Stochastic Continuous Normalizing Flows: Training SDEs as ODEs

Liam Hodgkinson[1,2] **Chris van der Heide**[3] **Fred Roosta**[2,3] **Michael W. Mahoney**[1,2]

[1]Department of Statistics, University of California Berkeley, Berkeley, CA, USA
[2]International Computer Science Institute, Berkeley, CA, USA
[3] School of Mathematics and Physics, University of Queensland, Australia

Abstract

We provide a general theoretical framework for *stochastic continuous normalizing flows*, an extension of continuous normalizing flows for density estimation of stochastic differential equations (SDEs). Using the theory of rough paths, the underlying Brownian motion is treated as a latent variable and approximated. Doing so enables the treatment of SDEs as random ordinary differential equations, which can be trained using existing techniques. For scalar loss functions, this approach naturally recovers the stochastic adjoint method of Li et al. [2020] for training neural SDEs, while supporting a more flexible class of approximations.

1 INTRODUCTION

Normalizing flows [Rezende and Mohamed, 2015] are probabilistic models constructed as a sequence of successive transformations applied to some initial distribution. Building on the change-of-variables formula for densities, normalizing flows enjoy significant expressive power as generative models, while possessing an explicitly computable form of the likelihood function evaluated over the transformed space. This makes them especially well-suited for variational inference (VI).

Continuous normalizing flows (CNF) were soon after developed by Chen et al. [2018] as a means of performing density estimation for probabilistic models derived from ordinary differential equations (ODEs). Extending the change-of-variables method from discrete to continuous time, the CNF framework stems from the Liouville equation: an expression for the evolving density of an ODE with random initial values, as the solution to another ODE. The jump to continuous-time dynamics affords a few computational benefits over its discrete-time counterpart, namely the presence of a trace in place of a determinant in the evolution formulae for the density, as well as the adjoint method for memory-efficient backpropagation. Motivated by deep learning, a family of ODEs, called *neural ordinary differential equations* were constructed, whose Euler discretizations resembles layer-wise transformations of residual neural networks. Further algorithmic improvements to the framework were presented by Grathwohl et al. [2018], enabling effectively arbitrary choices of parameterized classes of ODEs. Doing all this involves some technical subtlety, and effective neural ODE architectures remain the subject of ongoing research — see for example [Dupont et al., 2019, Gholami et al., 2019, Zhang et al., 2019].

There has also been recent interest in extending these frameworks to a stochastic scenario; that is, training probabilistic models derived from *stochastic* differential equations (SDEs). For physical models, where the evolution of a dynamical system is no longer deterministic, or microscopic fluctuations are dependent on components changing too rapidly to quantify, an SDE can be more appropriate. Stochastic extensions of neural ODEs have been considered in [Tzen and Raginsky, 2019, Liu et al., 2019, Jia and Benson, 2019, Peluchetti and Favaro, 2019] as limits of deep latent Gaussian models, where they have been suggested to show increased robustness to noisy or adversarial data. Furthermore, unlike deterministic flows, there is a foolproof recipe for constructing a family of SDEs that are ergodic with respect to some target density [Ma et al., 2015]. Such SDEs are prime candidates for the construction of stochastic MCMC algorithms, by generating sample paths via approximate stochastic integration methods.

However, developing an analogue of the continuous normalizing flows framework for flows constructed from SDEs—in particular, one that comes with simple and rigorous mathematical theory and that does not rely on ad hoc or problem-specific assumptions—is far from trivial. In fact, density estimation for SDEs is a notoriously challenging task in general — see Hurn et al. [2007] for a summary of existing techniques, each of which are limited in scope. A common approach for conducting VI with SDEs relies upon Gir-

Accepted for the 37th Conference on Uncertainty in Artificial Intelligence (UAI 2021).

sanov's theorem. This allows one to estimate the Kullback-Leibler divergence between densities of solutions to two SDEs (for the prior and posterior distributions) with differing drift coefficients [Beskos et al., 2006, Tzen and Raginsky, 2019]. Following this approach, Li et al. [2020] developed a stochastic adjoint method which scales well to high dimensions, and enables SDEs as latent models in variational autoencoders. Notable difficulties with these previous approaches include an incompatibility with higher-order adaptive SDE solvers, and a complex mechanism of reconstructing Brownian motion paths from random number generator seeds.

CONTRIBUTIONS

We provide a general theoretical framework for training models constructed from SDEs using existing techniques applied to random ODE approximations. By this process, we find that theoretical and practical developments concerning continuous normalizing flows and neural ODEs extend readily to the stochastic setting, without the need of an independent framework. The key tools underlying our strong results and simple analysis is the *theory of rough paths* [Friz and Hairer, 2014], an alternative stochastic calculus that enables approximation and pathwise treatment of SDEs. Our approach

1. enables density estimation for *arbitrary* SDE models: an extension of CNF that we refer to as *stochastic continuous normalizing flows* (SCNF); and
2. recovers the stochastic adjoint method of Li et al. [2020], but is sufficiently flexible to be used in conjunction with arbitrary higher-order numerical ODE solvers.

Under our framework, SCNF can be easily implemented using any general CNF implementation, such as that of Grathwohl et al. [2018]. Moreover, our approach allows any existing neural ODE framework (such as Zhang et al. [2019]) to be extended to SDEs, simply by the addition of a few extra terms.

Following a review of background material in §2, the SCNF framework is introduced and discussed in §3, with our main approximation result presented as Theorem 2. Some numerical investigations are conducted in §4.

2 BACKGROUND REVIEW

2.1 CONTINUOUS NORMALIZING FLOWS

We begin by reviewing the continuous normalizing flow framework for training ODE models, from which we will develop random and stochastic continuous normalizing flows. Consider a parameterized class of models $\{Z_\theta\}_{\theta \in \mathbb{R}^m}$ of the following form: for $f : \mathbb{R}^d \times [0,T] \times \mathbb{R}^m \to \mathbb{R}^d$, let $Z = Z_\theta \in \mathbb{R}^d$ satisfy the ODE with random initial condition (often called a *random ordinary differential equation*)

$$\frac{\mathrm{d}}{\mathrm{d}t} Z(t) = f(Z(t), t, \theta), \quad Z(0) \sim p_0(\theta). \tag{1}$$

In a general machine learning context, one might choose f such that the Euler discretization of (1) resembles layer-wise updates of a residual neural network [Lu et al., 2017, Chen et al., 2018], or one may parameterize f as a neural network itself [Grathwohl et al., 2018]. The resulting ODEs constitute the class of so-called *neural ordinary differential equations*. The following theorem is a consequence of the Liouville equation (equivalently, Fokker-Planck equation) applied to the solution $Z(t)$ of the random ODE (1). It provides an ODE for the log density of $Z(t)$ evaluated at $Z(t)$.

