoes [9, 28] datasets. We pre-processed the CelebA dataset as in [12].

Models and hyperparameters In order to ensure the fair comparison, we separate the evaluation of a particular estimator of conditional score from the evaluation of a particular neural network model. To this end, we train the same neural network architecture for all estimators. The architecture is based on the DDPM model used in [7, 23]. We used the variance-exploding SDE [23] given by:

$$dx = \sqrt{\frac{d}{dt}\sigma^2(t)}dw, \quad \sigma(t) = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t$$

Likelihood weighting was employed for all experiments. For CMDE, the diffusion speed of y was controlled by picking an appropriate σ_{\max}^y , which we found by trial-and-error. The performance of CMDE could be potentially improved by performing a systematic hyperparameter search for optimal σ_{\max}^y . Details on hyperparameters and architectures used in our experiments can be found in Appendix B.

Inverse problems The tasks of inpainting, super-resolution and edge to image translation are special cases of inverse problems [2, 15]. In each case, we are given a (possibly random) forward operator A which maps our data x (full image) to an observation y (masked image, compressed image, sketch). The task is to come up with a high-quality reconstruction \hat{x} of the image x based on an observation y . The problem of reconstructing x from y is typically ill-posed, since y does not contain all information about x . Therefore, an ideal algorithm would produce a reconstruction \hat{x} , which

looks like a realistic image (i.e. is a likely sample from $p(x)$) and is consistent with the observation y (i.e. $A\hat{x} \approx y$). Notice that if a conditional score model learns the conditional distribution correctly, then our reconstruction \hat{x} is a sample from the posterior distribution $p(x|y)$, which satisfies bespoke requirements. This strategy for solving inverse problems is generally referred to as *posterior sampling*.

Evaluation: Reconstruction quality Ill-posedness often means that we should not strive to reconstruct x perfectly. Nonetheless reconstruction error does correlate with the performance of the algorithm and has been one of the most widely-used metrics in the community. To evaluate the reconstruction quality for each task, we measure the Peak signal-to-noise ratio (PSNR) [26], Structural similarity index measure (SSIM) [26] and Learned Perceptual Image Patch Similarity (LPIPS) [29] between the original image x and the reconstruction \hat{x} .

Evaluation: Consistency In order to evaluate the consistency of the reconstruction, for each task we calculate the PSNR between $y := Ax$ and $\hat{y} := A\hat{x}$.

Evaluation: Diversity We evaluate diversity of each approach by generating five reconstructions $(\hat{x})_{i=1}^5$ for a given observation y . Then for each y we calculate the average standard deviation for each pixel among the reconstructions $(\hat{x})_{i=1}^5$. Finally, we average this quality over 5000 test observations.

Evaluation: Distributional distances If our algorithm generates realistic reconstructions while preserving diversity, then the distribution of reconstructions $p(\hat{x})$ should be similar to the distribution of original images $p(x)$. Therefore, we measure the Fréchet Inception Distance (FID) [6] between unconditional distributions $p(x)$ and $p(\hat{x})$ based on 5000 samples. Moreover, we calculate the FID score between the joint distributions $p(\hat{x}, y)$ and $p(x, y)$, which allows us to simultaneously check the realism of the reconstructions and the consistency with the observation. We use abbreviation UFID to refer to FID between unconditional distributions and JFID to refer to FID between joints. In our judgement, FID and especially the JFID is the most principled of the used metrics, since it measures how far $p_\theta(x|y)$ is from $p(x|y)$.

4.2.1 Inpainting

We perform the inpainting experiment using CelebA dataset. In inpainting, the forward operator A is an application of a given binary mask to an image x . In our case, we made the task more difficult by using randomly placed (square) masks. Then the conditional score model is used to obtain a reconstruction \hat{x} from the masked image y . We select the position of the mask uniformly at random and cover 25% of the image. The quantitative results are summarised in Table 3 and samples are presented in Figure 5. We ob-

Table 1. Multiscale and Vanilla model comparison on ImageNet 128x128

	Iterations	Parameters	Training (hours) ↓	Sampling (secs) ↓	FID ↓
Vanilla	1M	100M	191.0	53.5	79.21
Multiscale	2M	100M	136.6	12.1	70.87
Multiscale +	2M	200M	151.0	18.7	65.50

Table 2. Multiscale and Vanilla model comparison on CelebA-HQ 128x128

	Iterations	Parameters	Training (hours) ↓	Sampling (secs) ↓	FID ↓
Vanilla	0.67M	100M	128	53.5	54.3
Multiscale	2.54M	100M	128	12.1	31.8
Multiscale +	1.76M	200M	128	18.7	33.5

serve that CDE and CMDE significantly outperform CDiffE in all metrics, with CDE having a small advantage over CMDE in terms of reconstruction error and consistency. On the other hand, CMDE achieves the best FID scores.

4.2.2 Super-resolution

We perform 8x super-resolution using the CelebA dataset. A high resolution 160x160 pixel image x is compressed to a low resolution 20x20 pixels image y . Here we use bicubic downscaling [10] as the forward operator A . Then using a score model we obtain a 160x160 pixel reconstruction image \hat{x} . The quantitative results are summarised in Table 3 and samples are presented in Figure 6. We find that CMDE and CDE perform similarly, while significantly outperforming CDiffE. CMDE achieves the smallest reconstruction error and captures the distribution most accurately according to FID scores.

4.2.3 Edge to image translation

We perform an edge to image translation task on the Edges2shoes dataset. The forward operator A is given by a neural network edge detector [27], which takes an original photo of a shoe x and transforms it into a sketch y . Then a conditional score model is used to create an artificial photo of a shoe \hat{x} matching the sketch. The quantitative results are summarised in Table 3 and samples are presented in Figure 4. Unlike in inpainting and super-resolution where CDiffE achieved reasonable performance, in edge to image translation, it fails to create samples consistent with the condition (which leads to inflated diversity scores). CDE and CMDE are comparable, but CDE performed slightly better across all metrics. However, the performance of CMDE could be

potentially improved by tuning the diffusion speed $\sigma^y(t)$.

5. Conclusions and future work

In this article, we explored non-uniform diffusion models which rely on the idea of diffusing different parts of the tensor with different speeds or more generally according to different SDEs. We show that non-uniform diffusion leads to multiscale diffusion models which are more efficient and effective than standard uniform diffusion models for unconditional generation. More specifically, multiscale diffusion models achieve improved FID score with significantly faster sampling speed and for less training time.

