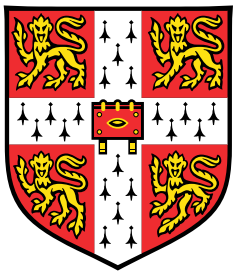
Improving Representation Learning through Generative Modeling

From Diffusion Models to Riemannian Geometry



Georgios Batzolis

Department of Applied Mathematics and Theoretical Physics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Churchill College October 2024

I would like to dedicate this thesis to my loving parents ...

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Georgios Batzolis
October 2024

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures **xiii**

List of tables **xv**

1 Conditional Image Generation with Score-Based Diffusion Models **1**
1.1 Introduction 1
1.2 Notation 3
1.3 Methods 4
1.3.1 Background: Score matching through Stochastic Differential Equations 4
1.3.2 Conditional generation 6
1.4 Experiments 11
1.4.1 Inpainting 13
1.4.2 Super-resolution 14
1.4.3 Edge to image translation 15
1.5 Comparison with state-of-the-art 15
1.6 Conclusions and future work 16
1.7 Acknowledgements 17

References **19**

Appendix A Conditional Image Generation with Score-Based Diffusion Models **21**
A.1 Proofs 21
A.1.1 Equality of minimizers for CDE 21
A.1.2 Consistency of CDE 23
A.1.3 Likelihood weighting for multi-speed and multi-sde models 25
A.1.4 Mean square approximation error 28
A.1.5 Architectures and hyperparameters 34

List of figures

1.1 Results from our conditional multi-speed diffusive estimator. 2
1.2 Sources of error for different estimators 9
1.3 Diversity of five different CMDE reconstructions for a given masked image. 12
1.4 Inpainting results. 14
1.5 Super-resolution results. 15
1.6 Edge to image translation results. 16

List of tables

1.1 Results of conditional generation tasks. 13

Chapter 1

Conditional Image Generation with Score-Based Diffusion Models

Score-based diffusion models have emerged as one of the most promising frameworks for deep generative modelling. In this work we conduct a systematic comparison and theoretical analysis of different approaches to learning conditional probability distributions with score-based diffusion models. In particular, we prove results which provide a theoretical justification for one of the most successful estimators of the conditional score. Moreover, we introduce a multi-speed diffusion framework, which leads to a new estimator for the conditional score, performing on par with previous state-of-the-art approaches. Our theoretical and experimental findings are accompanied by an open source library `MSDiff` which allows for application and further research of multi-speed diffusion models.

1.1 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 [22], [16], [14], [5], [34], [27]. In recent years much progress has been made in unconditional and conditional image generation. The most prominent approaches

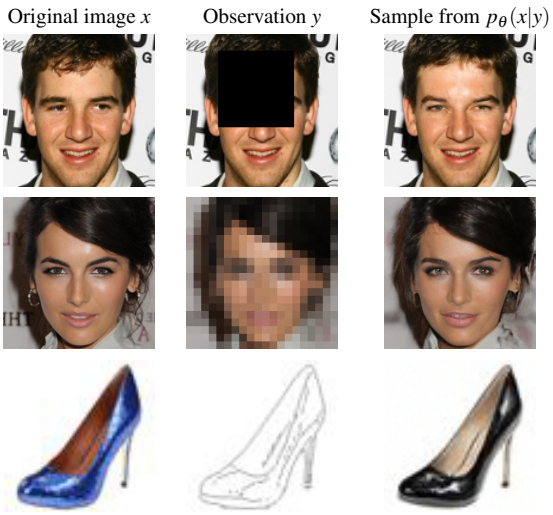


Fig. 1.1 Results from our conditional multi-speed diffusive estimator.

are auto-regressive models [3], variational auto-encoders (VAEs) [12], normalizing flows [21] and generative adversarial networks (GANs) [6].

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 require a lot of computational resources and 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 [9] and diffusion-based [23] generative models have been revived and improved in [25] and [8]. The connection between the two frameworks in discrete-time formulation has been discovered in [29]. Recently in [26], both frameworks have been unified into a single continuous-time approach based on stochastic differential equations [26] and called score-based diffusion models. These approaches have recently received a lot of attention, achieving state-of-the-art performance in likelihood estimation [26] 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, 26]. Moreover, their time complexity in high resolutions is much

better than that of auto-regressive models [4]. This makes score-based diffusion models very attractive for deep generative modelling.

