Differentiable Bayesian inference for SDEs using a pathwise series expansion of Brownian motion

Anonymous Author(s)
Affiliation
Address
email

Abstract

By invoking a pathwise series expansion of Brownian motion, we propose to approximate a stochastic differential equation (SDE) with an ordinary differential equation (ODE). This allows us to reformulate Bayesian inference for a SDE as the parameter estimation task for an ODE. Unlike a nonlinear SDE, the likelihood for an ODE model is tractable and its gradient can be obtained using adjoint sensitivity analysis. This reformulation allows us to use an efficient sampler, such as NUTS, that rely on the gradient of the log posterior. Applying the reparameterisation trick, variational inference can also be used for the same estimation task. We illustrate the proposed method on two biological SDE models. We obtain similar parameter estimates when compared to data augmentation techniques. Finally, we introduce a Bayesian neural SDE model to test out approach for a system identification task. We obtain competitive performance on a benchmark motion capture dataset when compared to latent ODE models.

1 Introduction

The solution to a stochastic differential equation – a diffusion process – that evolves randomly over time is a flexible and useful tool in modelling stochastic systems. Naturally, SDEs are widely used in modelling continuously evolving systems of real valued quantities subjected to noise or stochastic fluctuations.

The task of estimating the parameters of a SDE observed at discrete times is highly challenging. One has to deal with the estimation of a high dimensional latent diffusion in addition to the governing parameters. Moreover, the exact transition densities given by the forward Kolmogorov equation, required to calculate the likelihood, are intractable for a nonlinear SDE. In a Bayesian paradigm the estimation task requires repeated sampling of the parameters as well as the latent diffusion, wherein the proposals for the latter rely on the Euler-Maruyama discretisation of the SDE. The primary challenge here is the non-trivial sampling of the latent diffusion path. Additionally any sampling scheme has to also deal with the strong correlation between the diffusion path and the model parameters. Slow convergence of MCMC incurs a high computational cost due to the iterative nature of sampling a discretised diffusion path.

With the widespread availability of automatic differentiation (AD) techniques, Bayesian inference is increasingly carried out using Markov chain Monte Carlo (MCMC) algorithms that traverse the parameter space based on gradient of the target density. Such algorithms have proposal generating mechanism that can rapidly explore the parameter space when compared to traditional random-walk MCMC. Moreover, optimisation based alternatives to MCMC such as black-box variational inference has the potential to further expedite the inference process. AD has made it possible to apply such algorithms to many complex models with a differentiable target density. Many probabilistic

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

programming platforms, that rely on AD, includes such algorithms and have vastly automated the entire process of Bayesian inference.

Application of the aforementioned efficient algorithms to a SDE is highly non-trivial, as this requires the extension of AD to a SDE. Firstly, existing approaches (Gobet and Munos, 2005; Yang and Kushner, 1991; Giles and Glasserman, 2006) to differentiate a diffusion process generally scales poorly with the parameter dimension, or with memory. Additionally, the intractable likelihood of a SDE is often approximated using non-differentiable simulation methods, which in turn renders any attempt to use the gradient of the target density futile.

In case of an ODE, in contrast to a SDE, the likelihood is tractable and there exists an efficient numerical sensitivity analysis technique that lends itself to AD (Chen et al., 2018). Thus, we propose to approximate a SDE by an equivalent ODE using a pathwise truncated series expansion of Brownian motion (Lyons et al., 2012, 2014). The resulting ODE contains the same SDE parameters in addition to extra parameters that are the expansion coefficients. We estimate all these parameters jointly where the marginal density of the model parameters is the desired quantity of interest. We carry out this estimation task using the No-U-Turn sampler as well as using black-box variational inference, both inherently capable of handling the inflated parameter dimension efficiently.

2 Related work

Series expansion of Brownian motion for inference, in conjunction with a Gibbs sampling scheme, was first introduced in Lyons et al. (2012) for systems with additive noise. Lyons et al. (2014) applied series expansion for filtering in nonlinear state-space models. By marrying series expansion with adjoint sensitivity analysis, our approach can be applied to both additive and multiplicative noise models, the latter is prevalent in biology. Moreover, our approach is targeted towards differentiable inference, which accommodates faster and more efficient inference algorithms.

AD for SDEs was introduced in Li et al. (2020) by deriving an adjoint sensitivity analysis technique for Stratonovich SDEs, thus enabling more efficient AD for a SDE, when compared to approaches in Gobet and Munos (2005); Yang and Kushner (1991); Giles and Glasserman (2006). We instead focused on approximating a SDE with an ODE, the latter being fully integrated with AD (Chen et al., 2018; Ghosh et al., 2021). Additionally, working with an ODE we can apply familiar tools for inference and model building. Unlike Chen et al. (2018), our approach can handle both classical inference as well as system identification (say using a Neural SDE).

Variational inference for a SDE, exploiting AD, was recently proposed in Ryder et al. (2018). Li et al. (2020) used the path integral approach for this purpose. In contrast to these approaches our method do not require the usage of a neural network for inference. Thus, our approach is more conducive for mechanistic modelling, especially the calibration task.

ODE approximation for an SDE can also be derived using a Gaussian approximation of the transition density (Golightly and Gillespie, 2013; Fearnhead et al., 2014). Our approach rely on a pathwise approximation of the diffusion. So, in our approach we can accommodate non-Gaussian transition densities.

3 Background

We begin by first introducing the Bayesian inference framework for a diffusion process described by a SDE. Consider a K-dimensional diffusion process that satisfies the following SDE:

$$d\boldsymbol{X}_t = \boldsymbol{a}(\boldsymbol{X}_t, \boldsymbol{\theta})dt + \sqrt{\boldsymbol{B}(\boldsymbol{X}_t, \boldsymbol{\theta})}d\boldsymbol{W}_t, \quad \boldsymbol{X}_0 = \boldsymbol{x}_0, \tag{1}$$

where $\boldsymbol{X}_t$ denotes the value of the process at time t, $\boldsymbol{a}$ is a K-dimensional *drift* vector, $\boldsymbol{B}$ a $K \times K$ diffusion matrix and the driving noise $\boldsymbol{W}_t$ is a K-dimensional Brownian motion which is treated in the Itô sense. Both the drift and diffusion matrix depends on an unknown parameter vector $\boldsymbol{\theta} \in \mathbb{R}^D$. If the initial condition $\boldsymbol{x}_0$ is unknown then this becomes an element of $\boldsymbol{\theta}$.

We assume that $\boldsymbol{a}(\cdot)$ and $\boldsymbol{B}(\cdot)$ are sufficiently regular functions such that Equation (1) has a weak non-explosive solution (Oksendal, 2013).

Consider a set of noisy experimental observations $\boldsymbol{y} \in \mathbb{R}^{M \times K}$ observed at M experimental time points, $\{t_i\}_{i=1}^M$, for the K states. Within the Bayesian inferential paradigm we want to place a

prior distribution on the unknown parameters $p(\boldsymbol{\theta})$ and intend to obtain the corresponding posterior distribution

$$p(\boldsymbol{X}, \boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})p(\boldsymbol{X})p(\boldsymbol{\theta}) \tag{2}$$

of the latent path $\boldsymbol{X}$ and the unknown parameters $\boldsymbol{\theta}$, where $p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ is the likelihood and $p(\boldsymbol{X})$ is the distribution of the diffusion $\boldsymbol{X}$ defined from the SDE given by Equation (1). For a nonlinear SDE, $p(\boldsymbol{X})$ is intractable. To work around this intractability an Euler-Maruyama discretisation is generally used, yielding a Gaussian approximation to the transition density. Inference proceeds by generating samples of $\boldsymbol{X}$ and $\boldsymbol{\theta}$.