We further discovered that non-uniform diffusion leads to CMDE, a novel estimator of the conditional score which can interpolate between conditional denoising estimator (CDE) and conditional diffusive estimator (CDiffE). We conducted a systematic comparison of different estimators of the conditional score and concluded that CMDE and CDE perform on par, while significantly outperforming CDiffE. This is particularly apparent in edge to image translation, where CDiffE fails to produce samples consistent with the condition image. Furthermore, CMDE outperformed CDE in terms of FID scores in inpainting and super-resolution tasks, which indicates that diffusing the condition at the appropriate speed can have beneficial effect on the optimization landscape, and yield better approximation of the posterior distribution. Furthermore, we provided theoretical analysis of the estimators of conditional score. More importantly, we proved the consistency of the conditional denoising estimator, thus providing a firm theoretical justification for using it in future research.



Figure 3. Diversity of five different CMDE reconstructions for a given masked image.

Table 3. Results of conditional generation tasks.

	Estimator	PSNR/SSIM \uparrow	LPIPS \downarrow	UFID/JFID \downarrow	Consistency \uparrow	Diversity \uparrow
Inpainting	CDE	25.12/0.870	0.042	13.07/18.06	28.54	4.79
	CDiffE	23.07/0.844	0.057	13.28/19.25	26.61	6.52
	CMDE ($\sigma_{max}^y = 1$)	24.92/0.864	0.044	12.07/17.07	28.32	4.98
Super-resolution	CDE	23.80/0.650	0.114	10.36/15.77	54.18	8.51
	CDiffE	23.83/0.656	0.139	14.29/20.20	51.90	7.41
	CMDE ($\sigma_{max}^y = 0.5$)	23.91/0.654	0.109	10.28/15.68	53.03	8.33
Edge to image	CDE	18.35/0.699	0.156	11.87/21.31	10.45	14.40
	CDiffE	10.00/0.365	0.350	33.41/55.22	7.78	43.45
	CMDE ($\sigma_{max}^y = 1$)	18.16/0.692	0.158	12.62/22.09	10.38	15.20

6. Acknowledgements

GB acknowledges the support from GSK and the Cantab Capital Institute for the Mathematics of Information. JS acknowledges the support from Aviva and the Cantab Capital Institute for the Mathematics of Information. CBS acknowledges support from the Philip Leverhulme Prize, the Royal Society Wolfson Fellowship, the EPSRC advanced career fellowship EP/V029428/1, EPSRC grants EP/S026045/1 and EP/T003553/1, EP/N014588/1, EP/T017961/1, the Wellcome Innovator Award RG98755, the Leverhulme Trust project Unveiling the invisible, the European Union Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 777826 NoMADS, the Cantab Capital Institute for the Mathematics of Information and the Alan Turing Institute. CE acknowledges support from the Wellcome Innovator Award RG98755 for part of the work that was done at Cambridge.

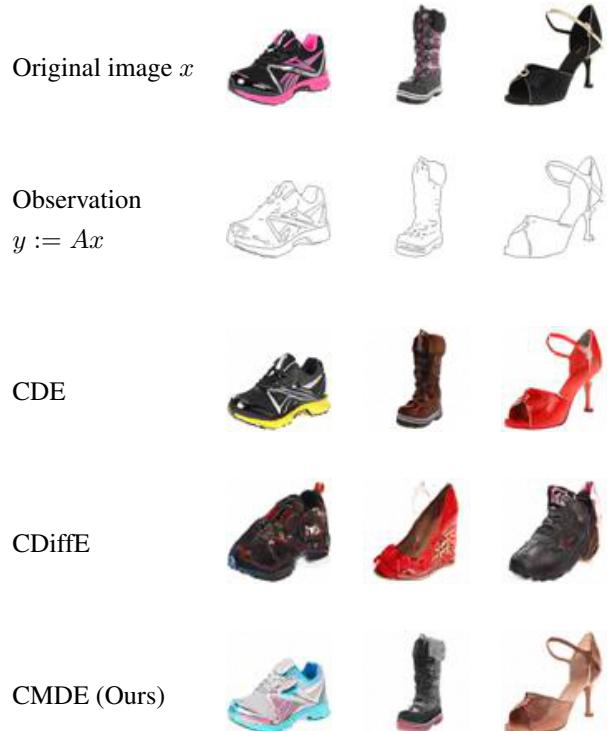


Figure 4. Edge to image translation results.

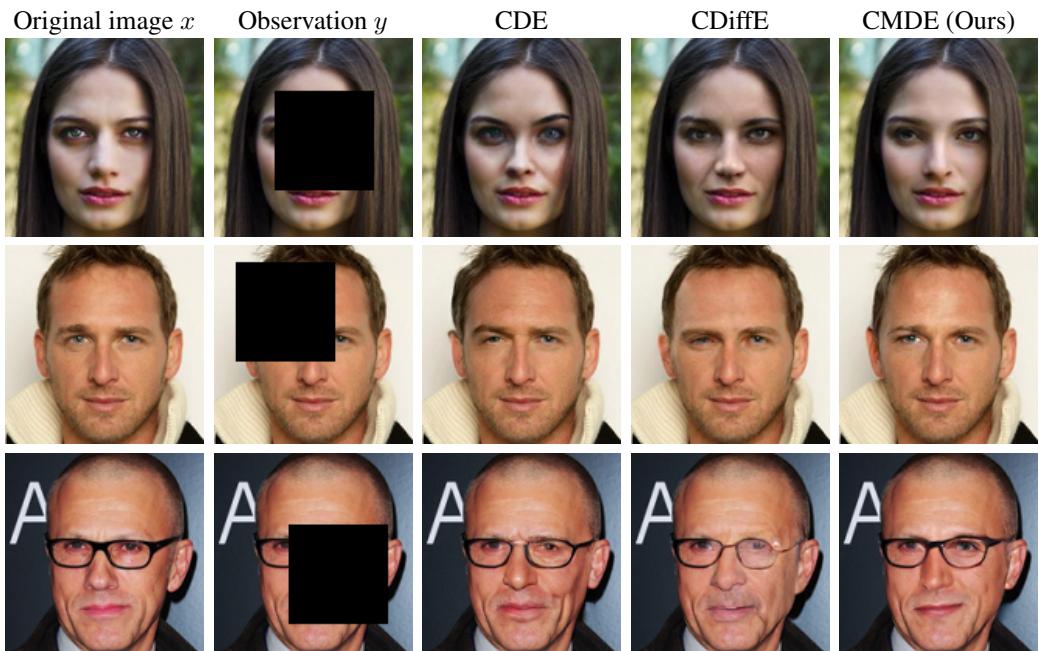


Figure 5. Inpainting results.