Theorem 1 (Chen et al. [2018])**.** *Suppose that $Z(t)$ satisfies (1). The distribution of $Z(t)$ is absolutely continuous with respect to Lebesgue measure, with probability density p_t satisfying*

$$\frac{\mathrm{d}}{\mathrm{d}t} \log p_t(Z(t)) = -\nabla_z \cdot f(Z(t), t, \theta) \tag{2}$$

Naively computing the divergence in (2) with automatic differentiation is of quadratic complexity in the dimension d. As pointed out by Grathwohl et al. [2018], this can be improved to linear complexity using a trace estimator [Roosta and Ascher, 2015]:

$$\nabla_z \cdot f(z) = \mathrm{tr}\left(\frac{\partial f}{\partial z}\right) \approx \frac{1}{n} \sum_{k=1}^{n} \epsilon_k^\top \frac{\partial f}{\partial z} \epsilon_k, \tag{3}$$

where each ϵ_k is an independent and identically distributed copy of a random vector $\epsilon \in \mathbb{R}^d$ with zero mean and $\mathbb{E}[\epsilon\epsilon^\top] = I$. Common choices for ϵ_k include standard normal and Rademacher random vectors.

2.2 THE ADJOINT METHOD

Training continuous normalizing flows often involves minimizing a scalar loss function involving Z and/or the log-density computed via Theorem 1 with respect to the parameters θ. For this, we require gradients of $Z(t)$ with respect to θ for $t \in [0,T]$. The most obvious approach is to directly backpropagate through a numerical integration scheme such as in Ryder et al. [2018], but this does not scale well in T. A more elegant alternative is the *adjoint method*, which computes derivatives of a scalar loss function by solving an appropriate differential equation in reverse time. Letting L denote a scalar loss depending on $Z(T)$, the *adjoint* given by $a(t) = \frac{\partial L}{\partial Z(t)}$, as well as the gradient of L in θ, satisfy

[Pontryagin, 2018, §12]

$$\frac{\mathrm{d}}{\mathrm{d}t}a(t) = -\nabla_z f(Z(t),t,\theta)a(t), \tag{4a}$$

$$\nabla_\theta L = \int_0^T \nabla_\theta f(Z(t),t,\theta)a(t)\mathrm{d}t. \tag{4b}$$

Together with (1), the equations (4) are solved in reverse time, starting from the terminal values $Z(T)$ and $\nabla L(Z(T))$. By augmenting $Z(t)$ together with (2), this method also allows for loss functions depending on $p_T(Z(T))$.

Solving (1), (2), and (4) can be achieved using off-the-shelf numerical integrators. Adaptive solvers prove particularly effective, although the backward solve (4) can often run into stability issues [Gholami et al., 2019], suggesting a Rosenbrock or other implicit approach [Hairer and Wanner, 1996]. The same is true in stochastic settings; see Hodgkinson et al. [2019], for example. For further implementation details concerning continuous normalizing flows, we refer to Grathwohl et al. [2018].

More recently, a *stochastic adjoint method* for SDEs was developed in [Li et al., 2020]. The principle is the same, with systems of ODEs replaced by corresponding SDEs. However, their theoretical justification is both delicate and complex due to its reliance on classical stochastic calculus, which is ill-suited for analyzing backward (approximate) solutions to SDEs. One critical attribute — perhaps mysterious at first — is that integration must be performed in the Stratonovich calculus (§3.1). It turns out that these results are better explained using rough path theory: because Stratonovich SDEs can be arbitrarily well-approximated by ODEs, the adjoint method for ODEs extends naturally. Indeed, the stochastic adjoint method is contained in equation (15) in our Theorem 2 as a byproduct of our framework.

2.3 ROUGH PATH THEORY

The theory of rough paths was first introduced in [Lyons, 1998] to provide a supporting pathwise theory for SDEs. It has since flourished into a coherent pathwise alternative to stochastic calculus, facilitating direct stochastic generalizations of results from classical ODE theory — we refer to Friz and Hairer [2014] for a gentle introduction, and Friz and Victoir [2010] for a thorough treatment of the topic. For reasons we soon describe, Hölder continuity is critical to rough path theory — in the sequel, we equip the space of α-Hölder functions with the usual α-Hölder norm,

$$\|X\|_\alpha := \sup_{t\in[0,T]}\|X_t\| + \sup_{\substack{s,t\in[0,T]\\ s\neq t}}\frac{\|X_t - X_s\|}{|t-s|^\alpha}.$$

Suppose that we would like to prescribe meaning to the infinitesimal limit of the sequence of iterates

$$Z_{t+h} = Z_t + f(Z_t)(X_{t+h} - X_t), \quad \text{as } h \to 0^+. \tag{5}$$

In the case of SDEs, X_t is a sample path of Brownian motion, so that each $X_{t+h} - X_t$ is a realization of a normal random vector with zero mean and covariance hI. Unfortunately, a strong limit of (5) fails to exist if X_t is too "rough". In particular, suppose that X_t is α-Hölder continuous for $\alpha \in (0,1)$, that is, there exists some $C > 0$ such that $\|X_s - X_t\| \leq C|s-t|^\alpha$ for any $s,t \geq 0$. Since the limit (5) is only well-defined if $\alpha \geq 1/2$ [Young, 1936], a function on $[0,T]$ is called *rough* if it is Hölder-continuous only for $\alpha < 1/2$. This is significant, since sample paths of Brownian motion constitute rough paths under this definition, as they are known to be α-Hölder continuous for any $\alpha < 1/2$ [Friz and Hairer, 2014, pg. 27].

The problem with establishing a strong limit is that the discretization (5) invokes the *zeroth-order* approximation $f(Z_{t+s}) \approx f(Z_t)$ for $0 \leq s \leq h$, which proves too poor. By instead taking a *first-order* approximation

$$\begin{aligned} f(Z_{t+s}) &\approx f(Z_t) + \nabla_z f(Z_t)(Z_{t+s} - Z_t) \\ &\approx f(Z_t) + \nabla_z f(Z_t) f(Z_t)(X_{t+s} - X_t), \end{aligned}$$

we arrive at the *Davie scheme* [Davie, 2008]

$$Z_{t+h} = Z_t + f(Z_t)(X_{t+h} - X_t) + \nabla_z f(Z_t) f(Z_t)\mathbb{X}_{t,t+h}, \tag{6}$$

where $\mathbb{X}_{s,t}$ represents the "integral" $\int_s^t X_r \mathrm{d}X_r^\top$. Once again, we cannot uniquely define $\mathbb{X}$ from the path X itself, so instead we prescribe it. In fact, each choice of $\mathbb{X}$ satisfying Chen's relations

$$\mathbb{X}_{s,t} - \mathbb{X}_{s,u} - \mathbb{X}_{u,t} = (X_s - X_u)(X_t - X_u)^\top,$$

for any $s,u,t \geq 0$, will reveal a *different* limit for (6) as $h \to 0^+$, provided $\alpha \geq 1/3$ (for smaller α, higher-order approximations are necessary). The pair $\boldsymbol{X} = (X, \mathbb{X})$ is referred to as a *rough path*, and the limit of (6) as $h \to 0^+$ is the solution to the *rough differential equation* (RDE)

$$\mathrm{d}\boldsymbol{Z}_t = f(Z_t)\mathrm{d}\boldsymbol{X}_t. \tag{7}$$

The definition of Hölder continuity extends to the iterated integral $\mathbb{X}$ by replacing X_t and $X_t - X_s$ with $\mathbb{X}_{0,t}$ and $\mathbb{X}_{s,t}$, respectively.