In this work, we examine how score-based diffusion models can be applied to conditional image generation. 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 [22, 28] without justification), and we thereby provide a firm theoretical foundation for using it in future research.

Moreover, we extend the original framework to support *multi-speed diffusion*, where different parts of the input tensor diffuse according to different speeds. This allows us to introduce a novel estimator of the conditional score and opens an avenue for further research.
The contributions of this paper are as follows:

1. 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.

2. We provide a proof of consistency for the *conditional denoising estimator* - one of the most successful approaches to estimating the conditional score.

3. We introduce a multi-speed diffusion framework which 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 an open-source library `MSDiff`, to facilitate further research on conditional and multi-speed diffusion models. [1]

1.2 Notation

In this work we will use the following notation:

- **Functions of time**

$$f_t := f(t)$$

[1] The code will be released in the near future.

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

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

  cf. Section 1.3.2.

- **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**

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

1.3 Methods

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

1.3.1 Background: Score matching through Stochastic Differential Equations

Unconditional generation

In a recent work [26] score-based [9, 25] and diffusion-based [23, 8] generative models have been unified into a single continuous-time score-based framework with diffusion driven by stochastic differential equations. This continuous-time score-based diffusion technique relies on Anderson's Theorem [1], which states that (under certain assumptions on $\mu : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$ and $\sigma : \mathbb{R} \to \mathbb{R}$) a forward diffusion process

$$dx = \mu(x,t)dt + \sigma(t)dw \tag{1.1}$$

has a reverse diffusion process governed by the following SDE:

$$dx = [\mu(x,t) - \sigma(t)^2 \nabla_x \ln p_{X_t}(x)]dt + \sigma(t)d\bar{w}, \tag{1.2}$$

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

The forward diffusion process transforms the *target distribution* $p(x_0)$ to a *diffused distribution* $p(x_T)$. 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 $p(x_T)$ approximates a simple distribution, such as $\mathcal{N}(0,I)$. We refer to this simple distribution as the *prior distribution*, denoted by π.

If we have access to the score of the marginal distribution, $\nabla_{x_t} \ln p(x_t)$, for all t, we can derive the reverse diffusion process and simulate it to map p_T to p_0. In practice, we approximate the score of the time-dependent distribution by a neural network $s_\theta(x_t,t) \approx \nabla_{x_t} \ln p(x_t)$ and map the prior distribution $\pi \approx p(x_T)$ to $p_\theta(x) \approx p(x_0)$ by solving the reverse-time SDE from time T to time 0. One can integrate the reverse SDE using standard numerical SDE solvers such Euler–Maruyama or other discretisation strategies. The authors propose to couple the standard integration step with a fixed number of Langevin MCMC steps to leverage the knowledge of the score of the distribution at each intermediate timestep. The MCMC correction step improves sampling; the combined algorithm is known as a predictor-corrector scheme. We refer to [26] for details.

In order to fit a neural network model $s_\theta(x_t,t)$ to approximate the score $\nabla_{x_t} \ln p(x_t)$, we minimize the weighted Fisher's divergence

$$\mathscr{L}_{SM}(\theta) := \frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_t \sim p(x_t)}} [\lambda(t) \, \| \nabla_{x_t} \ln p(x_t) - s_\theta(x_t,t) \|_2^2] \tag{1.3}$$

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

The above quantity cannot be optimized directly since we don't have access to the ground truth score $\nabla_{x_t} \ln p(x_t)$. Therefore in practice, a different objective has to be used [9, 25, 26]. In [26], the continuous denoising score-matching objective is chosen, which is equal to $\mathscr{L}_{SM}(\theta)$ up to an additive term, which does not depend on θ and is defined as

$$\begin{aligned} &\mathscr{L}_{DSM}(\theta) := \\ &\frac{1}{2} \mathbb{E}_{\substack{t \sim U(0,T) \\ x_0 \sim p(x_0) \\ x_t \sim p(x_t|x_0)}} [\lambda(t) \, \| \nabla_{x_t} \ln p(x_t|x_0) - s_\theta(x_t,t) \|_2^2] \end{aligned} \tag{1.4}$$