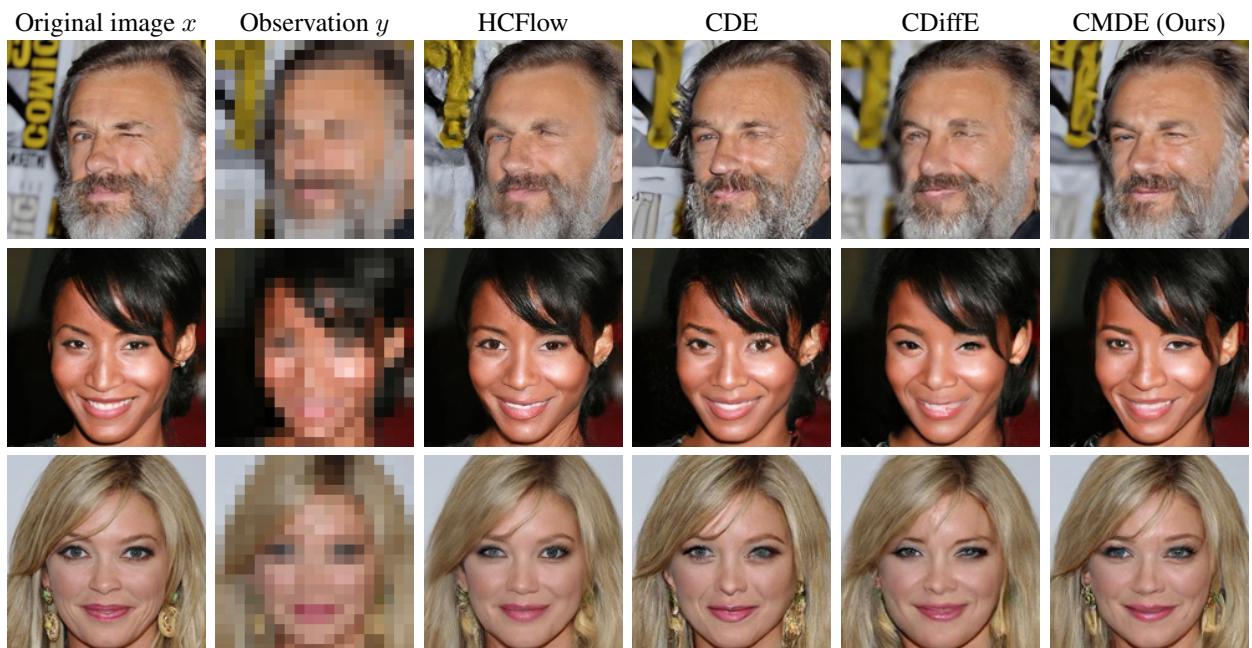


Figure 6. Super-resolution results.

References

- [1] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982. [3](#), [15](#)
- [2] Simon Arridge, Peter Maass, Ozan Öktem, and Carola-Bibiane Schönlieb. Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174, 2019. [8](#)
- [3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155, Mar. 2003. [1](#)
- [4] Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. [2](#), [3](#), [7](#), [8](#)
- [5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. [1](#)
- [6] 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, 2018. [8](#)
- [7] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. [2](#), [3](#), [7](#), [8](#)
- [8] Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. [2](#), [3](#)
- [9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks, 2018. [8](#)
- [10] R. Keys. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160, 1981. [9](#)
- [11] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2014. [1](#)
- [12] Jingyun Liang, Andreas Lugmayr, Kai Zhang, Martin Danelljan, Luc Van Gool, and Radu Timofte. Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling, 2021. [8](#)
- [13] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. [8](#)
- [14] Christian Léonard. Some properties of path measures, 2013. [15](#)
- [15] Jennifer L. Mueller and Samuli Siltanen. Linear and nonlinear inverse problems with practical applications. In *Computational science and engineering*, 2012. [8](#)
- [16] Whitney K. Newey and Daniel McFadden. Chapter 36 large sample estimation and hypothesis testing. volume 4 of *Handbook of Econometrics*, pages 2111–2245. Elsevier, 1994. [14](#)
- [17] Bernt Oksendal. *Stochastic Differential Equations (5th Ed.): An Introduction with Applications*. Springer-Verlag, Heidelberg, 2003. [16](#)
- [18] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference, 2021. [1](#)
- [19] Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J. Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement, 2021. [2](#), [4](#), [5](#), [8](#), [19](#)
- [20] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015. [2](#), [3](#)
- [21] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. [3](#), [4](#), [7](#), [15](#)
- [22] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution, 2020. [2](#), [3](#), [5](#), [6](#)

- [23] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. [2](#), [3](#), [5](#), [6](#), [8](#), [15](#), [19](#)
- [24] Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation, 2021. [2](#), [5](#)
- [25] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. [2](#), [3](#), [5](#), [14](#), [16](#)
- [26] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. [8](#)
- [27] Saining Xie and Zhuowen Tu. Holistically-nested edge detection, 2015. [9](#)
- [28] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Computer Vision and Pattern Recognition (CVPR)*, Jun 2014. [8](#)
- [29] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. [8](#)

A. Proofs

A.1. Equality of minimizers for CDE

Lemma 1. For a fixed $y \in \mathbb{R}^d$ and $t \in \mathbb{R}$ we have

$$\begin{aligned} & \mathbb{E}_{\substack{x_0 \sim p(x_0|y) \\ x_t \sim p(x_t|x_0,y)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0,y) - s_\theta(x_t,y,t)\|_2^2] \\ &= \mathbb{E}_{x_t \sim p(x_t|y)} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t,y,t)\|_2^2] \end{aligned}$$

Proof. Since y and t are fixed, we may define $\psi(x_t) := s_\theta(x_t,y,t)$, $q(x_0) := p(x_0|y)$ and $q(x_t|x_0) = p(x_t|x_0,y)$. Therefore, by the Tower Law, the statement of the lemma is equivalent to

$$\begin{aligned} & \mathbb{E}_{x_0, x_t \sim q(x_0, x_t)} [\|\nabla_{x_t} \ln q(x_t|x_0) - \psi(x_t)\|_2^2] \\ &= \mathbb{E}_{x_t \sim q(x_t)} [\|\nabla_{x_t} \ln q(x_t) - \psi(x_t)\|_2^2] \end{aligned}$$

Which follows directly from [25, Eq. 11]. \square

Theorem 1. The minimizer of

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t|x_0)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t,y,t)\|_2^2]$$

in θ is the same as the minimizer of

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{x_t, y \sim p(x_t, y)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t,y,t)\|_2^2].$$