It is useful to identify a calculus which satisfies the usual chain and product rules. This occurs precisely when the rough path $\boldsymbol{X}$ is *geometric*, that is,

$$\mathbb{X}_{s,t} - \mathbb{X}_{t,s} = \tfrac{1}{2}(X_t - X_s)(X_t - X_s)^\top, \quad \forall s,t \geq 0. \tag{8}$$

Every continuous and piecewise differentiable α-Hölder function X is canonically lifted to an α-Hölder geometric rough path by taking $\mathbb{X}_{s,t} = \int_s^t X_r \frac{\mathrm{d}}{\mathrm{d}r}X_r^\top \mathrm{d}r$, where the derivative is interpreted in the weak sense. In these cases, (7) equates to the ODE $\frac{\mathrm{d}}{\mathrm{d}t}Z_t = f(Z_t)\frac{\mathrm{d}}{\mathrm{d}t}X_t$.

Geometric rough paths have two key properties of interest:

I. (**Approximable**) The canonical lifts of any sequence of smooth approximations $X^{(n)}$ which converge to X as $n \to \infty$ in the α-Hölder norm, also converge in the α-Hölder rough path metric

$$\varrho_\alpha((X,\mathbb{X}),(Y,\mathbb{Y})) := \|X-Y\|_\alpha + \|\mathbb{X}-\mathbb{Y}\|_{2\alpha},$$

to a geometric rough path $(X,\mathbb{X})$. Conversely, any geometric rough path can be approximated by some sequence of smooth paths [Friz and Hairer, 2014, Proposition 2.5].

II. (**Reversible**) The reverse-time process $\tilde{Z}_t = Z_{T-t}$ of a solution Z_t to any rough differential equation (7) with Lipschitz f, itself satisfies the reversed rough differential equation $\mathrm{d}\tilde{\boldsymbol{Z}}_t = -f(T-t,\tilde{Z}_t)\mathrm{d}\boldsymbol{X}_{T-t}$ *if and only if* $\boldsymbol{X}$ is geometric.

By property I, any solution to RDEs driven by a geometric rough path can be approximated by solutions to appropriately chosen ODEs. Property II, which follows readily from the definition (8) in the limit (6), enables our analogue of the adjoint method for rough differential equations driven by a geometric rough path.

3 STOCHASTIC CONTINUOUS NORMALIZING FLOWS

Let Z_t satisfy the Itô SDE

$$\mathrm{d}Z_t = \mu_t(Z_t,\theta)\mathrm{d}t + \sigma_t(Z_t,\theta)\mathrm{d}B_t, \quad Z_0 \sim p_0(\theta), \quad (9)$$

where B_t is an m-dimensional Brownian motion, and $\mu_t : \mathbb{R}^d \to \mathbb{R}^d$, $\sigma_t : \mathbb{R}^d \to \mathbb{R}^{d\times m}$ are the drift, and diffusion coefficients, respectively. Analogous to neural ODEs, neural SDEs choose μ_t to resemble a single layer of a neural network [Tzen and Raginsky, 2019]. The dropout-inspired construction of Liu et al. [2019] suggests taking $\sigma_t \propto \mathrm{diag}(\mu_t)$. Alternatively, one can parameterize both μ_t and σ_t by multilayer neural networks.

The reliance of stochastic calculus on non-anticipating processes as well as the lack of continuity for solution maps of Itô SDEs necessitates complicated and delicate arguments for extending each piece of the continuous normalizing flow framework from §2 to SDEs. We bypass the intricacies of existing theoretical treatments of neural SDEs by an approximation argument: for a smooth approximation $\tilde{B}_t$ of Brownian motion B_t, we estimate solutions of an SDE by a random ODE involving $\tilde{B}_t$. One must take great care with such approximations. For example, geometric Brownian motion, that is, the solution to $\mathrm{d}Z_t = \sigma Z_t \mathrm{d}B_t$, has the explicit expression $Z_t = Z_0 \exp(-\frac{\sigma^2}{2}t + \sigma B_t)$, which is not well-approximated by the solution $\tilde{Z}_t = Z_0 \exp(\sigma \tilde{B}_t)$ to $\frac{\mathrm{d}}{\mathrm{d}t}\tilde{Z}_t = \sigma \tilde{Z}_t \frac{\mathrm{d}\tilde{B}_t}{\mathrm{d}t}$. Verification of this approach using traditional stochastic calculus is challenging due to the irregularity of solution maps. In order to circumvent this, we rely on rough path theory — particularly properties I and II of geometric rough paths.

In the rough path framework, one can reconstruct the Itô stochastic calculus via the rough path $\boldsymbol{B}^{\text{Itô}} = (B, \mathbb{B}^{\text{Itô}})$, where $\mathbb{B}^{\text{Itô}}_{s,t} = B_t(B_t - B_s)^\top - \frac{t-s}{2}I$. Indeed, by Friz and Hairer [2014, Theorem 9.1], letting $\boldsymbol{B}^{\text{Itô}}(\omega)$ denote a realization of the Itô Brownian motion rough path, the solution to the rough differential equation

$$\mathrm{d}\boldsymbol{Z}_t = \mu_t(Z_t,\theta)\mathrm{d}t + \sigma_t(Z_t,\theta)\mathrm{d}\boldsymbol{B}^{\text{Itô}}_t(\omega) \qquad (10)$$

is a realization of the strong solution to (9). Likewise, the Davie scheme (6) corresponds to the Milstein integrator for SDEs [Kloeden and Platen, 2013, §10.3].

Unfortunately, $\boldsymbol{B}^{\text{Itô}}(\omega)$ is not a geometric rough path, and so Theorem 2 cannot be directly applied. Instead, we shall proceed according to the following steps:

(i) Convert the Itô SDE to a Stratonovich SDE (§3.1).

(ii) Interpret the Stratonovich SDE pathwise as an RDE driven by a geometric rough path $\boldsymbol{B}^{\text{Strat}}$ (12).

(iii) Approximate the pathwise Stratonovich RDE by a random ODE (§3.2).

(iv) Train the random ODE as a continuous normalizing flow with added latent variables (§3.4).

Consequently, the RDE (10) is estimated by the ODE $\frac{\mathrm{d}Z_t(\omega)}{\mathrm{d}t} = F_\omega(Z_t(\omega),t,\theta)$ where

$$F_\omega(z,t,\theta) = \underbrace{\tilde{\mu}_t(z,\theta)}_{\text{Stratonovich drift}} + \sigma_t(z,\theta)\underbrace{\frac{\mathrm{d}B_t(\omega)}{\mathrm{d}t}}_{\text{approximation}}.$$

3.1 STRATONOVICH CALCULUS

The unique geometric rough path formed from Brownian motion $\mathbb{B}^{\text{Strat}}_{s,t} = B_t(B_t - B_s)^\top$ yields the Stratonovich calculus. A Stratonovich differential equation is commonly written in the form $\mathrm{d}Z_t = \mu_t(Z_t)\mathrm{d}t + \sigma_t(Z_t) \circ \mathrm{d}B_t$, where $\circ$ denotes Stratonovich integration: for a process Y_t adapted to the filtration generated by B_t,

$$\int_0^T Y_t \circ \mathrm{d}B_t = \lim_{|\mathcal{P}|\to 0} \sum_{k=1}^N \frac{1}{2}(Y_{t_k} + Y_{t_{k-1}})(B_{t_k} - B_{t_{k-1}}),$$

where $\mathcal{P} = \{0 = t_0 < \cdots < t_N = T\}$ is a partition with mesh size $|\mathcal{P}| = \max_k |t_k - t_{k-1}|$, and the limit is in L^2. This should be compared with Itô integration which is defined by

$$\int_0^T Y_t \,\mathrm{d}B_t = \lim_{|\mathcal{P}|\to 0} \sum_{k=1}^N Y_{t_{k-1}}(B_{t_k} - B_{t_{k-1}}).$$