Within an MCMC scheme, samples of $\boldsymbol{X}$ and $\boldsymbol{\theta}$ can be drawn by using a Gibbs sampling (Golightly and Wilkinson, 2008) scheme. As an alternative to such Gibbs proposal mechanism both $\boldsymbol{X}$ and $\boldsymbol{\theta}$ can be jointly updated using a particle MCMC scheme wherein a particle filter is used to draw the diffusion path and evaluate an unbiased estimate of the likelihood. Particle MCMC methods, despite their higher computational cost, produce a faster mixing Markov chain and thus have become the state-of-the-art for inference in SDEs. Nonetheless, both these methods are inherently non-differentiable and rely crucially on a random-walk proposal for updating $\boldsymbol{\theta}$.

4 Pathwise series approximation

Within an interval $[0, T]$ a standard Brownian motion has the following series expansion (Lyons et al., 2012, 2014):

$$W_t = \sum_{i=1}^{\infty} \Big(\int_0^T \phi_i(s) dW(s) \Big) \int_0^t \phi_i(u) dW(u), \tag{3}$$

where $\{\phi_i\}_{i\geq 1}$ is an orthonormal basis of $L^2[0, T]$. For example this can be the Karhunen–Loève (KL) expansion of a Gaussian process given by

$$\phi_i(t) = (2/T)^{1/2} \cos\{(2i-1)\pi t/2T\}. \tag{4}$$

We will use the shorthand $Z_i = \int_0^T \phi_i(s) dW(s)$. Since the basis functions $\{\phi_i\}$ are deterministic and orthonormal, it follows from standard results of Itô calculus that $Z_i \sim \mathcal{N}(0, 1)$.

The infinite series in Equation 3 can be truncated after N terms to obtain an approximation of $\hat{W}_t$ Brownian motion. Taking derivative with respect to time we obtain the following approximation to the differential of Brownian motion given by

$$\frac{d\hat{W}}{dt} = \sum_{i=1}^{N} Z_i \phi_i(t). \tag{5}$$

Consider now a scalar SDE driven by the approximate Brownian motion:

$$dX_t = a(X_t, \theta)dt + b(X_t, \theta)d\hat{W}_t, \tag{6}$$

where a, b are some scalar drift and diffusion term, that ensure a weak solution of this SDE exists. As $N \to \infty$ the solution of the above ODE X_t will converge to the solution had it been driven by the exact Brownian motion. The seminal work of Wong and Zakai (1965) shows that as $N \to \infty$ the solution actually converges to the solution of a Stratonovich SDE given by

$$dX_t = a(X_t, \theta)dt + b(X_t, \theta) \circ dW_t. \tag{7}$$

Thus, we can approximate the above Stratonovich SDE as

$$dX_t = a(X_t, \theta)dt + b(X_t, \theta) \circ d\hat{W}_t, \tag{8}$$

which is just the ODE:

$$\frac{d\hat{X}}{dt} = a(\hat{X}, \theta) + b(\hat{X}, \theta) \sum_{i=1}^{N} Z_i \phi_i(t). \tag{9}$$

Similarly, we can approximate a K-dimensional Brownian motion by applying the truncated series approximation to each of the K dimensions. Substituting such a K-dimensional approximation in Equation (1) we obtain the following ODE:

$$\frac{d\hat{\boldsymbol{X}}}{dt} = \boldsymbol{a}(\hat{\boldsymbol{X}}, \boldsymbol{\theta}) + \boldsymbol{B}(\hat{\boldsymbol{X}}, \boldsymbol{\theta}) \sum_{i=1}^{N} \boldsymbol{Z}_i \Phi_i(t), \tag{10}$$

where $\boldsymbol{Z}_i, \Phi_i \in \mathbb{R}^K$. In the multivariate case the convergence to the Stratonovich SDE is not guaranteed in general (Lyons et al., 2014). However, if one chooses to expand the Brownian motion using Haar wavelets then convergence is indeed guaranteed (Lyons et al., 2014; McShane, 2020). The latter is the Levy-Ciecelski construction of Brownian motion. In our experiments we did not observe a failure to converge to the Stratonovich limit while using the Karhunen-Loeve basis functions. Similar findings were reported in Lyons et al. (2012).

The first few coefficients in general captures the large scale oscillations whereas the remaining coefficients determine the small-scale (high frequency) oscillations. Thus, even without using a really large number of coefficients a good approximation to the true time t marginal distribution $p(\hat{\boldsymbol{X}}(t))$ can be achieved. In Figure 1 we compare the marginal density of the states of a Lotka-Volterra model (see section 8.1) obtained through the Euler-Maruyama and the series expansion introduced here. Clearly, with even 10 coefficients both the approximations match well.

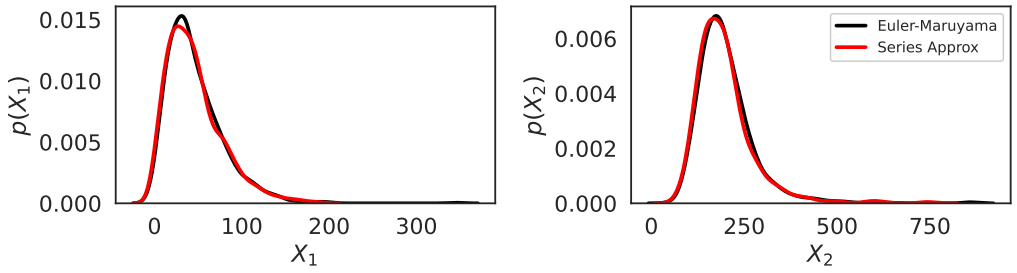

Figure 1: We compare the marginal density of the Lotka-Volterra model states at time T=30. We used 1000 samples to evaluate these densities.

5 Inference using MCMC

In order to use the ODE approximation of the SDE for estimating the SDE parameters $\boldsymbol{\theta}$, we first assume that $\boldsymbol{y}$ is the noisy observation, at the M time-points, of the solution $\hat{\boldsymbol{X}}(\boldsymbol{\theta}) := \hat{\boldsymbol{X}}(\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0) \in \mathbb{R}^K$ of the ODE given by Equation (5), which approximates the actual diffusion path $\hat{\boldsymbol{X}} \approx \boldsymbol{X}$, where $\mathbf{Z} := \{\boldsymbol{Z}_i\}_{i=1}^N$ and $\boldsymbol{x}_0$ is the initial value. Note that for a state-dependent diffusion term the Itô and Stratonovich formulation differ in their drift functions. Since the series approximation converges to the Stratonovich SDE, thus an Itô SDE needs to be converted to its equivalent Stratonovich form before applying the approximation. This can be done by modifying the drift term (see appendix A for details). The task then is to infer the posterior distribution of all the unknown parameters of the ODE: $\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0$. Since the random variables $\mathbf{Z}$ are distributed as standard Normal, we use the same as the prior on these variables. Thus, we have the prior: $p(\boldsymbol{Z}_i) = \mathcal{N}(\mathbf{0}, \mathbb{I}_{K\times K})$.

We can now write the joint posterior distribution of $(\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0)$ as

$$p(\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0|\boldsymbol{y}) \propto p(\boldsymbol{y}|\hat{\boldsymbol{X}}(\boldsymbol{\theta}))p(\mathbf{Z})p(\boldsymbol{\theta}, \boldsymbol{x}_0), \tag{11}$$

where the likelihood $p(\boldsymbol{y}|\hat{\boldsymbol{X}}(\boldsymbol{\theta}))$ is now both tractable and differentiable, with respect to $\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0$.