Proof. First, notice that x_t is conditionally independent of y given x_0 . Therefore, by applying the Tower Law we obtain

$$\begin{aligned} & \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t|x_0)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t,y,t)\|_2^2] \\ &\stackrel{(1)}{=} \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{y \sim p(y) \\ x_0 \sim p(x_0|y) \\ x_t \sim p(x_t|x_0,y)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t,y,t)\|_2^2] \\ &\stackrel{(2)}{=} \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{y \sim p(y) \\ x_0 \sim p(x_0|y) \\ x_t \sim p(x_t|x_0,y)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0,y) - s_\theta(x_t,y,t)\|_2^2] \\ &= \mathbb{E}_{t \sim U(0,T)} [f(t,y)] =: (*) \end{aligned}$$

where

$$\begin{aligned} f(t,y) &:= \\ & \mathbb{E}_{\substack{x_0 \sim p(x_0|y) \\ x_t \sim p(x_t|x_0,y)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0,y) - s_\theta(x_t,y,t)\|_2^2]. \end{aligned}$$

Now fix y and t . By Lemma 1, it follows that

$$\begin{aligned} & f(t,y) \\ &= \mathbb{E}_{x_0 \sim p(x_0|y)} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|x_0,y) - s_\theta(x_t,y,t)\|_2^2] \\ &\stackrel{(3)}{=} \mathbb{E}_{x_t \sim p(x_t|y)} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t,y,t)\|_2^2] \end{aligned}$$

Since t and y were arbitrary, this is true for all t and y . Therefore, substituting back into $(*)$ we get that

$$\begin{aligned} (*) &= \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{y \sim p(y) \\ x_t \sim p(x_t|y)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t,y,t)\|_2^2] \\ &\stackrel{(1)}{=} \mathbb{E}_{\substack{t \sim U(0,T) \\ \substack{x_t, y \sim p(x_t, y)}}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t|y) - s_\theta(x_t,y,t)\|_2^2]. \end{aligned}$$

(1) Tower Law, (2) Conditional independence of x_t and y given x_0 , (3) Lemma 1. \square

A.2. Consistency of CDE

In order to prove the consistency, in this subsection we make the following assumptions:

Assumption 1. The space of parameters Θ and the data space \mathcal{X} are compact.

Assumption 2. There exists a unique $\theta^* \in \Theta$ such that $s_{\theta^*}(x,y,t) = \nabla_{x_t} \ln p(x,y,t)$.

First we state some technical, but well-known lemmas, which will be useful in proving our consistency result.

Lemma 2 (Uniform law of large numbers). [16, Lemma 2.4]

Let z_i be i.i.d from a distribution $q(z)$ and suppose that:

- Θ is compact.
- $f(z, \theta)$ is continuous for all $\theta \in \Theta$ and almost all z .
- $f(\cdot, \theta)$ is a measurable function of z for each θ .
- There exists $d : \mathcal{Z} \rightarrow \mathbb{R}$ such that $\mathbb{E}[d(z)] < \infty$ and $\|f(z, \theta)\| \leq d(z)$ for each θ .

Then $\mathbb{E}_z[f(z, \theta)]$ is continuous in θ , and $\frac{1}{n} \sum_{i=1}^n f(z_i, \theta)$ converges to $\mathbb{E}_z[f(z, \theta)]$ uniformly in probability, i.e.:

$$\sup_{\theta} \left\| \frac{1}{n} \sum_{i=1}^n f(z_i, \theta) - \mathbb{E}_z[f(z, \theta)] \right\| \xrightarrow{P} 0$$

Lemma 3 (Consistency of extremum estimators). [16, Theorem 2.1]

Let Θ be compact and consider a family of functions $\mathcal{L}^{(n)} : \Theta \rightarrow \mathbb{R}$. Moreover, suppose there exists a function $\mathcal{L} : \Theta \rightarrow \mathbb{R}$ such that

- $\mathcal{L}(\theta)$ is uniquely minimized at θ^* .
- $\mathcal{L}(\theta)$ is continuous.
- $\mathcal{L}^{(n)}(\theta)$ converges uniformly in probability to $\mathcal{L}(\theta)$.

Then

$$\theta_n^* := \arg \min_{\theta \in \Theta} \mathcal{L}^{(n)}(\theta) \xrightarrow{P} \theta^*.$$

Corollary 1. Let θ_n^* be a minimizer of a n -sample Monte Carlo approximation of

$$\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t | x_0)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t | x_0) - s_\theta(x_t, y, t)\|_2^2].$$

Then under assumptions 1 and 2, the conditional denoising estimator $s_{\theta_n^*}(x, y, t)$ is a consistent estimator of the conditional score $\nabla_{x_t} \ln p(x_t | y)$, i.e.

$$s_{\theta_n^*}(x, y, t) \xrightarrow{P} \nabla_{x_t} \ln p(x_t | y),$$

as the number of Monte Carlo samples n approaches infinity.

Proof. By conditional independence and the Tower Law, we get

$$\begin{aligned} & \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t | x_0)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t | x_0) - s_\theta(x_t, y, t)\|_2^2] \\ &= \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, y \sim p(x_0, y) \\ x_t \sim p(x_t | x_0)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t | x_0) - s_\theta(x_t, y, t)\|_2^2] \\ &= \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0, x_t, y \sim p(x_0, x_t, y)}} [\lambda(t) \|\nabla_{x_t} \ln p(x_t | x_0) - s_\theta(x_t, y, t)\|_2^2]. \end{aligned}$$

Let $z = (t, x_0, x_t, y)$ and denote by $q(z) := p(t, x_0, x_t, y)$ the joint distribution. Moreover, define $f(z, \theta) := \lambda(t) \|\nabla_{x_t} \ln p(x_t | x_0) - s_\theta(x_t, y, t)\|_2^2$. Since $t \sim U(0, T)$ is independent of $(x_0, x_t, y) \sim p(x_0, x_t, y)$, the above is equal to

$$\mathbb{E}_{z \sim q(z)} [f(z, \theta)]$$

Therefore by Lemma 2, the Monte Carlo approximation of 12: $\mathcal{L}^{(n)}(\theta) = \frac{1}{n} \sum_{i=1}^n f(z_i, \theta)$ converges uniformly in probability to $\mathcal{L}(\theta) = \mathbb{E}_{z \sim q(z)} [f(z, \theta)]$. Let θ^* be the minimizer of $\mathcal{L}(\theta)$, by Lemma 3 we get that $\theta_n^* \xrightarrow{P} \theta^*$. Finally by Theorem 2, θ^* is also a minimizer of the Fisher divergence between $s_{\theta^*}(x_t, y, t)$ and $\nabla_{x_t} \ln p(x_t | y)$ and by Assumption 2 this implies that $s_{\theta^*}(x_t, y, t) = \nabla_{x_t} \ln p(x_t | y)$. Hence $s_{\theta_n^*}(x, y, t) \xrightarrow{P} \nabla_{x_t} \ln p(x_t | y)$. \square