Stratonovich differential equations were recognized in Li et al. [2020] to be the correct setting for extending the adjoint method to SDEs. However, the adherence to classical stochastic calculus and its reliance on adaptedness, somewhat complicates their arguments. In our setting, the advantages of Stratonovich differential equations are clear. Because Stratonovich differential equations can be arbitrarily well-approximated by random ODEs, *all* methods of training continuous normalizing flows immediately extend to them, including the adjoint method. Any Itô SDE can be converted into a Stratonovich SDE by adjusting the drift [Evans, 2012, p. 123], a fact readily seen by comparing limits of (6) with $\mathbb{B}^{\text{Itô}}$ and $\mathbb{B}^{\text{Strat}}$. The following formula is particularly amenable to implementation with automatic differentiation: the Itô SDE $\mathrm{d}Z_t = \mu_t(Z_t)\mathrm{d}t + \sigma_t(Z_t)\mathrm{d}B_t$ is equivalent to the Stratonovich SDE $\mathrm{d}Z_t = \tilde{\mu}_t(Z_t)\mathrm{d}t + \sigma_t(Z_t) \circ \mathrm{d}B_t$ provided that for each $i = 1, \dots, d$,

$$\tilde{\mu}_t^i(x) = \mu_t^i(x) - \tfrac{1}{2}\nabla_x \cdot (\sigma_t(x)\sigma_t^\top(x^*))_i, \qquad (11)$$

where x^* is an independent copy of x, and the subscript denotes the i-th row. Once again, we can make use of the trace estimator (3) to increase performance in higher dimensions. In the rough path theory, Stratonovich SDEs are interpreted pathwise according to the RDE

$$\mathrm{d}\boldsymbol{Z}_t = \tilde{\mu}_t(Z_t, \theta)\mathrm{d}t + \sigma_t(Z_t, \theta)\mathrm{d}\boldsymbol{B}_t^{\text{Strat}}(\omega), \qquad (12)$$

which is equivalent to (10).

3.2 WONG–ZAKAI APPROXIMATIONS

A random ODE $\frac{\mathrm{d}}{\mathrm{d}t}Z_t^{(n)} = \mu_t(Z_t^{(n)}) + \sigma_t(Z_t^{(n)})\frac{\mathrm{d}B_t^{(n)}}{\mathrm{d}t}$ estimating a Stratonovich SDE $\mathrm{d}Z_t = \mu_t(Z_t)\mathrm{d}t + \sigma_t(Z_t) \circ \mathrm{d}B_t$ is commonly referred to as a *Wong–Zakai approximation* [Twardowska, 1996], named after the authors of the seminal paper [Wong and Zakai, 1965], where this concept for one-dimensional Brownian motion was first introduced. We shall consider two types of Wong–Zakai approximations: a Karhunen-Loève expansion, and a piecewise linear function. These approximations are compared in Figure 1. To ease notation, our discussion will focus on estimating one-dimensional Brownian motion. The n-dimensional setting immediately follows as a vector of n independent copies of the one-dimensional case. In practice, we have found that the Karhunen-Loève expansion with $4 \leq n \leq 10$ terms works well for training, while the piecewise linear approximation is preferable for testing. However, we note that practitioners are not limited to these two choices, as our framework admits any other form of Wong–Zakai approximation.

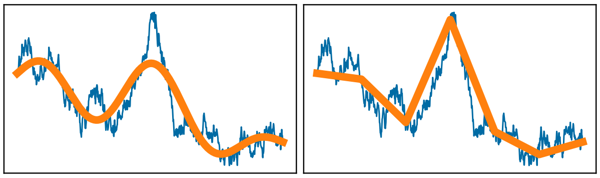

Figure 1: Karhunen-Loève (left) and piecewise linear (right) approximations of a Brownian motion sample path with $n = 6$ and $\Delta t = \frac{1}{6}$ respectively.

3.2.1 Piecewise linear

The simplest and most common approximation of Brownian motion involves exact simulation on a discrete set of times $\{0 = t_0, t_1, t_2, \dots, t_n\}$, followed by linear interpolation. More precisely, letting $\Delta t_k = t_{k+1} - t_k$, for each $k = 0, \dots, n-1$, we let

$$B_{t_{k+1}}^{(n)} = B_{t_k}^{(n)} + \sqrt{\Delta t_k}\omega_k, \quad \omega_k \sim \mathcal{N}(0,1),$$

and consider the approximation

$$B_t^{(n)} = B_{t_k}^{(n)} + \frac{t - t_k}{t_{k+1} - t_k}(B_{t_{k+1}}^{(n)} - B_{t_k}^{(n)}), \quad t \in [t_k, t_{k+1}].$$

Integrating the resulting Wong–Zakai approximation using Euler's method on the same set of time points is equivalent to performing the Euler–Maruyama method for solving the Stratonovich SDE. By Friz and Victoir [2010, Corollary 13.22], as the mesh size $\delta = \max_k \Delta t_k \to 0$, the piecewise linear approximation converges almost surely to Brownian motion in the α-Hölder norm for any $\alpha < 1/2$, as $\|B^{(n)} - B\|_\alpha \leq C_{\alpha,\eta}\, \delta^{1/4-(\alpha+\eta)/2}$ for small $\eta > 0$.

3.2.2 Karhunen-Loève expansion

For any zero-mean Gaussian process X_t on $\mathbb{R}^d$ with $t \in [0,T]$, the covariance function $K(s,t) = \mathbb{E}[X_s X_t^\top]$ is a positive-definite kernel. If K is also continuous, Mercer's theorem [Minh et al., 2006] guarantees the existence of an orthonormal basis on $L^2([0,T], \mathbb{R}^d)$ of eigenfunctions $\{e_k\}_{k=1}^\infty$ with corresponding positive eigenvalues $\{\lambda_k\}_{k=1}^\infty$ such that $K(s,t) = \sum_{j=1}^\infty \lambda_j e_j(s) e_j(t)$. The process X_t can be expanded in terms of these eigenfunctions as

$$X_t = \sum_{k=1}^\infty \sqrt{\lambda_k}\omega_k e_k(t), \quad \omega_k \sim \mathcal{N}(0,1),$$

where each ω_k is independent. This is called the *Karhunen-Loève expansion* of X. Truncating the series after n terms yields the n-th order Karhunen-Loève approximation, and has the smallest mean squared error over all expansions with n orthogonal basis elements [Brown, 1960]. Recalling that we are primarily interested in the endpoints of the solution, instead of expanding Brownian motion itself, we consider an approximation $B_t^{(n)}$ derived from the Karhunen-Loève expansion of the Brownian bridge $B_t - B_T \frac{t}{T}$:

$$B_t^{(n)} = \omega_0 \frac{t}{\sqrt{T}} + \sum_{k=1}^{n-1} \omega_k \frac{\sqrt{2T}\sin(k\pi t/T)}{k\pi}, \quad n = 2, 3, \dots.$$

Using this approximation ensures that the terminal density for SDEs with constant drift and diffusion coefficients is computed exactly. By Friz and Victoir [2010, Theorem 15.51], $(B_t^{(n)})_{t\in[0,T]}$ converges almost surely as $n \to \infty$ to Brownian motion in the α-Hölder norm for any $\alpha < 1/2$. Furthermore, since $B_t^{(n)}$ is smooth, Wong–Zakai approximations involving $\frac{\mathrm{d}B_t^{(n)}}{\mathrm{d}t}$ may be readily solved using adaptive ODE solvers.