The above expression involves only $\nabla_{x_t} \ln p(x_t|x_0)$ which can be computed analytically from the transition kernel of the forward diffusion process (provided that μ and σ are sufficiently simple). The expectation can be approximated using Monte Carlo estimation in the following way: First, we sample time points t_i from the uniform distribution on $[0,T]$, then we sample points $\tilde{x}_i$ from the target distribution p_0 (available via training set). Next, for each $\tilde{x}$, we sample points x_i from the transition kernel $p(x_t|\tilde{x})$. Finally, we average the expression inside the expectation over all samples obtained in this way.

1.3.2 Conditional generation

The continuous score-matching framework can be extended to conditional generation, as shown in [26]. 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 1.1) to obtain a family of diffused distributions $p(x_t|y)$ and apply Anderson's Theorem to derive the *conditional reverse-time SDE*

$$dx = [\mu(x,t) - \sigma(t)^2 \nabla_x \ln p_{X_t}(x|y)]dt + \sigma(t)d\tilde{w}. \tag{1.5}$$

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 [26] 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.

Conditional denoising estimator (CDE)

The conditional denoising estimator (CDE) is a way of estimating $p(x_t|y)$ using the denoising score matching approach [29, 25]. In order to approximate $p(x_t|y)$, the conditional denoising estimator minimizes

$$\frac{1}{2}\mathbb{E}_{\substack{t\sim U(0,T)\\ x_0,y\sim p(x_0,y)\\ x_t\sim p(x_t|x_0)}}[\lambda(t)\,\|\nabla_{x_t}\ln p(x_t|x_0) - s_\theta(x_t,y,t)\|_2^2] \tag{1.6}$$

This estimator has been shown to be successful in previous works [22, 28], also confirmed in our experimental findings (cf. Section 1.4).

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 [29], 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 1.3.1. *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.1.1. Using the above theorem, the consistency of the estimator can be established.

Corollary 1.3.2. *Let θ^* be a minimizer of a Monte Carlo approximation of (1.6), then (under technical assumptions, cf. Appendix A.1.2) 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.1.2.

Conditional diffusive estimator (CDiffE)

Conditional diffusive estimators (CDiffE) have first been suggested in [26]. 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 $\nabla_x \ln p(x)$ [25, 26], 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

$$\frac{1}{2}\mathbb{E}_{\substack{t\sim U(0,T)\\ z_0\sim p_0(z_0)\\ z_t\sim p(z_t|z_0)}}[\lambda(t)\,\|\nabla_{z_t} \ln p(z_t|z_0) - s_\theta(z_t,t)\|_2^2]. \tag{1.7}$$

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.

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.

Fig. 1.2 Sources of error for different estimators

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.

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).

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 1.2). This can lead to a better performance, as indicated by our experimental findings (cf. Section 1.4).

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

$$dx = \mu(x,t)dt + \sigma^x(t)dw$$
$$dy = \mu(y,t)dt + \sigma^y(t)dw.$$

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 $\lambda(t) : \mathbb{R} \to \mathbb{R}_+$ with a positive definite weighting matrix $\Lambda(t) : \mathbb{R} \to \mathbb{R}^{(n_x+n_y)\times(n_x+n_y)}$. Hence, the new training objective becomes

$$\frac{1}{2}\mathbb{E}_{\substack{t\sim U(0,T) \\ z_0\sim p_0(z_0) \\ z_t\sim p(z_t|z_0)}} [v^T \Lambda(t) v], \tag{1.8}$$

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

In [24] authors derive a likelihood weighting function $\lambda^{\mathrm{MLE}}(t)$, 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 $\Lambda^{\mathrm{MLE}}(t)$ with the same properties.

Theorem 1.3.3. *Let $\mathscr{L}(\theta)$ be the CMDE training objective (Equation 1.8) with the following weighting:*

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

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

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

The proof can be found in Appendix A.1.3.