A.3. Likelihood weighting for multi-speed and multi-sde models

In this section we derive the likelihood weighting for multi-sde models (Theorem 3). First using the framework in [23, Appendix A] we present the Anderson's theorem for multi-dimensional SDEs with non-homogeneous covariance matrix (without assuming $\Sigma(t) \neq \sigma(t)I$) and generalize the main result of [21] to this setting. Then, we cast the problem of multi-speed and multi-sde diffusion as a special case of multi-dimensional diffusion with a particular covariance matrix $\Sigma(t)$ and thus obtain the likelihood weighting for multi-sde models (Theorem 3).

Consider an Ito's SDE

$$dx = \mu(x, t)dt + \Sigma(t)dw$$

where $\mu : \mathbb{R}^{n_x} \times [0, T] \rightarrow \mathbb{R}^{n_x}$ and $\Sigma : [0, T] \rightarrow \mathbb{R}^{n_x \times n_x}$ is a time-dependent positive-definite matrix. By multidimensional Anderson's Theorem [1] the corresponding reverse time SDE is given by

$$dx = \tilde{\mu}(x, t)dt + \Sigma(t)dw \quad (15)$$

$$\text{where } \tilde{\mu}(x, t) := \mu(x, t) - \Sigma(t)^2 \nabla_x \ln p_{X_t}(x).$$

If we train a score-based diffusion model to approximate $\nabla_x \ln p_{X_t}(x)$ with a neural network $s_\theta(x, t)$ we will obtain the following approximate reverse-time sde

$$dx = \tilde{\mu}_\theta(x, t)dt + \Sigma(t)dw \quad (16)$$

$$\text{where } \tilde{\mu}_\theta(x, t) := \mu(x, t) - \Sigma(t)^2 s_\theta(x, t)$$

Now we generalize [21, Theorem 1] to multi-dimensional setting.

Theorem 6. Let $p(x_t)$ and $p_\theta(x_t)$ denote marginal distributions of 15 and 16 respectively. Then under regularity assumptions of [21, Theorem 1] we have that

$$\begin{aligned} KL(p(x_0) | p_\theta(x_0)) &\leq KL(p(x_T) | \pi(x_T)) \\ &\quad + \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_t \sim p(x_t)}} [v^T \Sigma(t)^2 v], \end{aligned}$$

$$\text{where } v = \nabla_{x_t} \ln p(x_t) - s_\theta(x_t, t).$$

Proof. We proceed in close analogy to the proof of [21, Theorem 1] but we use a more general diffusion matrix $\Sigma(t)$. Let P be the law of the true reverse-time sde and let P_θ be the law of the approximate reverse-time sde. Then by [14, Theorem 2.4] (generalized chain rule for KL divergence) we have

$$\begin{aligned} KL(P | P_\theta) &= KL(p(x_0) | p_\theta(x_0)) \\ &\quad + \mathbb{E}_{z \sim p(x_0)} [KL(P(\cdot | x_0 = z) | P_\theta(\cdot | x_0 = z))]. \end{aligned}$$

Since $\mathbb{E}_{z \sim p(x_0)} [KL(P(\cdot | x_0 = z) | P_\theta(\cdot | x_0 = z))] \geq 0$, this implies

$$KL(p(x_0) | p_\theta(x_0)) \leq KL(P | P_\theta)$$

Using the fact that $p_\theta(x_T) = \pi$ and applying [14, Theorem 2.4] again, we obtain

$$\begin{aligned} KL(P | P_\theta) &= KL(p(x_T) | \pi) \\ &\quad + \mathbb{E}_{z \sim p(x_T)} [KL(P(\cdot | x_T = z) | P_\theta(\cdot | x_T = z))]. \end{aligned}$$

Let $P^z := P(\cdot | x_T = z)$ and $P_\theta^z := P_\theta(\cdot | x_T = z)$

$$\begin{aligned} \mathbb{E}_{z \sim p(x_T)} [KL(P(\cdot | x_T = z) | P_\theta(\cdot | x_T = z))] \\ = -\mathbb{E}_{z \sim p(x_T)} \left[\mathbb{E}_{P^z} \left[\ln \frac{dP_\theta^z}{dP^z} \right] \right] \end{aligned}$$

Using Girsanov Theorem [17, Theorem 8.6.5] and the fact that $\Sigma(t)$ is symmetric and invertible

$$= \mathbb{E}_{z \sim p(x_T)} \left[\mathbb{E}_{P^z} \left[\int_0^T \Sigma(t) v(x_t, t) dw_t + \frac{1}{2} \int_0^T v(x_t, t)^T \Sigma(t)^2 v(x_t, t) dt \right] \right]$$

where $v(x_t, t) = \nabla_{x_t} \ln p(x_t) - s_\theta(x_t, t)$. Since $\int_0^T \Sigma(t) v(x_t, t) dw_t$ is a martingale (Ito's integral wrt Brownian motion)

$$\begin{aligned} &= \frac{1}{2} \mathbb{E}_{z \sim p(x_T)} \left[\mathbb{E}_{P^z} \left[\int_0^T v(x_t, t)^T \Sigma(t)^2 v(x_t, t) dt \right] \right] \\ &= \frac{1}{2} \int_0^T \mathbb{E}_{x \sim p(x_t)} [v(x_t, t)^T \Sigma(t)^2 v(x_t, t)] dt \\ &= \frac{1}{2} \mathbb{E}_{t \sim U(0, T)} [v(x_t, t)^T \Sigma(t)^2 v(x_t, t)]. \end{aligned}$$

□

A.3.1 Multi-sde and multi-speed diffusion

Now we consider again the multi-speed and the more general multi-sde diffusion frameworks from sections 3.3 and 3.4. Suppose that we have two tensors x and y which diffuse according to different SDEs

$$\begin{aligned} dx &= \mu^x(x, t) dt + \sigma^x(t) dw \\ dy &= \mu^y(y, t) dt + \sigma^y(t) dw \end{aligned}$$