3.3 MAIN RESULT

Using Wong–Zakai approximations, a Stratonovich SDE can be uniformly approximated in Hölder norm by random ODEs. In Theorem 2, we show that the log-densities and loss function gradients for these random ODEs also converge appropriately. More generally, geometric rough paths with random initial conditions (including the Stratonovich paths (12)) can be approximately trained as random ODEs.

Theorem 2. *Let $\boldsymbol{X} = (X, \mathbb{X})$ be an α-Hölder geometric rough path, and $\{X^{(n)}\}_{n=1}^{\infty}$ a sequence of piecewise smooth functions on $[0,T]$ that approximate X under the β-Hölder norm for $\beta \in (\frac{1}{3}, \frac{1}{2})$, that is, $\|X^{(n)} - X\|_\beta \to 0$ as $n \to \infty$. Let $\boldsymbol{Z}, Z^1, Z^2, \dots$ be solutions to the differential equations*

$$\mathrm{d}\boldsymbol{Z}_t = f(Z_t, t, \theta)\mathrm{d}\boldsymbol{X}_t, \qquad Z_0 \sim p_0, \tag{13a}$$

$$\frac{\mathrm{d}Z_t^{(n)}}{\mathrm{d}t} = f(Z_t^{(n)}, t, \theta)\frac{\mathrm{d}X_t^{(n)}}{\mathrm{d}t} \qquad Z_0^{(n)} = Z_0. \tag{13b}$$

Here $f : \mathbb{R}^d \times [0,T] \times \mathbb{R}^m \to \mathbb{R}$ is a four times continuously differentiable bounded function, and p_0 is a strictly positive continuous density on $\mathbb{R}^d$. Denote by $p_t^{(n)}$ the probability density of $Z_t^{(n)}$ at time t, given by (2). Then the distribution of Z_t is absolutely continuous with respect to Lebesgue measure with corresponding continuous density p_t and satisfies:

1. *For any $x \in \mathbb{R}^d$, $\sup_{t\in[0,T]} |\log p_t^{(n)}(x) - \log p_t(x)| \to 0$ as $n \to \infty$.*
2. *The path $t \mapsto \log p_t(Z_t)$ is the unique solution to the rough differential equation*
$$\mathrm{d}\log p_t(Z_t) = -\nabla_z \cdot (f(Z_t, t, \theta)\mathrm{d}\boldsymbol{X}_t). \tag{14}$$
3. *For any smooth loss function $L : \mathbb{R}^{d+1} \to \mathbb{R}$ and $t \geq 0$, as $n \to \infty$,*
$$\nabla_\theta L(Z_t^{(n)}, \log p_t^{(n)}(Z_t^{(n)})) \to \nabla_\theta L(Z_t, \log p_t(Z_t)). \tag{15}$$
Furthermore, if $\log p_0$ and L are also Lipschitz continuous, the limit (15) converges at rate $\mathcal{O}(\|X^{(n)} - X\|_\beta)$ as $n \to \infty$.

Proof of Theorem 2. We begin with some analytic background. The existence of the Lyons lift map [Friz and Victoir, 2010, Theorem 9.5] asserts that each $X^{(n)}$ can be lifted canonically to a rough path $\boldsymbol{X}^{(n)}$ such that $\rho_\beta(\boldsymbol{X}^{(n)}, \boldsymbol{X}) \leq C_\beta \|X^{(n)} - X\|_\beta \to 0$ as $n \to \infty$, for some $C_\beta > 0$. For an arbitrary rough path $\boldsymbol{Y}$, we let $\Phi_t(\boldsymbol{Y}, \xi)$ and $\Psi_t(\boldsymbol{Y}, \ell)$ denote the solution maps for the rough differential equations

$$\begin{aligned} \mathrm{d}\boldsymbol{Z}_t &= f(Z_t, t, \theta)\mathrm{d}\boldsymbol{Y}_t, & Z_0 &= \xi, \\ \mathrm{d}\boldsymbol{L}_t &= -\nabla_z \cdot f(Z_t, t, \theta)\mathrm{d}\boldsymbol{Y}_t, & L_0 &= \ell, \end{aligned}$$

respectively.

Existence of Densities. By Friz and Hairer [2014, Theorem 8.10], $\Phi_t(\boldsymbol{Y}, \cdot)$ is a $\mathcal{C}^1$-diffeomorphism, and hence, for $Z_0 \sim p_0(\theta)$ and any $t \in [0,T]$, $Z_t = \Phi_t(\boldsymbol{X}, Z_0)$ is an absolutely continuous random variable, whose corresponding density we denote by p_t. In fact, denoting by $\Phi_{-t}(\boldsymbol{Y}, \cdot)$ the inverse of $\Phi_t(\boldsymbol{Y}, \cdot)$, via a changes of variables

$$p_t^{(n)}(x) = p_0(\Phi_{-t}(\boldsymbol{X}^{(n)}, x)) \left|\det \frac{\partial \Phi_{-t}(\boldsymbol{X}^{(n)}, x)}{\partial x}\right|, \tag{17}$$

$$p_t(x) = p_0(\Phi_{-t}(\boldsymbol{X}, x)) \left|\det \frac{\partial \Phi_{-t}(\boldsymbol{X}, x)}{\partial x}\right|, \tag{18}$$

and so both $p_t^{(n)}$ and p_t are continuous.

Four Essential Results. Furthermore, by Friz and Hairer [2014, Theorem 8.5], for any $\frac{1}{3} < \gamma < \beta$, there exist constants $C_\gamma^\Phi, C_\gamma^\Psi$ such that for any β-Hölder continuous rough paths $\boldsymbol{X}, \boldsymbol{Y}$ and $\xi, \tilde{\xi} \in \mathbb{R}^d$, $\ell, \tilde{\ell} \in \mathbb{R}_+$,

$$\|\Phi(\boldsymbol{X}, \xi) - \Phi(\boldsymbol{Y}, \tilde{\xi})\|_\gamma \leq C_\gamma^\Phi(\|\xi - \tilde{\xi}\| + \varrho_\beta(\boldsymbol{X}, \boldsymbol{Y})) \tag{19}$$

$$\|\Psi(\boldsymbol{X}, \ell) - \Psi(\boldsymbol{Y}, \tilde{\ell})\|_\gamma \leq C_\gamma^\Psi(|\ell - \tilde{\ell}| + \varrho_\beta(\boldsymbol{X}, \boldsymbol{Y})). \tag{20}$$

This lets us deduce the following facts as $n \to \infty$, for any $t \in [0,T]$ and $x \in \mathbb{R}^d$:

(i) $\|Z^{(n)} - Z\|_\gamma \to 0$ by (19);

(ii) using (i) and continuity of p_t, $\log p_t(Z_t^{(n)}) \to \log p_t(Z_t)$;

(iii) continuity guarantees that (19) implies $p_t^{(n)}(x) \to p_t(x)$ via (17), (18), and estimates from and Friz and Hairer [2014, Theorem 8.10];

(iv) combining (ii) and (iii), $\log p_t^{(n)}(Z_t^{(n)}) \to \log p_t(Z_t)$.