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 1.3.4. *Fix t, x_t and y. Under mild technical assumptions (cf. Appendix A.1.4) there exists a function $E : \mathbb{R} \to \mathbb{R}$ monotonically decreasing to 0, such that*

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

The proof can be found in Appendix A.1.4.

Thus we see that the objective of CMDE approaches that of CDE as $\sigma^y(t) \to 0$, and CMDE coincides with CDiffE when $\sigma^y(t) = \sigma^x(t)$ (cf. Figure 1.2).

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 1.4.

`MSDiff`: Beyond multi-speed diffusion

Based on this work, we provide our open source library `MSDiff`, which generalizes the original framework of [26] and allows to diffuse x_t and y_t not only at different diffusion rates, but with two entirely different forward SDEs (i.e. with different diffusion coefficients *and* different drifts):

$$dx = \mu^x(x,t)dt + \sigma^x(t)dw$$
$$dy = \mu^y(y,t)dt + \sigma^y(t)dw$$

Moreover, the likelihood weighting of Theorem 1.3.3 holds in this more general case, allowing for principled training of multi-sde diffusion models (cf. Appendix A.1.3). This flexibility opens room for further research into multi-speed diffusion based training of score-based models, which we intend to examine in a future study.

1.4 Experiments

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. Moreover, we compare the most successful score-based diffusion approaches for super-resolution with HCFlow [13] – a state-of-the-art method in super-resolution.

Datasets In our experiments, we use the CelebA [15] and Edges2shoes [32, 10] datasets. We pre-processed the CelebA dataset as in [13].

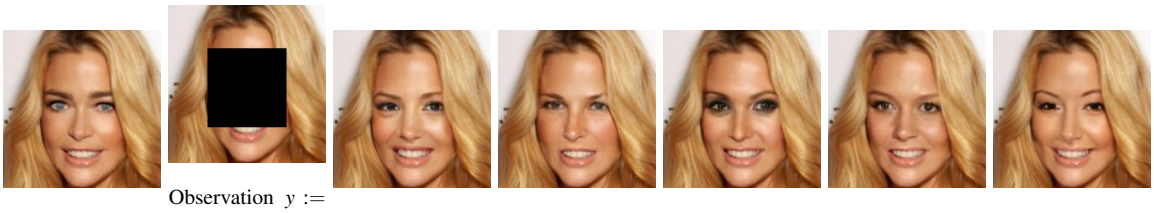


Fig. 1.3 Diversity of five different CMDE reconstructions for a given masked image.

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 [8, 26]. We used the variance-exploding SDE [26] given by:

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

Likelihood weighting was employed for all experiments. For CMDE, the diffusion speed of y was controlled by picking an appropriate $\sigma^y_{\max}$, which we found by trial-and-error. The performance of CMDE could be potentially improved by performing a systematic hyperparameter search for optimal $\sigma^y_{\max}$. Details on hyperparameters and architectures used in our experiments can be found in Appendix A.1.5.

Inverse problems The tasks of inpainting, super-resolution and edge to image translation are special cases of inverse problems [2, 18]. 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

Table 1.1 Results of conditional generation tasks.

	Estimator	PSNR/SSIM ↑	LPIPS ↓	UFID/JFID ↓	Consistency ↑	Di
Inpainting	CDE	**25.12/0.870**	**0.042**	13.07/18.06	**28.54**	
	CDiffE	23.07/0.844	0.057	13.28/19.25	26.61	
	CMDE ($\sigma^y_{max} = 1$)	24.92/0.864	0.044	**12.07/17.07**	28.32	
Super-resolution	CDE	23.80/0.650	0.114	10.36/15.77	54.18	
	CDiffE	23.83/0.656	0.139	14.29/20.20	51.90	
	CMDE ($\sigma^y_{max} = 0.5$)	23.91/0.654	0.109	**10.28/15.68**	53.03	
	HCFLOW	**24.95/0.702**	**0.107**	14.13/19.55	**55.31**	
Edge to image	CDE	**18.35/0.699**	**0.156**	**11.87/21.31**	**10.45**	
	CDiffE	10.00/0.365	0.350	33.41/55.22	7.78	
	CMDE ($\sigma^y_{max} = 1$)	18.16/0.692	0.158	12.62/22.09	10.38	

(PSNR) [30], Structural similarity index measure (SSIM) [30] and Learned Perceptual Image Patch Similarity (LPIPS) [33] 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) [7] 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 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 $p(x|y)$.