We can obtain samples from this posterior distribution using a MCMC, sampler that targets gradient of the log posterior: $\nabla_{\boldsymbol{\theta},\mathbf{Z},\boldsymbol{x}_0}\{\log p(\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0|\boldsymbol{y})\}$, such as the No-U-Turn (Hoffman and Gelman, 2014) sampler. The marginal density $p(\boldsymbol{\theta}|\boldsymbol{y})$ can be easily obtained by collecting the corresponding samples. We can also obtain the approximation to the latent diffusion's sample paths, conditioned on the data, $\hat{\boldsymbol{X}}(\boldsymbol{\theta})|_{\boldsymbol{\theta},\mathbf{Z},\boldsymbol{x}_0\sim p(\boldsymbol{\theta},\mathbf{Z},\boldsymbol{x}_0|\boldsymbol{y})}$ by solving the ODE using the parameter samples.

6 Variational inference

Let us first introduce the shorthand $\boldsymbol{\Theta} := (\boldsymbol{\theta}, \mathbf{Z}, \boldsymbol{x}_0)$ to denote the vector containing all the unknown quantities that we want to estimate. Using variational inference we can approximate $p(\boldsymbol{\Theta}|\boldsymbol{y})$ with a

tractable distribution $q(\boldsymbol{\Theta}|\boldsymbol{\lambda})$ from a family of distributions $q(\cdot|\boldsymbol{\lambda})$, indexed by $\boldsymbol{\lambda}$, by maximising the *evidence lower bound* (ELBO) (Jordan et al., 1999) given by

$$\mathcal{L}(\boldsymbol{\lambda}) = \mathbb{E}[\log p(\boldsymbol{y}|\hat{\boldsymbol{X}}(\boldsymbol{\Theta}))p(\boldsymbol{\Theta})] - \mathbb{E}[\log q(\boldsymbol{\Theta}|\boldsymbol{\lambda})] \tag{12}$$

where the above expectations are with respect to $q(\boldsymbol{\Theta}|\boldsymbol{\lambda})$. If gradient of the ELBO, w.r.t the variational parameter $\boldsymbol{\lambda}$, is available then variational inference can be formulated as a simple gradient descent problem as follows:

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma \nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}), \tag{13}$$

where γ is a learning rate. However, for an ODE the above expectations are intractable. Following Ghosh et al. (2021), we apply the reparameterisation trick (Kingma et al., 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014) to obtain a Monte Carlo (MC) estimate, using L samples, of the gradient of the ELBO:

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{MC}(\boldsymbol{\lambda}) = \frac{1}{L} \sum_{l=1}^{L} \nabla_{\boldsymbol{\Theta}} \left[\left\{ \log p(\boldsymbol{y}|\hat{\boldsymbol{X}}(\boldsymbol{\Theta}^l))p(\boldsymbol{\Theta}^l) - \log q(\boldsymbol{\Theta}^l|\boldsymbol{\lambda}) \right\} \right] \nabla_{\boldsymbol{\lambda}} g(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^{(l)}), \tag{14}$$

where $\boldsymbol{\Theta}^l$ is the output of an invertible, differentiable function $g(\boldsymbol{\lambda}, \boldsymbol{\epsilon}^{(l)})$ and $\boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$–a parameter free distribution. Substituting this MC estimate in Equation (13), we can find the optimal $\boldsymbol{\lambda}^*$ using the following stochastic optimisation update:

$$\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \gamma \nabla_{\boldsymbol{\lambda}} \mathcal{L}_{MC}(\boldsymbol{\lambda}). \tag{15}$$

6.1 Choice of the approximation

To place prior distributions with support on positive reals we need to transform the support of $\boldsymbol{\Theta}$ to the unconstrained real line $\mathbb{R}^D$: $T : \mathbb{R}^D_{>0} \rightarrow \mathbb{R}^D$, and subsequently obtain a transformed parameter vector $\boldsymbol{\xi} = T(\boldsymbol{\Theta})$. The posterior density $p(\boldsymbol{\Theta}|\boldsymbol{y})$, with the above transformation, is given by

$$p(\boldsymbol{\Theta}|\boldsymbol{y}) \propto p(\boldsymbol{y}|\boldsymbol{X}(T^{-1}(\boldsymbol{\xi})))p(T^{-1}(\boldsymbol{\xi})) \mid \det J_{T^{-1}}(\boldsymbol{\xi}) \mid, \tag{16}$$

where $J_{T^{-1}}(\boldsymbol{\xi})$ is the Jacobian of the inverse of T. This transformation lets us choose an approximating distribution $q(\boldsymbol{\xi}|\boldsymbol{\lambda})$ with unconstrained support, such as a Gaussian.

Gaussian approximation: Parameters of a nonlinear ODE such as Equation (10) are often strongly correlated. Moreover due the nature of our approximation we also expect correlations to exist between the expansion coefficients $\mathbf{Z}$ and $\boldsymbol{\Theta}$. For nonlinear ODEs, a full-rank Gaussian density as the variational approximation was shown in Ghosh et al. (2021) to be able to capture the correlation structure of the posterior amply. Thus, folowing Ghosh et al. (2021) we also chose a full-rank Gaussian approximation: $q(\boldsymbol{\xi}|\boldsymbol{\lambda}) = \mathcal{N}(\boldsymbol{\xi}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\lambda} = \left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ are the variational parameters. To ensure that the covariance matrix $\boldsymbol{\Sigma}$ remains positive semidefinite, we parameterise the covariance using Cholesky factorisation, $\boldsymbol{\Sigma} = \boldsymbol{L}\boldsymbol{L}^{'}$. To ensure uniqueness we take the logarithm of the diagonal elements of $\boldsymbol{L}$. The required reparameterisation, $\boldsymbol{\xi} = g(\boldsymbol{\lambda}, \boldsymbol{\epsilon})$, then simply follows as the affine transform $\boldsymbol{\xi} = \boldsymbol{\mu} + \boldsymbol{L}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$.

Normalizing flows: The full-rank Gaussian, although is able to capture strong correlations, lacks richness to adequately approximate a complex and possibly multi-modal posterior density if we model the drift and diffusion functions using a neural network, as required for a system identification task (see section 9). A series of invertible nonlinear transformations, a normalizing flow, can be applied to a simple base density (say a Gaussian) to generate a richer approximating density. In particular we use a neural autoregressive flow which has a compact parameterisation: block neural autoregressive flow (BNAF) (De Cao et al., 2020), and is proven to be an universal density approxmator.

7 Gradient evaluations

To carry out inference we need to obtain the gradient of a scalar function: i) the log posterior in the MCMC case or, ii) the ELBO for variatonal inference with respect to the model parameters $\boldsymbol{\Theta}$ or the variational parameters $\boldsymbol{\lambda}$. This in turn requires the propagation of gradients through the ODE. This can be efficiently carried out using the *adjoint sensitivity* analysis (Chen et al., 2018; Rackauckas et al., 2018), which is the continuous formulation of reverse-mode automatic differentiation. We

describe this for obtaining the gradient of the ELBO w.r.t $\boldsymbol{\Theta}$. Note that the downstream gradients, $\frac{d\boldsymbol{\Theta}}{d\boldsymbol{\lambda}}$, can be trivially obtained using AD. Gradient of the prior and likelihood densities can be obtained analogously.

