Constrained Diffusion: Applications to Image Generation, Manifold Learning, and Motion Planning

by

Spencer Ryan Szabados

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2024

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

This thesis consists, in part, of the author's previous work:

- [64], [65]. These works characterize the sufficient and necessary conditions for both diffusion models and diffusion bridge models to preserve isometry group invaraiances present in data distributions. The theoretical proofs in the cited works are attributed to Haoye Lu, who built up my initial sketches into rigorous theorems. Writing of these works was done equally by Haoye and myself. Both, Haoye and I contributed a significant amount of time to model implementation and evaluation. Specifically, in terms of model development in [64], I primarily implemented: SPDM+WT and SPDM+REG having proposed the regularization method, see [65, Appx. C]; I fixed the FID metric evaluation of SP-GAN by [3]; and for [65] I implemented (DDBM) SPDM+WT[1] and all other model comparisons for Table 5.2 and Table 5.3 excluding (DDBM) SPDM+FA for the LYSTO and ANHIR datasets which were implemented by Haoye alone.

- In this thesis various theoretical results are presented; those that are not my work appear with the relevant citation either in the statement block or nearby. The primary theoretical contributions made in this work are the proof of Lemma 1, and both Corollary 1 and Corollary 3. Outside of these primary results, various minor contributions serve to link together the different sections and aid in motivating the use of select techniques; e.g., within Section 3.3 a brief study of the variance of the trace estimate is given to motivate the use of the Rademacher distribution for sampling, following which we introduce a stochastic scaler term that proved valuable in stabilizing model training. Empirical contributions are detailed at the end of each section. Lastly, much of the work around motion planning is build incrementally atop existing work, which are cited whenever an existing technique is introduced, barring some rederivation of results. The major contribution of this section is in the exploratory work combining reflected diffusion models with motion planning.

[1] This model was not presented in the final paper.

Abstract

This thesis delves into the theoretical foundations, extensions, and applications of diffusion modeling in generative tasks. Diffusion models have garnered significant attention due to their stability during training and superior performance compared to competing methods.

In an attempt to make this work approachable for those not already familiar with diffusion, we begin by developing diffusion models from the ground up, starting with continuous diffusion processes and later deriving popular discrete diffusion models via discretization, providing insights into their mechanics. Motivated by work in the physical sciences, where datasets reside on curved surfaces, we describe extensions to Riemannian manifolds by redefining Brownian motion in these domains and formulating stochastic differential equations that describe continuous diffusion processes on manifolds. In much the same vein, as many real-world datasets are constrained within specific boundaries, we explore reflected diffusion processes. These processes describe diffusion processes that are constrained to a bounded region without absorption at the boundary, ensuring that generated data remains within a desired support. At the end of each of these chapters, we address the numerous practical challenges in training neural diffusion models on these different processes, as well as developing a few techniques that improve training stability of such models.

Further, we investigate structure-preserving diffusion models that respect inherent symmetries present in data, such as rotational invariance in imaging applications. We provide a complete characterization on the form of drift and diffusion terms required to ensure the diffusion processes, and diffusion model, accurately preserve affine group invariances present within target distributions. Three core techniques are discussed for achieving such group invaraince, with each being evaluated over a set of datasets focused on applications in Medical imaging. In closing out this section, we discuss in detail extensions of this work to reflected diffusion processes and Riemann manifolds.

Finally, we highlight some proof-of-concept work on applying reflected diffusion models to the domain of robotic motion planning. Focusing on generating collision-free paths for robot navigation and multi-segment robotic arms, we demonstrate how diffusion models can address the complexities inherent in planning under motion constraints. This application showcases the practical utility of the extended diffusion modeling framework in solving real-world problems.

Acknowledgements

I would first like to thank my supervisor, Yaoliang Yu, for his excellent mentorship and wealth of knowledge; and, Haoye Lu, who proved to be a great collaborator and mentor in his own right. My graduate experience has been all the better due to you both. Secondly, I want to extend my thanks to my examining committee members, Justin Wan, and, Victor Zhong, for their time and feedback. Lastly, I would like to thank both, Stephane Durocher, and, Alexandre Leblanc, for motivating me to apply to graduate studies, an experience I surely would have missed.

Dedication

To my family and friends.

Table of Contents

Chapter 1

Introduction

The central topic of this thesis is diffusion modeling, its definition, constrains, and application to select generative modeling tasks. Considerable effort has been put into structuring this work to include sufficient background on the underlying dynamics that govern diffusion models so that a reader who follows the chapter listings should be able to comprehend each section without much preliminary knowledge.

In the last several years, diffusion models [91, 90, 33, 48, 102], and diffusion bridge models [16], have gained significant interest due to their relative ease of training, in comparison to other contemporary models such as generative adversarial networks (GANs) [27] – that commonly suffer mode collapse while offering state-of-the-art performance. Diffusion models also benefit from a rich mathematical legacy within statistical physics that characterize the dynamics underlying various diffusion models that are commonly used in practice, making them easier to understand and design. Thus, diffusion based methods have shown to be attractive to not just practitioners but also mathematically minded individuals in machine learning.

We begin in Chapter 2 by describing the fundamental mechanics on which diffusion models operate, which is necessary for understanding some of the results given in Section 5.1, and discuss how neural networks can be parametrized and trained using these dynamics. Chapter 4 contains discussion around a constrained version of diffusion that offers several theoretical advantages for applications seeking to use diffusion for optimization tasks, which are touched on in Chapter 6. Lastly, the Appendix contains many of the proof contributions, additional experimental results, and qualitative sample comparison across various empirical experiments in the earlier chapters. Much of the mathematical formalisms and definitions that are used throughout the text are relegated to Appendix A and should be consulted if the reader is unsure of any unfamiliar notations.

To assist the reader in navigating this thesis and to allow them to allocate their attention to chapters more inline with their interests, a chapter dependency graph is provided in Fig. 1.1. This map illustrates the topic dependence between chapters and sections.

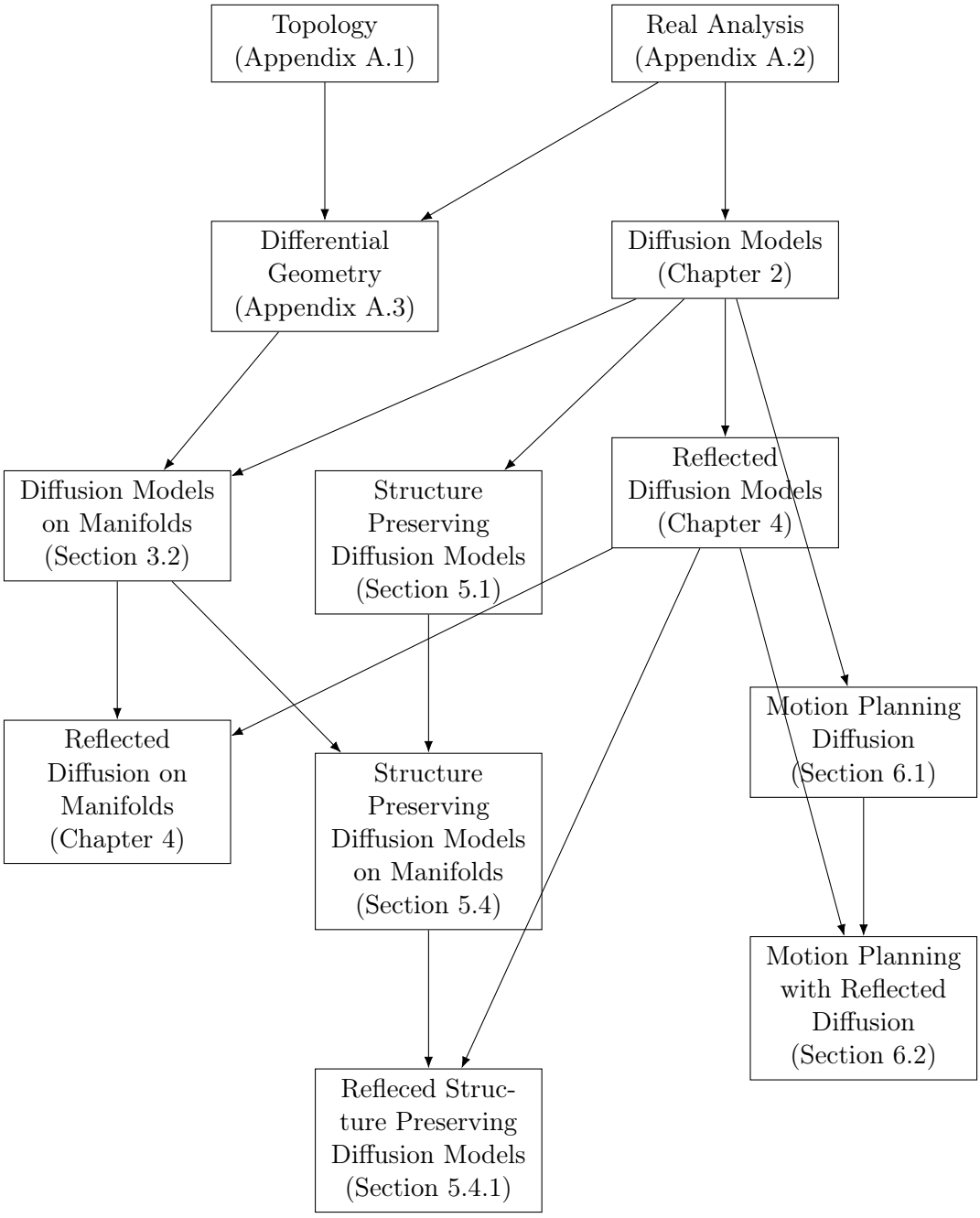


Figure 1.1: Chapter Dependency Graph

Chapter 2

Diffusion models

Diffusion based (generative) models have emerged as a highly capable neural framework for image synthesis [33, 34, 50, 74, 90, 48, 83], audio generation, robotic motion planning [43, 8], molecule conformation [12, 35, 46, 81, 101, 102], which are stable to train in comparison to competing methods, notably GANs [27], at the expense of training times. Informally, diffusion models follow a noisy process where input data is gradually corrupted and the model learns to reconstruct the original data from the noisy counterparts.

In the following section we will develop diffusion models from a (somewhat) bottom up fashion, starting from the theoretical underpinnings of continuous diffusion models and then recovering original discrete step diffusion models via discretization; this being what I feel to be a more natural derivation albeit one not commonly shown. First and foremost, we begin this chapter by formalizing the noise used within diffusion processes.

2.1 Brownian motion

The concept of Brownian motion, also called a Wiener process – the result of considering a continuous random walk, must be understood to properly characterize the dynamics governing diffusion models. Consequently, we lead by formally defining Brownian motion over Euclidean space.

Definition 1 (Wiener process)**.** *Let $(\Omega, \mathcal{B}, P)$ be a probability space. A (one-dimensional) Wiener process (or Brownian motion) is a stochastic process $\{B_t\}_{t\geq 0}$, for $t \in \mathbb{R}_{\geq 0}$, that satisfy:*

1. *$B_0 = 0$;*
2. *Almost surely $t \to B_t$ is continuous in t; i.e., the Wiener process is a continuous stochastic process where for all $t \geq 0$*

$$P(\{z \in \Omega \mid \lim_{s \to t} |B_s(z) - B_t(z)|\} = 0) = 1;$$

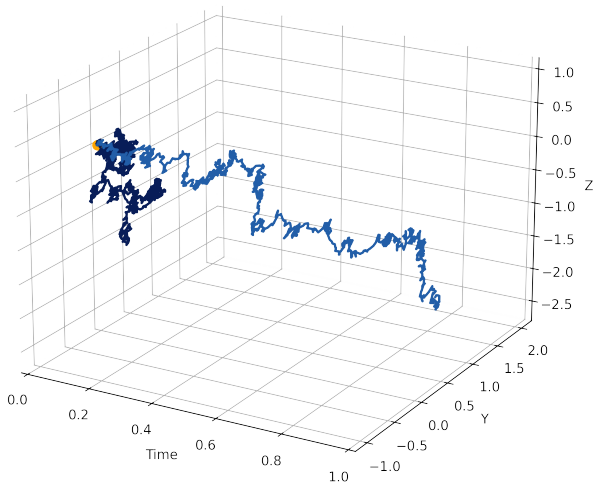


Figure 2.1: An illustration of 1000 steps of random walk originating from an orange point in the plane (black) and this same random walk viewed as a time series trajectory (blue).

3. *The process $\{B_t\}_t$ has stationary (fixed) independent increments; with the increment constraint $B_{t+s} - B_s \sim \mathcal{N}(0, \sigma t)$, with diffusion coefficient σ. (We assume $\sigma = 1$ throughout the remainder of the text.)*

While the above axiomatic definition of Brownian motion is constructive and easy to work with, it may not be clear (as it wasn't when historically introduced – people questioned if such a model was justifiable in reality) that any processes exists that realizes these conditions. As it turns out, such a process originates as the solution to the (stochastic) heat equation.

For those readers not familiar with the effects of Brownian motion, a trajectory of a particle in $\mathbb{R}^2$ evolving through time under Brownian motion is simulated for 1000 discrete steps and visualized in Fig. 2.1. As can be seen from the figure, trajectories following Brownian motion are very rough, in fact such trajectories are nowhere differentiable, and as such are most frequently studied using probabilistic concentration arguments.

Background on the heat kernel. From [28], recall the heat kernel in $\mathbb{R}^d$ is the (unique) positive solution of, $u : [0, T] \times \mathbb{R}^d \to \mathbb{R}$, of the time varying Cauchy problem

$$\begin{cases} \frac{\partial u(t,x)}{\partial t} = \nabla \cdot \nabla u(t, x) \\ u(0, x) = \delta(x - x_0), \end{cases}$$

with $x_0 \in \mathbb{R}^d$ being the initial starting point, and $\delta(x - x_0)$ is a Dirac density centered at x_0. Under this probabilistic view, we can derive the solution to this problem of the form

$$u(t, x) = \int_{\mathbb{R}^d} p(t, x, y)\delta(x - y)\,\mathrm{d}y,$$

in particular, in the Euclidean setting, it is know the Gaussian transition density

$$p(t, x, y) = \frac{1}{(4\pi t)^{d/2}} \exp\left\{ - \frac{\|x - y\|_2^2}{4t} \right\}$$

satisfies this problem.

Brownian motion and SDEs The most fundamental diffusion processes, is that based on the heat equation, which in Euclidean space takes the form of a Stochastic Differential Equation (SDE). Specifically, consider a sequence $(\overrightarrow{X}_t)_{t\geq 0}^T$ of time-indexed random variables in $\mathbb{R}^d$ with $\overrightarrow{X}_0 \sim p_0(x) = \delta(x - x_0)$ defined up till some time $T > 0$. The Fokker-Planck equation (forward Kolmogorov equation) [49, 79, 76, 87] describes how the probability distribution of $\overrightarrow{X}_t$, denoted p_t, evolves from p_0 under Brownian motion, in this case with covariance (time paramteizer) $\Sigma_t : [0, \infty) \to \mathbb{R}^{d\times d}$, as

$$\frac{\partial}{\partial t} p_t(\overrightarrow{x_t}) = \sum_{i,j}^{d} \frac{\partial^2}{\partial x_i \partial x_j} \left[\frac{[\Sigma_t \Sigma_t^\intercal]_{i,j}}{2} p_t(\overrightarrow{x_t}) \right]$$

with the dynamics of $\overrightarrow{X_t}$ expressed as the (Ito) SDE

$$\mathrm{d}\overrightarrow{X}_t = \Sigma_t \,\mathrm{d}\overrightarrow{B}_t,$$

where $(\overrightarrow{B_t})_{t\geq 0}$ is a d dimensional Brownian motion resulting from the heat transition kernel, that describes the dynamics of the random variables in space. Corresponding to this is a time reverse (backward) SDE, stated in terms of $\overleftarrow{X}_t = \overrightarrow{X}_{T-t}$,

$$\mathrm{d}\overleftarrow{X}_t = -\frac{1}{2}\Sigma_t \Sigma_t^\intercal \nabla_x \log p_t(\overleftarrow{X}_t)\,\mathrm{d}t + \Sigma_t \,\mathrm{d}\overleftarrow{B}_t$$

that transports $\overrightarrow{X}_T$ backwards in time to $\overrightarrow{X}_0$ (in a point-wise sense).

2.2 Diffusion processes

To begin, we will introduce (continuous) diffusion models, also referred to as score based generative models - the only distinction between these naming conventions being the chosen model paramterization outlined in Section 2.3, defined over Euclidean space following the works of [33, 90, 92, 48, 54]. To expedite the delivery, we will only detail the unconditional diffusion setting; the conditional setting is not a difficult extension and will be used without explicit introduction later on in the text.

Continuous diffusion processes The design and application of diffusion neural models is based on the central assumption[1] that for a given (ground truth) distribution p_0, with mean μ_0 and standard deviation Σ_0, if we progressively add Gaussian noise $\epsilon \sim \mathcal{N}(\mu_t, \Sigma_t)$, for $t = 1, \ldots, T$, with $\Sigma_T \succ\succ \Sigma_0$[2] the resulting mollified distribution p_T is approximately normal; i.e., $p_T(x) \approx \mathcal{N}(x; \mu_T, \Sigma_T)$.

This idea of gradually corrupting data can be formalized in terms of SDEs, as introduced above in Section 2.1. In particular, let $(\overrightarrow{X}_t)_{t\geq 0}^T$ denote a sequence of time-indexed random variables in $\mathbb{R}^d$ such that $X_t \sim p_t$, where p_t is the marginal distribution induced by an underlying SDE of the form

$$\mathrm{d}\overrightarrow{X}_t = \mu(\overrightarrow{X}_t, t)\,\mathrm{d}t + \Sigma(\overrightarrow{X}_t, t)\,\mathrm{d}\overrightarrow{B}_t,$$

where $\mu : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$ and $\Sigma : \mathbb{R}^d \times [0, T] \to \mathbb{R}^{d\times d}$ are prescribed drift and diffusion coefficients, commonly used value are discussed in [92, 48]. As outlined in Section 2.1, the Fokker-Planck equation (forward Kolmogorov equation) describes how the probability distribution of X_t evolves starting from p_0 through the dynamics described by a general SDE of the forgoing form, with

$$\frac{\partial}{\partial t}p_t(x_t) = -\sum_{i=1}^{d}\frac{\partial}{\partial x_i}[\mu_i(x_t, t)p_t(x_t)] + \sum_{i,j}^{d}\frac{\partial^2}{\partial x_i \partial x_j}\left[\frac{[\Sigma(x_t, t)\Sigma(x_t, t)^\intercal]_{i,j}}{2}p_t(x_t)\right], \tag{2.1}$$

an approximation, as written, that is accurate up to the first two modes. Correspondingly there exists a unique family of probability transition kernels $p(\overrightarrow{X}_t|\overrightarrow{X}_s)$, for $0 \leq t < s \leq T$;

[1] I use the term "assumption" here since often experimentally this property is not checked and the time scale of the diffusion process is heuristically determined.

[2] The matrix inequality expresses that $\Sigma_T - \Sigma_0$ is positive definite with a large positive lower bound. Where "large" here is dependent on the covariance of the data distribution, which is often experimentally estimated.

there is a special case when μ and Σ are affine (e.g., scalar matrices) where the transition kernels remain always Gaussian and can be derived in closed form due to (2.1) becoming exact. (Derivation of this is postponed till Section 2.2, likewise, for the ensuing reverse direction.)

Provided the initial, ground truth, distribution p_0 is sufficiently conditioned [1], there exists a time-reversed (also called backward) process from $t = T$ to $t = 0$ given as

$$\begin{aligned} \mathrm{d}\overleftarrow{X}_t = \Big[\mu(\overleftarrow{X}_t, t) - \frac{1}{2}\nabla \cdot [\Sigma(\overleftarrow{X}_t, t)\Sigma(\overleftarrow{X}_t, t)^\intercal] \\ - \frac{1}{2}\Sigma(\overleftarrow{X}_t, t)\Sigma(\overleftarrow{X}_t, t)^\intercal \nabla_x \log p_t(\overleftarrow{X}_t)\Big]\,\mathrm{d}t + \Sigma(\overleftarrow{X}_t, t)\,\mathrm{d}\overleftarrow{B}_t. \end{aligned} \tag{2.2}$$

where $\nabla \cdot \Sigma(X_t, t) = (\nabla \cdot g_1(X_t, t), \ldots, \nabla \cdot g_d(X_t, t))$ with $\nabla \cdot g_i$ denoting the divergence of $g_i : \mathbb{R}^d \times [0, T] \to \mathbb{R}^d$. As above, this backwards SDE characterizes how the probability distribution p_T, which recall we take $p_T \sim \mathcal{N}(0, \Sigma_T)$, is transported towards p_0; with

$$\frac{\partial}{\partial x} p_t(\overleftarrow{x_t}) = -\mu(\overleftarrow{x_t}, t)\frac{\partial}{\partial x} p_t \overleftarrow{x_t}) - \frac{\Sigma(\overleftarrow{x_t}, t)\Sigma(\overleftarrow{x_t}, t)^\intercal}{2} \sum_{i,j}^{d} \frac{\partial^2}{\partial x_i, \partial x_j} p_t(\overleftarrow{x_t}).$$

For the remainder of this text, whenever we refer to a "diffusion process," we mean a stochastic process that can be described via the above dynamics.

Affine continuous diffusion processes Within the aforementioned affine case, which is the setting used in practice [48], we take $\mu(X_t, t) = A_t X_t + b_t$ for $A_t \in \mathbb{R}^{d\times d}$, $b_t \in \mathbb{R}^d$, $\Sigma(X_t, t) = \sigma_t \sigma_t^\intercal$ with $\sigma_t \in \mathbb{R}^d$, the (forward) transition kernels, $t \geq s$, via [87, Theorem 5.10], are of the following general form:

$$\begin{aligned} \frac{\partial}{\partial t} p(\overrightarrow{X}_t | \overrightarrow{X}_s) &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}[\mu_i(\overrightarrow{X}_t, t) p_t(\overrightarrow{X}_t | \overrightarrow{X}_s)] \\ &\quad + \sum_{i,j}^{d} \frac{\partial^2}{\partial x_i \partial x_j}\left[\frac{[\Sigma(\overrightarrow{X}_t, t)\Sigma(\overrightarrow{X}_t, t)^\intercal]_{i,j}}{2} p_t(\overrightarrow{X}_t | \overrightarrow{X}_s)\right], \\ &= -\sum_{i=1}^{d} \frac{\partial}{\partial x_i}([A_t \overrightarrow{X}_t + b_t] p_t(\overrightarrow{X}_t | \overrightarrow{X}_s)) \\ &\quad + \sum_{i,j}^{d} \frac{[\Sigma_t \Sigma_t^\intercal]_{i,j}}{2} \frac{\partial^2}{\partial x_i \partial x_j}\left[p_t(\overrightarrow{X}_t | \overrightarrow{X}_s)\right]. \end{aligned}$$

Making use of the affine assumptions, using [87, Eq. 5.50, Eq. 5.51], we can compute the mean, $\eta_t = \mathbb{E}[\overrightarrow{X}_t]$, and covariance $\mathrm{H}_t = \mathbb{E}[(\overrightarrow{X}_t - \eta_t)(\overrightarrow{X}_t - \eta_t)^\intercal]$ of $p(X_t|X_s)$ to be respectively

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\eta_t &= \mathbb{E}[\mu(\eta_t, t)] \\ &= A_t\eta_t + b_t,\end{aligned}$$

and

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\mathrm{H}_t &= \mathbb{E}[\mu(\overrightarrow{X}_t, t)(\overrightarrow{X}_t - \eta_t)^\intercal] + \mathbb{E}[(\overrightarrow{X}_t - \eta_t)\mu(\overrightarrow{X}_t, t)^\intercal] + \mathbb{E}[\Sigma_t\Sigma_t^\intercal] \\ &= A_t\mathrm{H}_t + \mathrm{H}_tA_t^\intercal + \Sigma_t\Sigma_t^\intercal.\end{aligned}$$

Then given the initial conditions $\eta_s = \overrightarrow{X}_s$ and $\mathrm{H}_s = 0$, we can derive the general conditional solutions to these equations, namely

$$\eta_{t|s} = \Phi_{t,s}\overrightarrow{X}_s + \int_s^t \Phi_{t,\tau}b_\tau \,\mathrm{d}\tau$$

and

$$\mathrm{H}_{t|s} = \int_s^t \Phi_{t,\tau}\Sigma_\tau\Sigma_\tau^\intercal\Phi_{t,\tau}^\intercal \,\mathrm{d}\tau,$$

where $\Phi_{t,s}$ is the (continuous) Markov transition matrix (See [87] for characterization), which in general has no closed form and even for the affine case will depend on the form of A_t. Now, this solution is an affine transformation of Brownian motion (Gaussian process) it must follow a Gaussian distribution itself, namely;

$$p(\overrightarrow{X}_t|\overrightarrow{X}_s) = \mathcal{N}(\overrightarrow{X}_t; \eta_{t|s}, \mathrm{H}_{t|s}).$$

A natural question to then ask, the answer of which would simplify numerical simulations, is under what values of μ_t, and Σ_t does a closed form solution exist? As it turns out, if A_t, b_t, and Σ_t are smooth (bounded) scalar functions, i.e., $A_t = a_t : [0,T] \to \mathbb{R}$, $b_t = 0$, and $\Sigma_t = \sigma_t : [0,T] \to \mathbb{R}_{\geq 0}$, then we can solve the transition matrix in closed form using the method of integrating factors; in particular, for the drift and diffusion terms $a(t), \sigma(t) : [0,T] \to \mathbb{R}$ (smooth), picking the integrating factor $u_t = \exp\{-\int_{-\infty}^t a(\tau)\,\mathrm{d}\tau\}$ gives the transition matrix

$$\Phi_{t|s} = \exp\left\{\int_s^t a(\tau)\,\mathrm{d}\tau\right\},$$

and the transition kernel above reduces to

$$p(\overrightarrow{X}_t|\overrightarrow{X}_s) = \mathcal{N}\left(\overrightarrow{X}_t; \Phi_{t|s}\overrightarrow{X}_s, \mathbb{I}\int_s^t \Phi_{\tau|s}^2\sigma(\tau)^2\,\mathrm{d}\tau\right).$$

Connection to discrete diffusion processes We are now ready to derive the connection between continuous diffusion models and the commonly used discrete frameworks.

In order to simplify the delivery, we will focus on the DDPM framework [33] (a special case of the later DDIM framework [92] – which introduced non-Markovian forward processes making it harder to recover going from the continuous to discrete case). In practice, as discussed more in Section 2.3, the most commonly used drift and diffusion coefficient selection is $a(t) = \frac{1}{2}\frac{d}{dt}\ln(1-\beta(t))$ and $\sigma(t) = \sqrt{-\frac{d}{dt}\ln(1-\beta(t))}$, where $\beta(t) : [0,T] \to [0,1]$ is a smooth increasing function. Then under N discretization steps of the (forwards SDE or) transition kernel, and time partition $0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = T$ for step size $\Delta t \ll 1$, we have under discretization and successive Taylor series approximation

$$\begin{aligned}
\Phi_{t_i|t_{i-1}} &= \sqrt{\frac{1-\beta(t_i)}{1-\beta(t_{i-1})}} & H_{t_i|t_{i-1}} &= 1 - \frac{1-\beta(t_i)}{1-\beta(t_{i-1})} \\
&\approx \sqrt{1-\beta(t_{i-1}+\Delta t)}, & &\approx \frac{\beta(t_{i-1}+\Delta t) - \beta(t_{i-1})}{1-\beta(t_{i-1})} \\
& & &\approx \beta(t_{i-1}+\Delta t)
\end{aligned}$$

and we approximately recover the formulation given in [33, 90] with forwards noising kernels

$$p_T(\overrightarrow{X}_T|\overrightarrow{X}_0) = \prod_{i=1}^{T} p_t(\overrightarrow{X}_{t_i}|\overrightarrow{X}_{t_{i-1}}), \text{ with } p_{t_i}(\overrightarrow{X}_{t_i}|\overrightarrow{X}_{t_{i-1}}) = \mathcal{N}(\overrightarrow{X}_{t_i}; \sqrt{1-\beta_{t_i}}\overrightarrow{X}_{t_{i-1}}, \beta_{t_i}\mathbb{I}). \tag{2.3}$$

Likewise, for the time reversed SDE we have the backwards Markov chain

$$p_T(\overleftarrow{X}_0) = \prod_{i=1}^{T} p_{t_i}(\overleftarrow{X}_{t_{i-1}}|\overleftarrow{X}_{t_i})$$

where backwards transition densities depend on the score function, which is not known in advance. This is the central learning objective proposed in [33], paramterizing the time-reverse process with $p_\theta(\overleftarrow{X}_{t_{i-1}}|\overleftarrow{X}_{t_i}) = \mathcal{N}(\overleftarrow{X}_{t_{i-1}}; \eta_\theta(\overleftarrow{X}_{t_i}, t_i), \mathrm{H}_\theta(\overleftarrow{X}_{t_i}, t_i))$; the details of the learning task are developed below.[3]

[3]In subsequent sections, the sub-indexing of discrete time steps is suppressed and should be inferred from context.