1.4.1 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

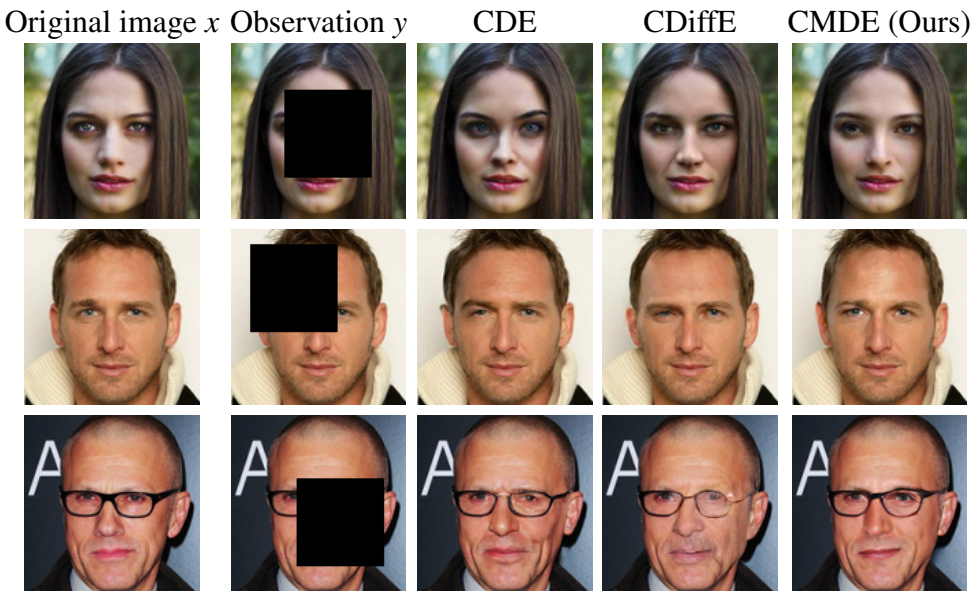


Fig. 1.4 Inpainting results.

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 1.1 and samples are presented in Figure 1.4. We observe 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.

1.4.2 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 [11] 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 1.1 and samples are presented in Figure 1.5. 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.