Consider a cost function, such as the ELBO, that depends on the ODE solution at the measurement times:

$$C(\hat{\boldsymbol{X}}) = \sum_i c(\hat{\boldsymbol{X}}_{t_i}), \tag{17}$$

where the sum appears due the factorisation of the likelihood over the time axis. In *adjoint sensitivity* analysis (Rackauckas et al., 2018) the gradient of the above scalar-valued cost function $C(\cdot)$, whose input is the ODE solution, can be computed directly. The first step is to solve a backwards ODE, the adjoint problem:

$$\frac{d\boldsymbol{a}(t)}{dt} = -\boldsymbol{a}(t)^{\mathrm{T}} \frac{\partial \boldsymbol{f}}{\partial \hat{\boldsymbol{X}}}, \tag{18}$$

where we use the shorthand $\boldsymbol{f}$ to denote denote the velocity field of the ODE in Equation (10). Furthermore, at each experimental time point t_i this backward ODE is perturbed by $\frac{\partial c(\hat{\boldsymbol{X}}_{t_i})}{\partial \hat{\boldsymbol{X}}}$. The gradient of the cost function with respect to the ODE parameters can be evaluated by another quadrature as follows:

$$\frac{dC}{d\boldsymbol{\theta}} = \boldsymbol{a}(t_1)^{\mathrm{T}} \frac{\partial \boldsymbol{f}\left(\hat{\boldsymbol{X}}_{t_1}, \boldsymbol{\Theta}\right)}{\partial \boldsymbol{\Theta}} + \sum_i \int_{t_i}^{t_{i+1}} \boldsymbol{a}(t)^{\mathrm{T}} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\Theta}} dt. \tag{19}$$

Note that a continuous solution of the system and the adjoint states is required for the integration above. Alternatively, the system, the adjoint and the cost function ODEs can be solved simultaneously backward in time (Chen et al., 2018). We have used this reverse-time simultaneous-solution technique introduced in Chen et al. (2018).

8 Benchmarking: parameter estimation in mechanistic models

We begin by testing the efficacy of the series approximation for the task of parameter estimation. In this regard we use two biological SDEs: (i) the Lotka-Volterra model, and (ii) the stochastic SIR epidemic model. For the Lotka-Volterra model we have used simulated data but for the SIR model we used real data. For both these models we first apply a particle marginal Metropolis-Hastings (PMMH) algorithm (Andrieu et al., 2010; Golightly and Wilkinson, 2011) to obtain a *gold standard* estimate of the posterior of $\boldsymbol{\theta}, \boldsymbol{X}$ that we treat as a proxy for the true unknown posterior. We used a Bootstrap particle filter (Gordon et al., 1995; Golightly and Wilkinson, 2011) which uses the Euler-Maruyama discretisation as the proposal (Golightly and Wilkinson, 2011) for $\boldsymbol{X}$ and we update $\boldsymbol{\theta}$ using an adaptive random-walk Metropolis-Hastings algorithm. Having run the PMMH, we then compare with this gold standard the estimates of the posterior obtained by applying MCMC using the No-U-Turn (NUTS) algorithm (Hoffman and Gelman, 2014) and variational inference, both using the proposed series approximation: which we denote as **SA-ODE**. For, variational inference we used the RMSprop (Tieleman and Hinton, 2016) optimisation algorithm with a step size of 10^{-3}. Our choice of RMSprop is motivated by the findings in Ghosh et al. (2021). By VI we denote variational inference.

For both the models we have used $N = 10$ basis functions, each being the KL expansion function (see Equation (4)). In appendix B we have compared the posterior estimates for the Lotka-Volterra, as obtained by NUTS, between the KL and the Haar wavelet basis. Although we obtain similar estimates using both type of basis functions, the wavelet bases require stricter error tolerances for the ODE solver. Furthermore, we have also carried out a sensitivity analysis to the choice of N, the results can be found in appendix B. Finally, we have set the value of T in Equation (4) to be the end-point of the chosen time interval for each of the models.

We used the `numpyro` probabilistic programming library to apply NUTS and variational inference for the ease of comparison. Furthermore, we used `google jax` implementation of the Dormand-Prince adaptive ODE solver for integrating the **SA-ODE**. This solver provides an implementation of adjoint sensitivity that is needed for applying NUTS and variational inference. For the PMMH algorithm we implemented a vectorised particle filter in `google jax` and implemented the adaptive MCMC in `Python`. The code is available at `retracted`

8.1 Stochastic Lotka-Volterra model

The stochastic Lotka–Volterra model (Wilkinson, 2018) has been widely used for benchmarking (see Fearnhead et al. (2014); Giagos (2010)). This model describes a population comprising of two competing species: *predators* which die with rate c_2 and reproduce with rate c_1 by consuming prey, which in turn reproduce with rate c_3. This system can be defined using the following drift and diffusion terms (see (Wilkinson, 2018) for derivation):

$$\boldsymbol{a}(\boldsymbol{X}_t, \boldsymbol{\theta}) = \begin{bmatrix} c_1 X_1 - c_2 X_1 X_2 \\ c_2 X_1 X_2 - c_3 X_2 \end{bmatrix}, \quad \boldsymbol{B}(\boldsymbol{X}_t, \boldsymbol{\theta}) = \begin{bmatrix} c_1 X_1 & c_2 X_1 X_2 \\ -c_2 X_1 X_2 & c_3 X_2 + c_2 X_1 X_2 \end{bmatrix}, \tag{20}$$

where we denote by X_1, X_2 the prey and predator species respectively. The parameter vector is the rate constants: $\boldsymbol{\theta} = (c_1, c_2, c_3)$, and the task is to estimate these given a noise corrupted sample path from the above system. We generate such a sample path using the Euler-Maruyama discretisation, with initial values $\boldsymbol{x}_0 = (100., 100.)$, between the time interval $[0 : 0.1 : 50]$. A set of 10 evenly spaced values from this path corrupted with Gaussian noise with $\sigma = 10$ constitute the observations $\boldsymbol{y}$. Following Golightly and Wilkinson (2011) we consider $\boldsymbol{x}_0, \sigma$ to be known and thus we are left with estimating the rate constants and the expansion coefficients.

The likelihood, for the series approximation, is a Gaussian, $p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{Z}, \sigma) = \prod_i \mathcal{N}(X(t_i; \boldsymbol{\theta}, \mathbf{Z}), \sigma^2 \mathbb{I})$, where $\mathbb{I}$ is a 2×2 identity matrix. We place a $\mathrm{Gamma}(1, 2)$ prior on all the parameters. We placed the following priors on the rates: $c_1 \sim \mathrm{Beta}(2, 1)$ $c_2 \times 100 \sim \mathrm{Half}\,\mathcal{N}(0, 1)$ and $c_3 \sim \mathrm{Beta}(1, 2)$. The priors on the expansion coefficients are just standard Gaussians.

We ran two chains of PMMH, each for 100,000 iterations from slightly differing initial states. We discarded the first 50,000 iterations as burnin for each chain and thinned accordingly to have 1000 samples representing the gold standard posterior estimate. For the series approximation two chains of NUTS is run for 1000 iterations after an initial 500 warmup iterations. The NUTS samples, from the two chains, are then thinned to obtain 1000 samples. VI with the series approximation was run for 30,000 iterations with $L = 1$. For plotting and summarising the posterior distributions, and comparing to the gold standard we used 1000 samples from the variational approximation for this and the subsequent (SIR model) example.

We present summaries of the posterior marginals in Table 1. Plots of the posterior diffusion paths are given in appendix C. We notice close match between the posteriors estimated with the series approximation to that of the PMMH. Interestingly, the variational inference estimate did not show underestimation of posterior variance (similar observations were made in Ghosh et al. (2021)). NUTS produced a relative ESS (averaged over the three parameters) of 0.89 and PMMH produced 0.05. We repeated the inference process using two more realisations of the artificial noise. Additional results are summarised in appendix C.