2.3 Neural diffusion models

Understanding the dynamics of a diffusion process is all well and good, but unless we can devise a method of efficiently sampling from these dynamics, they are of no practical utility. This section is dedicated to the task of describing how diffusion processes can be practically modeled using neural networks and trained to perform a target objective.

Model parameterization and training Continuous diffusion models, as implemented using neural networks, are parameterized to learn a time-conditional approximation to the Stein score $s_\theta(X_t, t) \approx \nabla_x \log p_t(X_t)$ that appears in Eq. (2.2), in order to sample from the reverse process via (repeat) evaluation of

$$\mathrm{d}\overleftarrow{x_t} = [\mu(\overleftarrow{x_t}, t) - \sigma^2(t) s_\theta(\overleftarrow{x_t}, t)]\,\mathrm{d}t + \sigma(t)\,\mathrm{d}\overleftarrow{B_t} \tag{2.4}$$

approximating sampling $x_t \sim p_t$. Thus, in diffusion models, the terminal distribution p_T serves as the "initial" distribution, which for affine drift and diffusion terms converges to a Gaussian distribution which can easily be sampled from, and transported to p_0, which is often not known analytically. In [91], this is approximated using M steps of Langevin Metropolis-Hasting [73] to sequentially sample $x_t \sim p_t$ according to the equation[4]

$$\overleftarrow{x_t}^{(m)} = \overleftarrow{x_t}^{(m-1)} + \frac{\sigma_t^2}{2} s_\theta(\overleftarrow{x_t}^{(m-1)}, t) + \sigma_t z_t^{(m)}, \quad z_t^{(m)} \sim \mathcal{N}(0, \mathbb{I}), \text{ for } m = 1, \ldots, M; \tag{2.5}$$

that is, $\overleftarrow{x_t} \approx \overleftarrow{x_t}^{(M)}$. In order to avoid stochasticity in sampling, the authors of [92] derived the probability flow ODE (PF-ODE), below, as a counterpart to Eq. (2.4)

$$\mathrm{d}\overleftarrow{x_t} = [\mu(\overleftarrow{x_t}, t) - \frac{\sigma^2(t)}{2} s_\theta(\overleftarrow{x_t}, t)]\,\mathrm{d}t$$

which, under the DDIM[90] framework, admits the same marginal distributions, p_t, and can thus be used to sample a trained model deterministically.

In order to train these models, we must define a suitable loss function. It was proved in [39] the score matching (SM) objective

$$\mathcal{L}_{SM}(s_\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,T)} \mathbb{E}_{\overrightarrow{X}_t \sim p_t} [\| s_\theta(\overrightarrow{X}_t, t) - \nabla_x \log p_t(\overrightarrow{X}_t) \|_2^2]$$

[4]The exact form of the step size in this equation will depend on the chosen forwards process parameters.

guarantees the learned approximation will agree up to a constant factor difference. However, as p_t is typically not accessible, with unknown ground-truth mean and variance, surrogate objectives have been developed. Notably, implicit score matching (ISM)

$$\mathcal{L}_{ISM}(s_\theta) = \mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\overrightarrow{X}_t\sim p_t}\left[\frac{1}{2}\|s_\theta(\overrightarrow{X}_t, t)\|_2^2 + \nabla\cdot s_\theta(\overrightarrow{X}_t, t)\right], \tag{2.6}$$

where the expectation is taken w.r.t. the empirical distribution $\hat{p}_t \approx p_t$[5]. Discussion around ISM, and its approximation, is postponed till Section 4.2 where it is necessary. More commonly, the equivalent denoising score matching (DSM) loss [99] is used whenever feasible

$$\mathcal{L}_{DSM}(s_\theta) = \mathbb{E}_{X_0\sim p_0}\mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\overrightarrow{X}_t\sim p_t(\overrightarrow{X}_t|\overrightarrow{X}_0)}[\|s_\theta(\overrightarrow{X}_t, t) - \nabla_x \log p_t(\overrightarrow{X}_t|\overrightarrow{X}_0)\|_2^2] \tag{2.7}$$

where $p_t(\overrightarrow{X}_t|\overrightarrow{X}_0)$ is the forward transition probability conditioned on $\overrightarrow{X}_0$, following the recursive application of the diffusion transition kernel, which for affine drift, as derived in Section 2.2, is tractable to compute unlike $p_t(\overrightarrow{X}_t)$. It was found in [33] that parameterizing the learning task in terms of predicting the noise sample $\epsilon_t \sim \mathcal{N}(X_t; \eta_t\overrightarrow{X}_0, H_t)$, using a network $\epsilon_\theta(X_t, t)$, opposed to the score, s_θ, directly, resulted in more stable training under the corresponding reparameterization of the loss

$$\mathcal{L}^\epsilon_{DSM}(\epsilon_\theta) = \mathbb{E}_{\overrightarrow{X}_0\sim\hat{p}_0}\mathbb{E}_{t\sim\mathcal{U}(0,T)}\mathbb{E}_{\epsilon\sim\mathcal{N}(\epsilon_t;\eta_t\overrightarrow{X}_0,H_t)}[\|\epsilon - \epsilon_\theta(X_t, t)\|_2^2];$$

under which the Stein score is approximated via Tweedie's estimate [20] $\epsilon_\theta(X_t, t) \approx \mathbb{E}[\epsilon|X_t]$ as

$$\nabla_x \log p_t(X_t) \approx \frac{-\epsilon_\theta(X_t, t)}{\sigma_t^2}.$$

However, as documented in [48], this method is not ideal as the inter-sample variance is high unless the noise samples are normalized to unit variance (among other considerations). See [48] for a more detailed discussion on diffusion model network parameterizations.

Pseudocode versions of the general training and sampling procedures of discrete diffusion models are given in Algorithm 1 and Algorithm 2 respectively.

[5]It is common to ignore this notational difference and read all expectations as being approximated over the given empirical data distribution.

Algorithm 1: (Discrete) Diffusion Model Training and Sampling
Consider diffusion over $\mathbb{R}^d$ parameterized by θ, let p_0 be the data distribution and D the training dataset, $p(x_t|x_{t-1})$ the forward diffusion process (see Eq. (2.3)), $p_\theta(x_{t-1}|x_t)$ is the reverse process, T the time horizon, and $\eta > 0$ is the learning rate.

Data: D, T, η, θ
Result: θ
while *training* **do**
  $x_0 \sim D$
  $t_i \sim \mathcal{U}\{1, \dots, T\}$
  for $t = 1, \dots, t_i$ **do**
    $x_t \sim p(x_t|x_{t-1})$; /* Forward diffusion step */
  $\hat{x}_0 \sim p_\theta(x_0|x_{t_i})$; /* Approx sample reverse process from x_{t_1} */
  $\mathcal{L} \leftarrow \mathcal{L}_{DSM}(\theta)$; /* Given x_0 and $\hat{x}_0$ */
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
return θ

Algorithm 2: Sampling from (Discrete) Diffusion Model
After training, samples are generated using the learned reverse diffusion process $p_\theta(x_{t-1}|x_t)$, resulting in an approximate sample $\hat{x}_0$.

Data: θ, T
Result: $\hat{x}_0$
$x_T \sim \mathcal{N}(0, \mathbb{I})$
for $t = T, \dots, 1$ **do**
  $\hat{x}_{t-1} \sim p_\theta(x_{t-1}|x_t)$; /* Approx reverse diffusion step */
return $\hat{x}_0$

Noise scale selection and design One important aspect governing neural diffusion model performance, as seen empirically and as mentioned at the start of Section 2.2, is the chosen noise scale (i.e., the scale of drift and diffusion coefficients) used for a chosen learning task. The study of designing noise schedules is the topic of various works [33, 50, 74, 48, 83, 57], but among these, the most dominant are the variance exploding (VE) [92] and variance preserving (VP)[33, 90] schedules.

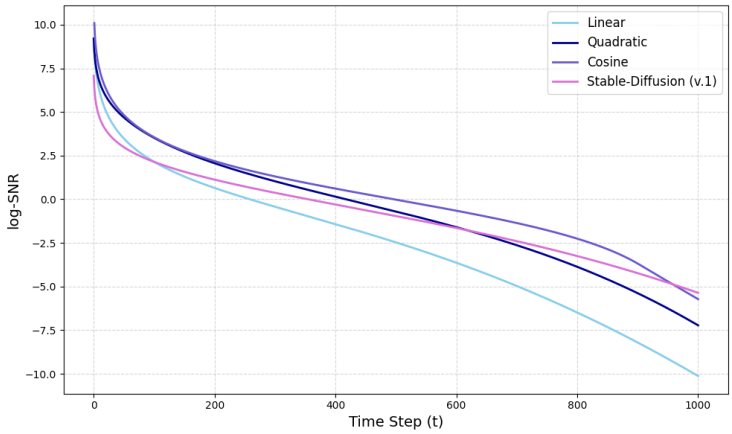


Figure 2.2: Comparison of log-SNR values between four commonly used noise schedules.

To better visualize the effect different noise schedules have, in Fig. 2.2 we plot the log signal to noise ratios (log-SNR), as popularized by [50] for analyzing these schedules, where

$$\mathrm{SNR}(t) = \frac{\prod_{i=1}^{t}(1-\beta_i)}{1-\prod_{i=1}^{t}(1-\beta_i)} \quad \text{or} \quad \frac{\sigma(t)^2}{1-\sigma(t)^2}$$

quantifies the relative intensity of a (date) signal compared to the added noise, values of the four most common DDPM noise schedules, which have equivalents in the continuous setting, with the common parameters $\beta_0 = 0.0001$, $\beta_T = 0.02$, $T = 1000$. Then in Fig. 2.3 we illustrate the progression of these noise schedules, plotting every 50th step, on a sample image taken from the CelebA dataset [60]. As evident from the figure, depending on which schedule is selected, high fidelity features of the image are preserved longer than others; in particular, in accordance with the purported design motivation, the cosine schedule [74] adds noise more progressively than the other illustrated methods. It is for this reason the authors suspect the schedule allowed diffusion models parameterized for noise prediction, Eq. (2.7), to perform better than its contemporaries. With this said, we are not currently aware of any existing analysis (theoretical or otherwise) that proposes a universally agreed upon metric for the "optimization" of noise schedules outside of post training evaluation.

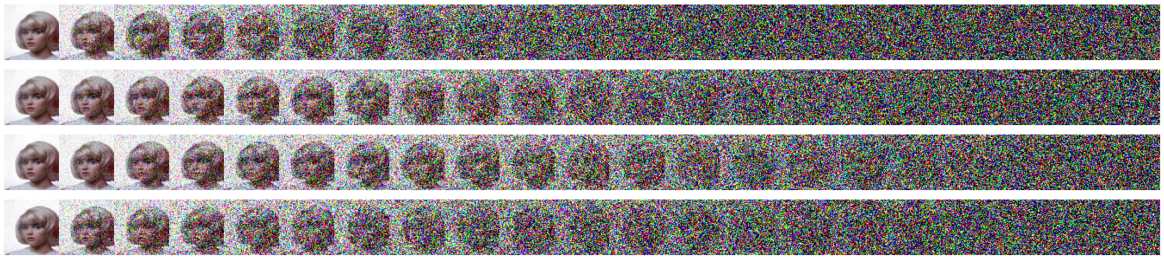


Figure 2.3: Illustration of the noise progression of different schedulers on a random sample from CelebA dataset. Top to bottom these follow the Linear, Quadratic, Cosine, and Stable-Diffusion schedules.

Example: DDPM learned score field To visualize the training and sampling dynamics, in the sample space sense, we train a DDPM using ISM loss and a linear noise schedule with $T = 50$ over the synthetic dataset constructed in Section 5.4. Progressive samples are drawn from the model at time steps $t = 0, 20, 40, 49$ along with the learned vector field which is evaluated over a finite grid $[1.5, 1.5] \times [-1.5, 1.5]$ with cell size $\Delta = 0.015$. These samples are displayed in Fig. 2.4, and shows how diffusion models transport Gaussian samples to a learned density.

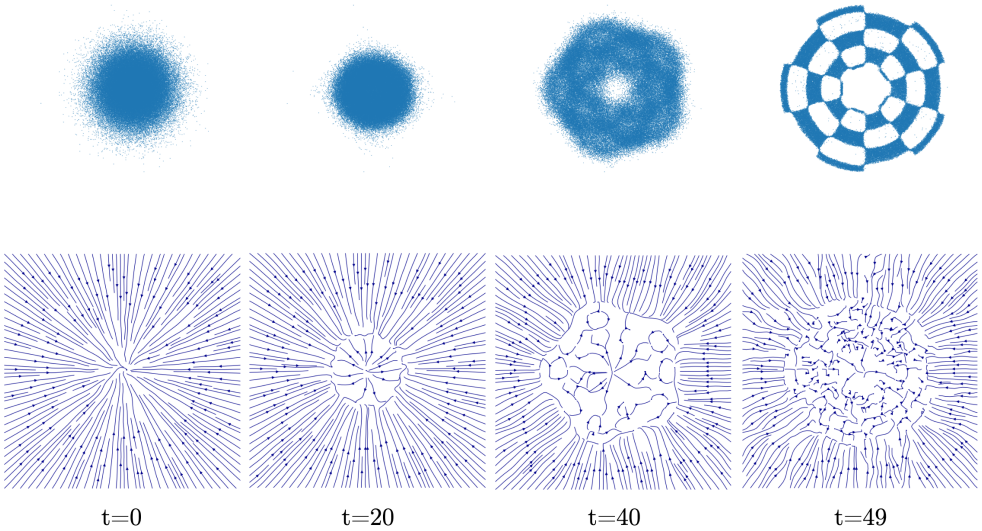


Figure 2.4: (top) Incremental samples drawn from a trained DDPM at time steps $t = 0, 20, 40, 49$, and (bottom) corresponding flow visualization of learned Stein score.

Chapter 3

Manifold diffusion models

As detailed in Chapter 2 diffusion models, in the Euclidean setting, can effectively model complex data distribution, however, many dataset naturally originate on non-Euclidean settings where the diffusion theory and techniques we have seen break down; e.g., data that is constrained to Riemann manifolds prove to be difficult to learn by standard diffusion model architectures. In response to this, diffusion models must be extended to operate directly on non-Euclidean settings in order to be applicable to a broader domains, particular those in the physical sciences where data often resides on manifolds.

Here we will outline how the underlying diffusion processes can be extended to Riemannian manifolds, beginning with revisiting the definition of Brownian motion in these domains, and later expressing stochastic differential equations on manifolds that express a continuous diffusion process. After introducing diffusion models in this setting, we detail methods of approximating the various quantities needed to simulate these processes and train newly parametrized neural diffusion models. These extensions enhance diffusion models applicability to datasets that are best described by curved surfaces, assuming the manifold hypothesis, allowing new problems to be attacked using these models.

3.1 Brownian motion on manifolds

Brownian motion has analogs on smooth manifolds, and can be defined (or simulated) in various ways; .e.g., for manifolds embedded into an ambient space, one can use a collection of charts to define Brownian motion over the manifolds local coordinates, or, similar to Section 2.1, define Brownian motion via the solution to the heat kernel on the manifold.

To begin, we will assume all (Riemannian) manifolds, $(\mathcal{M}, g)$, hereafter are compact, connected, and, for simplicity, are isometrically embedded in Euclidean space (under the Nash embedding theorem [72]) in order to define local coordinate charts; alternatively, and with increased generality, one can also make use of intrinsic coordinates. We will primarily

follow [28, 29] which develop estimates for the heat equation (special case of diffusion forward equation, see Section 2.1) under the perspective Brownian motion on Riemannian manifolds. We also draw from the appendixes of the works [14, 77, 37, 63].

Definition 2 (Laplace-Beltrami operator)**.** *Suppose $(\mathcal{M}, g)$ is a Riemann manifold, and $p \in \mathcal{M}$. If $\phi : \mathbb{R}^d \hookrightarrow U$, for $U \subseteq \mathcal{M}$ (open) and $p \in U$, we can define the Laplace operator on $\mathcal{T}_p\mathcal{M}$, called the Laplace-Beltrami operator in this setting, as:*

$$(\nabla \cdot \nabla)_{\mathcal{M}}(f) = \frac{1}{\sqrt{\det(g)}} \sum_{i,j=1}^{d} \frac{\partial}{\partial x_i}(\sqrt{\det(g)}(g_{i,j})^{-1} \frac{\partial}{\partial x_j}(f)),$$

for any function $f \in C^2(\mathcal{M})$ with $\frac{\partial}{\partial x_j} = \frac{\partial}{\partial \phi_j^{-1}}|_p$ and $g_{i,j} = g(\frac{\partial}{\partial x_i}, \frac{\partial}{\partial x_j})|_p$ defining the Riemannian tensor $g = (g_{i,j})$.

Then we can define Brownian motion (intrinsically) using semi-martingales.

Definition 3 (Semi-martingale, [14, Appx. C.2], [36])**.** *Let $(\mathcal{M}, g)$ be a given d-dimensional Riemann manifold of C^k. A a.s. continuous stochastic process $(X_t)_{t\geq 0}$ is called a $\mathcal{M}$-valued semi-martingale, meaning it is defined on $\mathcal{M}$, if $\forall f \in C^k(\mathcal{M}, \mathbb{R}^d)$, $(f(X_t))_{t\geq 0}$ is a real valued semi-martingale.*

Definition 4 (Brownian motion on manifolds, [14, Appx. C.3])**.** *Let $(B_t^{\mathcal{M}})_{t\geq 0}$ be a $\mathcal{M}$-valued semi-martingale; meaning it takes on value from $\mathcal{M}$. Then $(B_t^{\mathcal{M}})_{t\geq 0}$ is a Brownian motion on $\mathcal{M}$ if for any smooth vector field $\psi : \mathcal{M} \to \mathbb{R}^d$, the processes defined by*

$$M_t^{\psi} = \psi(B_t^{\mathcal{M}}) - \psi(B_0^{\mathcal{M}}) - \frac{1}{2}\int_0^t (\nabla \cdot \nabla)_{\mathcal{M}} \psi(B_s^{\mathcal{M}})\, \mathrm{d}s$$

is a (real-valued) local martingale.

Another important preliminary is that of the Stratanovich integral. Rather than making use of the Ito integral (as done in Euclidean space) we will deal with Stratonovich integrals (and SDEs) in order to exploit the chain rule over differentiable charts. A consequence of this is, unlike Ito's integral, the stochastic processes are not true martingales, and we must deal with semi-martingales and localization problems.

Definition 5 (Stratanovich integral on manifolds, [36])**.** *Let $(\mathcal{M}, g)$ be a given d-dimensional Riemannian manifold, $(\mathcal{M}, \mathcal{B}, (\mathcal{F}_t)_{t\geq 0}, P)$ be a filtered probability space, and $(\overrightarrow{B_t^{\mathcal{M}}})_{t\geq 0}$ be a Brownian motion (adapted process) on $\mathcal{M}$ defined upto a time T. Suppose $\{V^i\}_{i=1}^d$ be a*

collection of vector fields with $V_i : \mathcal{M} \to \mathbb{R}^d$*, meaning for a vector field* $V : \mathcal{M} \to \mathcal{T}\mathcal{M}$*,* $V = \sum_{i=1}^d V^i \frac{\partial}{\partial x^i}$*. A* $\mathcal{M}$*-valued semi-martingale* $(X_t)_{t\geq 0}$ *is a solution to the SDE*

$$\mathrm{d}\overrightarrow{X}_t = V(X_t) \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}$$

in the Stratanovich sense if for $\forall f \in C^k(\mathcal{M}, \mathbb{R})$ *and* $t \in [0, T]$

$$\begin{aligned} f(\overrightarrow{X}_t) &= f(\overrightarrow{X}_0) + \sum_{i=1}^d \int_0^t V^i(f)(\overrightarrow{X}_s) \circ \mathrm{d}\overrightarrow{(B_t^{\mathcal{M}})^i} \\ &= f(\overrightarrow{X}_0) + \sum_{i=1}^d \int_0^t V^i(\overrightarrow{X}_s) \frac{\partial f^i}{\partial x^i} \circ \mathrm{d}\overrightarrow{(B_t^{\mathcal{M}})^i}, \end{aligned}$$

under local charts $\{\phi_i\} : U_i \subseteq \mathbb{R}^d \hookrightarrow V_i \subseteq \mathcal{M}$ *with* $\phi_j^{-1} \circ \phi_i \in C^k(U_i, U_j)$*.*

We are now in a suitable position to develop more general diffusion processes on manifolds, mirroring the information presented in Section 2.2.

3.2 Diffusion processes on manifolds

In this section, we briefly summarize existing work that extends diffusion models to more general non-Euclidean geometries, in particular, Riemannian manifolds. While this extension includes additional complications and necessary imprecision from approximation, which are not present for problems that can be posed in Euclidean space, this setting becomes necessary (or more natural) for a variety of problems where the data does not inherently live on a flat. We primarily draw from the works [14, 77, 37, 63], citing results where required.

Continuous diffusion processes on manifolds Let $\sigma_t : [0, T] \to \mathbb{R}_{\geq 0}$ be a continuous smooth function and $(\overrightarrow{B_t^{\mathcal{M}}})_{t\geq 0}$ a Brownian motion over the d-dimensional orientable compact Riemann manifold $(\mathcal{M}, g)$, assume $\mathcal{M}$ is isometrically embedded into $\mathbb{R}^D$, for $D \geq d$, so we can define the global chart[1] $\phi : \mathbb{R}^D \hookrightarrow \mathcal{M}$. Then, if $\overrightarrow{X}_0 \sim p_0$, the analog to the Euclidean SDE on $\mathcal{M}$ is

$$\mathrm{d}\overrightarrow{X}_t = \sigma_t \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}. \tag{3.1}$$

[1]This condition can be slackened and is only assumed to make function multiplication easier notationally, and to avoid introducing additional definitions from differential geometry.

Then there exists an analogues time reversed equation, obeying similar initial conditions to Eq. (2.2) see [14], of the form

$$\begin{aligned}\mathrm{d}\overleftarrow{X}_t &= \sigma_t^2 \nabla_x \log p_t(\overleftarrow{X}_t)\,\mathrm{d}t - \sigma_t \circ \mathrm{d}\overleftarrow{B_t^{\mathcal{M}}} \\ &= \sigma_t^2 \left[\sum_{i,j}^{d} g_{i,j}(\overleftarrow{X}_t)^{-1} \frac{\partial \log p_t(\overleftarrow{X}_t)}{\partial x^j} \frac{\partial}{\partial x^i}\bigg|_{\overleftarrow{X}_t}\right] - \sigma_t \circ \mathrm{d}\overleftarrow{B_t^{\mathcal{M}}},\end{aligned}$$

more details for operating on manifolds is presented in Appendix A.3.

To simplify the delivery, and any proofs, we will primarily consider the diffusion process of the form

$$\mathrm{d}\overrightarrow{X}_t = \sigma_t \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}.$$

We can include a vector field drift term, but this necessitates further discussion around manifold connections and parallel transport and would significantly complicate the statement of results.

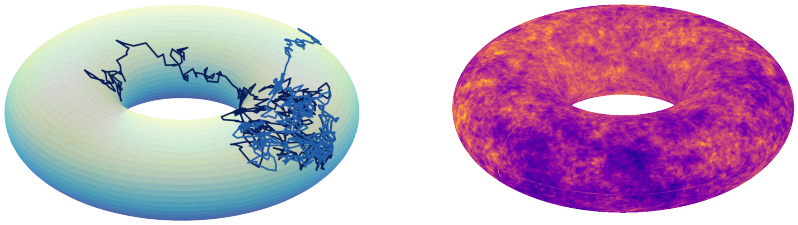
Sampling Brownian motion on manifolds While Definition 4 gives us a way to define Brownian motion through the use of charts, it is not particularly efficient to sample from, due to having to sample Brownian motion in higher dimensions, $\mathbb{R}^{D\times D}$ for $D \geq d$, and then project it onto the manifold of interest[2]. A (relatively) efficient and simple method of sampling Brownian motion on manifolds can be obtained from simulating geodesic random walk where we simulate curves on the tangent space $T_p\mathcal{M} \cong \mathbb{R}^d$ and project them onto the manifold, either using the exponential map – if known – or via orthogonal projection on the surface. A simple algorithm for simulating this kind of walk for the heat kernel is presented in [14] and written out in Algorithm 3.

In Fig. 3.1a we illustrate 600 simulation steps of Algorithm 3 atop a torus (black) as compared to simulating 600 steps of (standard) Brownian motion in tangent plane and projecting onto the torus. It is clear from the figure there is significant disagreement between these methods, due to proper adjustments to local curvature used in simulating the geodesic random walk but not in the latter. Fig. 3.1b contains a heatmap visualization of simulating 100,000,000 steps of Algorithm 3, which approximates the limiting uniform distribution of Brownian motion over the torus. These figures are intended to illustrate, to the reader, the distributional difference between Brownian motion in Euclidean space, as depicted in Fig. 2.1, and atop Manifolds.

[2]This is often referred to as the dimension cost of selecting an extrinsic view of the manifold.

Algorithm 3: Manifold Geodesic Random Walk.
Consider a Riemannian manifold $(\mathcal{M}, g)$. Let $T \geq 0$ be a given time, N the number of discretization steps, and X_0 a initial starting point.

Data: $\mathcal{M}, T, N, X_0$
Result: $\{\hat{X}_t\}_{t=0}^{T}$
$\Delta t \leftarrow T/N$
for $t = 0, 1, \ldots, N-1$ **do**
  $Z_{t+1} \leftarrow \mathcal{N}(0, \mathbb{I}_{d\times d})$
  $\hat{X}_{t+1} \leftarrow \exp_g(\hat{X}_t, \sqrt{\Delta t} Z_{t+1})$
return $\{\hat{X}_t\}_{t=0}^{T}$



(a) (b)

Figure 3.1: (a) An illustration of 600 steps of a geodesic random walk (black) and projected tangent plane random walk (blue) over the surface of a torus. (b) Heatmap approximation of limiting distribution of Brownian motion on torus resulting from simulating a geodesic random walk for 100,000,000 steps.

3.3 Neural diffusion models on manifolds

Standard diffusion model paramterizations, e.g., Section 2.3, cannot be directly employed to tasks where the data is constrained to lie on a manifold; when attempted without modification these models often fail to learn the task or under-perform. In order to enable these models to adapt to changes in curvature, which affects the learned score, techniques from [62, 77, 14, 37, 63] must be utilized.