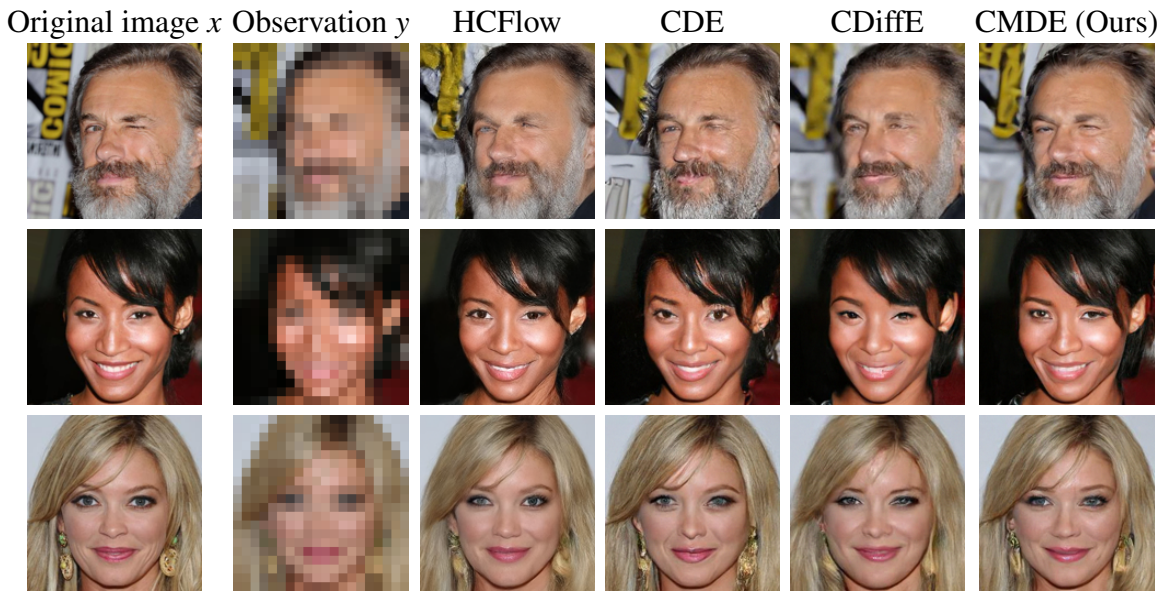


Fig. 1.5 Super-resolution results.

1.4.3 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 [31], 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 1.1 and samples are presented in Figure 1.6. 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 $\sigma^y(t)$.

1.5 Comparison with state-of-the-art

We compare score-based diffusion approaches with HCFlow [13] – a state-of-the-art method in super-resolution. To ensure a fair comparison, we used the data pre-processing and hyperparameters for HCFlow exactly as in the original paper [13]. We find that although HCFlow performs marginally better in terms of the reconstruction error, CDE and CMDE obtain a significantly better FID and diversity scores indicating better distribution coverage. We recall that a perfect reconstruction on a per-image basis is generally not desirable due to

Fig. 1.6 Edge to image translation results.

ill-posedness of the inverse problem and therefore in our view FID is the most principled of the used metrics. The FID scores suggest that CMDE was the most successful approach to approximating the posterior distribution.

1.6 Conclusions and future work

In this article, we conducted a systematic comparison of score-based diffusion models in conditional image generation tasks and provided an in-depth theoretical analysis of the estimators of conditional score. In particular, we proved the consistency of the conditional denoising estimator, thus providing a firm theoretical justification for using it in future research.

Moreover, we introduced a multi-speed diffusion framework, which led to CMDE, a novel estimator for conditional score which interpolates between conditional denoising estimator (CDE) and conditional diffusive estimator (CDiffE) by controlling the diffusion speed of the condition.

Our study showed 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.

We found that score-based diffusion models perform on par with prior state-of-the-art methods in super-resolution task and achieve better posterior approximation according to FID score.

1.7 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 Skodowska-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.

References

[1] Anderson, B. D. (1982). Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326.

[2] Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B. (2019). Solving inverse problems using data-driven models. *Acta Numerica*, 28:1–174.

[3] Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

[4] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis.

[5] Donahue, C., McAuley, J., and Puckette, M. (2019). Adversarial audio synthesis.

[6] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial networks.

[7] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2018). Gans trained by a two time-scale update rule converge to a local nash equilibrium.

[8] Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models.

[9] Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709.

[10] Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2018). Image-to-image translation with conditional adversarial networks.

[11] Keys, R. (1981). Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(6):1153–1160.

[12] Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes.

[13] Liang, J., Lugmayr, A., Zhang, K., Danelljan, M., Gool, L. V., and Timofte, R. (2021). Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling.

[14] Liu, G.-H., Vahdat, A., Huang, D.-A., Theodorou, E. A., Nie, W., and Anandkumar, A. (2023). I^2sb: Image-to-image schrödinger bridge.

[15] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.

[16] Lugmayr, A., Danelljan, M., Van Gool, L., and Timofte, R. (2020). Srflow: Learning the super-resolution space with normalizing flow. In *Computer Vision – ECCV 2020*.

[17] Léonard, C. (2013). Some properties of path measures.

[18] Mueller, J. L. and Siltanen, S. (2012). Linear and nonlinear inverse problems with practical applications. In *Computational science and engineering*.

[19] Newey, W. K. and McFadden, D. (1994). Chapter 36 large sample estimation and hypothesis testing. volume 4 of *Handbook of Econometrics*, pages 2111–2245. Elsevier.

[20] Oksendal, B. (2003). *Stochastic Differential Equations (5th Ed.): An Introduction with Applications*. Springer-Verlag, Heidelberg.

[21] Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference.

[22] Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D. J., and Norouzi, M. (2021). Image super-resolution via iterative refinement.