8.2 The SIR compartmental model

The SIR model (Anderson et al., 1992) of infectious disease models the number of susceptible (S), infected (I), and recovered (R) people in a population subjected to an epidemic. The stochastic version of the SIR model, for a population of N people, can be defined using an SDE with the following drift and diffusion terms (see Fuchs (2013) for derivation]):

$$\boldsymbol{a}(\boldsymbol{X}_t, \boldsymbol{\theta}) = \begin{bmatrix} -\beta SI \\ \beta SI - \gamma I \end{bmatrix}, \quad \boldsymbol{B}(\boldsymbol{X}_t, \boldsymbol{\theta}) = \begin{bmatrix} \beta SI & -\beta SI \\ -\beta SI & \beta SI + \gamma I \end{bmatrix}, \tag{21}$$

where the infection β and recovery γ rates are unknown parameters. Also, we have $N = S(t) + I(t) + R(t)$. We used this SDE to model an outbreak of influenza at a boys boarding school in 1978 (Jackson et al., 2013). This particular dataset was previously used for benchmarking in Ryder et al. (2018). This data consists of the number of infections $\boldsymbol{I_{obs}}$ for a period of 14 days. The population size is $N = 763$. In addition to β, γ we also estimated the fractional initial susceptibility, $s_0 = S(t = 0)/N$, assuming the initial recovered fraction $r_0 = 0$ and thus $i_0 = 1 - s_0$. As this is count data we have used a Poisson likelihood while using the series approximation, $p(y(t)|\beta, \gamma, s_0, \mathbf{Z}) = \mathrm{Poisson}(I(t))$, and placed the following priors: $\beta, \gamma \sim \mathrm{Gamma}(2, 2)$ and $s_0 \sim \mathrm{Beta}(2, 1)$.

For this problem we ran two chains of PMMH, each for 200,000 iterations, again started with slightly different initial states. In this case we discarded the first 100,000 iterations as burnin for each chain

Table 1: We summarise the **mean** $\pm$ **standard deviation** of the posterior distribution of each parameter for the two models. **VI** and **NUTS** are using the **SA-ODE** model. These were run on a 3.6 GHz machine with 16 GB memory

THE SIR MODEL				
$\boldsymbol{\theta}$	TRUE VALUE	**PMMH**	**VI**	**NUTS**
β	–	1.8427 ± 0.0719	1.8069 ± 0.1319	1.8479 ± 0.1413
γ	–	0.4875 ± 0.0190	0.4849 ± 0.0278	0.4851 ± 0.0258
s_0	–	0.9964 ± 0.0010	0.9957 ± 0.0010	0.9959 ± 0.0014

THE STOCHASTIC LOTKA-VOLTERRA MODEL				
$\boldsymbol{c}$	TRUE VALUE	**PMMH**	**VI**	**NUTS**
c_1	0.5	0.5081 ± 0.0261	0.4846 ± 0.0205	0.4961 ± 0.0219
$100 \times c_2$	0.25	0.2492 ± 0.0113	0.2417 ± 0.0092	0.2454 ± 0.0096
c_3	0.3	0.2965 ± 0.0159	0.2872 ± 0.0115	0.2924 ± 0.0126

and thinned accordingly to obtain 1000 samples that represent the gold standard. We used the same setup, as was used in the previous example, for applying NUTS and VI with the series approximation.

We present summaries of the posterior marginals in Table 1. The model fit plot is shown in appendix C. In this example NUTS produced a relative ESS (again averaged over the three parameters) of 0.95 and PMMH produced 0.05.

9 System identification task

So far we have concentrated on estimating the parameters of an SDE whose drift and diffusion terms are known. This is the context of classical inference in all mechanistic modelling scenarios. In the case of system identification we may have no knowledge about the drift or the diffusion functions. That is we are given a time-series dataset and the goal is then fit a black-box function that can predict the response at future time points. For this type of task Chen et al. (2018) proposed an ODE model whose velocity field is given by a neural network. Several extensions to this modelling framework have been proposed recently. We introduce here a (Stratonovich) SDE version of such a Neural ODE model, where the drift and the diffusion functions are modelled with neural networks. Such an SDE is given by

$$d\boldsymbol{X}_t = \boldsymbol{f}(\boldsymbol{X}_t, \boldsymbol{\eta}_m)dt + \boldsymbol{g}(\boldsymbol{X}_t, \boldsymbol{\eta}_s) \circ d\boldsymbol{W}_t, \quad \boldsymbol{X}_0 = \boldsymbol{x}_0, \tag{22}$$

where $\boldsymbol{f}, \boldsymbol{g}$ are neural networks with parameters $\boldsymbol{\eta}_m$ and $\boldsymbol{\eta}_s$ respectively. By applying the series approximation we get the following **SA-ODE**:

$$\frac{d\hat{\boldsymbol{X}}}{dt} = \boldsymbol{f}(\hat{\boldsymbol{X}}_t, \boldsymbol{\eta}_m) + \boldsymbol{g}(\hat{\boldsymbol{X}}_t, \boldsymbol{\eta}_s) \sum_{i=1}^{N} \boldsymbol{Z}_i \Phi_i(t), \tag{23}$$

where $\{\boldsymbol{Z}_i\}_{i=1}^N$ are expansion coefficients. Notice that if we make the diffusion term zero then we recover the usual Neural ODE (Chen et al., 2018). The diffusion term can be treated as an additional regularisation from a Neural ODE perspective. Interestingly, the series approximation provides a straightforward approach to regularise Neural ODE models.

Unlike a latent ODE (a time-series VAE) formulation as proposed in Chen et al. (2018)], we intend to use the variational inference formulation that we have setup in this paper. That is we place priors on $\boldsymbol{\eta}_m, \boldsymbol{\eta}_s$ and try to infer their posterior distribution. Once, a variational approximation to the joint posterior $p(\boldsymbol{\eta}_m, \boldsymbol{\eta}_s, \boldsymbol{Z}|\boldsymbol{y})$ is learnt, we can simply generate future predictions using the posterior predictive distribution. In this case we have used a BNAF as the variational approximation (see appendix D for details).

Here our goal is to investigate that whether the diffusion term and its series approximation can regularise the Neural ODE enough to stop overfitting, without any other additional regularisation.

Table 2: Average MSE on future time points.

MODEL	TEST ERROR	REFERENCE
GPDM	126.46 ± 34	WANG ET AL. (2007)
VGPLVM	142.18 ± 1.92	DAMIANOU ET AL. (2011)
DTSBN-S	80.21 ± 0.04	GAN ET AL. (2015)
NPODE	45.74	HEINONEN ET AL. (2018)
NEURAL ODE	87.23 ± 0.02	CHEN ET AL. (2018)
ODE^{2}VAE	93.07 ± 0.72	YILDIZ ET AL. (2019)
ODE^{2}VAE-KL	15.99 ± 4.16	YILDIZ ET AL. (2019)
BNSA-ODE	**41.21** $\pm$ 36.07	THIS WORK

9.1 Motion capture task

To demonstrate the efficacy of our Bayesian neural SA-ODE (which we denote as **BnSA-ODE**) model we have used a dataset extracted from the CMU motion capture library. This dataset consists of 43 walking sequences of several subjects, each of which is fitted separately. Heinonen et al. (2018) used this dataset to demonstrate an ODE model, npODE, whose velocity field was modelled using a Gaussian process. In Yildiz et al. (2019) this dataset was used to demonstrate a second order extension, ODE^{2}VAE, to the latent ODE model of Chen et al. (2018). In Yildiz et al. (2019) the same preprossessing as was done in Wang et al. (2007) was used which left a 50-dimensional time-series. This preprossessed dataset was made available in Yildiz et al. (2019) and this is what we have used. Additionally, we have scaled the data to be within a $[-1, 1]$ range. We followed the convention in Yildiz et al. (2019) of using the first two third of each sequence for training/validation and the rest for testing purpose i.e evaluating a MSE score.