Loss selection: Unfortunately, in the manifold setting one must be careful to select an appropriate loss function formulation as the regular DSM, Eq. (2.7), may no longer converge. However, provided the manifold is smooth, we may utilize ISM, Eq. (2.6), without significant issues. Nonetheless, utilizing implicit score matching comes with computational overhead, it being necessary to estimate the divergence $\nabla \cdot s_\theta(X_t, t)$ over $\mathcal{M}$. One method of doing so is by using auto-differentiation and approximating the trace of the Jacobian using the Hutchinson estimate [38]

$$\begin{aligned} \nabla \cdot s_\theta^{\mathcal{D}} &= \mathrm{Tr}(\nabla s_\theta) \\ &\approx \mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbb{I})}[\nabla(\epsilon^\intercal s_\theta)\epsilon] \\ &\approx \frac{1}{N}\sum_{i=1}^{N} \nabla(\epsilon_i^\intercal s_\theta)\epsilon_i \quad \text{for } N \text{ i.i.d. }, \epsilon_i \sim \mathcal{N}(0,\mathbb{I}), \text{ random samples.} \end{aligned} \tag{3.2}$$

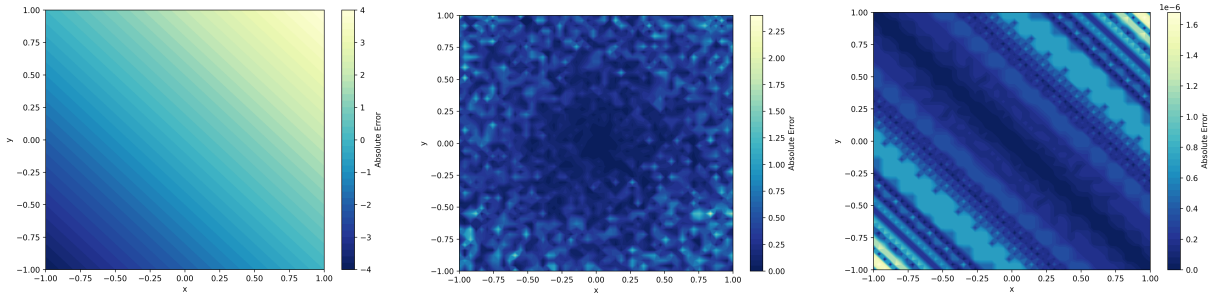
Example: Divergence estimate To demonstrate the divergence estimate given by Eq. (3.2) we use this method to approximate the divergence of the vector function $f(x, y) = (x^2, y^2)$ over a finite grid $[-1, 1] \times [-1, 1]$ with cell size $\Delta = 1/50$. Fig. 3.2 contains (a) the plot of the functions analytical divergence, (b) the Hutchinson divergence estimate for 30 random Gaussian samples, and (c) the Hutchinson divergence estimate for 30 random Rademacher samples – a distribution that is commonly used in practice. The estimate derived using Rademacher samples achieves a much lower absolute error – top right of the figure illustrates the scale change to $1e^{-6}$ – in comparison to the historically used Gaussian estimate. This highlights the importance of selecting an appropriate distribution to use for computing the Hutchinson estimator.

Now, in most applications, we don't have access to the analytical score, so determining which distribution is best to sample from may not be clear, but in a number of cases the Rademacher distribution will yield lower variance estimates. To see why, notice for $A \in \mathbb{R}^{n\times m}$, with i.i.d. random samples ϵ from a distribution s.t. $E[\epsilon] = 0$ and $E[\epsilon_i^4] = \gamma$, the Hutcheison estimate of $Tr(A)$ can be decomposed as

$$Var[\hat{Tr}(A)] = \frac{1}{N}\left[\sum_{i\neq j} A_{ij}^2 + (\gamma - 1)\sum_i A_{ii}^2\right].$$

Consequently, picking ϵ from the Rademacher distribution yields $E[\epsilon^4] = 1$ and the variance of the estimate reduces to

$$Var[\hat{Tr}(A)] = \frac{1}{N}\left[\sum_{i\neq j} A_{ij}^2\right].$$



(a) (b) (c)

Figure 3.2: (a) plot of analytical function divergence field along with absolute error between Hutchinson divergence estimates computed using (b) Gaussian and (c) Rademacher random projections. Note the change in scale between plots: (a) $(-4, 4)$, (b) $(0.00, 2.25)$, and (c) $(0.0, 1.6e^{-6})$.

Whereas, if $\epsilon \sim \mathcal{N}(0, \mathbb{I})$ then $E[\epsilon^4] = 3$ and the variance estimate becomes

$$Var[\hat{Tr}(A)] = \frac{1}{N}\left[\sum_{i \neq j} A_{ij}^2 + 2\sum_i A_{ii}^2\right].$$

In fact, provided $E[\epsilon_i^4] \geq 1$ and above distribution assumptions hold, the Rademacher distribution is guaranteed to lower the variance of the trace estimate for a given matrix.

ISM stochastic scaler In addition to the above, we found empirically during the experimentation on Section 5.4 the scale difference between the norm and divergence terms in Eq. (2.6) contributed to slow model convergence. Consequently, we devised the following (stochastic) normalizing constant

$$\hat{c}_t = -\frac{1}{N}\sum_{i=1}^{N} \frac{\mathbb{E}_{\epsilon \sim \mathcal{N}(0,\mathbb{I})}[\nabla(\epsilon^\intercal s_\theta(X_t^{(i)}, t))\epsilon]}{\|s_\theta(X_t^{(i)}, t)\|_2^2}, \quad \text{for a batch} \quad \{X_t^{(i)}\}_{i=1}^N \sim p_t$$

and the modified implicit score matching loss

$$\mathcal{L}_{ISM}(s_\theta, \hat{c}_t) = \mathbb{E}_{X_t \sim p_t}\left[\frac{1}{2}\|\hat{c}_t \cdot s_\theta(X_t, t)\|_2^2 + \hat{c}_t \cdot \nabla \cdot s_\theta(X_t, t)\right]. \tag{3.3}$$

In practice, this term is only recomputed after several thousand training iterations (>1000), and we make the coarse approximation $\hat{c}_t \approx \hat{c}_{t'}$ for all t' sampled between computations, and has shown, for the dataset Section 5.4, to improve training stability. See Section 5.4 for an example.

Model parametrization and training Having discussed some of the issues around loss selection and approximation on manifolds, it remains to discuss methods for paramterizing the score learned by a diffusion model to ensure the manifold curvature is properly incorporated into the model output; for brevity, we will limit our discussion to two general methods.

The simplest method, that is only applicable when the underlying manifold is generated by a set of parametric equations (e.g., sphere, torus, etc), is to map the model score onto the tangent plane of the manifold, say to $\mathcal{M}$ using a projection mapping $\textsc{proj}(p, \cdot) : \mathbb{R}^D \times \mathbb{R}^d \to T_p\mathcal{M}$, and perform the loss estimate within the corresponding tangent space $T_p\mathcal{M}$. In particular, for a score parametrized model s_θ, not necessarily constrained to $\mathcal{M}$, the loss computation – model training – can carried out by following Algorithm 4. This method can suffer from approximation error in the projection step if the projection map (e.g., exponential map) is only known approximately, but generally this method is easy to implement and performs suitably well for symmetric manifolds.

An alternative approach, that is more flexible but involves more implementation, is to utilize the basis approximation technique from [63, 9], which itself borrows from a longstanding technique in computer graphics [56], where a spectral basis is constructed from the k smallest eigen value-function pairs of the Laplace-Beltrami operator over $\mathcal{M}$. For instance, given $k > 0$, we may approximate the true basis of $\mathcal{M}$ via the spectral basis $\{\psi_1, \ldots, \psi_k\}$ where $(\nabla \cdot \nabla)_{\mathcal{M}}\psi_i = \lambda_i\psi_i$ for $i = 1, \ldots, k$. Under this basis the manifold heat kernel and DSM objective can be approximated, under sufficiently large k, as

$$\nabla_x \log p_t(X_t|X_0) \approx \nabla_x \log \sum_{i=1}^{k} e^{-\lambda_i}\psi_i(X_0)\psi_i(X_t),$$

thereby enabling one to train a diffusion model in a fashion similar to the standard Euclidean setting using Algorithm 1.

Algorithm 4: Manifold (projected) diffusion training.
Consider diffusion over the Riemannian manifold $\mathcal{M}$. Let s_θ be a score parametrized diffusion model, DATA the training dataset, σ diffusion coefficient, N the number of discretization steps, and $\eta > 0$ the learning rate.

Data: $\mathcal{M}$, DATA, N, s_θ, σ, η
Result: θ
while *training* **do**
  $x_0 \sim$ DATA
  $t_i \sim \mathcal{U}[0, 1]$
  $k \leftarrow \lfloor \sigma(t_i)/\sigma(T) \rfloor N$
  $x_t \leftarrow$ RANDOM-WALK$(\mathcal{M}, \sigma(t_i), k, x_0)$
  $\mathcal{L} \leftarrow \mathcal{L}_{ISM}(\text{PROJ}(x_t, s_\theta))$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
return θ

Chapter 4

Reflected diffusion

In the preceding sections, the support of the diffusion process was assumed connected and without boundary, however, this assumption is unrealistic as data generated from physical processes are often constrained within a narrow range of values. Consequently, if one desires guarantees on a diffusion model generating data within a particular support, we must consider the effects of constraining the diffusion process to lie within a bounded region of the domain. Such bounded Brownian motions are referred to as reflected Brownian motions without absorption at the boundary, and are the topic of this chapter.

We will begin by discussing bounded Euclidean (manifolds) regions and then move onto a discussion around Riemann manifolds with boundaries. It might, at first inspection, seem odd to position this chapter after Brownian motion on manifolds as opposed to before. This was done with good reason, as reflected diffusion naturally generalizes to the manifold setting and, in the author's opinion, is best understood by treating the constrained domains from the beginning as (Euclidean or Riemann) manifolds atop which Brownian motion is defined. This chapter ends with a discussion on practicalities of training neural diffusion models on reflected diffusion processes, and the challenges therein.

4.1 Diffusion processes on constrained domains

The seminal works [58, 69, 6, 5] have previously established, in combination, the expected convergence behaviour and uniqueness of solutions to the forwards diffusion equation over bounded smooth path connected domains, while the most recent works [61, 24] have established that such diffusion processes admit well defined backwards equations that can be effectively learned using neural networks. In this section, we elucidate the critical theorems of the forgoing papers in a unified and self contained manner, as well as providing some incite into the problem and key considerations necessary to effectively train neural diffusion models under this paradigm.

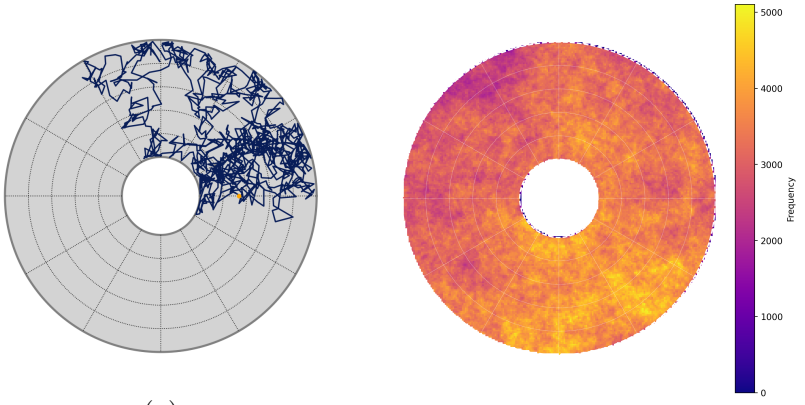
Forwards reflected diffusion process For this section we build off a stochastic process $(X_t)_{t\geq 0}^T$ in $\mathbb{R}^d$ governed by the SDE

$$\mathrm{d}\overrightarrow{X}_t = \mu(\overrightarrow{X}_t, t)\,\mathrm{d}t + \Sigma(\overrightarrow{X}_t, t)\,\mathrm{d}\overrightarrow{B}_t$$

where μ and Σ are Lipschitz in X_t and t, and $(\overrightarrow{B}_t)_{t\geq 0}^T$ is a Brownian motion process. We are interested in studying the existence and uniqueness of a corresponding process

$$\mathrm{d}\overrightarrow{X}_t = \mu(\overrightarrow{X}_t, t)\,\mathrm{d}t + \Sigma(\overrightarrow{X}_t, t)\,\mathrm{d}\overset{\rightsquigarrow}{B}_t$$

over a constrained manifold $\mathcal{D}$ where $(\overset{\rightsquigarrow}{B}_t)_{t\geq 0}^T$ is a Brownian motion process that does not leave the manifold $\mathcal{D}$; i.e., a version of Section 3.1 over outer-bounded subspaces.



(a) (b)

Figure 4.1: (a) An illustration of 1000 steps of a reflected random walk starting at a orange point bounded within a unit annulus. (b) Heatmap approximation of limiting distribution of reflected Brownian motion within annulus resulting from simulating 100,000,000 discrete steps.

To picture the difference between regular Brownian motion, Section 2.1, and reflected Brownian motion, an example is depicted in Fig. 4.1a where the domain is restricted to a (flat) annulus. Informally, much like how manifold Brownian motion, Section 3.1, must be constrained to lie atop the prescribed manifold, reflected Brownian motion is constrained within a domain by nullifying the portion of the random perturbation that would result in pushing the particle outside the domain by reflecting this motion inwards. Fig. 4.1a shows 1000 steps of a (discretized) reflected motion process. Then, just as depicted for

the manifold setting, Fig. 4.1b shows an approximation of the limiting distribution of this reflected motion process, which converges to the uniform distribution after sufficiently many steps.

We now summarize some key results from [58] that establish the existence and uniqueness of solutions to stochastic differential equation with (normal) reflection boundary conditions. The basis for these results lies with the following problem definition, which is discussed in detail in [78].

Definition 6 (Skorokhod problem, [58])**.** *Let $\mathcal{D} \subseteq \mathbb{R}^d$ be a smooth bounded open set and $(W_t)_t$ be a family of functions $W_t \in C([0,\infty), \mathbb{R}^d)$ – we are interested in the case where W_t is a Brownian motion $\overrightarrow{B}_t$ – with $W(0) \in \overline{\mathcal{D}}$, we desire a unique solution (coupling) (X_t, K_t) that satisfies:*

1. *$X_t \in C([0,\infty), \overline{\mathcal{D}})$, $K_t \in C([0,\infty), \mathbb{R}^d)$ with $K_t \in BV[0,T]$ with;*
2. *$X_t + K_t = W_t$, $\forall t \geq 0$;*
3. *where*

$$K_t = \int_0^t n(X_s)\,\mathrm{d}|K|_s \quad \text{and} \quad |K|_t = \int_0^t \mathbf{1}_{\{X_s \in \partial\mathcal{M}\}}\,\mathrm{d}|K|_s;$$

i.e,. $(K_t)_{t\geq 0}$ is a stochastic process of bounded total variation that cancels out any Brownian force that would cause X_t to leave $\mathcal{D}$.

For the remainder let $\mathcal{X} \subseteq \mathbb{R}^d$ compact and path connected and $(\mathcal{X}, \mathcal{B}, (F_t)_t, P)$ be a probability space with increasing filtration $(F_t)_t$ of all sub σ-fields of $\mathcal{X}$. Additionally, suppose $\mathcal{D} \subseteq \mathcal{X}$ is a smooth open bounded domain that satisfies the assumptions in Appendix A.4 and is equip with a vector field $n : \partial\mathcal{D} \to \mathcal{T}\partial\mathcal{D}$ such that $\forall x \in \partial\mathcal{D}$, $\|n(x)\| = 1$, where $n(x)$ is the unit outwards normal (not necessarily singular valued) at $x \in \partial\mathcal{D}$. Moreover, as $\mathcal{D}$ is assumed to be a Euclidean manifold[1] we have $n : \partial\mathcal{D} \to \mathcal{T}\partial\mathcal{D} \cong \partial\mathcal{D} \times \mathcal{X}$, which is the standard setting.

Note, while the assumptions stated in Appendix A.4 include non-convex domains, it is assumed (unless otherwise stated) for the remainder that $\mathcal{D}$ is at least simply connected. Additionally, the assumption can be slackened to a piecewise construction with finitely many non-differentiable points (joins).

[1]Equivalently assume there exists a isometric embedding of $\mathcal{D}$ into $\mathbb{R}^d$ via the Nash embedding theorem [72] for the case that $\mathcal{D}$ is a Riemann manifold, with appropriate alterations to derivatives to correct for local curvature.

Suppose $(\overrightarrow{B_t})_{t\geq 0}^T$ is a F_t-Brownian motion, then under the forgoing domain assumptions in Appendix A.4, our objective can be formally stated as trying to find an almost surely continuous semi-martingale $(\overrightarrow{X}_t)_{t\geq 0}^T$ of the form

$$\overrightarrow{X}_t = x_0 + \int_0^t f(\overrightarrow{X}_s)\,\mathrm{d}s + \int_0^t \sigma(\overrightarrow{X}_s)\,\mathrm{d}\overrightarrow{B_s} - K_t$$

with $\overrightarrow{X}_t \in \mathcal{D}$, $\forall t \geq 0$ and K_t satisfying the Skorokhod problem restrictions. The next theorem tells us when such a martingale exists.

Theorem 1 (Restated from [58]). *Let $\mathcal{D}$ be a domain that satisfies the above assumptions and the drift f_i and diffusion coefficients σ_{ij} are bounded continuous functions on $\mathbb{R}^d$, and $x_0 \in \overline{\mathcal{D}}$; then for the filtered probability space $(\mathcal{X}, \mathcal{B}, (F_t)_t, P)$ there exists an unique (coupled) stochastic process $(X_t, K_t)_{t\geq 0}^T$ such that:*

1. *$(K_t)_t$ is a stochastic processes of bounded variation on $[0,\infty]$ almost surely;*
2. *$(X_t)_t \in C([0,\infty], \mathcal{M})$;*
3. *$\forall t \geq 0$, $X_t = X_0 = \int_0^t f(X_s, s)\,\mathrm{d}s + \int_0^t \Sigma(X_s, s)\,\mathrm{d}\overrightarrow{B_s} - K_t$, where $|K|_t = \int_0^t \mathbf{1}\{x_s \in \partial\mathcal{M}\}\,\mathrm{d}|K|_s$.*

For the particular case where we are given a continuous F_t-local martingale $(\overrightarrow{B_t})_{t\geq 0}$, a continuous bounded variation adapted process $(K_t)_{t\geq 0}$, and $f_i(x,t)$ and $\sigma_{ij}(x,t)$ satisfy a uniform Lipschitz condition in x and are progressively measurable (under the filtration), then the above are also satisfied.

Under the (simple) reflected process

$$\mathrm{d}\overrightarrow{X}_t = \mathrm{d}\overset{\rightsquigarrow}{B_t},$$

which is that primarily considered in [24], we have the following results, which can be seen to originate from characterization of Brownian motion via the heat kernel given in Section 2.1 with Neumann boundary conditions. The first result combines the forgoing assumptions and lemmas to explicitly cover non-convex domains with smooth boundaries.

Theorem 2 ([69, Theorem.3]). *Let $\mathcal{D} \subseteq R^d$ be a given bounded smooth path-connected domain. If $\mathcal{D}$ satisfies Assumption 1 then there exists a unique limiting distribution (stochastic process $(X_t, K_t)_{t\geq 0}$ that satisfies Definition 6.*

I provide a sketch of the overall proof of this theorem, along with some minor notes to give the reader some intuition, as the original proof is quite dense and only a special case of the result is needed.

Proof. For diffusion process $(\overrightarrow{X}_t)_{t\geq 0}^T$ with reflection at the boundary $\partial\mathcal{D}$ in the direction of the normal n and zero jump (i.e., $\gamma(\overrightarrow{X}_t, z) = 0$ for $\forall \overrightarrow{X}_t$ – in the original text) with Lipschitz continuous drift and diffusion terms μ and Σ, then by [69, Theorem 3] problem Definition 6 admits a unique limiting (in probability) solution over $\overline{\mathcal{D}}$. Then by noting the diffusion processes

$$\mathrm{d}\overrightarrow{X}_t = \mu(\overrightarrow{X}_t)\,\mathrm{d}t + \Sigma(\overrightarrow{X}_t)\,\mathrm{d}\overrightarrow{B}_t$$

can be equivalently[2] treated as one in which the drift and diffusion terms vary in time, e.g.,

$$\mathrm{d}Y_t = \mu(\overrightarrow{Y}_t, t)\,\mathrm{d}t + \Sigma(\overrightarrow{Y}_t, t)\,\mathrm{d}\overrightarrow{Z}_t$$

where we take $\overrightarrow{B}_t = (\overrightarrow{Z}_t, t)$ with $(\overrightarrow{Z}_t)_{t\geq 0}$ to be $(d-1)$-dimensional Brownian motion and $\mu(\overrightarrow{X}_t) = \mu(\overrightarrow{Y}_t, t)$ and $\Sigma(\overrightarrow{X}_t) = \Sigma(\overrightarrow{Y}_t, t)$ (under some slight abuse of notation). Thus, the above result recovers the affine diffusion process we considered in Section 4.1. □

It is worth pointing out the results of [69, 6, 5] consider Corollary 1 where the boundary of the domain vary in time, however, our discussion is limited to fixed boundary domains.

Corollary 1 (See [5])**.** *Let $(\overleftrightarrow{B}_t)_{t\geq 0}$ be a Brownian motion over the constrained domain $\mathcal{D}$. For any t the distribution of $\overleftrightarrow{B}_t$ under the Lebesgue measure $\mathrm{d}\lambda$ is uniquely determined by a density*

$$\begin{cases} \frac{\partial}{\partial t}p_t(x) = \frac{1}{2}(\nabla\cdot\nabla)_{\mathcal{M}}p_t(x) \\ \frac{\partial}{\partial n}p_t(x_0) = 0 \end{cases}$$

where n is the outwards normal vector field of $\partial\mathcal{D}$.

[2]Under a suitably chosen dimension lift function.

Backwards diffusion process Having established the conditions for the existence and uniqueness of a solution to the forwards (normal) reflected SDE, in particular the form of the forwards transition kernels, we now discuss conditions for the matching backwards process $(\overleftarrow{\widetilde{B}}_t)_{t\geq 0}^T$ to be well posed.

The most general formulation of the two is that presented within [24] which approaches the problem from a topological perspective over (Euclidean) manifolds, building on the work of [14, 37]. We will begin by setting up existing results for the backwards diffusion process on domains possessing smooth boundaries consisting of one path connected component.

Lemma 1 (Extension of [5, Theorem 2.6] by [24])**.** *Let $u(s,x)$ be $C^1((0,T),\mathbb{R})$ in terms of s, $C^2(\mathcal{D},\mathbb{R})$ in x and $C^1(\overline{\mathcal{D}},\mathbb{R})$. Let $\mathcal{D}$ obey the conditions of Assumption 1. Then for any $s,t\in[0,T]$, $s\leq t$, we have*

$$\mathbb{E}\bigg[\int_s^t u(r,\overleftarrow{\widetilde{B}}_r)\,\mathrm{d}|K|_r\bigg] = \frac{1}{2}\int_s^t\int_{\partial\mathcal{D}} u(r,x)p_r(x)\,\mathrm{d}\mu(x)\,\mathrm{d}r,$$

where p_r is the density admitted by the Brownian process $(\overrightarrow{B}_r)_{r\geq 0}$ under the Lebesgue measure $\mathrm{d}\lambda$, and μ is the volume measure for the surface $\partial\mathcal{D}$.

The statement of this lemma is slightly modified from its statement in [5] by replacing the smoothness assumptions with Assumption 1, which we believe to be equivalent. Proof of this result was not provided in [24], to the best of our searching, so we reconstruct one below based on the proof of [5, Theorem. 2.6]. We opt to include this result in the main text, despite its length, as it is a central result to the validity of applying these techniques to training diffusion models in constrained settings.

Proof. Let $\mathcal{D}$ be a C^2 smooth bounded (Euclidean) manifold, a subspace of $\mathbb{R}^D$. Let $\epsilon>0$ and define $\psi:\mathbb{R}\to\mathbb{R}_{\geq 0}$ with

$$\psi_\epsilon(x) = \begin{cases} \frac{(\epsilon-x)^2}{2} & \text{if } 0\leq x\leq\epsilon, \\ 0 & \text{otherwise.}\end{cases}$$

Now for a distance measure $d:\mathbb{R}^D\times\mathbb{R}^D\to\mathbb{R}_{\geq 0}$, set $f_\epsilon(x)=\psi_\epsilon(\inf_{p\in\mathcal{D}^c} d(x,p))$, which for convince we will write as $\psi_\epsilon(d(x,\mathcal{D}^c))$. We will also make use of the shorthand notation $\mathcal{D}_\epsilon=\{x\mid 0\leq d(x,\mathcal{D})\leq\epsilon\}$. As $\partial\mathcal{D}$ is smooth we observe the following: $\forall x\in\mathbb{R}^D$

1. $0\leq f_\epsilon\leq\epsilon^2$;

2. $\|\nabla f_\epsilon\|_2 \leq c\epsilon$ for some constant $c > 0$;
3. $\nabla f_\epsilon(x_r) = \begin{cases} -\epsilon n(x_r) & \text{if } x_r \in \partial\mathcal{D}, \\ -(\epsilon - d(x, \mathcal{D}^c))n(x_r) & \text{if } 0 < d(x_r, \partial\mathcal{D}) \leq \epsilon, \\ 0 & \text{otherwise.} \end{cases}$
4. $\nabla \cdot \nabla f_\epsilon(x_r) = (1 + O(\epsilon))\mathbf{1}\{d(x_r, \mathcal{D}^c) \leq \epsilon\}$.