[23] Sohl-Dickstein, J., Weiss, E. A., Maheswaranathan, N., and Ganguli, S. (2015). Deep unsupervised learning using nonequilibrium thermodynamics.

[24] Song, Y., Durkan, C., Murray, I., and Ermon, S. (2021a). Maximum likelihood training of score-based diffusion models.

[25] Song, Y. and Ermon, S. (2020). Generative modeling by estimating gradients of the data distribution.

[26] Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. (2021b). Score-based generative modeling through stochastic differential equations.

[27] Tang, H., Liu, H., Xu, D., Torr, P. H. S., and Sebe, N. (2021). Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks.

[28] Tashiro, Y., Song, J., Song, Y., and Ermon, S. (2021). Csdi: Conditional score-based diffusion models for probabilistic time series imputation.

[29] Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674.

[30] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

[31] Xie, S. and Tu, Z. (2015). Holistically-nested edge detection.

[32] Yu, A. and Grauman, K. (2014). Fine-grained visual comparisons with local learning. In *Computer Vision and Pattern Recognition (CVPR)*.

[33] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

[34] Zhu, J.-Y., Park, T., Isola, P., and Efros, A. A. (2020). Unpaired image-to-image translation using cycle-consistent adversarial networks.

Appendix A

Conditional Image Generation with Score-Based Diffusion Models

A.1 Proofs

A.1.1 Equality of minimizers for CDE

Lemma A.1.1. *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 [29, Eq. 11]. □

Theorem 1. *The minimizer of*

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

in θ is the same as the minimizer of

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

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

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

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 A.1.1, it follows that

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

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

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

(1) Tower Law, (2) Conditional independence of x_t and y given x_0, (3) Lemma A.1.1. □

A.1.2 Consistency of CDE

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

Assumption A.1.3. The space of parameters Θ and the data space $\mathscr{X}$ are compact.

Assumption A.1.4. 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 A.1.5 (Uniform law of large numbers)**.** *[19, 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 $d : \mathscr{Z} \to \mathbb{R}$ such that $\mathbb{E}[d(z)] < \infty$ and $\|f(z,\theta)\| \le d(z)$ 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 A.1.6 (Consistency of extremum estimators)**.** *[19, Theorem 2.1]*
Let Θ be compact and consider a family of functions $\mathscr{L}^{(n)} : \Theta \to \mathbb{R}$. Moreover, suppose there exists a function $\mathscr{L} : \Theta \to \mathbb{R}$ such that

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

Then

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

Corollary A.1.7. *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 A.1.3 and A.1.4, 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

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

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 A.1.5, the Monte Carlo approximation of 1.6: $\mathscr{L}^{(n)}(\theta) = \frac{1}{n}\sum_{i=1}^n f(z_i,\theta)$ converges uniformly in probability to $\mathscr{L}(\theta) = \mathbb{E}_{z\sim q(z)}[f(z,\theta)]$. Let θ^* be the minimizer of $\mathscr{L}(\theta)$, by Lemma A.1.6 we get that $\theta_n^* \xrightarrow{P} \theta^*$. Finally by Theorem 1.3.1, 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 A.1.4 this implies that $s_{\theta^*}(x_t, y, t) = \nabla_{x_t} \ln p(x_t|y)$. Hence $s_{\theta_n^*}(x, y, t) \xrightarrow{P} \nabla_{x_t} \ln p(x_t|y)$. □

A.1.3 Likelihood weighting for multi-speed and multi-sde models

In this section we derive the likelihood weighting for multi-sde models (Theorem 1.3.3). First using the framework in [26, 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 [24] 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 1.3.3).

Consider an Ito's SDE

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

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

$$\begin{aligned} dx &= \tilde{\mu}(x,t)dt + \Sigma(t)dw \\ \text{where } \tilde{\mu}(x,t) &:= \mu(x,t) - \Sigma(t)^2 \nabla_x \ln p_{X_t}(x). \end{aligned} \tag{A.1}$$

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

$$\begin{aligned} dx &= \tilde{\mu}_\theta(x,t)dt + \Sigma(t)dw \\ \text{where } \tilde{\mu}_\theta(x,t) &:= \mu(x,t) - \Sigma(t)^2 s_\theta(x,t) \end{aligned} \tag{A.2}$$

Now we generalize [24, Theorem 1] to multi-dimensional setting.