We may cast this system of two SDEs, as a single SDE

$$dz = \mu^z(z, t) dt + \Sigma_z(t) dw$$

where $z = (x, y)$, $\mu^z(z, t) = (\mu^x(x, t), \mu^y(y, t))$ and

$$\Sigma_z(t) = \begin{cases} \sigma^x(t), & \text{if } i = j, i \leq n_x \\ \sigma^y(t), & \text{if } i = j, n_x < i \leq n_y \\ 0, & \text{otherwise} \end{cases}.$$

If we train a score-based diffusion model for $z_t = (x_t, y_t)$, then by Theorem 6

$$KL(p(z_0) | p_\theta(z_0)) \leq C_1 + \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_t \sim p(z_t)}} [v^T \Sigma_z(t)^2 v],$$

where $C_1 := KL(p(x_T) | \pi(x_T))$ does not depend on θ . Because Λ_{MLE} (from Theorem 3) is equal to $\Sigma_z(t)^2$, we may rewrite the above as

$$KL(p(z_0) | p_\theta(z_0)) \leq C_1 + \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_t \sim p(z_t)}} [v^T \Lambda_{MLE}(t)^2 v],$$

and since by denoising score matching [25]

$$\begin{aligned} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_t \sim p(z_t)}} [v^T \Lambda_{MLE}(t) v] &= \\ \mathbb{E}_{\substack{t \sim U(0, T) \\ z_0 \sim p_0(z_0) \\ z_t \sim p(z_t | z_0)}} [v^T \Lambda_{MLE}(t) v] + C_2 \end{aligned}$$

where C_2 is another term constant in θ . We conclude that

$$KL(p(z_0) | p_\theta(z_0)) \leq \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_0 \sim p_0(z_0) \\ z_t \sim p(z_t | z_0)}} [v^T \Lambda_{MLE}(t) v] + C_3$$

where $C_3 := C_1 + C_2$. Now recall that the term on the RHS is exactly the training objective of a multi-sde score-based diffusion model with likelihood weighting

$$\mathcal{L}(\theta) := \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0, T) \\ z_0 \sim p_0(z_0) \\ z_t \sim p(z_t | z_0)}} [v^T \Lambda_{MLE}(t) v].$$

Therefore

$$KL(p(z_0) | p_\theta(z_0)) \leq \mathcal{L}(\theta) + C_3.$$

Finally, since $KL(p(z_0) | p_\theta(z_0)) = \mathbb{E}_{(x, y) \sim p(x, y)} [\ln p(x, y)] - \mathbb{E}_{(x, y) \sim p(x, y)} [\ln p_\theta(x, y)]$, we have

$$-\mathbb{E}_{(x, y) \sim p(x, y)} [\ln p_\theta(x, y)] \leq \mathcal{L}(\theta) + C$$

where $C := C_3 - \mathbb{E}_{(x, y) \sim p(x, y)} [\ln p(x, y)]$ is independent of θ . Thus the Theorem 3 is established.

A.4 Mean square approximation error

Assumption 3. $p(x, y) \in C^2(\mathcal{X})$

Assumption 4. $p(x, y) > 0$ for all x, y .

Assumption 5. The data space \mathcal{X} is compact.

Lemma 4. Under assumptions 3 and 5 we have

$$\begin{aligned} p_{Y_t | X_t}(y_t | x_t) &= (p_{Y | X_t}(\cdot | x_t) * \varphi_\sigma)(y_t) \\ \partial_{x_t} p_{Y_t | X_t}(y_t | x_t) &= (\partial_{x_t} p_{Y | X_t}(\cdot | x_t) * \varphi_\sigma)(y_t) \end{aligned}$$

Proof. For this proof, we drop our convention of denoting the probability distribution of a random variable via the name of its density's argument.

$$\begin{aligned} p_{Y_t | X_t}(y_t | x_t) &= \\ &= \int p_{Y, Y_t | X_t}(y, y_t | x_t) dy \\ &= \int p_{Y | X_t}(y | x_t) p_{Y_t | Y, X_t}(y_t | y, x_t) dt \\ &= \int p_{Y | X_t}(y | x_t) p_{Y_t | Y}(y_t | y) dy \end{aligned}$$

Since $Y_t|Y$ has normal distribution with mean y and variance $\sigma^y(t)^2$:

$$\begin{aligned} &= \int p_{Y|X_t}(y|x_t) \varphi_\sigma(y_t - y) dy \\ &= (p_{Y|X_t}(\cdot|x_t) * \varphi_\sigma)(y_t) \end{aligned}$$

where φ_σ is a Gaussian kernel with variance $\sigma^y(t)^2$. Moreover, under the assumptions of the lemma we can exchange the differentiation and integration. Therefore

$$\begin{aligned} \partial_{x_t} p_{Y|X_t}(y_t|x_t) &= \partial_{x_t} \int p_{Y|X_t}(y|x_t) \varphi_\sigma(y_t - y) dy \\ &= \int \partial_{x_t} p_{Y|X_t}(y|x_t) \varphi_\sigma(y_t - y) dy \\ &= (\partial_{x_t} p_{Y|X_t}(\cdot|x_t) * \varphi_\sigma)(y_t) \end{aligned}$$

□

Lemma 5. Let f be a C^1 -function on a compact domain \mathcal{X} and let φ_σ be a Gaussian kernel with variance σ^2 . Then there exists a function $E : \mathbb{R} \rightarrow \mathbb{R}$, which is monotonically decreasing to zero, such that

$$\|(f * \varphi_\sigma) - f\|_\infty \leq E(1/\sigma).$$

Proof.

$$\begin{aligned} &|(f * \varphi_\sigma)(y) - f(y)| \\ &= \left| \int f(z) \varphi_\sigma(z - y) dz - \int f(y) \varphi_\sigma(z - y) dz \right| \\ &\leq \int |f(z) - f(y)| \varphi_\sigma(z - y) dz \end{aligned}$$