The validity of 4., which is needed in order to apply Ito's lemma later on, is not obvious as the second spacial derivative of f is not defined, so we will provide a distributional argument. Define a mollifier $\eta : \mathbb{R}^D \to \mathbb{R}_{\geq 0}$ with $\eta \in C_c^3$ and compactly supported on $B_1(0)$, with $\int_{\mathbb{R}^D} \eta(x)\,\mathrm{d}x = 1$, and $\int_{\mathbb{R}^D} \nabla \cdot \nabla(x)\,\mathrm{d}x\eta = 0$. Set

$$\eta_\delta(x) = \frac{1}{\delta^D}\eta\left(\frac{x}{\delta}\right), \quad \text{and} \quad f_{\epsilon,\delta}(x) = (f_\epsilon * \eta_\delta)(x).$$

First we claim that $f_{\epsilon,\delta} \to f_\epsilon$ uniformly as $\delta \to 0$ on the compact support $B_\delta(0)$. To see this, note that $f_{\epsilon,\delta}$ is compactly supported and due to the continuity of f_ϵ, for $\epsilon' > 0$ there exists a $\delta' > 0$ s.t.,

$$\|y\| \in B_{\delta'}(x) \quad \text{then} \quad |f_\epsilon(x - y) - f_\epsilon(y)| \leq \epsilon'.$$

Picking $\delta = \delta'$, gives for all y in the compact support $B_\delta(0)$

$$\begin{aligned} |f_{\epsilon,\delta}(x) - f_\epsilon(x)| &= |\int_{\mathbb{R}^D} f_\epsilon(x - y)\eta_\delta(y)\,\mathrm{d}y - f_\epsilon(x)\int_{\mathbb{R}^D} \eta_\delta(y)\,\mathrm{d}y| \\ &= |\int_{\mathbb{R}^D} [f_\epsilon(x - y) - f_\epsilon(y)]\eta_\delta(y)\,\mathrm{d}y| \\ &\leq \int \epsilon'\eta_\delta(y)\,\mathrm{d}y \\ &= \epsilon' \end{aligned}$$

and so $f_{\epsilon,\delta} \to f_\epsilon$ uniformly as $\delta \to 0$. Now, let $g \in C_c^\infty(\mathbb{R}^D, \mathbb{R})$ be a test function, and recall the definition of a generalized function derivative

$$\langle \nabla \cdot \nabla f_{\epsilon,\delta}, g\rangle = \langle f_{\epsilon,\delta}, \nabla \cdot \nabla g\rangle.$$

Now, since $f_{\epsilon,\delta} \in L^1(B_\delta(0))$, from [29, Lem 11.2], we have

$$\langle f_{\epsilon,\delta}, \nabla \cdot \nabla g\rangle \to^* \langle f_\epsilon, \nabla \cdot \nabla g\rangle$$

as $\delta \to 0$. The final expression for 5. follows from differentiating f_ϵ point-wise. Returning to the main result, for $\mathrm{d}x_r = \mathrm{d}\overrightarrow{B}_r$, i.e., for the forward dynamics from Corollary 1, we get after applying Ito's lemma that

$$\begin{aligned}
u(t,x_t)f_\epsilon(x_t) = u(s,x_s)f_\epsilon(x_s) & \\
&+ \int_s^t [u(r,x_r)\nabla f_\epsilon(x_r) + f_\epsilon(x_r)\nabla u(r,x_r)]\,\mathrm{d}\overrightarrow{B}_t \\
&+ \int_s^t u(r,x_r)\frac{\partial}{\partial n(x_r)}f_\epsilon(x_r)\,\mathrm{d}|K|_t \\
&+ \int_s^t [f_\epsilon \frac{\partial}{\partial r}u(r,x_r) + \frac{1}{2}u(r,x_r)\nabla\cdot\nabla f_\epsilon(x_r) + \frac{1}{2}f_\epsilon(x_r)\nabla\cdot\nabla u(r,x_r) \\
&\qquad + \nabla u(r,x_r)\cdot\nabla f_\epsilon(x_r)]\,\mathrm{d}r.
\end{aligned}$$

Dividing everything by ϵ, observe that for the component terms in the above expression we have (1)

$$\frac{f_\epsilon(x_r)}{\epsilon} = \frac{\psi_\epsilon(d(x_r,p))}{\epsilon} = \begin{cases} \frac{(\epsilon - d(x_r,p))^2}{2\epsilon} & \text{if } 0 \le d(x_r,p) \le \epsilon \\ 0 & \text{otherwise,} \end{cases}$$

and consequently $u(t,x_t)f_\epsilon(x_t) - u(0,x_0)f_\epsilon(x_0) \in O(\epsilon)$ for $\epsilon \to 0$; for (2) we leave it as is; in (3) we get

$$\frac{\partial}{\partial n}f_\epsilon(x_r) = \begin{cases} \frac{\partial\psi_\epsilon(x_r)}{\partial d(x_r,p)}\frac{\partial d(x,p)}{\partial n} & \text{for } x_r \in \mathcal{D}_\epsilon \\ 0 & \text{otherwise.} \end{cases} = \begin{cases} -(\epsilon - d(x_r,p)) & \text{if } x_r \in \mathcal{D}_\epsilon \\ 0 & \text{otherwise.} \end{cases}$$

since $\frac{\partial\psi_\epsilon(x_r)}{\partial d(x_r,p)} = -(\epsilon - d(x_r,p))$ for $x_r \in \mathcal{D}_\epsilon$ and $\frac{\partial d(x,p)}{\partial n} = 1$ for $x = vx'$ for some $x' \in \mathcal{D}_\epsilon$ and $v \in \mathbb{S}^{d-1}$; and for (4) by Taylor approximation and definition of the function

$$\begin{aligned}
\frac{1}{2\epsilon}\int_s^t \nabla\cdot\nabla f_\epsilon(x_r)\,\mathrm{d}r &= \frac{1}{2\epsilon}\int_s^t (1+O(\epsilon))\mathbf{1}\{x_r \in \mathcal{D}_\epsilon\}\,\mathrm{d}r \\
&= O(1)\int \mathbf{1}\{x_r \in \mathcal{D}_\epsilon\}\,\mathrm{d}r + \frac{1}{2\epsilon}\int \mathbf{1}\{x_r \in \mathcal{D}_\epsilon\}\,\mathrm{d}r.
\end{aligned}$$

Rearranging and simplifying by plugging in the above observations we get

$$\int_s^t u(r,x_r)\,\mathrm{d}|K|_t = -\frac{1}{\epsilon}u(t,x_t)f_\epsilon(x_t) + \frac{1}{\epsilon}u(0,x_0)f_\epsilon(x_0) + \frac{1}{\epsilon}\int_s^t u(r,x_r)\nabla f_\epsilon(x_r)\,\mathrm{d}\overrightarrow{B}_r$$

$$+\frac{1}{\epsilon}\int_s^t f_\epsilon(x_r)\nabla u(r,x_r)\,\mathrm{d}\overrightarrow{B}_r+\frac{1}{\epsilon}\int_s^t f_\epsilon(x_r)\frac{\partial}{\partial t}u(r,x_r)\,\mathrm{d}r$$
$$+\frac{1}{2\epsilon}\int_s^t u(r,x_r)\mathbf{1}\{x_r\in\mathcal{D}_\epsilon\}\,\mathrm{d}r+\frac{1}{\epsilon}O(1)\int_s^t u(r,x_r)\mathbf{1}\{x_r\in\mathcal{D}_\epsilon\}\,\mathrm{d}r$$
$$+\frac{1}{2\epsilon}f_\epsilon(x_r)\nabla\cdot\nabla u(r,x_r)\,\mathrm{d}r-\frac{1}{2\epsilon}\int_s^t\nabla u(r,x_r)\cdot\nabla f_\epsilon(x_r)\,\mathrm{d}r.$$

Now under expectation and collecting like terms this reduces to

$$\begin{aligned}\mathbb{E}[\int_s^t u(r,x_r)\,\mathrm{d}|K|_t]=&\mathbb{E}[\frac{1}{\epsilon}\int_s^t u(r,x_r)\nabla f_\epsilon(x_r)\,\mathrm{d}\overrightarrow{B}_r\\&+\frac{1}{\epsilon}\int_s^t f_\epsilon(x_r)\nabla u(r,x_r)\,\mathrm{d}\overrightarrow{B}_r\\&+\frac{1}{2\epsilon}\int_s^t u(r,x_r)\mathbf{1}\{x_r\in\mathcal{D}_\epsilon\}\,\mathrm{d}r]+O(\epsilon)\end{aligned}$$

this further simplifies, for sufficiently small step size of $(\overrightarrow{B_t})_{t\geq 0}$, to

$$\mathbb{E}[\int_s^t u(r,x_r)\,\mathrm{d}|K|_t]=\mathbb{E}[\frac{1}{2\epsilon}\int_s^t u(r,x_r)\mathbf{1}\{x_r\in\mathcal{D}_\epsilon\}\,\mathrm{d}r]+O(\epsilon).$$

Then by definition of the surface measure μ on $\partial\mathcal{D}$, and the definition of $|K|_t$ in Definition 6, as $\epsilon\to 0$ we replace the indicator and transition kernel to yield the desired expression

$$\mathbb{E}[\int_s^t u(r,\overleftrightarrow{B_r})\,\mathrm{d}|K|_r]=\frac{1}{2}\int_s^t\int_{\partial\mathcal{D}}u(r,x)p_r(x)\,\mathrm{d}\mu(x)\,\mathrm{d}r.$$

□

Having established the validity of Lemma 1 we now state the main result of [24], the poof of which relies on the forgoing result.

Theorem 3 (Time reversed SDE, see[24])**.** *Under the forgoing assumptions, there exists a coupled process* $(\overleftarrow{B_t},K_t)_{t\geq 0}$ *where* $K_t\in BV[0,T]$ *with*

$$\overleftarrow{X_t}=x_0+\overleftarrow{B_t}+\int_0^t\nabla_x\log p_{T-s}(\overleftarrow{X_s})\,\mathrm{d}s-K_t.$$

Consequently, we can say the reversed process $(\overleftarrow{\overleftarrow{B_t}})_{t\geq 0}$ is well defined over the selected domains and the reverse SDE can be expressed in the familiar form:

$$\mathrm{d}\overleftarrow{X_t} = \nabla_x \log p_t(\overleftarrow{X_t})\,\mathrm{d}t + \mathrm{d}\overleftarrow{\overleftarrow{B_t}}.$$

Before concluding this section, we state a straightforwards generalization of the above result to a forwards SDE that has a non-fixed scaling term[3].

Corollary 2. *Under all the same assumptions as in Theorem 3, and for $\sigma(t) : [0, T] \to \mathbb{R}_{>0}$ smooth, the SDE*

$$\mathrm{d}\overrightarrow{X}_t = \sigma(t) \circ \mathrm{d}\overrightarrow{\overrightarrow{B_t}}$$

admits a well defined time-reversed (reflected) equation.

Proof. Let $\sigma(t) \in C^2([0, T], \mathbb{R}_{>0})$ and set $\overrightarrow{Y}_t = \sigma(t) \cdot \overrightarrow{B_t}$, for the adapted process $(\overrightarrow{\overrightarrow{B_t}})_{t\geq 0}^T$ with $\overrightarrow{\overrightarrow{B_t}} = \overrightarrow{B_t} + \overrightarrow{K_t}$, so that we get a stochastic process $(Y_t)_{t\geq 0}^T$. Now consider the dynamics of X_t where

$$\mathrm{d}\overrightarrow{X}_t = \mathrm{d}\overrightarrow{Y}_t.$$

Following the same proof as for Theorem 3 but with $|K|_t = \int_0^t \mathbf{1}\{\overrightarrow{Y}_t \in \partial\mathcal{D}\}\,\mathrm{d}|K|_s$, absorbing the $\sigma(t)$ into K_t, and $K_t = \int_0^t n(\overrightarrow{Y}_t)\,\mathrm{d}|K|_s$ gives the claimed result. □

Example: Smooth constrained domain To close out this section, we give a toy example (defined below) that characterizes the form of assumptions discussed implicitly til now, which are detailed in Appendix A.4. As setup, suppose $\mathbb{R}^2$ is our ambient space. Let $r_1, r_2 \in \mathbb{R}_{>0}$ with $r_2 > r_2$, $p \in \mathbb{R}^2$, and set

$$\mathcal{D} = \{x \in \mathbb{R}^2 \mid r_2 \geq d(x, p) \geq r_1\}$$

to be our constrained manifold of interest; which is nothing more than an annulus. It is clear that $\partial\mathcal{D}$ is composed of finitely many smooth functions, so $\mathcal{D}$ is a smooth bounded

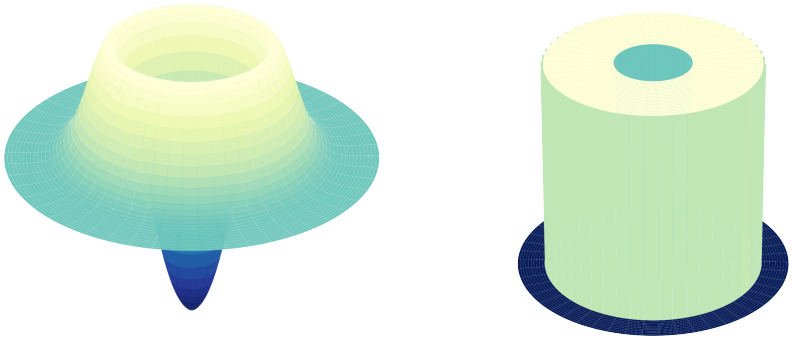
[3]While this result is not stated explicitly in [24], we believe it can fit into the proof provided by selection of the function u in the forgoing lemma. Nonetheless, we state it explicitly for clarity.

domain (in the topological sense Definition 13). Moreover, $\mathcal{D}$ satisfies the uniform (external and internal) sphere condition for radius $r = r_1/2$.

To show this domain satisfies Assumption 1, and thus admits a limiting reflected diffusion distribution, we may construct $\mathcal{D}$ based around Bolzman functions, which are of the form

$$f(x) = c - \exp\left\{-\frac{\|x-p\|_2^\gamma}{2r^2}\right\},$$

where the variables $c, r, \gamma \in \mathbb{R}$ control the described disc; in particular, c controls the intercept, γ the bound of derivative norm, and r the radius.



(a) Image of ϕ in $\mathbb{R}^3$. (b) Image of $\phi(x) \geq 0$ in $\mathbb{R}^3$.

Figure 4.2: Plot of (a) function ϕ and its (b) threshold image which generates domain D.

For simplicity of presentation, assume wlg $p = 0$, which can be achieved by appropriate translation of x, and let

$$\phi(x) = \exp\left\{-\left(\frac{\sqrt{x_1^2+x_2^2}-r_2}{r_1}\right)^2\right\} - \exp\left\{-\left(\frac{\sqrt{x_1^2+x_2^2}}{r_2}\right)^2\right\}.$$

An illustration of ϕ is depicted in Fig. 4.2. Moreover, below we show this $\phi(x)$ satisfies all the conditions in [93, 58] necessary to ensure the (forward) diffusion process - the forwards stochastic differential equations - with (normal) reflected boundary conditions possesses a well defined and unique solution. Observe, ϕ satisfies:

1. Boundedness: For any $B(0,r) \subseteq \mathbb{R}^2$, $\forall x \in B(0,r)$, $\exists L \in \mathbb{R}_{\geq 0}$ such that $\|\phi(x)\| \leq L$;

2. Is twice continuously differentiable: As ϕ is the sum of exponential functions it is (at least) twice continuously differentiable;
3. The gradient is lower bounded on the boundary, in particular, $\|\nabla\phi(x)\| \geq 1$ for $x \in \partial\mathcal{D}$: Firstly, with $d = \sqrt{x_1^2 + x_2^2}$, we have

$$\nabla\phi(x) = \left[\frac{2x_1}{r_1^2}\exp\left\{-\left(\frac{d}{r_1}\right)^2\right\} - \frac{2x_1(d-r_2)}{r_1^2 d}\exp\left\{-\left(\frac{d-r_2}{r_1}\right)^2\right\},\right.$$
$$\left.\frac{2x_2}{r_1^2}\exp\left\{-\left(\frac{d}{r_1}\right)^2\right\} - \frac{2x_2(d-r_2)}{r_1^2 d}\exp\left\{-\left(\frac{d-r_2}{r_1})^2\right\}\right].$$

Now for $x \in \partial\mathcal{D}$, if $d = r_1$ the above reduces to

$$\nabla\phi(x) = \left[\frac{2x_1}{r_1^2}\exp\{-1\} - \frac{2x_1(r_1-r_2)}{r_1^3}\exp\left\{-\left(\frac{r_1-r_2}{r_1}\right)^2\right\},\right.$$
$$\left.\frac{2x_2}{r_1^2}\exp\{-1\} - \frac{2x_2(r_1-r_2)}{r_1^3}\exp\left\{-\left(\frac{r_1-r_2}{r_1}\right)^2\right\}\right].$$

Now observe, since $r_1 \leq x \leq r_2$ and the inequality $e^x \geq 1 + x$, we have for the first term

$$\begin{aligned}\left[\frac{2x_1}{r_1^2}\exp\{-1\} - \frac{2x_1(r_1-r_2)}{r_1^3}\exp\left\{-\left(\frac{r_1-r_2}{r_1}\right)^2\right\}\right]^2 &\geq \left[\frac{2x_1(r_1-r_2)}{r_1^3}\exp\left\{-\left(\frac{r_1-r_2}{r_1}\right)^2\right\}\right]^2 \\ &\geq \frac{4x_1^2(r_1-r_2)^6}{r_1^{10}}\end{aligned}$$

and applying the same reduction to the second term we get

$$\begin{aligned}\|\nabla\phi(x)\| &\geq \left[4(x_1^2+x_2^2)\frac{(r_1-r_2)^6}{r_1^{10}}\right]^{\frac{1}{2}} \\ &\geq 2\left[\frac{(r_1-r_2)^6}{r_1^9}\right]^{\frac{1}{2}}.\end{aligned}$$

As a result of this approximation not being tight, the parameter space (r_1, r_2) is not convex; with approximate lower bound $r_2 \geq (0.5 + r_1)^2$. An exemplar set of parameters that does satisfy the condition are $r_1 = 1, r_2 = 2$. It was implicitly assumed $c = 0$ and $\gamma = 2$ throughout.

4. The boundary $\partial\mathcal{D}$ satisfies the uniform sphere condition: It is evident for $0 < r \leq r_1$ the uniform sphere condition is satisfied.

With this motivating example in mind, it is easy to see how one can generalize the forgoing results to encompass domains constructed by various ϕ functions of the above form. Examples of training diffusion models over this kind of domain are postponed till Section 5.4.1.

4.2 Neural reflected diffusion models

Particular care must be given to the parameterization chosen for neural reflected diffusion models and the score matching loss, since the noise is no longer (globally) Gaussian which results in a few additional complications to the learning task. We begin by discussing some loss selection and evaluation considerations necessary for training these models, and then introduce specific methods of constraining the model output for domains with boundary.

Loss selection: It was shown in the works [24, 61] that the ISM loss, Eq. (2.6), will converge (up to a constant difference) to the correct score for reflected diffusion. Consequently, ISM loss remains a viable option for training, alongside direct score parameterization of the model as discussed in Section 3.3. Concurrent to this, [61] goes on to show, thanks to an assumed[4] global diffeomorphism mapping between $\mathcal{D}$ and the unit hypercube $\mathbb{H}^d$, denoising score matching Eq. (2.7) has an analogous extension to constrained domains, namely

$$\mathcal{L}_{CDSM}(\theta) = \mathbb{E}^{\mathcal{D}}_{\overrightarrow{X}_0 \sim p_0} \mathbb{E}^{\mathcal{D}}_{\overrightarrow{X}_t \sim p_t(\overrightarrow{X}_t|\overrightarrow{X}_0)} [\| s_\theta(\overrightarrow{X}_t) - \nabla_x \log p_t(\overrightarrow{X}_t|\overrightarrow{X}_0) \|_2^2]. \tag{4.1}$$

Methods for evaluating the ISM loss over manifolds, which $\mathcal{D}$ is a subset of, were discussed in Section 3.3 so we will not repeat this discussion here. The only modification necessary is the incorporation of the above boundary parameterization given below in Section 4.2. On the other hand, when using Eq. (4.1), we cannot compute the conditional marginals $p_t(X_t|X_0)$ in closed form, thus, in order to compute the constrained denoising score matching objective [61], for "small" $\sigma(t)$ we may utilize a mixture of Gaussian's approximation [68]

$$p_t(X_t|X_0) \approx \sum_{i=1}^{N} w_i(X_0) \frac{1}{(2\pi\sigma_t^2)^{d/2}} \exp\left\{ -\frac{1}{2\sigma_t^2} \|X_t - x_i\|_2^2 \right\}$$

[4]Equivalent to the isometric embedding assumptions used in Section 3.2 for convex domains.

where the sum is evaluating the pdf of $\mathcal{N}(X_0, \sigma_t^2 \mathbb{I})$ over $\{x_1, \ldots, x_N\} \subseteq \mathcal{D}$ a set of samples drawn following Algorithm 6, and $w_1, \ldots, w_N$ are weighting functions used to model the conditional behaviour on X_0.

Model parametrization and training: In either loss formulation, it was empirically found in [61] that it is necessary to enforce the the boundary condition by scaling the output of the neural network by a monotone decreasing (smooth) function $h : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ with $h(x) = 0$ for all $x \in \partial\mathcal{D}$. The paramterization chosen in [24] is

$$s_\theta^{\mathcal{D}}(X_t, t) = \min\{1, \text{ReLU}(d(X_t, \partial\mathcal{D}) - \delta)\} \cdot s_\theta(X_t, t), \tag{4.2}$$

where $\delta > 0$ is a chosen boundary margin and $d : \mathcal{D} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$ is a distance function. This function ensures that as points approach the boundary the magnitude of the score goes to zero within the set margin from the boundary. Thus, when sampling using Eq. (2.5) points can't be pushed outside of the domain.

Just as done in Section 3.3, when these methods are applied outside of the Euclidean domain, depending on the chosen parametrization, the score must be mapped (projected) onto the tangent plane of the manifold. In particular, for a score parametrized model s_θ, not necessarily constrained to $\mathcal{D}$, the loss computation – model training – is carried out following Algorithm 5.

Algorithm 5: Manifold (reflected) diffusion training.
Consider diffusion over the manifold $\mathcal{D}$ which lies on the Riemannian manifold $\mathcal{M}$. Let s_θ be a score parametrized diffusion model, DATA the training dataset, σ diffusion coefficient, N the number of discretization steps, $\eta > 0$ the learning rate, and $\{\phi_i\}_{i \in I}$ a set of constraints that define $\partial\mathcal{D}$.

Data: $\{\phi_i\}_{i \in I}$, DATA, N, s_θ, σ, η
Result: θ
while *training* **do**
  $x_0 \sim$ DATA
  $t_i \sim \mathcal{U}[0, 1]$
  $k \leftarrow \lfloor \sigma(t_i)/\sigma(T) \rfloor N$
  $x_t \leftarrow$ MANIFOLD-REFLECTED-STEP$(\{\phi_i\}_{i \in I}, \sigma(t_i), k, x_0)$
  $\mathcal{L} \leftarrow \mathcal{L}_{ISM}(\text{PROJ}(x_t, s_\theta))$
  $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}$
return θ

Sampling To sample the (forwards) reflected diffusion process,

$$\mathrm{d}\overrightarrow{X}_t = \mathrm{d}\overrightarrow{\sim}_{B_t}$$

that we started with at the beginning of this section, which is a necessary step in Algorithm 5, we will follow the Metropolis-Hasting version of the Euler-Maruyama scheme proposed in [24], which we now restate as Algorithm 6. This algorithm, which as written in [24, Appx.C], is a discretization of the Riemann manifold heat equation given in Section 3.1[5].

Algorithm 6: (Reflected step proxy) Metropolis-Hasting Random Walk.
Consider diffusion over the manifold $\mathcal{D}$ which lies on the Riemannian manifold $\mathcal{M}$. Let $T \geq 0$ be a given time, N the number of discretization steps, X_0 a starting point, and $\{\phi_i\}_{i \in I}$ a set of constraints that define $\partial\mathcal{D}$. [24].

Data: $\{\phi_i\}_{i \in I}, T, N, X_0$
Result: $\{X_k\}_{k=0}^{N}$
$\Delta t \leftarrow T/N$
for $k = 0, 1, \ldots, N-1$ **do**
  $Z_{k+1} \leftarrow \mathcal{N}(0, \mathbb{I})$
  $X' \leftarrow \exp_g(X_k, \sqrt{\Delta t} Z_{k+1})$
  if $\min_{i \in I} \phi_i(X') \geq 0$ **then**
    $X_{k+1} \leftarrow X'$
  else
    $X_{k+1} \leftarrow X_k$
return $\{\tilde{X}_k\}_{k=0}^{N}$

After training, the model can be sampled using a projected form of Langrangien dynamics, Eq. (2.5), as mentioned in prior sections where points are transported along the domain through dscretized projection integration steps.

[5]This differs from the SDE given in the algorithm caption stated Appx.C of [24]; if one discretizes the SDE in the caption they will not recover the correct algorithm update step. We have corrected this error.

Chapter 5

Structure preserving diffusion models

In keeping with the theme presented when discussing reflected diffusion, the notion of structure preserving diffusion models is best conveyed by observations to properties present in solutions to physical systems, or data generated from such systems. Such data tend to posses certain forms of symmetry indicative of the problem considered, e.g., a drug molecule is unchanged by its orientation within a fluid and when rotated around certain points remains unchanged in form. Consequentially, if one has prior knowledge of the symmetries to solution of a known problem, these symmetries can be used to constrain the list of candidates to only those that satisfy these symmetries. This often results in considerable speedup in under-posed tasks where the search space can be quite large, containing many different orientations of the same principle candidate solution.

As a result of this exchangability between solution and symmetry constraints, there has been considerable work attempting to constrain neural networks to respect group-invariant (or equivariant) distributions [88], particularly, for generations tasks, e.g., drug discovery, where typically multiple solutions exist for any given set of input parameters. This is underscored by the widespread utilization of diverse forms of data augmentation; however, achieving perfect group invariance (or equivariance) by data augmentation alone necessitates infeasibly many training samples, with models often falling short of being adequately conditioned through data augmentation alone [22, 25]. Consequently, various more principled approaches have started to gain popularity in the last few years.

Notably within diffusion models [91, 90, 33, 48, 102], and diffusion bridge models [16], have focused primarily on applications in molecule generation (e.g., molecular conformation, and protein backbone generation) [89, 101, 35, 102, 46, 12, 66]. Most of these approaches can be broadly described as conditioning the diffusion process on a graph prior that represents the unconformed molecule, and employing a transformation (applied to the inner molecular atomic distances - such as the relative torsion angle coordinates [46]) that produces a group-invariant form (or one that is more robust to the selected group transformations). This, thereby, results in a representation that is sufficient to ensure the diffusion process is equivariant. More generally, [15, 67, 102], investigate distribution invariance over

more general geometries (e.g., Riemannian manifolds generated by Lie groups). The study of distribution invariance comes about naturally as a result of finding a limiting probability distribution over the geometry in these settings, a requirement for the diffusion process to be well-defined.

In the works [64, 65] we extend existing theoretical results, developed within the forgoing works, by providing a complete characterization of the necessary and sufficient conditions on the drift and diffusion terms to ensure a diffusion process preserves invariants; in particular, invariances that can be expressed in terms of a group of isometry transformations. I will now summarize these results and provide some extensions to alternative diffusion processes, reflected diffusion in constrained settings, and manifold domains.

5.1 Invariant diffusion processes

To put this discussion on more solid terms, we must formalize the notion of an "invariant" distribution. Throughout the following, we are interested in sampling from a $\mathcal{G}$-invariant distribution with a smooth density function $p : \mathbb{R}^d \to \mathbb{R}_{>0}$ that is non-zero almost everywhere[1] which satisfies:

Definition 7 ($\mathcal{G}$-invaraint distribution)**.** *For a given group $\mathcal{G}$, a distribution p is $\mathcal{G}$-invariant if for all closed balls $\overline{B} \subseteq \mathbb{R}^d$ and $h \in \mathcal{G}$,*

$$\int_{\overline{B}} p(x)\,\mathrm{d}x = \int_{h(\overline{B})} p(z)\,\mathrm{d}z,$$

where $h(\overline{B}) = \{h(x) \mid x \in \overline{B}\}$.

In this study we restrict our discussion to groups comprised of linear isometries. This restriction is made to exclude transformations that would, for pixel-space diffusion models, cause value shifts away from the assumed mean value; e.g., transformations that scalar multiply the pixel values in the image by a scalar amount – for images with non-zero mean, avoiding aberrant scaling factors within the learned score and proof of Theorem 4.

[1] This assumption can be easily satisfied by convolving the target (or empirical proxy) distribution with a Gaussian kernel, to produce a mollified distribution.

5.2 Invariance conditioning

In the works [65, 64] we consider a general diffusion bridge SDE of the form

$$\mathrm{d}\overrightarrow{X}_t = f(\overrightarrow{X}_t, y, t)\,\mathrm{d}t + g(t)\,\mathrm{d}\overrightarrow{B}_t, \quad X_0 \sim p(x_0 \mid y), \tag{5.1}$$

that bridges a (probability) path q_t between (data) distributions $q_0(x) = q_{\text{data}}(x)$ and $q_T(y) = q_{\text{data}}(y)$, where $(x, y) \sim q_{\text{data}}(x, y)$. DBs leverage the distribution p_t induced by Eq. (2.1) with $X_0 = x$ and $X_T = y$ to sample X_t. In this way,

$$q_t(X_t) = \mathbb{E}_{(X,Y)\sim q_{\text{data}}(X,Y)}\big[p_t(X_t \mid X_0 = x, X_T = y)\big].$$

Such a process encompass regular diffusion processes seen in Section 2.2 by setting $q_T = \mathcal{N}(0, \Sigma_t)$ to recover the standard, conditional, diffusion equations. In fact, Eq. (5.1) is a slightly more general version of the standard diffusion bridge equation due to the inclusion of the conditioning variable y which we use to encode other factors affecting the process which need not have the same shape as X_t.

The central result from the aforementioned works is the following theorem, which says, given a data distribution, that is assumed to be $\mathcal{G}$-invariant, describes sufficient and necessary conditions on the form of drift terms of diffusion processes that preserve the $\mathcal{G}$-invariance throughout the diffusion trajectory. While sufficient conditions have been established for the case where Y lies in the same space as X_t, this result extents these to more general conditional variables and provides new necessary conditions which convey design insights for equivariant bridge models.

Theorem 4 (Diffusion invariance characterization, [65])**.** *Given a forward diffusion process, such as in Section 2.2 or a more general diffusion bridge process, with $\mathcal{G}$-invariant $p_0(X \mid Y)$, let $[0]_{p_t}$ be the set of ODE drifts that preserve the distribution p_t; meaning they preserve the drift. Then $p_t(X_t|Y)$ is $\mathcal{G}$-invariant for all $t \geq 0$ if and only if*

$$\kappa_1^{-1} \circ f(\kappa_1 X_t, \kappa_2 Y, t) - f(X_t, Y, t) \in [0]_{p_t} \tag{5.2}$$

for all $t > 0$, $X \in \mathbb{R}^m$, $Y \in \mathbb{R}^n$ and $\kappa \in \mathcal{G}$.