Evolution of $\log$**-densities (Theorem 2.2).** Since $\Psi_t(\boldsymbol{X}^{(n)}, \log p_0(Z_0)) = \log p_t^{(n)}(Z_t^{(n)})$ by Theorem 1, (iv) and (20) give $\log p_t(Z_t) = \Psi(\boldsymbol{X}, \log p_0(Z_0))$, whence (14).

Uniform Convergence (Theorem 2.1). Let $x \in \mathbb{R}^d$ be arbitrary. To show that $\log p_t^{(n)}(x)$ converges uniformly in t, observe that

$$\log p_t(x) = \Psi(\boldsymbol{X}, \log p_0(\Phi_{-t}(\boldsymbol{X}, x))), \tag{21}$$

and similarly for $\log p_t^{(n)}(x)$. Together with property II of geometric rough paths, inequality (19) with $\boldsymbol{Y} \equiv \boldsymbol{0}$ reveals that $\Phi_{-t}(\boldsymbol{X}^{(n)}, x)$ and $\Phi_{-t}(\boldsymbol{X}, x)$ are uniformly bounded in $t \in [0,T]$. Since $\log p_0$ is continuous, $\log p_0(\Phi_{-t}(\boldsymbol{X}^{(n)}, x))$ converges to $\log p_0(\Phi_{-t}(\boldsymbol{X}, x))$ uniformly in $t \in [0,T]$. Applying (20) to (21),

$$\begin{aligned} &\sup_{t\in[0,T]} |\log p_t^{(n)}(x) - \log p_t(x)| \leq C_\gamma^\Psi(\varrho_\beta(\boldsymbol{X}^{(n)}, \boldsymbol{X}) \\ &+ \sup_{t\in[0,T]} |\log p_0(\Phi_{-t}(\boldsymbol{X}^{(n)}, x)) - \log p_0(\Phi_{-t}(\boldsymbol{X}, x))|) \to 0. \end{aligned} \tag{22}$$

Adjoint Method (Theorem 2.3). To show (15), by Friz and Hairer [2014, Proposition 5.6], we can write (θ, L) as the solution to the rough differential equation

$$\mathrm{d}\theta = 0 \tag{23a}$$

$$\begin{aligned} \mathrm{d}L(Z_t, \log p_t(Z_t)) &= \nabla_z L(Z_t, \log p_t(Z_t)) \cdot \mathrm{d}\boldsymbol{Z}_t \\ &\quad + \nabla_\ell L(Z_t, \log p_t(Z_t)) \mathrm{d} \log p_t(Z_t) \end{aligned} \tag{23b}$$

and similarly for $Z_t^{(n)}$ and $\log p_t^{(n)}(Z_t^{(n)})$, where ∇_z and ∇_ℓ denote the gradients with respect to Z_t and $\log p_t(Z_t)$, respectively. The derivative of L with respect to θ is a derivative of (23) with respect to its initial condition, and hence (15) follows from Friz and Hairer [2014, Theorem 8.10]. The same result, together with (19), (20), and (22), completes the theorem. □

Stochastic Adjoint Method. Theorem 2.3 is of particular value in practice. For $\mathbb{X} = (B, \mathbb{B}^{\text{Strat}})$, (13a) becomes an SDE, and (13b) is its Wong–Zakai approximation. Since Z_t^n is a random ODE, conditioned on the Brownian motion approximation, it can be trained using the adjoint method (4). The claim (15) implies that training Z_t^n to a smooth loss L in this way properly approximates training Z_t. An SDE can be trained as a latent ODE model; no further theory or special methodology is required. Indeed, by taking the limit $\nabla L(Z_t^n) \to \nabla L(Z_t)$ and integrating the ODE (4) using an Euler scheme, the stochastic adjoint method of Li et al. [2020] is recovered. However, with Theorem 2, one may use *any* valid approximation for Brownian motion, together with *any* ODE integrator, and also involve the density in the loss function.

Rates of Convergence. We would also like to briefly comment on the rate of convergence in Theorem 2.3, which suggests that the order of the error in the associated stochastic adjoint method is on par with that of the Wong–Zakai approximation. This presents one notable advantage over the approach in Li et al. [2020]. The use of Wong–Zakai approximations allows one to "split" the total approximation error into the sum of the integration error in solving the ODE, and the approximation error in the Brownian motion. Combining higher-order numerical integrators with a large number of terms in a Karhunen-Loève expansion (for instance), one can theoretically achieve arbitrarily fast rates of convergence in computing the gradients, without involving higher-order (expensive) approximations to the Lévy area for the SDE (9).

3.4 DENSITY ESTIMATION

By a conditioning argument, any random ODE, such as a Wong-Zakai approximation, may be treated as a continuous normalizing flow. Let Z_t be the solution to a random ODE of the form

$$\frac{\mathrm{d}}{\mathrm{d}t} Z_t = f(Z_t, \omega, t, \theta), \tag{24}$$

where $\omega = (\omega_1, \dots, \omega_n) \sim q(\omega)$ is a random vector independent of Z_t, t, and θ. The reduction of a random ODE to this form is in keeping with the reparameterization trick [Xu et al., 2019]. In particular, for the piecewise linear and Karhunen-Loève approximations, each $\omega_i \sim \mathcal{N}(0,1)$. After conditioning on ω, Theorem 1 applied to (24) provides a means of computing $\log p_t(Z_t|\omega)$, after sampling $Z_0 \sim p_0$. The density $p_t(Z_t)$ can be computed using a naive Monte Carlo estimator

$$p_t^\theta(Z_t) = \int p_t^\theta(Z_t|\omega) q(\omega) \mathrm{d}\omega \approx \frac{1}{N} \sum_{i=1}^n p_t^\theta(Z_t|\omega_i), \tag{25}$$

where the dependence on θ has been made explicit, and can be optimized over using the adjoint method. Analogously to Chen et al. [2018] and Grathwohl et al. [2018], the density of data $\boldsymbol{x}$ may be estimated along a single sample path $B_t(\omega)$ (denoted $p(\boldsymbol{x}|\omega)$) in the following way: letting $\Delta \log p_t^\omega = \log p_t(Z_t|\omega) - \log p(\boldsymbol{x}|\omega)$, we see that $\Delta \log p_t^\omega$ also satisfies (2). By solving (24) and the corresponding (2) in reverse time from the initial conditions $Z_T = \boldsymbol{x}$ and $\Delta \log p_T^\omega = 0$, we obtain Z_0 and $\Delta \log p_0^\omega$, and compute $\log p(\boldsymbol{x}|\omega) = \log p_0(Z_0) - \Delta \log p_0^\omega$. This is shown in Algorithm 1, which depends on an ODE solver ODESOLVE, and yields a density estimation procedure for SCNF when paired with (25). Note that by comparison to Grathwohl et al. [2018, Algorithm 1] which encompasses steps 6–12 of our Algorithm 1, we see much of the density estimation procedure can be accomplished using an existing continuous normalizing flow implementation. In variational settings where $\log p_T(Z_T)$ is required, the same procedure applies, where $\mathbf{x}$ becomes Z_T and is generated by the SDE as well.