Since f is a C^1 function on a compact domain, it is Lipschitz and bounded (in absolute value) by some constant M . Fix $\epsilon > 0$, and let L denote the Lipschitz constant of f . We have that $|f(z) - f(y)| < \epsilon$ whenever $\|z - y\| < \epsilon/L$. Let $B_y(\epsilon/L) := \{z \in \mathcal{X} : \|z - y\| < \epsilon/L\}$ be a ball of radius ϵ/L around y . Then

$$\begin{aligned} &\int |f(z) - f(y)| \varphi_\sigma(z - y) dz \\ &= \int_{B_y(\epsilon/L)} |f(z) - f(y)| \varphi_\sigma(z - y) dz \\ &\quad + \int_{\mathcal{X} \setminus B_y(\epsilon/L)} |f(z) - f(y)| \varphi_\sigma(z - y) dz \\ &\leq \epsilon + \int_{\mathcal{X} \setminus B_y(\epsilon/L)} 2M \varphi_\sigma(z - y) dz \\ &= \epsilon + 2MP\left(|Z_\sigma| > \frac{\epsilon}{L}\right) \end{aligned}$$

where Z_σ is a normally-distributed random variable with mean zero and variance σ^2 . By the Chernoff bound, we

have

$$\leq \epsilon + 4M \exp\left(-\frac{\epsilon^2}{2L^2\sigma^2}\right).$$

Define $E_\epsilon(1/\sigma) := \epsilon + 4M \exp\left(-\frac{\epsilon^2}{2L^2\sigma^2}\right)$. Observe that $E_\epsilon : \mathbb{R}_+ \rightarrow \mathbb{R}$ is monotonically decreasing to ϵ . Moreover

$$\|(f * \varphi_\sigma) - f\|_\infty \leq E_\epsilon(1/\sigma).$$

Now let $A := [0, 1]$ and define

$$E(1/\sigma) := \min_{\epsilon \in A} E_\epsilon(1/\sigma).$$

Notice that the above minimum is achieved, since A is compact and for a fixed σ , the function $\epsilon \mapsto E_\epsilon(1/\sigma)$ is continuous.

We will prove that E is a monotonically decreasing to zero and upper-bounds $\|(f * \varphi_\sigma) - f\|_\infty$. Firstly, it is clear that $E(x) \rightarrow 0$ as $x \rightarrow \infty$, since for all $\epsilon \in A$ we have $\lim_{x \rightarrow \infty} E(x) \leq \lim_{x \rightarrow \infty} E_\epsilon(x) = \epsilon$. Secondly, suppose $a < b$, and let ϵ_a be such that $E(a) = E_{\epsilon_a}(a)$. Then

$$E(b) = \inf_{\epsilon \in A} E_\epsilon(b) \leq E_{\epsilon_a}(b) < E_{\epsilon_a}(a) = E(a).$$

Therefore E is monotonically decreasing. Finally since for all $\epsilon > 0$

$$\|(f * \varphi_\sigma) - f\|_\infty \leq E_\epsilon(1/\sigma).$$

Taking minimum over $\epsilon \in A$ on both sides we obtain

$$\|(f * \varphi_\sigma) - f\|_\infty \leq E(1/\sigma).$$

□

Lemma 6. Let f be a C^1 function on a compact domain and let Z be a random variable with mean μ and variance σ^2 . Then

$$\mathbb{E}_Z[(f(\mu) - f(Z))^2] \leq L^2\sigma^2$$

where L denotes the Lipschitz constant of f .

Proof. Since f is a C^1 function on a compact domain it is Lipschitz with some Lipschitz constant L . Therefore

$$\mathbb{E}_Z[(f(\mu) - f(Z))^2] \leq L^2 \mathbb{E}_Z[(\mu - Z)^2] \leq L^2\sigma^2$$

□

Theorem 3. Fix t , x_t and y . Then under Assumptions 3, 4 and 5, there exists a function $E : \mathbb{R} \rightarrow \mathbb{R}$ which is monotonically decreasing to zero, such that

$$\begin{aligned} \mathbb{E}_{y_t \sim p(y_t|y)} [\|\nabla_{x_t} \ln p(x_t|y_t) - \nabla_{x_t} \ln p(x_t|y)\|_2^2] \\ \leq E(1/\sigma^y(t)). \end{aligned}$$

Proof. For this proof, we drop our convention of denoting the probability distribution of a random variable via the name of its density's argument.

$$\begin{aligned} & \|\nabla_{x_t} \ln p_{X_t|Y_t}(x_t|y_t) - \nabla_{x_t} \ln p_{X_t|Y}(x_t|y)\|_2^2 \\ &= \sum_{i=1}^{n_x} (\partial_{x_t}^i \ln p_{X_t|Y_t}(x_t|y_t) - \partial_{x_t}^i \ln p_{X_t|Y}(x_t|y))^2 \end{aligned}$$

Therefore it is sufficient to prove the theorem in each dimension separately. Hence, without loss of generality, we may assume that $x_t \in \mathbb{R}$ and show

$$\begin{aligned} \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p_{X_t|Y_t}(x_t|y_t) - \partial_{x_t} \ln p_{X_t|Y}(x_t|y))^2] \\ \leq E(1/\sigma^y(t)). \end{aligned}$$

By Bayes's rule we have

$$\begin{aligned} \partial_{x_t} \ln p_{X_t|Y_t}(x_t|y_t) &= \partial_{x_t} \ln p_{Y_t|X_t}(y_t|x_t) + \partial_{x_t} \ln p_{X_t}(x_t) \\ \partial_{x_t} \ln p_{X_t|Y}(x_t|y) &= \partial_{x_t} \ln p_{Y|X_t}(y|x_t) + \partial_{x_t} \ln p_{X_t}(x_t). \end{aligned}$$

Therefore,

$$\begin{aligned} & (\partial_{x_t} \ln p_{X_t|Y_t}(x_t|y_t) - \partial_{x_t} \ln p_{X_t|Y}(x_t|y))^2 \\ &= (\partial_{x_t} \ln p_{Y_t|X_t}(y_t|x_t) - \partial_{x_t} \ln p_{Y|X_t}(y|x_t))^2. \end{aligned}$$

To unclutter the notation, let $p(y|x) := p_{Y|X_t}(y|x)$ and $p_\sigma(y|x) := p_{Y_t|X_t}(y|x)$. Applying this notation:

$$\begin{aligned} \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p_{Y_t|X_t}(y_t|x_t) - \partial_{x_t} \ln p_{Y|X_t}(y|x_t))^2] \\ = \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p_\sigma(y_t|x_t) - \partial_{x_t} \ln p(y|x_t))^2] \end{aligned}$$

Adding and subtracting $\partial_{x_t} \ln p(y_t|x_t)$ and using the triangle inequality:

$$\begin{aligned} & \leq \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p_\sigma(y_t|x_t) - \partial_{x_t} \ln p(y_t|x_t))^2] \\ &+ \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p(y_t|x_t) - \partial_{x_t} \ln p(y|x_t))^2] \end{aligned}$$