Formalizing the above note, existing structure-preserving diffusion models mentioned above, are based on the special case that $\kappa_1^{-1} \circ f(\kappa_1 X_t, \kappa_1 Y, t) - f(\overrightarrow{X}_t, Y, t) = 0$, thus the above theorem is expressly more general in terms of conditioning and permissible drift terms. As a note, the above results reduces to the standard unconditional case when $y = \emptyset$, with $\mathcal{G}$ reducing to a non-coupled group of elements, and Eq. (5.2) becoming $\kappa_1^{-1} \circ f(\kappa_1 X_t, t) - f(X_t, t) \in [0]_{p_t}$.

5.2.1 Conditioning methods

Below we discuss three methods of constraining diffusion models over discrete groups of linear isometries. These methods are subsequently evaluated against each other in addition to some baseline methods in Section 5.3 over a variety of different groups and datasets.

Method: Weight-tied convolutions In [65, 64] we focused on diffusion models based on the U-Net backbone [86, 84] in which the only components that are not equivariant are the CNNs layers. Thus, we replace them with group-equivariant CNNs [11, 82, 23, 52, 51] to make the entire network equivariant.

For the particular case of linear isometries, we can construct equivariant CNN layers by constraining convolution kernel weights. In particular, for a given linear group $\mathcal{G}$, we can construct a group equivariant convolution kernel $k \in \mathbb{R}^{d\times d}$, of the form:

$$k = \begin{array}{|c|c|c|c|} \hline k_{1,1} & k_{1,2} & \cdots & k_{1,d} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & & \vdots \\ \hline k_{d-1,1} & k_{d-1,2} & \cdots & k_{d-1,d} \\ \hline k_{d,1} & k_{d,2} & \cdots & k_{d,d} \\ \hline \end{array},$$

such that for any $h \in \mathcal{G}$ and $X \sim p_0$, we have $h(k * X) = k * h(X)$ by constraining the individual kernel weights to obey a system of equalities set by the group invariance condition $h(k) = k$.

Example: The C_4 cyclic and D_4 dihedral group. Recall that the C_4 cyclic group is composed of planar $\pi/2$ rotations about the origin, and can be denoted as $C_4 = \{e, r_1, r_2, r_3\}$ where r_i represents a rotation by $i\pi/2$ radians. Taking a convolution kernel $k \in \mathbb{R}^{5\times 5}$ and constraining it to be C_4-equivariant results in k being of the form:

$$k = \begin{array}{|c|c|c|c|c|} \hline a & b & c & d & a \\ \hline d & e & f & e & b \\ \hline c & f & g & f & c \\ \hline b & e & f & e & d \\ \hline a & d & c & b & a \\ \hline \end{array}.$$

The D_4 dihedral group can then be constructed from C_4 by adding the vertical flipping operation to the past example; that is, $D_4 = \{e, r_1, r_2, r_3, f_x, f_x \circ r_1, f_x \circ r_2, f_x \circ r_3\}$. This

requires further constraints to k so that:

$$k = \begin{array}{|c|c|c|c|c|} \hline a & b & c & b & a \\ \hline b & e & f & e & b \\ \hline c & f & g & f & c \\ \hline b & e & f & e & b \\ \hline a & b & c & b & a \\ \hline \end{array}.$$

Naturally, constraining convolution kernels in this fashion has the computational advantage of reducing the number of model parameters – with a possible loss in expressiveness when the kernel size is relatively small in comparison to the size of the group and structure of the data. For a more general discussion on $\mathcal{G}$-equivariant convolution kernels in the context of CNNs see [11].

Method: Equivariance regularization Instead of achieving $\mathcal{G}$-equivalence by adopting specific model architectures, as in Section 5.2.1, we can also directly add a regularizer to the score-matching loss, Eq. (2.7), to inject this preference. Specifically, from Theorem 4 we know the estimated score $s_\theta(X_t, t)$ is equivariant if

$$s_\theta(\kappa X_t, \kappa Y, t) = \kappa s_\theta(X_t, Y, t)$$

for all $\kappa \in \mathcal{G}$; for the unconditional setting, an equivalent technique can be applied by omitting the second argument. Thus, we propose the following regularizer to encourage the two terms to match for all X_t:

$$\mathcal{R}(\theta, \overline{\theta}) = \mathbb{E}\left[\frac{1}{|\mathcal{G}|}\sum_{\kappa\in\mathcal{G}} \|s_\theta(\kappa X_t, \kappa Y, t) - \kappa s_{\overline{\theta}}(X_t, Y, t)\|_2^2\right]$$

where the expectation is taken over the same variables in Eq. (2.7) and $\overline{\theta}$ denotes the exponential moving average (EMA) of the model weights

$$\overline{\theta} \leftarrow \text{stopgrad}(\alpha\overline{\theta} + (1-\alpha)\theta) \text{ with } \alpha \in [0, 1),$$

which, at least empirically, has been shown to improve training stability [96, 33]. In practice, iterating over all elements in $\mathcal{G}$ may be computationally prohibitive, instead for each optimization step, $\mathcal{R}(\theta, \overline{\theta})$ is one-sample approximated by:

$$\mathcal{R}(\theta, \overline{\theta}) \approx \mathbb{E}\left[\|s_\theta(\kappa X_t, \kappa Y, t) - \kappa s_{\overline{\theta}}(X_t, Y, t)\|_2^2\right],$$

with randomly picked $\kappa \in \mathcal{G}$. This regularizer in practice, see Table 5.1 and Table 5.2 – the details of which are described in Section 5.3, was shown to outperform data-augmentation alone. We suspect this is due to it providing a more accurate one-sample estimate of the invariant gradient compared to that provided by an augmented batch alone. With that said, one group element sample does not appear to sufficiently condition the score, as can be seen by the Δx_0 reconstruction error.

This regularization technique is very similar in formulation to the more principled approach of frame averaging (FA) [80][2] which we ended up utilizing in [65] across all the experiments.

Method: Frame averaging When $\mathcal{G}$ contains finitely many elements, we can achieve $\mathcal{G}$-equivariance through frame averaging (FA) [80], leveraging the following fact: for any function $r : \mathbb{R}^d \to \mathbb{R}^d$,

$$\tilde{r}(x, y) = \frac{1}{|\mathcal{G}|} \sum_{k \in G} k^{-1} r(kx, ky)$$

is $\mathcal{G}$-equivariant. The second argument of r can be discarded for the approximation of the score not conditioned on y. Based on this fact, we can obtain an equivariant estimator $\tilde{s}_\theta$ of the score by setting $r = s_\theta$. Note that unlike other FA-based diffusion models, notably [66, 19], we train our models using regular denoising score-matching, Eq. (2.7), and only adopt FA during sampling. This design significantly saves training costs while theoretically sacrificing nothing.

5.3 Experiments

Here we summarize the key experiments and empirical results conducted in [65, 64], which were used to evaluate the effectiveness of the proposed methods in Section 5.2.1 at preserving symmetries present in the target distributions. The main experiments encompass a set of image generation, denoising, and style-transfer tasks detailed below in Section 5.3 and Section 5.3 respectively. A qualitative comparison of sample quality is given in Appendix B.2.

[2]We ended up rediscovering the frame averaging technique independently during development only to have the existance of FA pointed out to us by a reviewer.

All methods are trained using the same set of common data-augmentation techniques[3], unless otherwise stated to remove this as a factor in performance comparisons; thus, it is important to reiterate that we do not expect the proposed models to greatly exceed the baseline in perceptual image quality, as measured by Fréchet intercept distance (FID) [32] and structural similarity index measure (SSIM) [100], rather, these methods should be evaluated on retention of baseline quality while offering guarantees on learned invariances not offered by standard methods. To this end, we report the absolute reconstruction error between samples generated by cycling the condition through the invariance group. This evaluation metric, denoted $\Delta\hat{x}_0$, is minimized when a model is able to perfectly reconstruct the same sample for all orientations of the condition but is otherwise equal to the largest deviation between such samples. Apart from this, we introduce a metric called *Inv-FID* in an attempt to quantify the learned invariance in the sampling distribution. Inv-FID calculates the maximum FID between a set of samples $D_s \sim p_0$ and $\kappa_1(D_s)$ for $\kappa \in \mathcal{G}$. If p_0 is perfectly $\mathcal{G}$-invariant, applying any κ to its outcomes would leave the resulting distribution unchanged.

Emprical datasets Here we adopt C_4 Rotated MNIST [55], LYSTO [44], and ANHIR [4] datasets that have been used in the past to evaluate the equivaraince of generative models [18, 3]. We also validate our models effectiveness on the style transfer tasks of denoising LYSTO images, and converting CT scan images to PET scan images of the same patients from the dataset [26].

Rotated MNIST dataset contains random $\pi/2$ rotations of MNIST images [17], resulting in a C_4-invariant distribution[4]. This dataset has been used to evaluate group-invariant CNN models previously, as seen in [18, 3], with experiments performed on 1% (600), 5% (3000), and 10% (6000) of the dataset used to evaluate a model's robustness to limited data. The LYSTO dataset [44] includes 20,000 image patches from breast, colon, and prostate cancer samples stained with CD3 or CD8 dyes, exhibiting D_4 invariance. Following [3], models are trained on randomly selected (64x64x3) crops from the scaled-down (128x128x3) LYSTO dataset. ANHIR dataset [4] provides (15kx15k) images of lesions, lung-lobes, and mammary-glands, from which we extract random (64x64x3) patches from lung-lobes according to the method used in [18][5].

The CT-PET dataset [26] includes 1014 annotated whole-body paired FDG-PET/CT scans of patients with malignant lymphoma, melanoma, and non-small cell lung cancer.

[3]random rotation, noise perturbation, contrast adjustment, etc.

[4]Note, here there will be ambiguity between 6 and 9 digits.

[5]I would like to thank, Neel Dey, for providing the pre-processed ANHIR dataset used in [18] and for clarifying some details around how the computation of FID was carried out within the forgoing paper.

We processed this volumetric dataset by extracting a set of median 2D slices along the long axis of each patient scan. The resulting images were center cropped to a final resolution of (256x256x3), resulting in the final used dataset that is invariant under horizontal flipping[6].

Experiment: Image generation For the image generation tasks on the Roated MNIST, LUSTO, and ANHIR datasets, we train a standard diffusion model, VP-SDE [33, 90], as a baseline measure of performance and error in adherence to equivariance properties, alongside SP-GAN [3], the only GAN-based model with theoretical group invariance guarantees, and report the mean performance of GE-GAN [18]. These are pitted against the VP-SDE diffusion models conditioned using the techniques Section 5.2.1 referred to respectively as SPDM+WT, SPDM+Reg, and SPDM+FA.

The performance results of each model on Rotated MNIST are reported in Table 5.1 with LYSTO and ANHIR reported in Table 5.2. To ensure benchmark consistency, we reproduced the results of SP-GAN and GE-GAN. FID was computed using the standard InceptionV3 model with features averaged over the chosen invariance group; details of our FID calculation are provided in Appendix B.1. We note the reproduced FID values of GE-GAN are significantly higher than those self reported in [18]. This is due to the author's fine-tuning the InceptionV3 model on the LYSTO and ANHIR datasets. While we include these results in the table for reference, the scores are not comparable with other FIDs due to fine-tuning. All FIDs are based on 50,000 randomly generated images in order to ensure low variance in the FID computation.

As shown in Table 5.1, diffusion models with theoretical guarantees tend to achieve lower Inv-FID and $\Delta\hat{x}_0$ scores. Interestingly, the differences in Inv-FID scores across diffusion models are relatively small, corroborating the statement that VP-SDE diffusion models are structure preserving in principle. Therefore, in situations where invariance in the sampling distribution is not crucial, standard diffusion models may be sufficient. However, differences emerge between diffusion models when evaluating $\Delta\hat{x}_0$, where only the equivariant models prove capable of accurate equivariant sampling.

Experiment: Image denoising As image up-scaling, denoising, and sharpening are common applications for diffusion models, we propose a denoising (or deblurring) task for evaluating model performs on denoising images under a rotational invariance prior. To create this dataset, LYSTO (64x64x3) patches are downscaled to (16x16x3) using linear

[6]This dataset is under restricted license and due to patient privacy concerns, we are not able to distribute any related data or model checkpoints.

Table 5.1: SPDM Robustness to lack of data C_4 MNIST.

	FID↓				Inv-FID↓	$\Delta\hat{\mathbf{x}}_0 \downarrow$
Model	1%	5%	10%	100%	100%	100%
VP-SDE	5.97	**3.05**	3.47	2.81	2.21	0.2997
SPDM+WT	5.80	3.34	3.57	3.50	2.20	0.0004
SPDM+FA	**5.42**	3.09	**2.83**	**2.64**	**2.07**	**0.0002**
SPDM+Reg	**5.42**	3.69	**2.83**	2.75	2.09	0.1806
SP-GAN	149	99	88	81	–	–
SP-GAN (Reprod.)	16.59	11.28	9.02	10.95	19.92	–
GE-GAN	–	–	4.25	2.90	–	–
GE-GAN (Reprod.)	15.82	7.44	5.92	4.17	58.61	–

Table 5.2: SPDM Comparison on LYSTO and ANHIR.

	LYSTO			ANHIR		
Model	FID↓	Inv-FID↓	$\Delta\hat{\mathbf{x}}_0 \downarrow$	FID↓	Inv-FID↓	$\Delta\hat{\mathbf{x}}_0 \downarrow$
VP-SDE	7.88	0.66	20.77	8.03	0.57	39.82
SPDM+WT	12.75	**0.59**	**0.00**	11.73	0.43	**0.00**
SPDM+FA	**5.31**	0.6	**0.00**	**7.57**	**0.31**	**0.00**
SP-GAN	192	–	–	90	–	–
SP-GAN (Reprod.)	16.29	0.66	–	17.12	0.28	–
GE-GAN	3.90	–	–	5.19	–	–
GE-GAN (Reprod.)	23.20	27.84	–	14.16	6.87	–

interpolation and training pairs are formed, (I_{blur}, I_{ref}), between the downscaled condition and reference image.

For comparison, we train a diffusion bridge model (DDBM) [103], the popular style-transfer method Pix2Pix [40], and the unconditional diffusion bridge model I^{2}SB [59]. These are compared against an equivariant DDBM implementation, denoted SPDM+FA, equipped with a modified sampling procedure that utilizes FA and a precomputed noise sequence to fix image orientation during sampling. All models are implemented in pixel-space for this comparison. Model performance metrics are reported in Table 5.3. The SPDM+FA performed best across all evaluation metrics, particularly in $\Delta\hat{x}_0$ reconstruction.

Experiment: CT-PET style transfer For the CT to PET scan style-transfer task, we make use of the CT-PET dataset [26] by attempting to transform a patient's CT scan into the matching PET scan. As baselines for this task, we consider again implementations of DDBM, Pix2pix, and I^{2}SB. These are compared against a DDBM model that is modified in a similar way to that in the above image denoising task, denoted SPDM+FA, with the model utilizing FA to ensure invariance to horizontal image flipping. Due to GPU memory limitations, it was necessary to modify both DDBMs and I^{2}SB to act on latent spaces in the fashion proposed in [83]. See Section 5.5 for further implementation details.

As can be seen in the Table 5.3 the SPDM+FA implementation achieves the lowest FID and invariant reconstruction error $\Delta\hat{x}_0$ of all the tested models.

Table 5.3: SPDM Comparison on LYSTO denoising and CT-PET style transfer datasets.

	LYSTO				CT-PET			
Model	FID↓	$L_1 \downarrow$	SSIM↑	$\Delta\hat{\mathbf{x}}_0 \downarrow$	FID↓	$L_1 \downarrow$	SSIM↑	$\Delta\hat{\mathbf{x}}_0 \downarrow$
DDBM	17.28	0.076	0.696	0.8884	18.13	**0.041**	0.861	0.9233
SPDM+FA	**16.21**	**0.071**	0.721	**0.0001**	**17.74**	0.042	0.860	**0.0000**
Pix2Pix	78.43	0.087	0.654	0.8629	20.26	0.043	**0.862**	1.3196
I^2SB	20.45	0.073	**0.722**	0.8683	27.51	0.051	0.832	1.2123

5.4 Manifold structure preserving diffusion models

If the domain we happen to be operating within has a symmetric density under some group isometry, then, for Euclidean domains, we may apply the results and techniques from Section 5.1 to aid in task learning. Now, as much of our interest lies in learning tasks over manifolds, we should point out the proposed approach differs from [63] which exploits symmetry of the domain to improve score matching estimation for domains with curvature by instead exploiting symmetry of the distribution.

In setting up the example, we note the theorem in [65], as written, does not immediately extend to domains with local curvature. Thus, we will limit the immediate presentation to a Euclidean region that is encompassed by the developed theory without the need for modification.

Experiment: Symmetric density estimation on (Euclidean) disc Here we demonstrate the application of $+FA$, as outlined above, to the problem of learning a toy (symmetric) density over a symmetric domain; in particular, we consider the domain $\mathcal{D}$ from Section 4.1 equip with polar coordinates and density function $p(r,\theta) = \frac{1}{A}h(r,\theta)$ where

$$h(r,\theta) = \int_0^{2\pi}\int_{1/4}^{1} g(r',\theta')G(r-r',\theta-\theta')r'\,\mathrm{d}r'\,\mathrm{d}\theta'$$

with the component functions equal to

$$g(r,\theta) = \begin{cases}\rho_1 & \text{if } \cos(\kappa_\theta\theta)\sin(\kappa_r r) > 0\\ \rho_2 & \text{otherwise.}\end{cases}, \quad G(r',\theta') = \frac{1}{\pi\sigma^2}\exp\left\{\frac{-r'^2-\theta'^2}{2\sigma^2}\right\}$$

for $1 \geq r \geq 1/4$ and where $\rho_1, \rho_2, \kappa_\theta, \kappa_r, \sigma$ are chosen constants, and we have the normalizing factor

$$A = \int_{1/4}^{1} \int_{0}^{2\pi} h(r, \theta) r \, \mathrm{d}\theta \, \mathrm{d}r.$$

An illustration of this density, that has been smoothed using a non-zero Gaussian kernel, is illustrated in Fig. 5.1.

(a) (b)

Figure 5.1: Plot of proposed radial checkerboard density function over a unit disc with a hole punched through the center (a) along with (uniform) rejection sampled approximation (b) which is used for model training.

Here we train a (non-equivariant) DDPM and SPDM+FA model (without domain constraints – for $0 \leq r \leq 1$) on this dataset under the C_{κ_θ} group, for $\kappa_\theta = 5$, since the density is invariant under such rotations about the origin. All models are trained over $500,000$ steps using a batch size of $10,000$, and 100 diffusion steps. The learned densities are sampled and illustrated in Fig. 5.2. It is apparent in the figure the SPDM+FA model was better able to learn the high frequency information present towards the center of the distribution, moreover, the model converged faster on the macroscopic features based on incremental qualitative observations made every 20,000 steps.

5.4.1 Structure preserving reflected diffusion

As noted in [61], the addition of the reflected term does not alter the general form of the PF-ODE [90], so the core proposition from [65] remains true under reflected diffusion, at least

(a) (b) (c)

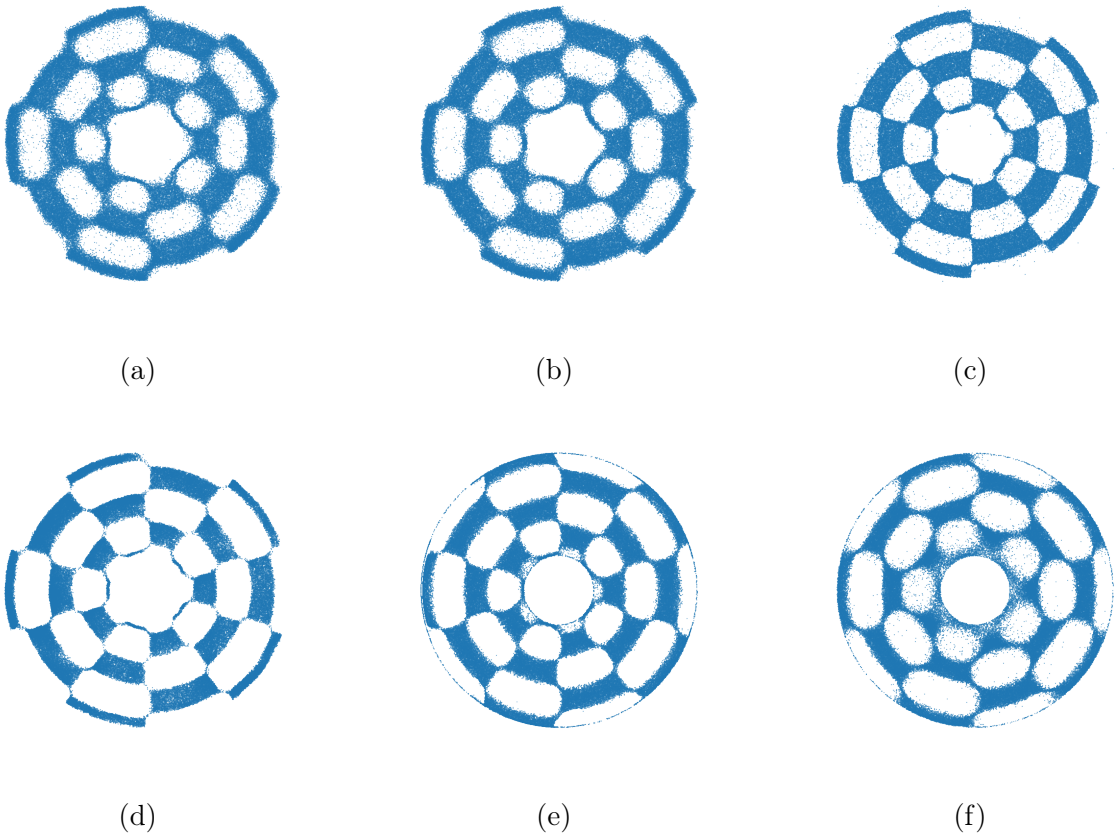
Figure 5.2: Sample comparison of radial checkerboard density distribution between (a) reference rejection sampled density, (b) DDPM (non-structure preserving), and (c) SPDM+FA.

when integrating. Consequently, provided the domain is Euclidean, we may immediately apply the techniques from Section 5.3 to the reflected setting, under appropriate alterations for training a using reflected noise, as discussed in Section 4.2.

Experiment: Symmetric density estimation on constrained disc Here we reuse the synthetic density from Section 5.4, but add the boundary constraints of the data domain, namely, defining a ϕ function according to Section 4.1 and incorporating this into the score prediction via Eq. (4.2).

As the limiting distribution of the reflected diffusion process is only locally Gaussian, we can no longer use the mean prediction denoising diffusion paramterization, see Eq. (2.7), and must either train the model to predict x_t using CDSM or predict the score directly using implicit score matching. As done in Section 5.4 we train a set of diffusion models to illustrate the differences between using Gaussian noise and reflected noise in the presence of boundary constants: (a) DDPM trained to predict x_t directly, (b) DDPM trained using Eq. (3.3) without and with (c) FA, (d) reflected variant of DDPM predicting x_t, and lastly (e) reflected DDPM using Eq. (3.3) without and with (f) FA. The model (a)-(d) where trained on 500,00 steps, while (e)-(f) proved much more sensitive to the training procedure requiring the use of gradient clipping and a lower learning rate and consequently additional training steps. Further implementation details can be found in Section 5.5.

A qualitative comparison between these models is shown in Fig. 5.3. To see the effect



(a) (b) (c)

(d) (e) (f)

Figure 5.3: Sample comparison of (a) DDPM, (b) DDPM+FA, (c) FFPM+cISM+FA, (d) reflected DDPM, (e) reflected DDPM+cISM, (f) reflected DDPM+cISM+FA.

of the boundary condition and reflected diffusion on the learned vector field, we plot the vector field learned by models (c) and (f) in Fig. 5.4 over a subsequence of time-steps. It is plane that (f), despite its comparably worse sample quality, faithfully captures the domain boundary constraints while (c) does not.

5.4.2 Structure preserving Riemann diffusion models

We will now show an extension of Theorem 4 to diffusion processes with zero drift to symmetric Riemann manifolds. While this setting is restrictive, it is worth noting that the zero drift manifold diffusion SDE Eq. (3.1) is the most commonly used in practice,

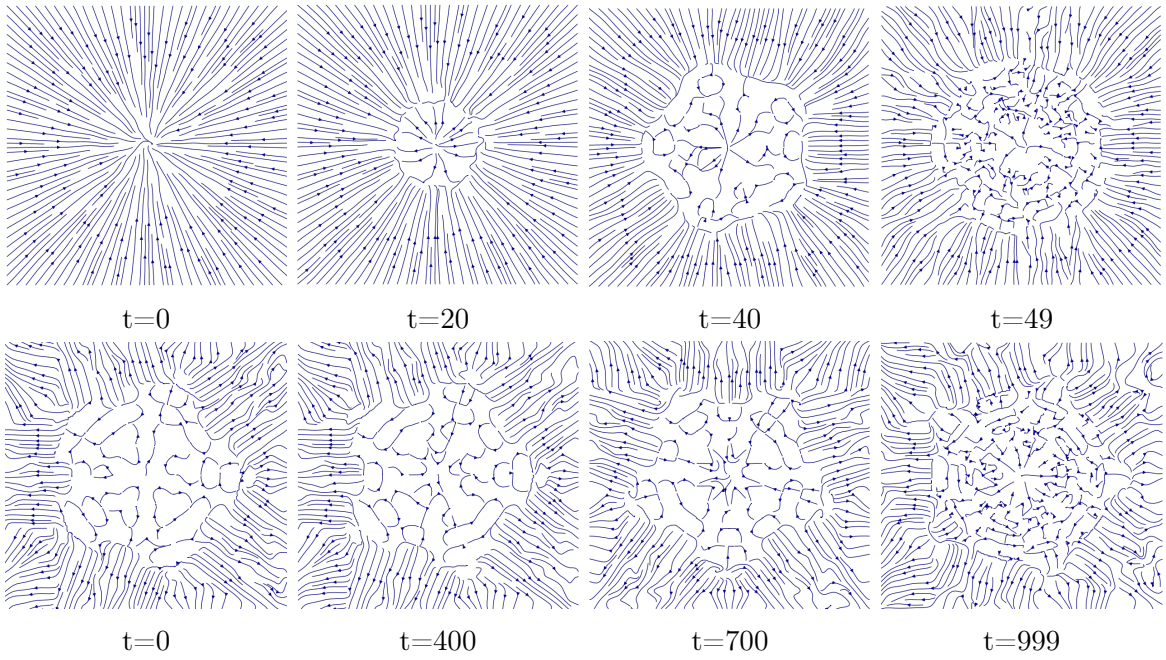


Figure 5.4: (top) Incremental flow visualization of learned Stein score from a trained DDPM at time steps $t = 0, 20, 40, 49$, and (bottom) reflected DDPM with boundary constraint at time steps $t = 0, 400, 700, 999$, both trained on the dataset described in Section 5.4. Inspecting these plots, there is a clear separation between the interior and exterior flow for the reflected DDPM while the regular DDPM has no such distinction.

[24, 63, 77, 14], due to the additional complications that would be incurred from having to compute the drift term w.r.t. local curvature; moreover, most application discussed in the forgoing works operate on symmetric spaces, so the restriction to this kind of domain is not impractical or unfounded.

Lemma 2. *For a symmetric Riemannian manifold $(\mathcal{M}, g)$, a distribution $p_0(x)$ that is invariant under the Isometry group $I_{\mathcal{M}}$ of $\mathcal{M}$, a manifold Brownian motion $(\overrightarrow{B_t^{\mathcal{M}}})_{t\geq 0}$ defined until some time $T > 0$, and a monotonically increasing fucntion $\sigma : [0, T] \to \mathbb{R}_{\geq 0}$, the marginal distributions p_t induced of the stochastic process $(\overrightarrow{X}_t)_{t\geq 0}$ characterized by the SDE*

$$\mathrm{d}\overrightarrow{X}_t = \sigma(t) \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}$$

are invariant under the Isometry group $I_{\mathcal{M}}$.