A number of techniques exist for debiasing the logarithm of (25) — see Rhee and Glynn [2015] and Rischard et al. [2018], for example. Alternatively, we lie in the setting of

Algorithm 1 Density estimation; single path

```
Input: drift function $\mu$, diffusion function $\sigma$, an initial density $p_0$, final time $T$, minibatch of samples $\boldsymbol{x}$, sample path $\tilde{B}_t(\omega)$ of Brownian motion approximation.
Output: an estimate of $\log p(\boldsymbol{x}|\omega)$
Generate $\epsilon = (\epsilon_1, \dots, \epsilon_d)$ for (3).
function ODEFUNC$(z,t)$
    Compute $\tilde{\mu}(z,t)$ via (11).        ▷ Itô correction
    return $\tilde{\mu}(z,t) + \sigma(z,t)\frac{\mathrm{d}\tilde{B}_t(\omega)}{\mathrm{d}t}$.
end function
function AUG$((z, \log p_t), t)$
    $f_t \leftarrow$ ODEFUNC$(z, t, \omega)$
    $J_t \leftarrow -\nabla_z(\epsilon \cdot f_t) \cdot \epsilon$   ▷ Trace estimator (3); $n = 1$
    return $(f_t, J_t)$
end function
$(z, \Delta \log p_t^\omega) \leftarrow$ ODESOLVE(AUG,$(x,0)$,0,$T$)
return $\log p_0(z) - \Delta \log p_t^\omega$
```

semi-implicit variational inference seen in Yin and Zhou [2018] and Titsias and Ruiz [2019], and those techniques directly extend to our case as well. Naturally, it would be easiest to instead optimize the upper bound

$$-\log p_t^\theta(Z_t) \leq -\mathbb{E}_\omega \log p_t^\theta(Z_t|\omega).$$

Observing that

$$\log p_t^\theta(Z_t) - D_{\mathrm{KL}}(q \| p^\theta_{\omega|Z_t}) = \mathbb{E}_\omega \log p_t^\theta(Z_t|\omega), \quad (26)$$

minimizing $-\mathbb{E}_\omega \log p_t^\theta(Z_t|\omega)$ maximizes the true log-likelihood regularized by the KL-divergence between the prior and posterior distributions for ω, which reduces the effect of noise on the model. At the same time, parameterizations of the diffusion coefficient that allow $\|\sigma\|$ to shrink to zero will often do so, resulting in overfitting, and should be avoided to remain distinct from a CNF.

4 NUMERICAL EXPERIMENTS

While our contributions are predominantly theoretical, in this section, we conduct simple numerical investigations into two properties of SCNF not covered by the theory: (1) whether trained time-homogeneous SDEs trained as SCNFs are stable; and (2) the regularization effect of SCNFs over CNFs. In both cases, we train a SCNF (9) — using Algorithm 1 with the upper bound (26) — to data generated from a specified target density. For our drift function, we adopt the same architecture used in the toy examples of Grathwohl et al. [2018]; a four-layer fully-connected neural network with 64 hidden units in each layer. Dependence on time is removed to ensure a time-homogeneous, and hence, potentially stable SDE after training. The Dormand–Prince integrator is used throughout, and all networks were trained using Adagrad [Duchi et al., 2011], with $p_0 \sim \mathcal{N}(0, I)$ and 1000 samples.

4.1 EFFICACY AS A SEQUENTIAL SAMPLER

In our first example, we investigate stability of SDEs trained using the SCNF framework. Data is generated from the banana-shaped density

$$p(x,y) \propto \exp(-\tfrac{1}{2}(x^2 + \tfrac{1}{2}(x^2+y)^2)).$$

Two choices of diffusion coefficient are considered: the first, where $\sigma = I$, yields a neural SDE that can be trained using the techniques of Li et al. [2020]. For the second, we choose

$$\sigma(x) = \lambda \begin{pmatrix} 1 & \sigma_1(x) \\ \sigma_2(x) & 1 \end{pmatrix}, \quad (27)$$

with $\lambda = 1$, and parameterize (σ_1, σ_2) by a two-layered neural network with 64 hidden units. This SDE can only be trained using our method. After training, to emulate the application of these SDEs as approximate samplers, a single sample path with 10,000 steps was simulated for each model using the Euler–Maruyama method. The resulting paths are compared in Figure 2. From data alone, both models appear to have constructed stable processes, reminiscent of an ergodic process with the target density as its invariant measure. The addition of a trainable diffusion coefficient led to improved adaptation of the sampler to the underlying curvature. This is perhaps unsurprising in light of the improved performance attained by Riemannian MCMC methods [Girolami and Calderhead, 2011, van der Heide et al., 2021].

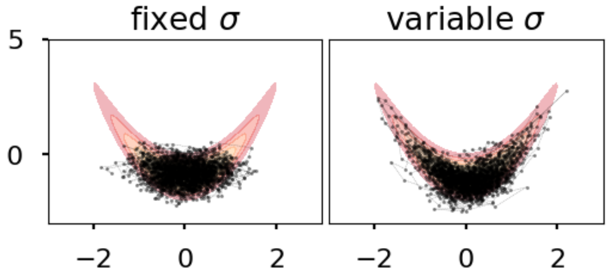

Figure 2: Sample paths from SDEs trained as stochastic continuous normalizing flows to a banana-shaped density.

4.2 VISUALIZING REGULARIZATION

As discussed in Liu et al. [2019], the stochastic noise injection in SDEs is a natural form of regularization, that can potentially improve robustness to noisy or adversarial data. We visualize this effect by considering the same stochastic continuous normalizing flows treated in §4.1 with diffusion coefficient (27), and adjusting the parameter $\lambda > 0$. Our

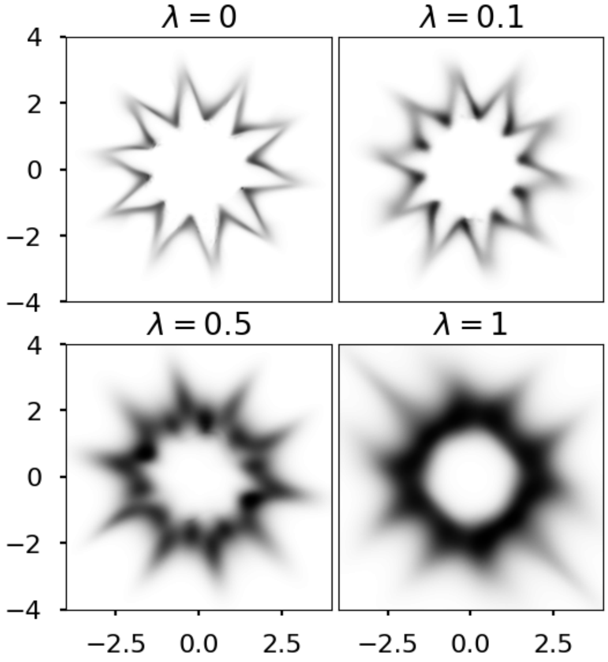

Figure 3: Density plots of stochastic continuous normalizing flows trained to a star-shaped density with varying diffusion coefficients.

data is generated in polar coordinates from a ten-pointed star-shaped density by

$$\theta \sim \mathrm{Unif}(-\pi, \pi), \quad r|\theta \sim \mathcal{N}\left(\tfrac{2}{\sqrt{1+\frac{1}{2}\sin(10\theta)}}, \tfrac{9}{400}\right).$$

In Figure 3, we plot the densities for $\lambda \in \{0, \frac{1}{10}, \frac{1}{2}, 1\}$ computed using Algorithm 1, noting that the $\lambda = 0$ case corresponds to a continuous normalizing flow. Increasing λ reveals generative models with expectedly higher variance, but with improved capacity to smooth out minor (potentially, unwanted) details.