We considered a 3-dimensional state-space and used a decoder network that maps the latent state-space to the 50-dimensional output. Moreover, we consider a Gaussian measurement model and we estimate the noise standard deviation on which we place a $\mathrm{Half}\,\mathcal{N}(0, 1)$ prior. We place a standard Gaussian prior on all the neural network parameters. We used a shallow architecture with 10 hidden units for the drift, diffusion and decoder networks (see appendix D for details). We retained the same time interval for numerical solution of the **BnSA-ODE** as was used in Yildiz et al. (2019). For the series expansion we have used $N = 15$ basis functions and due to the need of extrapolation we set the value of T for the KL basis functions (see Equation (4)) to be twice the end-point of the training interval. We ran **VI** for 1200 iterations with the RMSProp where the step-size was set to 10^{-2}. We set $L = 40$. Running further iterations results in over-fitting. We evaluated the MSE between the data and the mean of the posterior predictive distribution at the test time-points for each sequence. We evaluated the posterior predictive distribution pointwise using 50 samples from the variational approximation.

BnSA-ODE performs better than the Neural ODE on average and only ODE^{2}VAE with a KL penalty produces smaller average test error. However, **BnSA-ODE** performs poorly for some sequences and that is why it has a much higher variance of the test error. One way to enhance the **BnSA-ODE** model is to use a deeper architecture. Alternatively, **BnSA-ODE** can be integrated within a latent ODE framework.

10 Conclusion

We presented a method to carry out inference for a nonlinear SDE by approximating the SDE with an ODE. We do this using a truncated series expansion of Brownian motion. For the parameter estimation task, of mechanistic models, this method produce close approximation to the estimate obtained using a particle MCMC algorithm. Which is considered the state-of-the-art for this type of inference problems. Furthermore, our approach lets us easily convert a Neural ODE to a Neural SDE and vice-versa. We exploited this facet of our approach by approximating a Neural SDE using the series expansion for a system identification task. In future we want extend the proposed approach to develop a latent SDE model using the VAE formulation.

References

Roy M Anderson, B Anderson, and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992.

Christophe Andrieu, Arnaud Doucet, and Roman Holenstein. Particle markov chain monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3): 269–342, 2010.

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.

Andreas C. Damianou, Michalis K. Titsias, and Neil D. Lawrence. Variational gaussian process dynamical systems. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, page 2510–2518, 2011.

Nicola De Cao, Wilker Aziz, and Ivan Titov. Block neural autoregressive flow. In *Uncertainty in Artificial Intelligence*, pages 1263–1273. PMLR, 2020.

Paul Fearnhead, Vasilieos Giagos, and Chris Sherlock. Inference for reaction networks using the linear noise approximation. *Biometrics*, 70(2):457–466, 2014.

Christiane Fuchs. *Inference for diffusion processes: with applications in life sciences*. Springer Science & Business Media, 2013.

Zhe Gan, Chunyuan Li, Ricardo Henao, David Carlson, and Lawrence Carin. Deep temporal sigmoid belief networks for sequence modeling. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2467–2475, 2015.

Sanmitra Ghosh, Paul Birrell, and Daniela De Angelis. Variational inference for nonlinear ordinary differential equations. 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 2719–2727. PMLR, 13–15 Apr 2021.

Vasileios Giagos. *Inference for auto-regulatory genetic networks using diffusion process approximations*. PhD thesis, Lancaster University, 2010.

Mike Giles and Paul Glasserman. Smoking adjoints: Fast monte carlo greeks. *Risk*, 19(1):88–92, 2006.

Emmanuel Gobet and Rémi Munos. Sensitivity analysis using itô–malliavin calculus and martingales, and application to stochastic optimal control. *SIAM Journal on control and optimization*, 43(5): 1676–1713, 2005.

Andrew Golightly and Colin S Gillespie. Simulation of stochastic kinetic models. In *In Silico Systems Biology*, pages 169–187. Springer, 2013.

Andrew Golightly and Darren J Wilkinson. Bayesian inference for nonlinear multivariate diffusion models observed with error. *Computational Statistics & Data Analysis*, 52(3):1674–1693, 2008.

Andrew Golightly and Darren J Wilkinson. Bayesian parameter inference for stochastic biochemical network models using particle markov chain monte carlo. *Interface focus*, 1(6):807–820, 2011.

Neil Gordon, David Salmond, and Craig Ewing. Bayesian state estimation for tracking and guidance using the bootstrap filter. *Journal of Guidance, Control, and Dynamics*, 18(6):1434–1443, 1995.

Markus Heinonen, Cagatay Yildiz, Henrik Mannerström, Jukka Intosalmi, and Harri Lähdesmäki. Learning unknown ode models with gaussian processes. In *International Conference on Machine Learning*, pages 1959–1968. PMLR, 2018.

Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *Journal of Machine Learning Research*, 15(1):1593–1623, 2014.

Charlotte Jackson, Emilia Vynnycky, Jeremy Hawker, Babatunde Olowokure, and Punam Mangtani. School closures and influenza: systematic review of epidemiological studies. *BMJ open*, 3(2), 2013.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

D P Kingma, Max Welling, et al. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, volume 1, 2014.

Xuechen Li, Ting-Kam Leonard Wong, Ricky T. Q. Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3870–3882. PMLR, 26–28 Aug 2020.

Simon M. J. Lyons, Amos J. Storkey, and Simo Särkkä. The coloured noise expansion and parameter estimation of diffusion processes. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 1961–1969, 2012.

Simon MJ Lyons, Simo Särkkä, and Amos J Storkey. Series expansion approximations of brownian motion for non-linear kalman filtering of diffusion processes. *IEEE Transactions on Signal Processing*, 62(6):1514–1524, 2014.

EJ McShane. Stochastic differential equations and models of random processes. In *Contributions to Probability Theory*, pages 263–294. University of California Press, 2020.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Christopher Rackauckas, Yingbo Ma, Vaibhav Dixit, Xingjian Guo, Mike Innes, Jarrett Revels, Joakim Nyberg, and Vijay Ivaturi. A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions. *arXiv preprint arXiv:1812.01892*, 2018.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286. PMLR, 2014.

Tom Ryder, Andrew Golightly, A. Stephen McGough, and Dennis Prangle. Black-box variational inference for stochastic differential equations. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4423–4432, 2018.

T. Tieleman and G Hinton. Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning, 4*, 2016.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1971–1979. PMLR, 2014.

Jack M Wang, David J Fleet, and Aaron Hertzmann. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2):283–298, 2007.

Darren J Wilkinson. *Stochastic modelling for systems biology*. CRC press, 2018.

Eugene Wong and Moshe Zakai. On the Convergence of Ordinary Integrals to Stochastic Integrals. *The Annals of Mathematical Statistics*, 36(5):1560 – 1564, 1965. doi: 10.1214/aoms/1177699916. URL `https://doi.org/10.1214/aoms/1177699916`.

Jichuan Yang and Harold J Kushner. A monte carlo method for sensitivity analysis and parametric optimization of nonlinear stochastic systems. *SIAM journal on control and optimization*, 29(5): 1216–1249, 1991.