Theorem A.1.8. *Let $p(x_t)$ and $p_\theta(x_t)$ denote marginal distributions of A.1 and A.2 respectively. Then under regularity assumptions of [24, Theorem 1] we have that*

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

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

Proof. We proceed in close analogy to the proof of [24, 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 [17, Theorem 2.4] (generalized chain rule for KL divergence) we have

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

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 [17, Theorem 2.4] again, we obtain

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

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

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

Using Girsanov Theorem [20, Theorem 8.6.5] and the fact that $\Sigma(t)$ is symmetric and invertible

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

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)

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

□

Multi-sde and multi-speed diffusion

Now we consider again the multi-speed and the more general multi-sde diffusion frameworks from Sections 1.3.2 and 1.3.2. 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 $z = (x,y)$, $\mu^z(z,t) = (\mu^x(x,t), \mu^y(x,t))$ and

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

If we train a score-based diffusion model for $z_t = (x_t, y_t)$, then by Theorem A.1.8

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

where $C_1 := KL(p(x_T)|\pi(x_T))$ does not depend on θ. Because Λ_{MLE} (from Theorem 1.3.3) 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 [29]

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

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

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

Therefore

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

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

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

where $C := C_3 - \mathbb{E}_{(x,y)\sim p(x,y)}[\ln p(x,y)]$ is independent of θ. Thus the Theorem 1.3.3 is established.

A.1.4 Mean square approximation error

Assumption A.1.9. $p(x,y) \in C^2(\mathscr{X})$

Assumption A.1.10. $p(x,y) > 0$ for all x, y.

Assumption A.1.11. The data space $\mathscr{X}$ is compact.

Lemma A.1.12. *Under assumptions A.1.9 and A.1.11 we have*

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

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 differentiaion and integration. Therefore

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

□

Lemma A.1.13. *Let f be a C^1-function on a compact domain $\mathscr{X}$ and let φ_σ be a Gaussian kernel with variance σ^2. Then there exists a function $E : \mathbb{R} \to \mathbb{R}$, 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 $\varepsilon > 0$, and let L denote the Lipschitz constant of f. We have that $|f(z) - f(y)| < \varepsilon$ whenever $\|z - y\| < \varepsilon/L$. Let $B_y(\varepsilon/L) := \{z \in \mathscr{X} : \|z - y\| < \varepsilon/L\}$ be a ball of radius ε/L around y. Then

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

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

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

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

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

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

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

Notice that the above minimum is achieved, since A is compact and for a fixed σ, the function $\varepsilon \mapsto E_\varepsilon(1/\sigma)$ 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 $E(x) \to 0$ as $x \to \infty$, since for all $\varepsilon \in A$ we have $\lim_{x\to\infty} E(x) \leq \lim_{x\to\infty} E_\varepsilon(x) = \varepsilon$. Secondly, suppose $a < b$, and let ε_a be such that $E(a) = E_{\varepsilon_a}(a)$. Then

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

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

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

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

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

□

Lemma A.1.14. *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 A.1.9, A.1.10 and A.1.11, there exists a function $E : \mathbb{R} \to \mathbb{R}$ 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.

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

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 A.1.9 $(y_t, x_t) \to p(y_t|x_t)$ is C^2 and therefore $(y_t, x_t) \to \partial_{x_t} p(y_t|x_t)$ is C^1. Moreover, since $\mathscr{X}$ is compact, $y_t \to \partial_{x_t} p(y_t|x_t)$ is Lipschitz for some Lipschitz constant L. Therefore, by Lemma A.1.14,

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

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

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

By assumption A.1.9 and A.1.11 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 A.1.10 we obtain that $p_\sigma(\cdot|x_t)$ and $p(\cdot|x_t)$ are

bounded from below by some $\varepsilon > 0$. Therefore

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

Now by Lemma A.1.13 and A.1.12

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

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

A.1.5 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. [26] 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. [22]. 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 [26]. 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.