5 CONCLUSION

We have developed a general theoretical framework for extending the training processes of ODE-based models — including the continuous normalizing flows framework itself — to analogous SDE-based models. Constructed from rough path theory, our framework enables practitioners of neural ODEs to apply their existing implementation for training neural SDEs. This is advantageous, as neural SDEs have been suggested to be more robust than neural ODEs in high-dimensional real-world examples [Liu et al., 2019, Li et al., 2020]. However, several practical challenges remain. In particular, we have found the density estimators (25) to have relatively high variance. This could be overcome using control variates, for example. Additionally, the usual challenges of semi-implicit variational inference remain, with the troublesome tendency for (26) to overfit. Addressing these features is important for effective implementations of SCNF in real-world settings, and is the subject of future work.

Acknowledgements

This work has been supported by the Australian Research Council Centre of Excellence for Mathematical & Statistical Frontiers (ACEMS), under grant number CE140100049. We would also like to acknowledge DARPA, NSF, and ONR for providing partial support.

References

Alexandros Beskos, Omiros Papaspiliopoulos, Gareth O. Roberts, and Paul Fearnhead. Exact and computationally efficient likelihood-based estimation for discretely observed diffusion processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(3): 333–382, 2006.

John L. Brown. Mean square truncation error in series expansions of random functions. *Journal of the Society for Industrial and Applied Mathematics*, 8(1):28–32, 1960.

Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6571–6583, 2018.

Alexander M. Davie. Differential equations driven by rough paths: an approach via discrete approximation. *Applied Mathematics Research eXpress*, 2008, 2008.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12: 2121–2159, 2011.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems*, 2019.

Lawrence C. Evans. *An introduction to stochastic differential equations*, volume 82. American Mathematical Society, 2012.

Peter Friz and Martin Hairer. *A Course on Rough Paths*. Springer International Publishing, 2014.

Peter K. Friz and Nicolas B. Victoir. *Multidimensional stochastic processes as rough paths: theory and applications*, volume 120. Cambridge University Press, 2010.

Amir Gholami, Kurt Keutzer, and George Biros. ANODE: Unconditionally Accurate Memory-Efficient Gradients for Neural ODEs. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*, 2019.

Mark Girolami and Ben Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Will Grathwohl, Ricky T. Q. Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.

Ernst Hairer and Gerhard Wanner. *Solving Ordinary Differential Equations II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag Berlin Heidelberg, 1996.

Liam Hodgkinson, Robert Salomone, and Fred Roosta. Implicit Langevin algorithms for sampling from log-concave densities. *arXiv preprint arXiv:1903.12322*, 2019.

Stan Hurn, Joseph Jeisman, and Kenneth A. Lindsay. Seeing the wood for the trees: A critical evaluation of methods to estimate the parameters of stochastic differential equations. *Journal of Financial Econometrics*, 5(3):390–455, 2007.

Junteng Jia and Austin R. Benson. Neural Jump Stochastic Differential Equations. In *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.

Peter E. Kloeden and Eckhard Platen. *Numerical solution of stochastic differential equations*, volume 23. Springer Science & Business Media, 2013.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. *arXiv preprint arXiv:2001.01328*, 2020.

Xuanqing Liu, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. Neural SDE: Stabilizing Neural ODE Networks with Stochastic Noise. *arXiv preprint arXiv:1906.02355*, 2019.

Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. *arXiv preprint arXiv:1710.10121*, 2017.

Terry J. Lyons. Differential equations driven by rough signals. *Revista Matemática Iberoamericana*, 14(2):215–310, 1998.

Yi-An Ma, Tianqi Chen, and Emily Fox. A complete recipe for stochastic gradient MCMC. In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

Ha Quang Minh, Partha Niyogi, and Yuan Yao. Mercer's theorem, feature maps, and smoothing. In *International Conference on Computational Learning Theory*, pages 154–168. Springer, 2006.

Stefano Peluchetti and Stefano Favaro. Infinitely deep neural networks as diffusion processes. *arXiv preprint arXiv:1905.11065*, 2019.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Chang-han Rhee and Peter W. Glynn. Unbiased estimation with square root convergence for SDE models. *Operations Research*, 63(5):1026–1043, 2015.

Maxime Rischard, Pierre E. Jacob, and Natesh Pillai. Unbiased estimation of log normalizing constants with applications to Bayesian cross-validation. *arXiv preprint arXiv:1810.01382*, 2018.

Fred Roosta and Uri Ascher. Improved bounds on sample size for implicit matrix trace estimators. *Foundations of Computational Mathematics*, 15(5):1187–1212, 2015.

Thomas Ryder, Andrew Golightly, Stephen McGough, and Dennis Prangle. Black-box variational inference for stochastic differential equations. *arXiv preprint arXiv:1802.03335*, 2018.

Michalis K. Titsias and Francisco J. R. Ruiz. Unbiased implicit variational inference. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019.

Krystyna Twardowska. Wong-Zakai approximations for stochastic differential equations. *Acta Applicandae Mathematica*, 43(3):317–359, 1996.

Belinda Tzen and Maxim Raginsky. Neural Stochastic Differential Equations: Deep Latent Gaussian Models in the Diffusion Limit. *arXiv preprint arXiv:1905.09883*, 2019.

Chris van der Heide, Liam Hodgkinson, Fred Roosta, and Dirk Kroese. Shadow Manifold Hamiltonian Monte Carlo. In Arindam Banerjee and Kenji Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 1477–1485. PMLR, 13–15 Apr 2021.

Eugene Wong and Moshe Zakai. On the convergence of ordinary integrals to stochastic integrals. *The Annals of Mathematical Statistics*, 36(5):1560–1564, 1965.

Ming Xu, Matias Quiroz, Robert Kohn, and Scott A. Sisson. Variance reduction properties of the reparameterization trick. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2711–2720, 2019.

Mingzhang Yin and Mingyuan Zhou. Semi-implicit variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Laurence C. Young. An inequality of the Hölder type, connected with Stieltjes integration. *Acta Mathematica*, 67: 251–282, 1936.

Tianjun Zhang, Zhewei Yao, Amir Gholami, Kurt Keutzer, Joseph Gonzalez, George Biros, and Michael W. Mahoney. ANODEV2: A Coupled Neural ODE Evolution Framework. *Advances in Neural Information Processing Systems*, 2019.