The following proof sketch is a direct consequence of [29, Sec.9 Thm.9.12] in combination with Definition 21, and is provided to clarify the connection to the work presented above and setup further discussion and terminology.

Proof. To begin with, let $(\mathcal{M}, g)$ be a symmetric d-dimensional orientable Riemann manifold equipped with metric g isometrically embedded with $\phi : \mathbb{R}^D \hookrightarrow \mathcal{M}$; so that we might consider a weighted Riemann manifold $(\mathcal{M}, g, \mu)$ with the Lebesgue measure $d\lambda$. Let $G \subseteq I_{\mathcal{M}}$ be a semi-group of the isometry group $I_{\mathcal{M}}$ of $\mathcal{M}$. Recall from Corollary 1 the distribution of the stochastic process $(\overrightarrow{X_t})_{t\geq 0}$, with dynamics $\mathrm{d}\overrightarrow{X}_t = \sigma(t) \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}$ can be described by a transition kernel $p_t(x, y)$ that admits a density of the form:

$$\frac{\partial}{\partial t} p_t(x) = \frac{\sigma^2(t)}{2} (\nabla \cdot \nabla)_{\mathcal{M}} p_t(x),$$

which is unique over any bounded domain $\mathcal{D} \subseteq \mathcal{M}$

$$p_t(x) = \int_{\mathcal{D}} p_t(x, y) f(y) \, \mathrm{d}y$$

for any smooth compactly supported function f. As a consequence of [29, Sec.9 Thm.9.12] the transition kernel $p_t(x, y)$ is invarant under G; i.e., for all $\kappa \in G$

$$p_t(L_\kappa x, L_\kappa y) = p_t(x, y),$$

provided p_0 is invariant. Thus, we see for any $\kappa \in G$

$$\begin{aligned} p_t(L_\kappa x) &= \int_{\mathcal{D}} p_t(L_\kappa x, y) f(y) \, \mathrm{d}y \\ &= \int_{L_\kappa^{-1}\mathcal{D}} p_t(L_\kappa x, L_\kappa z) f(L_\kappa z) \, \mathrm{d}z \\ &= \int_{L_\kappa^{-1}\mathcal{D}} p_t(x, z) f(L_\kappa z) \, \mathrm{d}z \\ &= p_t(x). \end{aligned}$$

□

Given the above lemma, we can now state a corollary that shows the sufficient condition of Theorem 4 from [65] extends to zero-drift diffusion processes over Riemann manifolds.

Corollary 3. *Let $(\mathcal{M}, g)$ be a symmetric d-dimentional Riemannian manifold, $(\overrightarrow{B_t^{\mathcal{M}}})_{t\geq 0}$ a manifold Brownian motion defined until some time $T > 0$, and consider the SDE*

$$\mathrm{d}\overrightarrow{X}_t = \sigma(t) \circ \mathrm{d}\overrightarrow{B_t^{\mathcal{M}}}$$

where $\sigma : [0, T] \to \mathbb{R}_{\geq 0}$ is a smooth monotonically increasing function. This diffusion process is structure preserving if and only if the Stine score of the admitted density satisfies

$$\nabla_{L_\kappa x} \log p_t(L_\kappa x) = dL_k|_x \nabla_x \log p_t(x)$$

for all $\kappa \in G \subseteq I_{\mathcal{M}}$ and $x \in \mathcal{M}$.

Proof. From Lemma 2, it has already been established that p_t is invariant, thus, all that remains is to validate the truth of the equality above. In particular, assume $(\mathcal{M}, g)$ is a symmetric compact (or locally compact) orientable d-dimensional Riemann manifold equip with the Riemann metric g and is isometrically embedded via the map $\phi : \mathbb{R}^D \hookrightarrow \mathcal{M}$. Let $G \subseteq I_{\mathcal{M}}$ be a semi-group of the isometry group of $\mathcal{M}$. Then $\forall \kappa \in G$ and $\forall x \in \mathcal{M}$ by the Riemann chain rule

$$\begin{aligned} \nabla_x \log p_t(L_\kappa x) &= (dL_\kappa|_x)^* (\nabla_{L_\kappa x} \log p_t(L_\kappa x) \\ &= dL_\kappa|_x^{-1} \nabla_{L_\kappa x} \log p_t(L_\kappa x), \end{aligned}$$

which implies

$$\nabla_x \log p_t(L_\kappa x) = dL_k|_x \nabla_x \log p_t(x).$$

□

Table 5.4: Comparison on symmetric sphere densities.

	Dotted density			Checkerboard density		
	Init.	rNet	rNet+FA	Init.	rNet	rNet+FA
NLL (↓)	2.51	0.52	**0.31**	2.53	1.38	**1.30**

While the proof of Corollary 3 is somewhat trivial, due to the global assumptions on $\mathcal{M}$, the result is no less important. Many existing works implicitly assume this equivalence between score differentiation and transformation when utilizing FA on non-Euclidean domains, namely molecule conformation. Informally, is it easy to convince yourself why the above result is true; take $\mathbb{S}^2 = \{x \in \mathbb{R}^3 \mid \|x\| = 1\}$ defined as a subspace of $\mathbb{R}^3$, which possess the isometry group $I_{\mathbb{S}^2} = SO(3)$, the (Lie) group of all rotations in $\mathbb{R}^3$ which can be represented as orthogonal matrices of determinate one. For any $v \in T_x\mathbb{S}^2 \cong \mathbb{R}^2$ it can be seen that[7]

$$dL_\kappa|_x(v) = \frac{d}{dt} L_\kappa(x + vt)\bigg|_{t=0} = \kappa v.$$

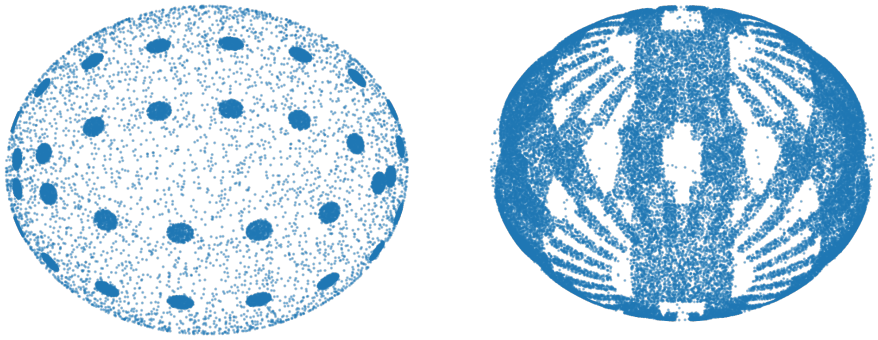
The uniformity of this result being due to the curvature symmetry of the space.

Experiment: Symmetric dotted density estimation on sphere Before attempting a challenging learning task on a sphere, we designed a toy dataset that consists of two equidistance bands of circles projected onto the unit sphere $\mathbb{S}^2$. This density is, by construction, C_{16} invariant about the z-axis. A scatter plot illustration of the constructed density is presented in Fig. 5.5a.

We train two models on this dataset, both based on a ResNet [30], using the score matching technique proposed in [63] with and without modification to include FA. We report the best achieved negative log likelihood (NLL) performance of each model, named rNet and rNet+FA respectively, along with a random initialization performance score as a basedline, denoted Init., in Table 5.4.

Experiment: Symmetric density estimation on sphere As another benchmark, we generalize the symmetric distribution $p(\omega, \nu) = \frac{1}{V} j(\omega, \nu)$ construction in Section 5.4 to a (unit) sphere $\mathbb{S}^2$, isometrically embedded into $\mathbb{R}^3$, for a checkerboard density distribution

[7]Here I just picked a simple geodesic curve parameterized using t.



(a) (b)

Figure 5.5: Scatter plot visualization of 100,000 randomly sampled points from (a) dotted spherical density and (b) checkerboard density on (unit) sphere.

$j : \mathbb{S}^2 \to \mathbb{R}_{\geq 0}$

$$j(\omega, \nu) = \begin{cases} \rho_1 & \text{if } \cos(k_\omega \omega) \cdot \cos(k_\nu(\omega)\nu) \geq 0 \\ \rho_2 & \text{otherwise,} \end{cases}$$

where $\rho_1, \rho_2, k_\omega \in \mathbb{R}$ are chosen constants, $k_\nu : \mathbb{R} \to \mathbb{R}_{>0}$ is a azimuthal frequency function that increased towards the pol defined as

$$k_\nu(\omega) = k_b \cdot 2^{n - \left\lfloor \frac{2n|\omega - \pi/2|}{\pi} \right\rfloor},$$

$k_b \in \mathbb{R}$ the base equatorial frequency, n the number of frequency doublings towards the pol, and V is the volume of integration given by integrating j over the torus w.r.t. the volume element $\mathrm{d}\mu = \sqrt{\det(g)}\,\mathrm{d}\lambda$; i.e.,

$$V = \int_0^{2\pi} \int_0^{2\pi} j(\omega, \nu) \sqrt{\det(g)}\,\mathrm{d}\omega\,\mathrm{d}\nu.$$

An illustration of this density is given in Fig. 5.5b. This distribution is C_{2k_b} invariant w.r.t. transporting points around the z-axis of the sphere.

As with the above experiment, we train two ResNet models and evaluate their NLL performance on this dataset, the values of which are reported in Table 5.4. Clearly, based on the achieved NLL scores, this dataset proved to be significantly more challenging to learn than the dot dataset, nonetheless, the incorporation of FA enabled the second model to learn a more accurate score.

5.5 Implementation details

We implemented various generative model architectures for the experiments presented in the preceding chapter. This includes regular diffusion models and bridge diffusion models (DDBM) based on VP-SDEs, which are structure-preserving with respect to C_4, D_4, and flipping, as per [65]. Except for SPDM-WT, all models are trained with data augmentation using randomly selected operators from their respective groups along with the standard slew of augmentation techniques intended to improve model robustness.

To boost the performance of SP-SDE and DDBM model architectures, we apply the boundary condition parameterization proposed in [48] and self-conditioning [10] to improve sample quality. The model code and training details are documented in the respective GitHub repositories for each experiment.

Specifically, for Rotated MNIST, LYSTO, and ANHIR, we use SPDM[8] along with modified versions of GE-GAN and SP-GAN[9].

For the style transfer tasks involving LYSTO and CT-PET, we implemented a modified version of DDBM designed to operate in latent space, as proposed in [83]. This implementation is based on the VAE used in Stable Diffusion v1-4, fine-tuned on the CT-PET dataset while applying flipping augmentation using FA to ensure the encoder is invariant to horizontal flips. This approach transforms the (256×256×3) data into a (32×32×4) latent space. The alteration was necessary due to memory constraints on the utilized NVIDIA L40S GPUs (40GB), which did not allow effective training of the previously used U-Net backbone at higher resolutions. This implementation of SPDM[10] is available alongside similarly modified implementations of Pix2Pix[11] and I^{2}SB[12].

For the toy dataset experiments with and without reflected noise (Section 5.4), we implemented an MLP[13] along with the training algorithms detailed in Section 4.2. Lastly, for the manifold extension experiment (Section 5.4.2), we developed a ResNet[14] based on the models used in [63].

[8] `https://github.com/SpencerSzabados/Group-Diffusion`
[9] `https://github.com/SpencerSzabados/SP-GAN`
[10] `https://github.com/SpencerSzabados/Group-Diffusion-Bridge`
[11] `https://github.com/SpencerSzabados/pix2pix`
[12] `https://github.com/SpencerSzabados/I2SB`
[13] `https://github.com/SpencerSzabados/reflected-diffusion-mlp`
[14] `https://github.com/SpencerSzabados/group-diffusion-manifold`

Chapter 6

Motion planning

The task of motion planning is to generate a safe path (or trajectory of motion) for a robot under a set of motion constraints to achieve a supplied task (or objective). Motion planning has its roots in long time horizon reinforcement learning [95] and control systems [13, 97]. While there are many domain specific variations to motion planning, the one we will be concerned with is that of generating collision free motion paths for robot navigation and multi-segment robot arms, as considered within [43, 8].

Specifically, within Euclidean space, let $w = [s, a] \in \mathbb{R}^{d\times 2}$ encode the state of the robot along with the actions needed to take the robot from its current state to the next (e.g., s might encode the state of a robot arm and a the joint velocities needed align the end effector at the next discrete position along an ascribed trajectory); that is, we assume the dynamics of the robot follow discrete-time steps with dynamics $s_{i+1} = f(s_i, a_i)$. Rather than attempting to encode a continuous trajectory it is common practice to approximate the trajectory using a sequence of way-points, $\tau = (w_1, \ldots, w_{H-1}, w_H) \in \mathbb{R}^{(d\times 2)\times H}$, at key points along the robot's path. In order to formulate planning tasks as a reinforcement learning problem, we must introduce a reward function and some way of encoding the optimally of a trajectory. To this end, let $\mathcal{O} = (O_i)_{i=1}^H$ be a time indexed sequence of binary random variables where O_i encodes the optimally of the way-point w_i with $\mathbb{P}(O_i = 1) \propto \exp\{R(s_i, a_i)\}$, where $R : \mathbb{R}^{d\times 2} \to \mathbb{R}_{\geq 0}$ is a (composite) reward function (or cost function) used to evaluate the robot's ability at reaching optimal positions along trajectories (along with the smoothness of the trajectory, collision avoidance, etc). Within the described environment, motion planning can be formulated as the maximization problem:

$$\tau^* = \arg\max_{\tau} \sum_{i=1}^{H} \lambda_i R(s_i, a_i), \tag{6.1}$$

where $1 \geq \lambda_i \geq 0$ are reward discounting weights; [95], these weights are often constructed to be monotonically increasing, putting more weight on achieving the long run task over immediate rewards near the initialization point.

Once a suitable trajectory is found, it is subsequently the task of a "low-level" (hardware specific) control system to execute and maneuver the physical robot. This separation between optimization and physical control, where a generated trajectory is plugged into an existing classical trajectory control routine, is the most common framework employed for sampling based planning methods, see [43, 8], largely due to its simplicity.

As eluded to, we will focus on sampling-based (or planning as inference) methods for generating (optimizing) trajectories [21], as opposed to the motion optimization setting discussed in [98], where trajectories are sampled from an informed prior distribution which is either refined from example dataset or heuristic approaches to result in a collision free path. That is, given an environment prior $p(\tau)$, which in some capacity represents boundaries of the environment, our goal is to sample from the posterior distribution:

$$p(\tau|\mathcal{O}) \propto p(\mathcal{O}|\tau)p(\tau), \tag{6.2}$$

where $p(\mathcal{O}|\tau)$ represents the likelihood of achieving the planning objective goals. In order to simplify sampling and construct more informed priors, it is assumed classically [98] that $p(\mathcal{O}|\tau)$ can be factored into independent components:

$$p(\mathcal{O}|\tau) \propto \prod_{i=1}^{H} p_i(O_i|\tau)^{\alpha_i}, \tag{6.3}$$

where $\alpha_i \geq 0$ are annealing temperatures for the different objective distributions [85, 47] with $p_i(O_i|\tau) \propto \exp\{R_i(\tau)\}$ for the index reward function R_i; e.g., these objective distributions might be used to preferentially optimize the long term goals of the trajectory opposed initial accuracy.

6.1 Motion planning using diffusion

Having elucidated the generic trajectory planning task considered, we now describe how this task can be approached using diffusion models. Under the planning as inference scheme, diffusion models are used to parameterize the objective prior $p(\mathcal{O}|\tau)$ in the trajectory sampling procedure. We follow the framework proposed in [43, 8][1] which makes use of the DDPM framework, outlined in Section 2.2, and a modified U-Net to generate multi-segment trajectories encoded as a sequence of time-correlated way-points with a specified time horizon.

[1]Aspects of the sampling procedure(s) were necessarily redervied below as some details were omitted in the aforementioned paper.

Begin by assuming $\tau_0 \sim p_0$, where p_0 is the data distribution for trajectories over a given domain $\mathcal{D} \subseteq \mathbb{R}^d$ which obey the planning objectives $\mathcal{O}$. Recall, per the diffusion process outlined in Section 2.2, the discrete step transition kernel admitted by a forward diffusion process, under the VP-SDE, can be expressed as

$$p(\tau_t|\tau_{t-1}) = \mathcal{N}(\tau_t; \sqrt{1-\beta_t}\tau_{t-1}, \beta_t\mathbb{I}).$$

Then in order to sample from Eq. (6.2), we consider a diffusion model parameterized as

$$p(\tau_0|\mathcal{O}) = p(\tau_T|\mathcal{O}) \prod_{t=1}^{T} p_\theta(\tau_{t-1}|\tau_t, \mathcal{O}) \tag{6.4}$$

starting from $\tau_T \sim \mathcal{N}(0, \mathbb{I})$; i.e., $p(\tau_T|\mathcal{O}) = \mathcal{N}(\tau_T; 0, \mathbb{I})$. In order to incorporate the planning objectives $\mathcal{O}$ into this task the authors of [8] modify the sampling procedure in a similar way to classifier-free-guidance [34], biasing the sampling towards loss cost regions by modifying the gradient updates, sampling from this distribution by iteratively sampling the task-conditioned posterior(s):

$$p_\theta(\tau_{t-1}|\tau_t, \mathcal{O}) \propto p_\theta(\tau_{t-1}|\tau_t) p_t(\mathcal{O}|\tau_{t-1}).$$

To derive the planning objective gradient updates, recall from Section 2.2, for scalar diffusion coefficients the backwards transitions kernels can be approximated as:

$$\begin{aligned} \log p_\theta(\tau_{t-1}|\tau_t) &= \log \mathcal{N}(\tau_{t-1}; \mu_\theta(\tau_t, t), \beta_t\mathbb{I}) \\ &\propto -\frac{1}{2}(\tau_{t-1} - \mu_\theta)^\intercal (\beta_t\mathbb{I})^{-1}(\tau_{t-1} - \mu_\theta) \end{aligned} \tag{6.5}$$

and, via a first order taylor expansion around μ_t

$$\log p(\mathcal{O}|\tau_{t-1}) \approx \log p(\mathcal{O}|\mu_t) + (\tau_{t-1} - \mu_t)\nabla_\tau \log p(\mathcal{O}|\mu_t); \tag{6.6}$$

Then under Eq. (6.3), factoring $p(\mathcal{O}|\tau)$, we get by combining Eqs. (6.5) and (6.6)

$$\begin{aligned} \log p_\theta(\tau_{t-1}|\tau_t, \mathcal{O}) &\propto -\frac{1}{2}(\tau_{t-1} - \mu_\theta)^\intercal (\beta_t\mathbb{I})^{-1}(\tau_{t-1} - \mu_\theta) + (\tau_{t-1} - \mu_t)\nabla_\tau \log p(\mathcal{O}|\mu_t) \\ &\quad + \log p(\mathcal{O}|\mu_t) \\ &\propto -\frac{1}{2}(\tau_{t-1} - \mu_\theta)^\intercal (\beta_t\mathbb{I})^{-1}(\tau_{t-1} - \mu_\theta) \ + (\tau_{t-1} - \mu_t)\nabla_\tau \log p(\mathcal{O}|\mu_t) \\ &\quad - \frac{1}{2}(\tau_{t-1} - \mu_t)^\intercal (\beta_{t-1}\mathbb{I})^{-1}(\tau_{t-1} - \mu_t) \quad \text{(assuming } p(\mathcal{O}|\tau) \text{ is Gaussian.)} \end{aligned}$$

$$
\begin{aligned}
&\propto -\frac{1}{2}(\tau_t - \mu_\theta - \beta_t \nabla_\tau \log p(\mathcal{O}|\mu_t))^\intercal (\beta_t \mathbb{I})^{-1} (\tau_t - \mu_\theta - \beta_t \nabla_\tau \log p(\mathcal{O}|\mu_t)) \\
&\quad - \frac{1}{2}(\tau_{t-1} - \mu_t)^\intercal (\beta_{t-1}\mathbb{I})^{-1}(\tau_{t-1} - \mu_t). \qquad \text{(assuming } \mu_t \approx \mu_\theta.\text{)} \\
&\propto -\frac{1}{2}(\tau_t - \mu_\theta - \beta_t \nabla_\tau \log p(\mathcal{O}|\mu_t))^\intercal (\beta_t \mathbb{I})^{-1} (\tau_t - \mu_\theta - \beta_t \nabla_\tau \log p(\mathcal{O}|\mu_t))
\end{aligned}
$$

Sampling is then performed by making use of the reparameterization trick, interactively evaluating the Langrangian dynamics:

$$
\tau_{t-1} = \mu_\theta(\tau_t, t) - \nabla_\tau \log p(\mathcal{O}|\mu_t) + \sqrt{\beta_t} z, \quad z \sim \mathcal{N}(0, \mathbb{I}).
$$

The gradient update, under the factorization assumption in Eq. (6.3), bias trajectory samples towards those that are more likely to satisfy the planning objectives; specifically, we have by construction

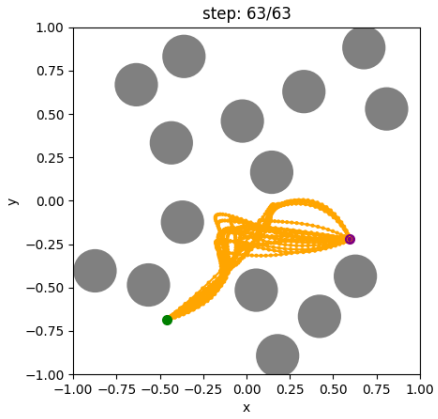
$$
\nabla_\tau \log p(\mathcal{O}|\mu_t) = \sum_{i=1}^{H} \lambda_i \nabla_\tau R_i(\tau).
$$

As an aside, it is because of the fact that we formulate the planning problem in terms of learning (diffusing) the entire trajectory (at a fixed length) as opposed to learning an optimal policy at each step, as done in the more recent work [41], which sequentially generates the trajectory with the score posing as the learned planning policy, we are able to guild the entire trajectory during sampling.

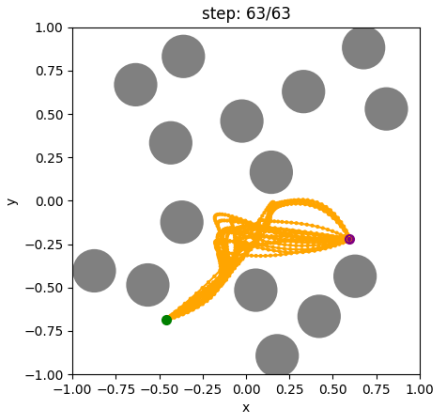
Experiment: Point-mass robot with circular objects. Here we reproduce the results from [8], training a model based on the configuration provided by the authors.

Fig. 6.1 contains trajectories generated from a trained diffusion model with reward guidance which is either enabled or disabled during sampling. The objects in (red) are extra objects added outside of model training, while the objects in (grey) are static objects present in both training and inference. Sample statistics are reported in Table 6.1.

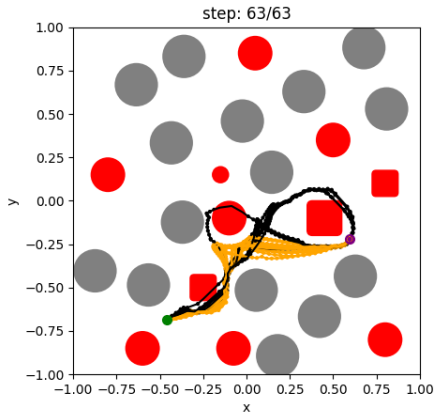
As can been seen from Fig. 6.1a and Fig. 6.1b, the model performs reasonably well at avoiding trajectories in its original training environment, but without guidance fails to respect the underlying geometry on which it was trained. This issue is further highlighted in Fig. 6.1c and Fig. 6.1d. A diffusion model that is geometry aware should elevate this problem for the unperturbed training environment, as no sampled trajectories will intersect obstacles.



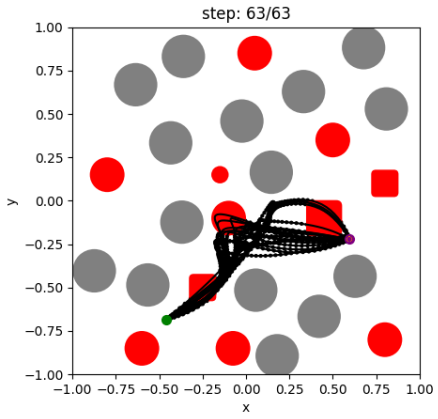
(a) Trajectories sampled using guidance.



(b) Trajectories sampled without guidance.



(c) Trajectories sampled using guidance.



(d) Trajectories sampled without guidance.

Figure 6.1: Sample trajectories drawn from motion planning diffusion model trained using reward guidance, as formulated in Section 6.1.

6.2 Motion planning on constrained (Euclidean) manifolds

We will now outline a hybrid RL reflected diffusion motion planning framework that offers to produce, in some sense, verification free trajectories. This has the primary benefit of reducing the number of required samples until a valid trajectory is produced to one.

Environment	Guidance	Non Coll. (%)	Coll. Intensity	Waypoint Var.	Lowest Cost
Circles2D	Yes	(99.8, 99.9, 100.0)	(0.0, 0.004, 0.004)	(0.73, 0.80, 0.82)	(2.32, 2.41, 2.49)
Circles2D	No	(97.6, 98.2, 98.7)	(0.11, 0.18, 0.21)	(0.69, 0.75, 0.78)	(2.31, 2.41, 2.48)
Circles2D+Extra	Yes	(35.4, 36.7, 34.8)	(1.25, 1.38, 1.48)	(0.33, 0.35, 0.45)	(2.59, 2.74, 2.76)
Circles2D+Extra	No	(0.2, 0.5, 0.8)	(20.12, 21.03, 21.25)	(0.01, 0.04, 0.80)	(2.76, 3.08, 3.13)

Table 6.1: Table includes sample statistics for a batch of 50 trajectories sampled with, see Figs. 6.1a and 6.1c, and without guidance, see Figs. 6.1b and 6.1d. The first two rows are from the unchanged sample environment, the last two correspond to the altered environment with additional obstacles added.

This framework operates by converting any fixed obstacle avoidance rewards into domain constraints which are avoided by replacing the standard diffusion sampling with reflected diffusion, Chapter 4, over the newly constructed domain; this hybrid approach is inspired by existing sampler-optimization based approaches, however, hard constrains are enforced by reflected steps rather than an optimizer. Given a set $\{b_i\}_{i\in I}$ of obstacles, corresponding to some partial rewards $r_i(s_t, a_t)$ for some waypoint $w_t = [s_t, a_t]$ on a trajectory, we construct a matching set of smooth boundaries $\{\phi_i(x)\}_{i\in I}$ that outerbound these obstacles (.e.g. performing the Minkowski sum with a ϵ-ball) and define the domain $\mathcal{D}$ as above. Letting $\mathcal{O}'$ and R' denote the modified set of planning objectives $\mathcal{O}$ and reward function R without fix obstacle avoidance criteria. Then, keeping everything else the same, the planning objective Eq. (6.1) outlined above can be expressed as the constrained problem:

$$\tau^* = \arg\max_{\tau\in\mathcal{D}} \sum_{t=1}^{T} \lambda_i R'(s_t, a_t).$$

Now as the transition kernels of reflected diffusion processes are only local martingales (e.g., local Brownian monition), the sampling formulation of Section 6.1 must be appropriately modified, in particular, we consider the forwards heat SDE

$$\mathrm{d}\overrightarrow{X}_t = \sigma_t \circ \mathrm{d}\overset{\rightsquigarrow}{B}_t;$$

which, under T discretization steps for a general domain $\mathcal{D}$, must be simulated (e.g., using Algorithm 6). It is necessary to alter the model to do score prediction directly to ensure the theoretical guarantees; however, to begin with, we offer a proximal solution in Section 6.2.

Proxy reflected motion planning. The following is performed under the assumption that for sparse objects the local distribution is sufficiently close to Gaussian so the model

will be able to learn the score via mean prediction (i.e., via τ_0 reconstruction).