Cagatay Yildiz, Markus Heinonen, and Harri Lähdesmäki. ODE2VAE: deep generative second order odes with bayesian neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 13412–13421, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section **??**.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [No]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Converting from Itô to Stratonovich formulation

The Stratonovich SDE given by

$$dX_t^i = \tilde{a}(X_t^i, \boldsymbol{\theta})dt + \sum_{j=1}^{K} \sqrt{b^{i,j}(X_t, \boldsymbol{\theta})} \circ dW_t^j, \quad i, j = 1, \dots K, \tag{24}$$

with the same solutions as the K-dimensional Ito SDE driven by a K-dimensional Wiener process given by

$$dX_t^i = a(X_t^i, \boldsymbol{\theta})dt + \sum_{j=1}^{K} \sqrt{b^{i,j}(X_t, \boldsymbol{\theta})} dW_t^j, \quad i, j = 1, \dots K, \tag{25}$$

has a drift coefficient that is defined component-wise as

$$\tilde{a}(X_t^i, \boldsymbol{\theta})dt = a(X_t^i, \boldsymbol{\theta})dt + \sum_{k=1}^{K} \sum_{j=1}^{K} \sqrt{b^{k,j}(X_t, \boldsymbol{\theta})} \frac{\partial\{\sqrt{b^{i,j}(X_t, \boldsymbol{\theta})}\}}{\partial X_t^k}. \tag{26}$$

The above partial derivatives can be evaluated either through hand-calculation or using AD itself. The later is what we did in our experiments.

B Wavelet basis function

For the multivariate case uniform convergence of the SA-ODE to the corresponding Stratonovich SDE is guaranteed if one chooses Haar wavelets as an orthonormal basis in which to expand the driving Brownian motion (Lyons et al., 2014). Thus, we wanted to compare the approximation achieved by using the KL expansion with that of Haar wavelets. But, before we delve into the comparison, let us briefly introduce the Haar wavelets: which is a complete orthonormal basis of $L^2[0, T]$.

The Haar wavelets are parameterised by two natural numbers: the scale $n \geq 0$ and the shift $0 \leq k < 2^n$. The first wavelet is defined as

$$\boldsymbol{\psi}_{0,0}(t) = \begin{cases} 1 & 0 \leq t < \frac{T}{2} \\ -1 & \frac{T}{2} < t \leq T \\ 0 & \text{otherwise} \end{cases}. \tag{27}$$

Further wavelets are defined by rescaling $\boldsymbol{\psi}_{0,0}$ so that it is non-zero only on some sub-interval of $[0, T]$, while ensuring that the wavelet still has unit norm. In general,

$$\boldsymbol{\psi}_{n,k}(t) = \frac{2^{n/2}}{\sqrt{T}} \boldsymbol{\psi}_{0,0}(2^n - kt). \tag{28}$$

Thus, $\boldsymbol{\psi}_{1,0}$ is a copy of $\boldsymbol{\psi}_{0,0}$ restricted to $[0, T/2]$, and $\boldsymbol{\psi}_{1,1}$ is a copy restricted to $[T/2, T]$. Furthermore, we add the constant function $\boldsymbol{\psi}_* = \frac{1}{\sqrt{T}}$ to form a complete basis. To be consistent with the notation introduced in section 4 we set $\boldsymbol{\phi}_1 = \boldsymbol{\psi}_*$, $\boldsymbol{\phi}_2 = \boldsymbol{\psi}_{0,0}$, $\boldsymbol{\phi}_3 = \boldsymbol{\psi}_{1,0}$ and so on.

To compare the wavelet and KL basis we ran NUTS and VI on the simulated dataset used in section 8.1, with the same algorithmic settings retained. We set $N = 10$ and $T = 50$ as was done in section 8.1. Marginal densities of the parameters are plotted in Figure 2. Although we get similar estimates, as in (Lyons et al., 2014, 2012), following which we used KL expansion throughout, the wavelets required stricter error tolerances for the ODE solver. We thus recommend the usage of a stiff solver when using the wavelet basis. Extension of AD for a stiff solver can be done using the custom op creation method of Ghosh et al. (2021). However, a JIT compiled solver, provided with `Jax`, is faster.

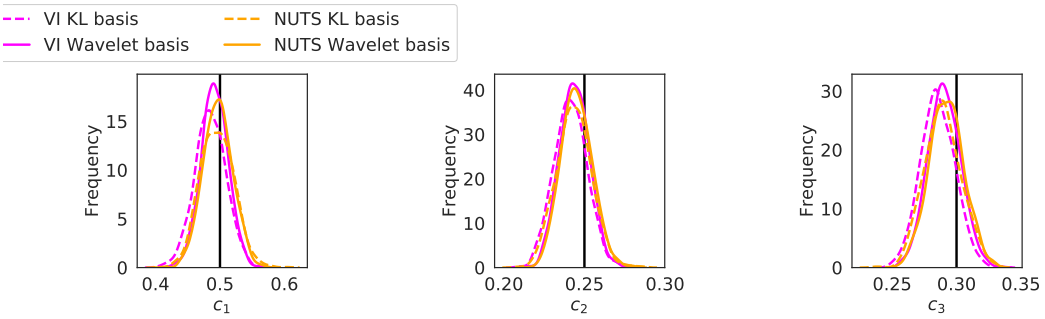

Figure 2: We compare the marginal density of the Lotka-Volterra parameters as obtained by NUTS and VI, using the wavelets (solid lines) and KL (dashed lines) basis. The black vertical lines indicate the true parameter values.

B.1 Sensitivity to truncation of the series: the choice of N

To carry out this sensitivity analysis we use three realisations of the simulated dataset. The first one is used in section 8.1 and the estimates on the rest are summarised in section C.1. We ran the NUTS algorithm with $N = 3, 5, 8, 10$ respectively and measured the maximum mean discrepancy (MMD) to the corresponding estimates obtained by PMMH for each dataset. We used the KL basis throughout and retained all algorithmic settings for NUTS as in section 8.1. Figure 3 compares the MMD for specific choices of N. It is apparent that with $N = 8$ the MMD plateaus.

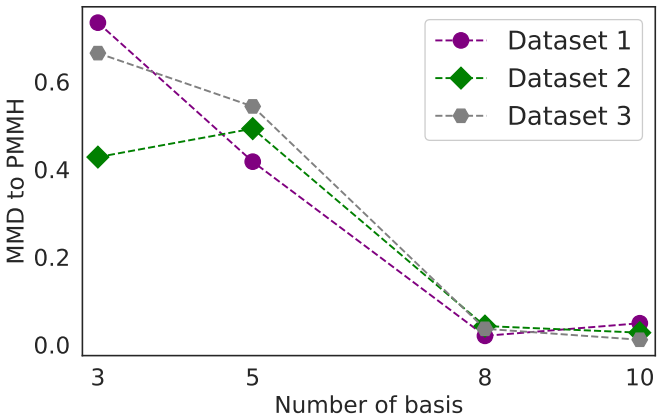

Figure 3: We compare the effect of increasing the number of expansion terms N on the MMD between the PMMH estimate and the estimates obtained from running NUTS, with $N = 3, 5, 8, 10$, for the Lotka-Volterra model. All estimates used 1000 samples from the posterior.

C Plots of the latent diffusion for Lotka-Volterra

Here we evaluate the posterior predictive distribution of the latent diffusion $\hat{\boldsymbol{X}}^*(\boldsymbol{\theta})|_{\boldsymbol{\theta},\mathbf{Z},\boldsymbol{x}_0 \sim p(\boldsymbol{\theta},\mathbf{Z},\boldsymbol{x}_0|\boldsymbol{y})}$, using the **SA-ODE** approximation, on a finer grid t^* than the observations. The posterior predictive distribution is evaluated pointwise using samples obtained from running NUTS or drawn from the VI approximation. We used the posterior distributions estimated using the simulated dataset that was used in section 8.1. The mean and the 95% credible intervals of $\hat{\boldsymbol{X}}^*$ are shown in Figure 4.