We may bound the expectation by the supremum norm

$$\begin{aligned} & \leq \|\partial_{x_t} \ln p_\sigma(\cdot|x_t) - \partial_{x_t} \ln p(\cdot|x_t)\|_\infty^2 \\ &+ \mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p(y_t|x_t) - \partial_{x_t} \ln p(y|x_t))^2] \end{aligned}$$

We will bound each of the summands separately. Firstly, by Assumption 3 $(y_t, x_t) \rightarrow p(y_t|x_t)$ is C^2 and therefore $(y_t, x_t) \rightarrow \partial_{x_t} p(y_t|x_t)$ is C^1 . Moreover, since \mathcal{X} is compact, $y_t \rightarrow \partial_{x_t} p(y_t|x_t)$ is Lipschitz for some Lipschitz constant L . Therefore, by Lemma 6,

$$\mathbb{E}_{y_t \sim p(y_t|y)} [(\partial_{x_t} \ln p(y_t|x_t) - \partial_{x_t} \ln p(y|x_t))^2] \leq L^2 \sigma^y(t)^2.$$

To finish the proof, we need to bound

$$\|\partial_{x_t} \ln p_\sigma(\cdot|x_t) - \partial_{x_t} \ln p(\cdot|x_t)\|_\infty^2$$

First, we apply the chain rule

$$\begin{aligned} & \|\partial_{x_t} \ln p_\sigma(\cdot|x_t) - \partial_{x_t} \ln p(\cdot|x_t)\|_\infty^2 \\ &= \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t)}{p_\sigma(\cdot|x_t)} - \frac{\partial_{x_t} p(\cdot|x_t)}{p(\cdot|x_t)} \right\|_\infty^2 \end{aligned}$$

Adding and subtracting $\frac{\partial_{x_t} p_\sigma(\cdot|x_t)}{p(\cdot|x_t)}$:

$$\begin{aligned} & \leq \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t)}{p_\sigma(\cdot|x_t)} - \frac{\partial_{x_t} p_\sigma(\cdot|x_t)}{p(\cdot|x_t)} \right\|_\infty^2 \\ &+ \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t)}{p(\cdot|x_t)} - \frac{\partial_{x_t} p(\cdot|x_t)}{p(\cdot|x_t)} \right\|_\infty^2 \\ &= \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t)[p(\cdot|x_t) - p_\sigma(\cdot|x_t)]}{p_\sigma(\cdot|x_t)p(\cdot|x_t)} \right\|_\infty^2 \\ &+ \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t) - \partial_{x_t} p(\cdot|x_t)}{p(\cdot|x_t)} \right\|_\infty^2 \end{aligned}$$

By assumption 3 and 5 we have that $\partial_{x_t} p_\sigma(\cdot|x_t)$, $p_\sigma(\cdot|x_t)$ and $p(\cdot|x_t)$ are continuous functions on a compact domain. Therefore, $\partial_{x_t} p_\sigma(\cdot|x_t)$ is bounded from above by some constant M . Moreover, by adding assumption 4 we obtain that $p_\sigma(\cdot|x_t)$ and $p(\cdot|x_t)$ are bounded from below by some $\epsilon > 0$. Therefore

$$\begin{aligned} & \leq \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t)[p(\cdot|x_t) - p_\sigma(\cdot|x_t)]}{p_\sigma(\cdot|x_t)p(\cdot|x_t)} \right\|_\infty^2 \\ &+ \left\| \frac{\partial_{x_t} p_\sigma(\cdot|x_t) - \partial_{x_t} p(\cdot|x_t)}{p(\cdot|x_t)} \right\|_\infty^2 \\ &\leq \frac{M}{\epsilon^2} \|p(\cdot|x_t) - p_\sigma(\cdot|x_t)\|_\infty^2 \\ &+ \frac{1}{\epsilon} \|\partial_{x_t} p_\sigma(\cdot|x_t) - \partial_{x_t} p(\cdot|x_t)\|_\infty^2 \end{aligned}$$

Now by Lemma 5 and 4

$$\leq \frac{M}{\epsilon^2} E_1(1/\sigma^y(t)^2) + \frac{1}{\epsilon} E_2(1/\sigma^y(t)^2)$$

where E_1 and E_2 are monotonically decreasing to zero. The theorem follows with $E(1/\sigma^y(t)^2) := \frac{M}{\epsilon^2} E_1(1/\sigma^y(t)^2) + \frac{1}{\epsilon} E_2(1/\sigma^y(t)^2) + L^2 \sigma^y(t)^2$, which monotonically decreases to zero as $\sigma^y(t)^2$ decreases to zero. \square

B. Architectures and hyperparameters

We used almost the same neural network architecture across all tasks and all estimators, so that we can compare the estimators fairly. The only difference between the score model for the diffusive estimators and the score model for the CDE estimator is that the former contains 6 instead 3 filters in the final convolution to account for the joint score estimation. This difference in the final convolution leads to

negligible difference in the number of parameters, which is highly unlikely to have impacted the final performance.

We used the basic version of the DDPM architecture with the following hyperparameters: channel dimension 96, depth multipliers [1, 1, 2, 2, 3, 3], 2 ResNet Blocks per scale and attention in the final 3 scales. The total parameter count is 43.5M. Song et al. [23] report improved performance with the NCSN++ architecture over the baseline DDPM when training with the VE SDE. This claim is also supported by the work of Saharia et al. [19]. Therefore, adopting this architecture is likely to improve the performance of all estimators and lead to even more competitive performance over state-of-the-art methods. For all estimators, we concatenate the condition image y or $y(t)$ with the diffused target $x(t)$ and pass the concatenated image as input to the score model for score calculation. In the super-resolution experiment, we first interpolate the condition to the same resolution as the target using nearest neighbours interpolation and then concatenate it with the target image.

We used exponential moving average (EMA) with rate 0.999 and the same optimizer settings as in [23]. Moreover, we used a batch size of 50 for the super-resolution and edge to image translation experiments and a batch size of 100 for the inpainting experiments.

C. Extended visual results

We provide additional samples in Figures 7, 8 and 9.

D. Potential negative impact

The potential of negative impact of this work is the same as that of any work that advances generative modeling. Generative modeling can be used for the creation of deep-fakes which can be used for malicious purposes such as disinformation and blackmailing. However, research on generative modeling can indirectly or directly contribute to the robustification of deep-fake detection algorithms. Moreover, generative models have proven very useful in academic research and in industry. The potential benefits of generative modeling outweigh the potential threats. Therefore, the research community should continue to conduct research on generative modeling.



Figure 7. Extended super-resolution results.



Figure 8. Extended inpainting results.



Figure 9. Extended edge to shoe synthesis results.