For national simplicity, let $q_t(\tau_t|\tau_{t-1})$ denote the approximate transition kernel that arises from simulating k steps of the reflected diffusion process. Then Eq. (6.4) takes the form

$$q(\tau_0|\mathcal{O}) = q(\tau_T|\mathcal{O}) \prod_{t=1}^{T} q_\theta(\tau_{t-1}|\tau_t, \mathcal{O})$$

starting from $\tau_T \sim q_T$. As before, we incorporate the remaining planning objectives by modifying the (backwards) sampling process via

$$q_\theta(\tau_{t-1}|\tau_t, \mathcal{O}) \propto q_\theta(\tau_{t-1}|\tau_t) q_t(\mathcal{O}|\tau_{t-1})$$

by treating the unconstrained $p_\theta(\tau_{t-1}|\tau_t)$ as a proxy to the true constrained dynamics $q_\theta(\tau_{t-1}|\tau_t)$ and iteratively evaluating rejecting samples that violate the constrains; training and sampling processes used are given in Algorithm 7 and Algorithm 8 respectively.

Algorithm 7: Reflected Motion Planning Diffusion Training.
Consider diffusion over the manifold $\mathcal{D}$ with $\{\phi_i\}_{i\in I}$ a set of constraints that define $\partial\mathcal{D}$, $\mathcal{J}$ be a set of collision-free trajectories in $\mathcal{D}$, $T \geq 0$ be a given stopping time, $N > 0$ the number of discretization steps, $\eta > 0$ a learning rate, set of model parameters θ to be optimized.

Data: $\{\phi_i\}_{i\in I}, \mathcal{J}, T, N, \eta, \theta$
Result: θ
while *training* **do**
  $\tau_0 \sim \mathcal{J}$, $t \sim \mathcal{U}(0, T)$; `/* Sample trajectory batch and time. */`
  $\tau_t \leftarrow \textsc{ReflectedStep}(\{\phi_i\}_{i\in I}, t, N, \tau_0)$; `/* Simulate SDE trajectory. */`
  $\mathcal{L} \leftarrow \mathcal{L}_{CDSM}(\tau_t, \theta)$; `/* Compute loss. */`
  $\theta \leftarrow \theta + \eta\nabla_\theta\mathcal{L}$
return θ

Experiment: Proxy reflected point-mass robot with circular obstacles. Here we reconsider the same environment motion planning problem from Section 6.1. As before the objects in (red) are extra objects added outside of model training, while the objects in (grey) are static objects present in both training and inference. Sample statistics from several configurations are reported where the (min, mean, max) statistics are computed

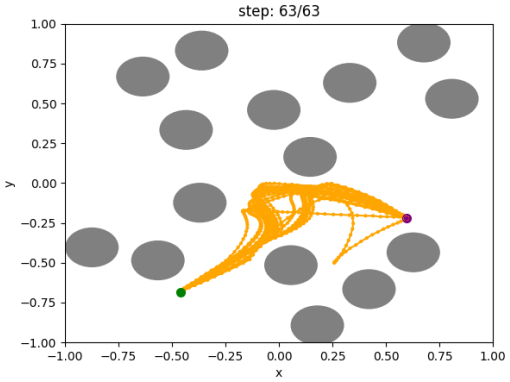
Algorithm 8: Reflected Motion Planning Sampling.
Consider diffusion over the manifold $\mathcal{D}$ with $\{\phi_i\}_{i\in I}$ a set of constraints that define $\partial\mathcal{D}$, set of trained model parameters θ, pair of desired start and end states s_0, s_T, $T \geq 0$ be a given stopping time, $N > 0$ the number of discretization steps, $\kappa > 0$ a number of warm-up steps, motion planning reward function R

Data: $\{\phi_i\}_{i\in I}, \theta, s_0, s_T, T, N, \kappa, R$
Result: τ_0

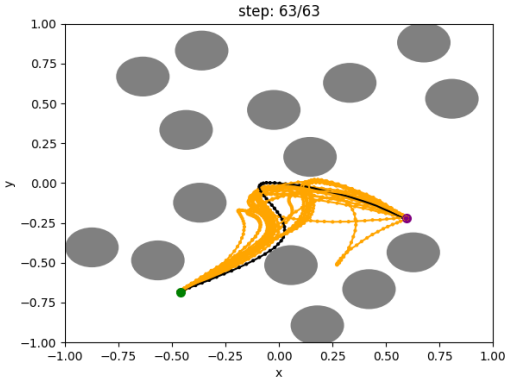
$p \sim \mathcal{U}(\mathcal{D})$; `/* Randomly sample trajectory in domain. */`
$\tau_T \leftarrow \textsc{ReflectedStep}(\{\phi_i\}_{i\in I}, T, N, p)$; `/* Generate noisy trajectory */`
$\tau_T[0] = s_0$, $\tau_T[T-1] = s_T$; `/* Enforce start and end states. */`
for $t = T, \ldots, 1$ **do**
    $\tilde{\mu}_t \leftarrow \mu_\theta(\tau_t, t)$
    $g \leftarrow -\sum_{i=1}^{N} \lambda_i \nabla_{\tau_{t-1}} R_i(\tau_{t-1})$; `/* Apply guidance to trajectory. */`
    $\tau_{t-1} \leftarrow \tilde{\mu}_t + g + \sqrt{\beta_t} z$ for $z \sim \mathcal{N}(0, \mathbb{I})$
    $\tau_{t-1}[0] = s_0$, $\tau_{t-1}[T-1] = s_T$
return τ_0

across the random seeds $\{0, 1, 11, 42, 4242, 438233955\}$ from which the (min, median, max) of each metric is computed. The start and end points of the desired trajectory are fixed across all evaluations.

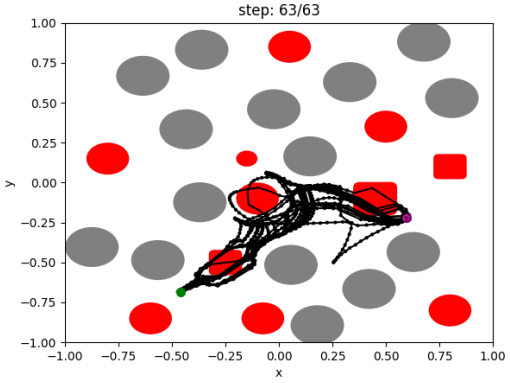
We trained two reflected proxy models on this planning task, the first being trained without velocity guidance, reported in in Table 6.3, and the second trained with velocity guidance, reported in Table 6.2.Fig. 6.2 contains sampled trajectories from the reflected diffusion model trained with velocity guidance, with this guidance enabled or disabled during sampling. As can be seen, these models – despite being proxies – perform on par with the baseline model for the first two scenarios; however, these models are not as able to adapt to domain changes with the addition of the (red) obstacles. It is our suspicion that this technique would perform far better once implemented in accordance with the discussion in Section 4.2. With that said, these experiments do serve to illustrate the potential for future work applying reflected diffusion to motion planning tasks.



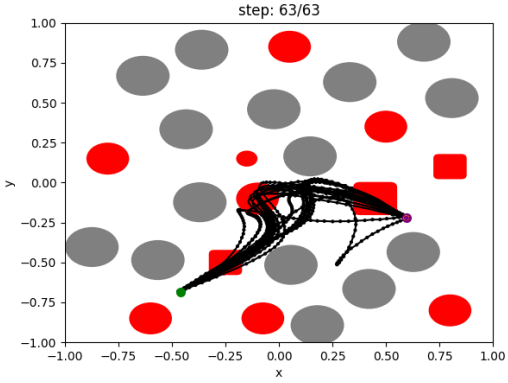
(a) Trajectories sampled using guidance.



(b) Trajectories sampled without guidance.



(c) Trajectories sampled using guidance.



(d) Trajectories sampled without guidance.

Figure 6.2: Sample trajectories drawn from motion planning (reflected) constrained diffusion model trained with and without using reward guidance, as formulated in Section 6.2.

Environment	Guidance	Non Coll. (%)	Coll. Intensity	Waypoint Var.	Lowest Cost
Circles2D	Yes	(99.2, 99.4, 99.5)	(0.02, 0.03, 0.03)	(0.29, 0.26, 0.28)	(3.30, 3.31, 3.37)
Circles2D	No	(73.7, 74.2, 75.2)	(1.29, 1.34, 1.52)	(0.302, 0.318, 0.362)	(3.19, 3.26, 3.37)
Circles2D+Extra	Yes	(0.0, 0.0, 0.4)	(8.11, 8.26, 8.36)	(1.22, 1.27, 1.64)	(4.51, 4.57, 4.38)
Circles2D+Extra	No	(0.0, 0.0, 0.0)	(34.94, 35.25, 35.27)	nan	nan

Table 6.2: Table includes (min, mean, max) sample statistics for a batch of 1000 trajectories sampled with, see Figs. 6.2a and 6.2c, and without guidance, see Figs. 6.2b and 6.2d. The first two rows are from the unchanged sample environment, the last two correspond to the altered environment with additional obstacles added.

Environment	Guidance	Non Coll. (%)	Coll. Intensity	Waypoint Var.	Lowest Cost
Circles2D	Yes	–	–	–	–
Circles2D	No	(94.1, 96.6, 97.2)	(0.207, 0.301, 0.450)	(0.345, 0.375, 0.398)	(2.89, 3.09, 3.28)
Circles2D+Extra	Yes	–	–	–	–
Circles2D+Extra	No	(0.0, 0.0, 0.1)	(28.25, 28.67, 28.96)	nan	(4.67, 4.67, 4.67)

Table 6.3: Table includes (min, mean, max) sample statistics for a batch of 1000 trajectories sampled with and without guidance. The first two rows are from the unchanged sample environment, the last two correspond to the altered environment with additional obstacles added.

6.3 Implementation details

Following the work of [43] a temporal U-Net is used to encode the trajectories as a chain of time correlated way-points. We incorporated a modified version of the score parameterization used in [61] into our U-Net [2] implementation and simulate the reflected noise as described in Section 4.2.

[2] `https://github.com/SpencerSzabados/Motion-Planning-Diffusion-Manifold`

Chapter 7

Conclusion

In summary, we have aimed to guide the reader through a range of concepts related to diffusion-based generative models, beginning with their foundational continuous processes and then considering their discrete counterparts. By extending the basic Euclidean framework to settings involving manifolds — whether constrained or not — we have sought to illustrate methods for more effectively modeling data that resides in non-Euclidean spaces or within bounded domains. Although this exploration is by no means exhaustive, it is the author's belief the material presented herein offers a reasonably thorough and accurate overview of the current state of diffusion modeling.

A central focus of our study was the notion of structure-preserving diffusion models in which known symmetries or invariants can be directly integrated into the modeling process provided given conditions on the form of the diffusion equation are satisfied. Beyond the illustrative toy examples for reflected and manifold diffusion, we have also touched upon practical applications, notably in image generation and reconstruction in and medical imaging, which were shown to benefit from the inclusion of the techniques developed herein. Beyond this, we briefly touched on applications of reflected diffusion to motion planning, highlighting some experimental results that suggest the broader applicability of non-standard diffusion models in robotics.

While there remains much ground to cover, we hope the contributions outlined here will serve as a meaningful foundation for further research, guiding ongoing efforts to refine and expand both the theoretical and applied dimensions of diffusion modeling.

References

[1] B. D. Anderson. "Reverse-time diffusion equation models". *Stochastic Processes and their Applications*, vol. 12, no. 3 (1982), pp. 313–326.

[2] T. M. Apostol. "Mathematical Analysis". Addison-Wesley Series in Mathematics. Addison-Wesley, 1973.

[3] J. Birrell, M. Katsoulakis, L. Rey-Bellet, and W. Zhu. "Structure-preserving GANs". In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 1982–2020.

[4] J. Borovec et al. "ANHIR: Automatic Non-Rigid Histological Image Registration Challenge". *IEEE Trans. Med. Imaging*, vol. 39, no. 10 (2020), pp. 3042–3052.

[5] K. Burdzy, Z.-Q. Chen, and J. Sylvester. "The heat equation and reflected Brownian motion in time-dependent domains". *The Annals of Probability*, vol. 32, no. 1B (2004), pp. 775–804.

[6] K. Burdzy, Z.-Q. Chen, and J. Sylvester. "The heat equation and reflected Brownian motion in time-dependent domains.: II. Singularities of solutions". *Journal of Functional Analysis*, vol. 204, no. 1 (2003), pp. 1–34.

[7] M. do Carmo. "Differential Geometry of Curves and Surfaces". Differential Geometry of Curves and Surfaces p. 2. Prentice-Hall, 1976.

[8] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters. "Motion Planning Diffusion: Learning and Planning of Robot Motions with Diffusion Models". 2024.

[9] R. T. Q. Chen and Y. Lipman. "Flow Matching on General Geometries". In: *The Twelfth International Conference on Learning Representations*. 2024.

[10] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. Zhang. "Sampling is as easy as learning the score: theory for diffusion models with minimal data assumptions". In: *The Eleventh International Conference on Learning Representations*. 2023.

[11] T. Cohen and M. Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F.

Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. 2016, pp. 2990–2999.

[12] G. Corso, H. Stärk, B. Jing, R. Barzilay, and T. S. Jaakkola. "DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking". In: *The Eleventh International Conference on Learning Representations*. 2023.

[13] I. J. Cox and G. T. Wilfong. "Autonomous Robot Vehicles". 1st ed. New York, New York: Springer, 1990.

[14] V. De Bortoli, E. Mathieu, M. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet. "Riemannian Score-Based Generative Modelling". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 2406–2422.

[15] V. De Bortoli, E. Mathieu, M. J. Hutchinson, J. Thornton, Y. W. Teh, and A. Doucet. "Riemannian Score-Based Generative Modelling". In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022.

[16] V. De Bortoli, J. Thornton, J. Heng, and A. Doucet. "Diffusion Schrödinger Bridge with Applications to Score-Based Generative Modeling". In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021.

[17] L. Deng. "The mnist database of handwritten digit images for machine learning research". *IEEE Signal Processing Magazine*, vol. 29, no. 6 (2012), pp. 141–142.

[18] N. Dey, A. Chen, and S. Ghafurian. "Group Equivariant Generative Adversarial Networks". In: *International Conference on Learning Representations*. 2021.

[19] A. A. Duval, V. Schmidt, A. Hernández-García, S. Miret, F. D. Malliaros, Y. Bengio, and D. Rolnick. "FAENet: Frame Averaging Equivariant GNN for Materials Modeling". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 9013–9033.

[20] B. Efron. "Tweedie's Formula and Selection Bias". *Journal of the American Statistical Association*, vol. 106, no. 496 (2011), pp. 1602–1614.

[21] M. Elbanhawi and M. Simic. "Sampling-Based Robot Motion Planning: A Review". *IEEE Access*, vol. 2 (2014), pp. 56–77.

[22] B. Elesedy and S. Zaidi. "Provably Strict Generalisation Benefit for Equivariant Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 2959–2969.

[23] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis. "Learning $SO(3)$ Equivariant Representations with Spherical CNNs". In: *ECCV*. 2018, pp. 54–70.

[24] N. Fishman, L. Klarner, V. De Bortoli, E. Mathieu, and M. J. Hutchinson. "Diffusion Models for Constrained Domains". *Transactions on Machine Learning Research* (2023).

[25] L. Gao, Y. Du, H. Li, and G. Lin. "RotEqNet: Rotation-equivariant network for fluid systems with symmetric high-order tensors". *Journal of Computational Physics*, vol. 461 (2022), p. 111205.

[26] S. Gatidis et al. "A whole-body FDG-PET/CT Dataset with manually annotated Tumor Lesions". *Scientific Data*, vol. 9 (2022), pp. 601–608.

[27] I. Goodfellow. "Neurips 2016 tutorial: Generative adversarial networks". *arXiv preprint arXiv:1701.00160* (2016).

[28] A. Grigor'yan. "Estimates of heat kernels on Riemannian manifolds". In: *Spectral Theory and Geometry*. Ed. by E. B. Davies and Y. Safarov. London Mathematical Society Lecture Note Series. Cambridge University Press, 1999, pp. 140–225.

[29] A. Grigor'yan. "Heat Kernel and Analysis on Manifolds". 1st ed. Vol. 47. AMS Studies in Advanced Mathematics. AMS, 2009.

[30] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition". *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778.

[31] S. Helgason. "Differential Geometry and Symmetric Spaces". AMS, 1962.

[32] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium".

In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[33] J. Ho, A. Jain, and P. Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[34] J. Ho and T. Salimans. "Classifier-Free Diffusion Guidance". In: *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*. 2021.

[35] E. Hoogeboom, V. G. Satorras, C. Vignac, and M. Welling. "Equivariant Diffusion for Molecule Generation in 3D". In: *Proceedings of the 39th International Conference on Machine Learning*. 2022, pp. 8867–8887.

[36] E. P. Hsu. "Stochastic Analysis on Manifolds". Ed. by S. G. Krantz, D. Saltman, D. Sattinger, and R. Stern. Vol. 38. Graduate Studies in Mathematics. American Mathematical Society AMS, 2002.

[37] C.-W. Huang, M. Aghajohari, J. Bose, P. Panangaden, and A. Courville. "Riemannian Diffusion Models". In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. Vol. 35. Curran Associates, Inc., 2022, pp. 2750–2761.

[38] M. Hutchinson. "A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines". *Communications in Statistics - Simulation and Computation*, vol. 19, no. 2 (1990), pp. 433–450.

[39] A. Hyvärinen. "Estimation of Non-Normalized Statistical Models by Score Matching". *Journal of Machine Learning Research*, vol. 6, no. 24 (2005), pp. 695–709.

[40] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "Image-To-Image Translation With Conditional Adversarial Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.

[41] V. Jain and S. Ravanbakhsh. "Learning to Reach Goals via Diffusion". In: *Proceedings of the 41st International Conference on Machine Learning*. 2024.

[42] K. Jänich. “Topology”. Ed. by F. Gehring, P. Halmos, and J. Ewing. Trans. by S. Levy. 2nd ed. Undergraduate texts in Mathematics. New York: Springer-Verlag, 1984.

[43] M. Janner, Y. Du, J. Tenenbaum, and S. Levine. “Planning with Diffusion for Flexible Behavior Synthesis”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 9902–9915.

[44] Y. Jiao et al. “LYSTO: The Lymphocyte Assessment Hackathon and Benchmark Dataset”. arXiv:2301.06304. 2023.

[45] K. Jin, X. Huang, J. Zhou, Y. Li, Y. Yan, Y. Sun, Q. Zhang, Y. Wang, and J. Ye. “FIVES: a Fundus Image Dataset for Artificial Intelligence based Vessel Segmentation”. Vol. 9 (), pp. 475–483.

[46] B. Jing, G. Corso, J. Chang, R. Barzilay, and T. S. Jaakkola. “Torsional Diffusion for Molecular Conformer Generation”. In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022.

[47] A. T. Kalai and S. Vempala. “Simulated Annealing for Convex Optimization”. *Mathematics of Operations Research*, vol. 31, no. 2 (2006), pp. 253–266.

[48] T. Karras, M. Aittala, T. Aila, and S. Laine. “Elucidating the Design Space of Diffusion-Based Generative Models”. In: *Advances in Neural Information Processing Systems*. Ed. by A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho. 2022.

[49] J. Keilson and J. E. Storer. “On Brownian Motion, Boltzmann’s Equation, and the Fokker-Planck Equation”. *Quarterly of Applied Mathematics*, vol. 10, no. 3 (1952), pp. 243–253.

[50] D. Kingma, T. Salimans, B. Poole, and J. Ho. “Variational Diffusion Models”. In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. Vol. 34. Curran Associates, Inc., 2021, pp. 21696–21707.

[51] D. M. Knigge, D. W. Romero, and E. J. Bekkers. “Exploiting Redundancy: Separable Group Convolutional Networks on Lie Groups”. In: *Proceedings of the 39th*

International Conference on Machine Learning. Ed. by K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato. Vol. 162. Proceedings of Machine Learning Research. PMLR, 2022, pp. 11359–11386.

[52] R. Kondor and S. Trivedi. "On the Generalization of Equivariance and Convolution in Neural Networks to the Action of Compact Groups". In: *Proceedings of the 35th International Conference on Machine Learning*. 2018.

[53] S. G. Krantz. "Complex Analysis: The Geometric Viewpoint". 2nd ed. Vol. 23. Mathematical Association of America, 2004.

[54] C.-H. Lai, Y. Takida, N. Murata, T. Uesaka, Y. Mitsufuji, and S. Ermon. "FP-Diffusion: Improving Score-based Diffusion Models by Enforcing the Underlying Score Fokker-Planck Equation". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 18365–18398.

[55] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. "An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation". In: *Proceedings of the 24th International Conference on Machine Learning*. 2007.

[56] B. Levy. "Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry". In: *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. 2006, pp. 13–13.

[57] S. Lin, B. Liu, J. Li, and X. Yang. "Common Diffusion Noise Schedules and Sample Steps are Flawed". In: *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2024, pp. 5392–5399.

[58] P. L. Lions and A. S. Sznitman. "Stochastic differential equations with reflecting boundary conditions". *Communications on Pure and Applied Mathematics*, vol. 37, no. 4 (1984), pp. 511–537. eprint: `https://onlinelibrary.wiley.com/doi/pdf/10.1002/cpa.3160370408`.

[59] G.-H. Liu, A. Vahdat, D.-A. Huang, E. Theodorou, W. Nie, and A. Anandkumar. "I^2SB: Image-to-Image Schrödinger Bridge". In: *Proceedings of the 40th International Conference on Machine Learning*. 2023, pp. 22042–22062.

[60] Z. Liu, P. Luo, X. Wang, and X. Tang. "Deep Learning Face Attributes in the Wild". In: *Proceedings of International Conference on Computer Vision (ICCV)*. 2015.

[61] A. Lou and S. Ermon. "Reflected Diffusion Models". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 22675–22701.

[62] A. Lou, D. Lim, I. Katsman, L. Huang, Q. Jiang, S. N. Lim, and C. M. De Sa. "Neural Manifold Ordinary Differential Equations". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 17548–17558.

[63] A. Lou, M. Xu, and S. Ermon. "Scaling Riemannian Diffusion Models". 2023.

[64] H. Lu, S. Szabados, and Y. Yu. "Structure Preserving Diffusion Models". 2024.

[65] H. Lu, S. Szabados, and Y. Yu. "Diffusion Models with Group Equivariance". In: *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*. 2024.

[66] K. Martinkus et al. "AbDiffuser: full-atom generation of in-vitro functioning antibodies". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[67] E. Mathieu, V. Dutordoir, M. J. Hutchinson, V. De Bortoli, Y. W. Teh, and R. E. Turner. "Geometric Neural Diffusion Processes". In: *Thirty-seventh Conference on Neural Information Processing Systems*. 2023.

[68] G. J. McLachlan and D. Peel. "Finite mixture models". Wiley Series in Probabiliy and Statistics. New York: Wiley, 2000.

[69] J.-L. Menaldi and M. Robin. "Reflected Diffusion Processes with Jumps". *The Annals of Probability*, vol. 13, no. 2 (1985), pp. 319–341.

[70] J. Munkre. "Analysis on Manifolds". Addison-Wesley Publishing Company, 1991.

[71] J. Munkres. "Topology". 2nd ed. Graduate Texts in Mathematics. Springer, Pearson, 2014.

[72] J. Nash. "The Imbedding Problem for Riemannian Manifolds". *Annals of Mathematics*, vol. 63, no. 1 (1956), pp. 20–63.

[73] R. M. Neal. "Bayesian Learning for Neural Networks". lecture Notes in Statistics. Springer, 1996.

[74] A. Q. Nichol and P. Dhariwal. "Improved Denoising Diffusion Probabilistic Models". In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8162–8171.

[75] C. Nour, R. J. Stern, and J. Takche. "Proximal Smoothness and the Exterior Sphere Condition". *Convex Analysis*, vol. 16, no. 2 (2009), pp. 501–514.

[76] B. Øksendal. "Stochastic Differential Equations: An Introduction with Applications". 6th ed. Universeitext. Springer Berlin, Heidelberg, 2003.

[77] S. W. Park, H. Kim, K. Lee, and J. Kwon. "Riemannian Neural SDE: Learning Stochastic Representations on Manifolds". In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 1434–1444.

[78] A. Pilipenko. "An Introduction to Stochastic Differential Equations with Reflection". Lectures in pure and applied mathematics. University Press, 2014.

[79] P. E. Protter. "Stochastic Integration and Differential Equations". Ed. by B. Rozovskii and M. Yor. Stochastic Modeling and Applied Probability. Springer, 2005.

[80] O. Puny, M. Atzmon, E. J. Smith, I. Misra, A. Grover, H. Ben-Hamu, and Y. Lipman. "Frame Averaging for Invariant and Equivariant Network Design". In: *International Conference on Learning Representations*. 2022.

[81] B. Qiang, Y. Song, M. Xu, J. Gong, B. Gao, H. Zhou, W.-Y. Ma, and Y. Lan. "Coarse-to-Fine: a Hierarchical Diffusion Model for Molecule Generation in 3D". In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 28277–28299.

[82] S. Ravanbakhsh, J. Schneider, and B. Póczos. "Equivariance Through Parameter-Sharing". In: *Proceedings of the 34th International Conference on Machine Learning*. 2017.

[83] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. "High-Resolution Image Synthesis With Latent Diffusion Models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.

[84] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi. Cham: Springer International Publishing, 2015, pp. 234–241.

[85] P. Salamon, P. Sibani, and R. Frost. "Facts, Conjectures, and Improvements for Simulated Annealing". Society for Industrial and Applied Mathematics, 2002.

[86] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma. "PixelCNN++: Improving the PixelCNN with Discretized Logistic Mixture Likelihood and Other Modifications". In: *International Conference on Learning Representations*. 2017.

[87] S. Särkkä and A. Solin. "Applied Stochastic Differential Equations". Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2019.

[88] J. Shawe-Taylor. "Symmetries and discriminability in feedforward network architectures". *IEEE Transactions on Neural Networks*, vol. 4, no. 5 (1993), pp. 816–826.

[89] C. Shi, S. Luo, M. Xu, and J. Tang. "Learning Gradient Fields for Molecular Conformation Generation". In: *International Conference on Machine Learning*. 2021.

[90] J. Song, C. Meng, and S. Ermon. "Denoising Diffusion Implicit Models". In: *International Conference on Learning Representations*. 2021.

[91] Y. Song and S. Ermon. "Generative Modeling by Estimating Gradients of the Data Distribution". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.

[92] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *International Conference on Learning Representations*. 2021.

[93] D. W. Stroock and S. Varadhan. "Diffusion Processes". In: vol. 3. 1972, pp. 361–368.

[94] C. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. Cardoso. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations". *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10553 LNCS (2017), pp. 240–248.

[95] R. S. Sutton and A. G. Barto. "Reinforcement Learning: An Introduction". Cambridge, MA, USA: A Bradford Book, 2018.

[96] A. Tarvainen and H. Valpola. "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.

[97] Y. Tassa, T. Erez, and E. Todorov. "Synthesis and stabilization of complex behaviors through online trajectory optimization". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012, pp. 4906–4913.

[98] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters. "Learning Implicit Priors for Motion Optimization". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 7672–7679.

[99] P. Vincent. "A connection between score matching and denoising autoencoders". *Neural Comput.*, vol. 23, no. 7 (2011), pp. 1661–1674.

[100] Z. 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*, vol. 13, no. 4 (2004), pp. 600–612.

[101] M. Xu, L. Yu, Y. Song, C. Shi, S. Ermon, and J. Tang. "GeoDiff: A Geometric Diffusion Model for Molecular Conformation Generation". In: *International Conference on Learning Representations*. 2022.

[102] J. Yim, B. L. Trippe, V. De Bortoli, E. Mathieu, A. Doucet, R. Barzilay, and T. Jaakkola. “SE(3) diffusion model with application to protein backbone generation”. In: *Proceedings of the 40th International Conference on Machine Learning*. Ed. by A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett. Vol. 202. Proceedings of Machine Learning Research. PMLR, 2023, pp. 40001–40039.