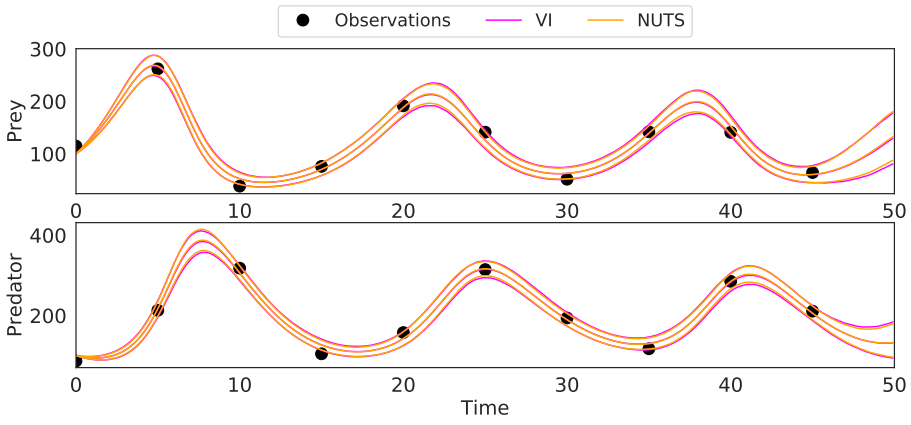

Figure 4: Mean and 95% credible intervals of the posterior predictive distribution of the latent diffusion $\hat{\boldsymbol{X}}^*$, for the Lotka-Volterra model.

Table 3: We summarise the **mean** $\pm$ **standard deviation** of the posterior distribution of each parameter for the The Stochastic Lotka-Volterra model. **VI** and **NUTS** are using the **SA-ODE** model.

THE STOCHASTIC LOTKA-VOLTERRA MODEL: DATASET 2				
$\boldsymbol{c}$	TRUE VALUE	**PMMH**	**VI**	**NUTS**
c_1	0.5	0.5194 ± 0.0273	0.5026 ± 0.0205	0.5093 ± 0.0224
$100 \times c_2$	0.25	0.2557 ± 0.0122	0.2477 ± 0.0092	0.2494 ± 0.0095
c_3	0.3	0.2937 ± 0.0163	0.2883 ± 0.0125	0.2906 ± 0.0132
THE STOCHASTIC LOTKA-VOLTERRA MODEL: DATASET 3				
$\boldsymbol{c}$	TRUE VALUE	**PMMH**	**VI**	**NUTS**
c_1	0.5	0.5172 ± 0.0213	0.4970 ± 0.0159	0.5210 ± 0.0204
$100 \times c_2$	0.25	0.2540 ± 0.0080	0.2432 ± 0.0067	0.2514 ± 0.0076
c_3	0.3	0.3140 ± 0.0117	0.3048 ± 0.0101	0.3130 ± 0.0111

C.1 Additional results for Lotka-Volterra

To benchmark the methods with this model we used simulated data for which the corresponding estimates were summarised in Table 1. In addition to this dataset we generated two more datasets using two new realisation of the artificial noise corruption. The results for these additional datasets are summarised in Table 3. All the algorithmic and model specific settings were kept the same as was used in section 8.1.

C.2 Model fit plot for the SIR model

We evaluated the posterior predictive distribution $p(\boldsymbol{y}^*|\boldsymbol{y})$ on a finer time grid t^*, using the **SA-ODE**. Samples of the posterior predictive distribution were evaluated pointwise using the posterior estimates obtained from running NUTS and VI. Figure 5 summarises the mean and 95% credible intervals of $p(\boldsymbol{y}^*|\boldsymbol{y})$.

D BNAF as approximating distribution

In BNAF each dimension of the transformed parameter $\boldsymbol{\xi}$ (see section 6.1), as a bijection from a standard Normal variable, is modelled using a neural network that (autoregressively) takes the preceding dimensions as inputs. Such bijections can be chained together (applied repeatedly) to form a flow. The dense layers of these networks have weight matrices constructed as a block-diagonal

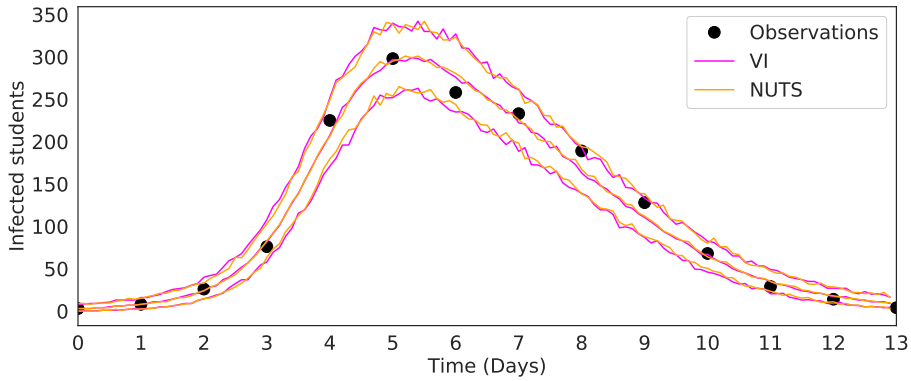

Figure 5: Mean and 95% credible intervals of the posterior predictive distribution for the SIR model.

matrix with the diagonal elements being $a \times b$ matrices themselves, for some chosen $a, b > 1$. We set $a, b = 8$. We used the validation data to choose between $a, b = 5, 8, 10$. $a, b = 8$ produced the best prediction errors. Furthermore, we set the number of flows to 1.

D.1 Architecture of the BnSA-ODE

We used a neural network with a single hidden layer to model the drift, diffusion and the decoder function. The architecture is shown in Figure 6. We used the validation data to experiment with $5, 10, 15, 25, 50$ hidden units. We obtained best prediction errors on the future frames using 10 units.

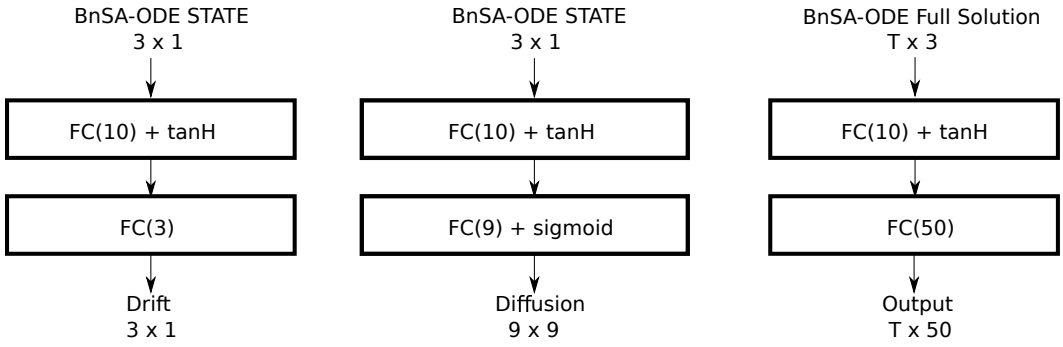

Figure 6: The **BnSA-ODE** architecture for the drift, diffusion and decoder networks that produced the lowest average validation error. We set the number of latent states $K = 3$. Note that by T here we denote the number of time points for training/validation/test data.