[103] L. Zhou, A. Lou, S. Khanna, and S. Ermon. “Denoising Diffusion Bridge Models”. In: *The Twelfth International Conference on Learning Representations*. 2024.

Appendix A

Definitions and background

Here we introduce various definitions and concepts from optimization and differential geometry that are used throughout the text. As there is a lot of subtly in the domains we are interested in and differences in notation between sources, we set forth some standard definitions and properties in an effort to make this thesis, at least in part, more self contained and digestible.

A.1 Topology

We will begin by introducing the notion of a topological space and coordinate charts.

Definition 8 (Topological space, see [71])**.** *A topological space is a set X along with a associated collection $\Omega \subseteq \mathcal{P}(X)$ of subsets of S defined to be the open sets of X, which define the topology on the space, satisfying:*

1. *$\emptyset \in \Omega$ and $X \in \Omega$;*
2. *if $A, B \in \Omega$ then $A \cap B \in \Omega$;*
3. *for any index set I, $\bigcup_{i \in I} A_i \in \Omega$; i.e., the arbitrary union of open sets is itself open.*

The topological space is the ordered pair (X, Ω).

In the following, we always assume we are dealing with the usual topology on $X \subseteq \mathbb{R}^d$ unless this does not make sense from context.

Definition 9 (Chart, see [71, 7])**.** *A chart (V, ϕ) on a topological space (X, Ω) is an open subset $V \subseteq X$ together with an open embedding, $\phi : U \hookrightarrow V$ where $U \subseteq \mathbb{R}^d$, that maps open sets to open sets of the respective topologies.*

Definition 10 (Connected)**.** *Let X be a topological space. A separation of X is a pair $U, V \subseteq X$, $U \cap V = \emptyset$ with $U \cup V = X$. The space X is called connected if no separating pair exists.*

Unless otherwise stated, we will implicitly assume all spaces considered are connected, and usually compact.

Definition 11 (Path connected)**.** *Let X be a topological space. The space X is called path connected if $\forall x_0, x_1 \in X$, $\exists f : [0,1] \to X$, continuous, with $f(0) = x_0$ and $f(1) = x_1$.*

The next definition (simply connected) serves to identify properties of a space in a similar notion to the idea of convexity. Informally speaking, a space X is simply connected if every closed curve (e.g., Jordan curve) in X can be contracted down to a point in X.

Definition 12 (Simply connected)**.** *A topological space X is simply connected if it is path connected and its fundamental group $\pi_1(X, x_0)$ is the trivial group (i.e., one element under Path homotopy equivalence).*

A.2 Real analysis

Following our brief discussion on topology, we immediately fall back into the standard Euclidean setting and use the preceding definitions to formalize some intuitive ways of describing different spaces.

Definition 13 (Smooth bounded domains, [53])**.** *A domain U, that is bounded, is said to have twice continuous differentiable boundary, denoted $\partial U \in C^2$, if the boundary is composed of finitely many pairwise disjoint, simple closed (Jordan), twice continuously differentiable curves $\gamma \in C^2$; i.e., $\gamma : [0,1] \to U$.*

Definition 14 (Exterior sphere condition, [75])**.** *Let U be a domain with boundary ∂U. The domain U is said to satisfy the (uniform) exterior sphere condition if there exists a $r > 0$ s.t. for all $x \in \partial U$ there exists a $z \notin U$ with $d(x, z) = r$ and*

$$B(z, r) \cap U = \emptyset.$$

The domain U is said to satisfy the (uniform) Interior sphere condition if $(\text{intU})^c$, the complement of the interior, satisfies the (uniform) exterior sphere condition.

Now we will define a very useful notation of how "rough" a function is on a given interval. This plays an important role in being able to analyze certain boundary value problems that popup when discussing diffusion processes.

Definition 15 (Total variation, [2]). *Let $a, b \in \mathbb{R}$ and $f \in C([a,b], \mathbb{R})$. The total variation of f over $[a,b]$, denoted $V_{a,b}(f)$, is equal to*

$$\sup_{\substack{(x_t)_{t=1}^n \\ n \in \mathbb{N}}} \left\{ \sum_{t=1}^{n} |f(x_{t+1}) - f(x_t)| \right\}$$

over all partitions, $(x_t)_t$, $a = x_0 < x_1 < \cdots < x_{n-1} < x_n = b$ for any $n \geq 0$.

Definition 16 (Functions of bounded variation, [2]). *Let $a, b \in \mathbb{R}$ and $f \in C([a,b], \mathbb{R})$. A function $f : [a,b] \to \mathbb{R}$ is of bounded variation, denoted $f \in BV[a,b]$, if there exists a $M \in \mathbb{R}_{\geq 0}$ such that $V_{a,b}(f) \leq M$.*

Functions of bounded variations can be better understood through the characterization, on compact intervals, of expressing them as the difference of two monotonic functions.

A.3 Differential geometry

As one of the core aspects of this thesis involves differential manifolds, we take time here to develop the absolute bare essentials needed to understand the forthcoming work. Fundamentally, the construction of (differential) manifolds, at least those that we will be considering, is centered around the definition of coordinate systems; these coordinate systems are occasionally referred to as curvilinear coordinates as, unlike standard Euclidean coordinates, they may posses curvature. It is useful to picture coordinate systems of differential manifolds as being a spacial indexed functional basis, where the given basis changes smoothly as you traverse the surface of the manifold.

Definition 17 (Smooth manifold, see [70]). *Let $(\mathcal{M}, d)$ be a metric space and $\{\phi_i : U_i \to V_i\}$ be a collection of (topological) charts (that is, a collection of homeomorphisms) where each U_i is open in $\mathbb{R}^k$, and V_i is open in $\mathcal{M}$, such that the sets $\{V_i\}$ form an open cover of $\mathcal{M}$. Further suppose, the maps $\{\phi_i\}$ overlap with class C^∞, meaning, $\phi_i^{-1} \circ \phi_j \in C^\infty$ provided $V_i \cap V_j \neq \emptyset$. Then the metric space together with the set of charts, say $(\mathcal{M}, d, \{\phi_i\})$, is a differentiable k-manifold of class C^∞.*

Note, there is some disagreement on the definition of charts between sources, particularly between topology textbooks [42, 70, 71] and those discussing differential geometry, stating a chart as written above or in terms of the pre-image of the above. Additionally,

I sometimes break from convention and write the local coordinate (functions) using subscripts rather than superscripts since that notation is likely to be more familiar to readers within the machine learning domain. Additionally, it is often assumed all manifolds of a specified dimension are of class C^∞ without it being explicitly stated.

We will now explore a quick example of how one might define a smooth manifold as subspace of a higher dimensional Euclidean ambient space, which is guaranteed to exist, for a sufficiently high dimensional ambient space, under the Nash embedding theorem [72].

Example: Smooth manifolds defined as subspaces A subset $\mathcal{M} \subset \mathbb{R}^d$ is a k-dimensional manifold in $\mathbb{R}^d$, for $d \geq k$, if and only if $\forall x \in \mathcal{M}$ there exists $U \subset \mathbb{R}^d, x \in U$, an open set $V \subset \mathbb{R}^k$, and a injective differentiable function $f : V \to \mathbb{R}^n$ such that:

1. $f(V) = \mathcal{M} \cap U$,
2. $\mathbb{J}_f(y)$, the Jacobian of f, has rank k for all $y \in V$,
3. $f^{-1} : f(V) \to V$ is continuous;

such a f is called a coordinate system around x.

Coordinate systems are the chosen basis of functions on that space and, as defined above, correspond to a special case of having smooth charts, as in Definition 17, labeling what points exist within local regions of the manifold.

We will primarily be concerned with the specific class of smooth manifolds that admit inner products, specifically Riemannian manifolds.

Definition 18 (Riemannian manifold, see [7, 70, 24])**.** *A* Riemannian manifold*, denoted* $(\mathcal{M}, g)$*, is comprised of a smooth manifold* $\mathcal{M}$ *and a metric* g *which defines an inner product over the tangent space(s)* $T_p(\mathcal{M})$ *to any point* $p \in \mathcal{M}$*. The collection of all tangent spaces, denoted* $\mathcal{TM}$*, is called the tangent bundle.*

Up to now, we have not discussed, at least in clear terms, how one can operate over manifolds using charts, or how to go about multiplying and adding functions that take on manifold values or even if such a notion is well posed. As Riemann manifolds are generally not vector spaces (globally), vector operations, at least using local charts – which while simple are not overly convent, are only defined locally. We now briefly provide such an example.

Example: manifold operations Suppose $(\mathcal{M}, g)$ is a Riemannian manifold equipped with a chart $\phi \in C^\infty$, if $p = \phi(x^1, x^2, \ldots, x^d) \in U$ the components $x^1, x^2, \ldots, x^d$ are called (local) coordinates of p w.r.t. the chart ϕ, more specifically, $x^j = \phi_j^{-1}(p)$ and the coordinate vector fields at p are defined using the differential

$$\frac{\partial}{\partial x^j}|_p = d\phi(\frac{\partial}{\partial x^j}|_x)$$

where $x = \phi^{-1}(p)$ and $\frac{\partial}{\partial x^j}|_x$ is the j-th standard basis vector in $\mathbb{R}^d$ differentiated through the chain rule. Now let's consider $f : \mathbb{R} \to \mathcal{M}$ and $g : \mathbb{R} \to \mathcal{M}$ are paths along $\mathcal{M}$, and assume for all $t \in \mathbb{R}$ both $f(t) = p$ and $g(t) = q$ lie within $V \subseteq \mathcal{M}$ (open) with local chart $\phi : U \hookrightarrow V \subset \mathcal{M}$ with $U \subseteq \mathbb{R}^d$ open (as $\mathcal{M}$ is a smooth manifold it is assumed ϕ is a diffeomorphism). Then, in local coordinates

$$\begin{aligned} x(t) &= \phi^{-1}(f^1(t), f^2(t), \ldots, f^d(t)), \\ y(t) &= \phi^{-1}(g^1(t), g^2(t), \ldots, g^d(t)). \end{aligned}$$

we have, with some abuse of notation for compactness, that

$$\begin{aligned} f(t) + g(t) &= \phi(x^1(t) + y^1(t), \ldots, x^d(t) + y^d(t)), \\ f(t) \odot g(t) &= \phi(x^1(t)y^1(t), \ldots, x^d(t)y^d(t)). \end{aligned}$$

There are different approaches in which standard vector and differentiation operations can be carried out on a manifold, which are used later on implicitly; .e.g., Riemannian exponential map via geodesics when available, etc.[1]Continuing this example, if we instead consider vector fields of the form $f, g : \mathbb{R} \to \mathcal{T}\mathcal{M}$; i.e., for each $t \in \mathbb{R}$ $f(t) \in T_p\mathcal{M}$, then by recalling the expression for coordinate vector fields, f takes on the local coordinate form

$$f(t) = \sum_{i=1}^{d} f^i(t) \frac{\partial}{\partial x^i}|_p.$$

Then, provided $f(t), g(t) \in T_p\mathcal{M}$, we can define

$$\begin{aligned} (f + g)(t) &= \sum_{i=1}^{d} (f^i(t) + h^i(t)) \frac{\partial}{\partial x^i}|_p \\ (\alpha f)(t) &= \sum_{i=1}^{d} (\alpha f^i(t)) \frac{\partial}{\partial x^i}|_p, \quad \forall \alpha \in \mathbb{R}. \end{aligned}$$

[1]It should also be noted in general the above construction has no guarantees of producing points that lie on the manifold unless $p, q \in V$.

Definition 19 (Constrained manifolds, see [24]). *Let $(\mathcal{M}, g)$ be a Riemannian manifold and $\{f_i\}_{i\in I}$ a family of real-valued functions from $C^3(\mathcal{M}, \mathbb{R})$. The manifold*

$$\mathcal{D} = \{x \in \mathcal{M} \mid f_i(x) \leq 0, \forall i \in I\}$$

equipped with g is called a constrained manifold.

Associated to each Riemannian manifold is an isometry group, which consists of a set of local (and global) isometries over the manifold.

Definition 20 (Manifold isometry group). *Let $(\mathcal{M}, g)$ be a Riemannian manifold. The isometry group of $\mathcal{M}$, denoted $I_{\mathcal{M}}$, is a Lie group of diffeomorphisms s.t.,*

$$\begin{aligned} I_{\mathcal{M}} \times \mathcal{M} &\to \mathcal{M} \\ (\kappa, p) &\mapsto L_\kappa p \end{aligned}$$

where L_κ is the (left) action of κ on the point p, and each $\kappa \in I_{\mathcal{M}}$ preserves the pull back $\kappa^* g = g$ metric, and satisfies the conditions: $\forall p \in \mathcal{M}$

1. *$\exists e \in I_{\mathcal{M}}$ s.t., $L_e p = p$*
2. *$\forall \kappa_1, \kappa_2 \in I_{\mathcal{M}}$, $L_{\kappa_1,\kappa_2} p = L_{\kappa_1}(L_{\kappa_2} p)$.*

Of particular interest is the manifold group of symmetric manifolds, as these manifolds appear most commonly in physical applications and process attractive group properties.

Definition 21 (Symmetric manifolds, [31]). *A Riemannian manifold, $(\mathcal{M}, g)$, is symmetric (or globally symmetric) if $\forall p \in \mathcal{M}$ there exists an isometry $\kappa : \mathcal{M} \to \mathcal{M}$ such that*[2] *$dL_\kappa|_p = -id|_p$.*

A.4 Reflected diffusion assumptions

Here we present some additional assumptions and results that are used implicitly within Chapter 4. These results serve to characterize the kinds of domains are considered within the aforementioned section, but do not aid the macroscopic delivery so were removed from the main text.

[2]Here I opt for a characterization given on [31, p.6]

Assumption 1 (Used in [58, 69, 5])**.** *For a given domain $\mathcal{D} \subseteq \mathcal{X}$, there exists $f \in C^3(\mathcal{X}, \mathbb{R})$ s.t.*

$$\begin{aligned} \mathcal{D} &= \{x \in \mathcal{X} \mid f(x) < 0\}; \\ \partial\mathcal{D} &= \{x \in \mathcal{X} \mid f(x) = 0\}; \\ \|\nabla f(x)\|_2 \geq 1 \quad &, \forall x \in \partial\mathcal{D} \qquad\qquad (\text{ or } \nabla f(x) \neq 0) \end{aligned}$$

and lastly $\mathcal{D}$ is outer-bounded – in the sense that $\exists r \geq 0$ s.t. $\mathcal{D} \subseteq B(x, r)$ for some $x \in \mathcal{D}$.

Lemma 3 (from [58])**.** *If $\mathcal{D} \in C^1$ is an open domain that satisfies the uniform (exterior) sphere condition for radius $r > 0$ equip with the vector field $n : \partial\mathcal{D} \to \mathcal{T}\partial\mathcal{D}$, defining the unit outwards normal vectors along $\partial\mathcal{D}$, then for $R \geq \frac{1}{2}r$ and $\forall z \in \overline{\mathcal{D}}$:*

$$\langle n(x), x - z\rangle + R\langle x - z, x - z\rangle \geq 0.$$

Assumption 2. *For the given smooth domain $\mathcal{D}$ that satisfies the uniform (exterior) sphere condition, assume there exists a function $\phi \in C_b^2(\mathbb{R}^d)$ such that $\exists \alpha > 0$, $\forall x \in \partial\mathcal{D}$, $\forall v \in n(x)$ with $\nabla\phi(x) \cdot v \leq -\alpha R$, where R is from Lemma 3.*

The results of Lemma 3 and Assumption 2 serve to describe the level of smoothness required by the domain boundary w.r.t. the rate of change of the outward normal vector field. The following assumption goes onto quantify the total change of the outwards vector field evaluated over a covering of the domain boundary.

Assumption 3 (From [58].)**.** *For domain $\mathcal{D}$, assume: $\exists n \geq 1$, $\exists \alpha > 0$, $\exists R > 0$ s.t.,*

1. *$\exists a_1, \ldots, a_n \in \mathbb{R}^d$ with $\|a_i\| = 1$, $\forall i$;*
2. *$\exists x_1, \ldots, x_n \in \partial\mathcal{D}$ with $\partial\mathcal{D} \subseteq \bigcup_{i=1}^n B(x_i, R)$ and $\forall i$, $\forall x \in \partial\mathcal{D} \cap B(x_i, 2R)$ we have $\langle n(x), a_i\rangle \geq \alpha$.*

Appendix B

Invariant diffusion additional material

B.1 Invariant FID computation

In Section 5.3 we report the Fréchet intercept distance (FID) [32] score of various models on the datasets described in Section 5.3, respectively under C_4 and D_4 groups. In order to ensure the FID score is invariant to these group operations applied to sample images, without modifying the underlying InceptionV3 model – meaning the features the underlying InceptionV3 model extracts from the reference dataset can be compared to those extracted from the generated samples – we compute the mean score over all group elements. That is, given a reference dataset $\mathcal{D}_{ref}$, a collection of generated samples $\mathcal{D}_s$ and a group $\mathcal{G}$, and if $\mathrm{T}(\cdot)$ denotes the FID model evaluation that returns the mean and covariance statistics of the features extracted from a dataset; i.e., $T(\mathcal{D}_s) = (\mu_s, \Sigma_s)$. Then we compute FID by first computing

$$\mathrm{T}_{\mathcal{G}}(\mathcal{D}_{ref}) = \frac{1}{|\mathcal{D}_{ref}|} \sum_{h \in \mathcal{G}} T(A_h \mathcal{D}_{ref}) = (\hat{\mu}, \hat{\Sigma}),$$

where $A_h \mathcal{D}_{ref} = \{A_h x \mid x \in \mathcal{D}_{ref}\}$, and then we report the FID score as

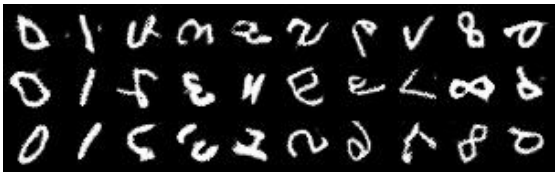
$$\mathrm{FID}_{\mathcal{G}} = \|\mathrm{T}_{\mathcal{G}}(\mathcal{D}_{ref}) - \mathrm{T}(\mathcal{D}_s)\|.$$

This formulation ensures that the reference statistics used in computing the FID score of a model is invariant to the group. All FID values reported in Table 5.1 and Table 5.2, potentially excluding those reported by other authors, were calculated in the above fashion.

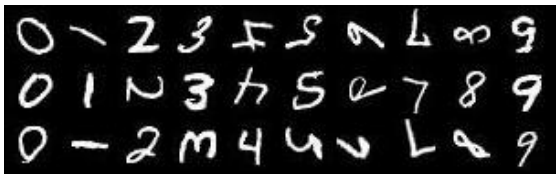
B.2 Structure preserving diffusion model samples

Here, we include a collection of image samples from the models discussed within the text across the various datasets in Section 5.3.

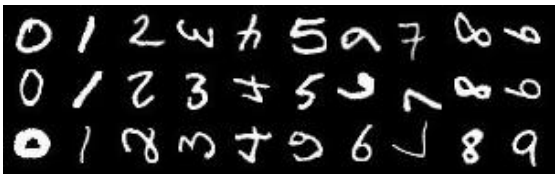
(a) Reference C_4 rotated MNIST images.



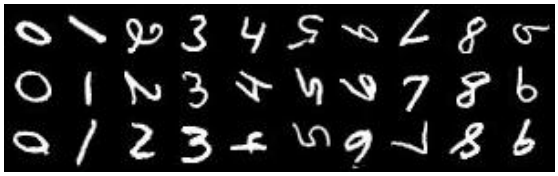
(b) Images generated from SP-GAN.



(c) Images generated from VP-SDE.

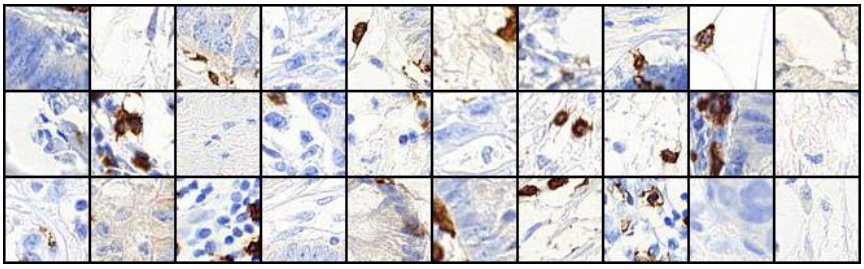


(d) Images generated from SPDM+WT

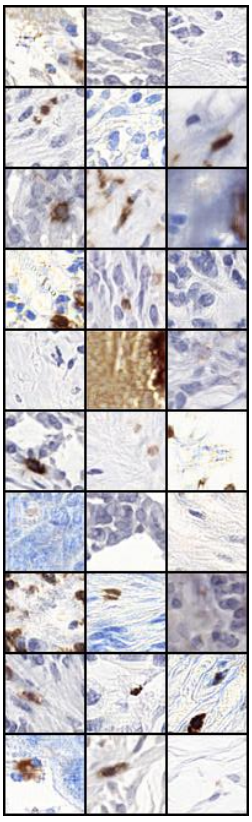


(e) Images generated from SPDM+FA

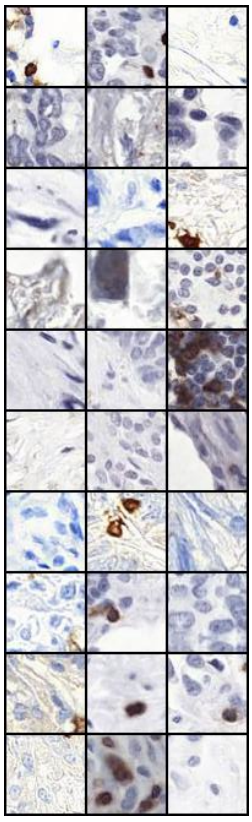
Figure B.1: Sample comparison between models trained on the Rotated MNIST (28x28x1) dataset.



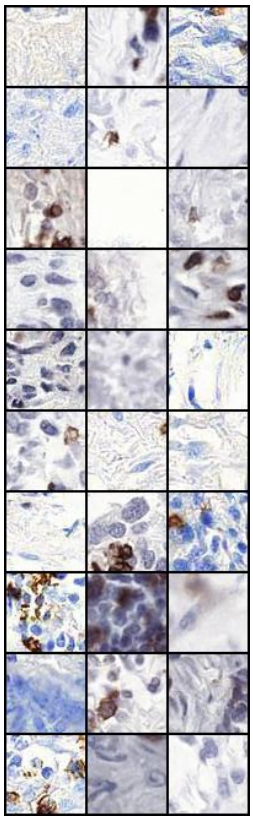
(a) Reference images.



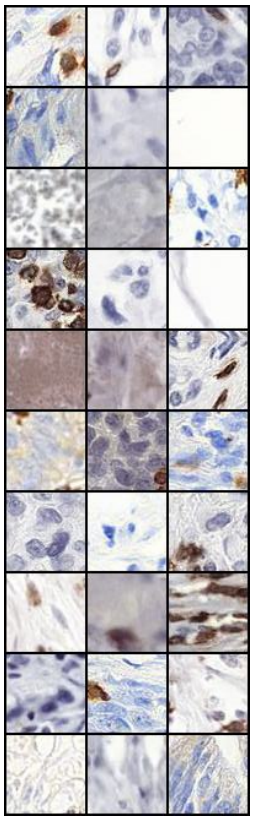
(b) SP-GAN samples.



(c) VP-SDE samples.



(d) SPDM+WT samples.



(e) SPDM+FA samples.

Figure B.2: Sample comparison between models trained on the LYSTO (64x64x3) dataset.

Appendix C

Extraneous experiments

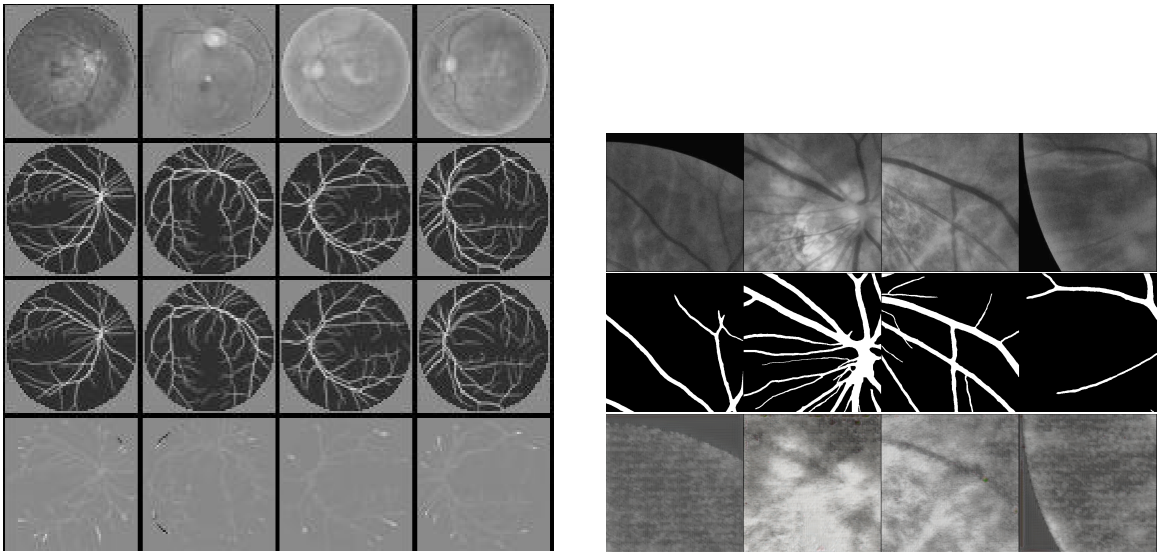
C.1 Structure preserving pixel mask generation

Experiment: Fundus image mask generation One of the original tasks we proposed as an evaluation of the models in Section 5.3 was to (conditionally) generate pixel (one-hot) image masks for fundus images of human eyes utilizing the FIVES dataset. The FIVES dataset [45] provides 800 fundus images at a resolution of (2048x2048x3) with matching pixel-wise annotations of eye vasculature from patients with diabetic retinopathy (DR), age-related macular degeneration (AMD), and a control with healthy eyes. This data is helpful for performing certain medial diagnosis of eye disease. As the captured images are circumscribed they exhibit natural SO(2) invariance.

Two methods where considered, both motivated by the memory constrains of attempting to train a U-Net based diffusion model on images exceeding resolutions beyond (128x128x3) in pixel-space on a NVIDIA L40S GPU (40GB).

First, we attempted scaling the images down to the resolution of (64x64x3) making direct pixel-space training feasible. I implemented a DDBM+WT[1] model and trained it under the C_4 group. However, this model was not able to produce reasonable results. Suspecting the sparsity of the generation task was the issue, a DICE loss [94] regularizer was added alongside the DSM, Eq. (2.7), loss to try improving the results. While this generated initially promising results, the model quickly converged, reaching a MSE loss of 0.201 on the training data and 0.301 on the test data with a dice loss of 0.201 and 0.205 respectively, and did not end up delivering sufficiently detailed image masks; I suspect this is due to the aliasing produced by the image scaling. Some example images are provided in Fig. C.1a.

[1]The implementation of this can be found on a branch of the central repository given above for [65].



(a) (b)

Figure C.1: (a) Reference and sample (64x64x3) images generated from DDBM+WT fundus model. The first row is the condition followed by the reference mask, generated mask, and lastly the MSE between the reference and generated masks. (b) Reference and sample (512x512x3) image patches generated from latent space DDBM+WT fundus model. The first row is the condition followed by the reference mask and lastly the generated mask.

Following this, we implemented a latent space diffusion model based on Stable Diffusion v1-4 [83] accepting image patches of size (512x512x3) and operating on a latent space of size (32x32x4). The VAE, was fine-tuned from a pre-trained checkpoint from the same model and learned to accurately reconstruct both the fundus images and pixel masks down to an accuracy of $1e^{-6}$; however, the DDBM+WT diffusion model struggled to produce meaningful results, see Fig. C.1b. Again a DICE loss regularizer was added to training in an attempt to correct this but this only marginally improved the initial results, and later hampered performance by perturbing the DSM, Eq. (2.7), estimate of the score.