Black-Box Inference for Non-Linear Latent Force Models

Wil O. C. Ward[1*] Tom Ryder[23] Dennis Prangle[3] Mauricio A. Álvarez[1]

Abstract

Latent force models are systems whereby there is a mechanistic model describing the dynamics of the system state, with some unknown forcing term that is approximated with a Gaussian process. If such dynamics are non-linear, it can be difficult to estimate the posterior state and forcing term jointly, particularly when there are system parameters that also need estimating. This paper uses black-box variational inference to jointly estimate the posterior, designing a multivariate extension to local inverse autoregressive flows as a flexible approximator of the system. We compare estimates on systems where the posterior is known, demonstrating the effectiveness of the approximation, and apply to problems with non-linear dynamics, multi-output systems and models with non-Gaussian likelihoods.

1 INTRODUCTION

Latent force models are a class of models that can be used to combine known mechanistic systems with non-parametric representations of unknown forces. Consider the example system, defined as a differential equation,

$$\alpha_0 x(t) + \alpha_1 \frac{\mathrm{d}}{\mathrm{d}t} x(t) + \frac{\mathrm{d}^2}{\mathrm{d}t} x(t) = u(t), \tag{1}$$

which describes a forced mass-spring-damper system. In practice, the unknown forcing term, $u(t)$, may need to be estimated given only observations of the system state, $x(t)$. It may be that parameters α_0 and α_1 are also unknown, and need to be simultaneously estimated. Parametric forms for $u(t)$ may be used to estimate the state; however, placing a Gaussian process prior over $u(t)$ gives rise to the latent force model (Alvarez et al., 2013). The resulting model can be transformed into a regression problem, and solved a probabilistic solution to $x(t)$ and $u(t)$ can be inferred. An alternaive inference scheme using Kalman filtering and Rauch-Tung-Streibel smoothing was introduced by Hartikainen and Särkkä (2010), which infers the state sequentially. Latent force models have been shown as effective ways to approximate latent forcing terms, constraining the solution with the defined mechanistic dynamics of the system.

Solutions can only be inferred exactly in the case that the underlying differential equaton is linear. Alvarez et al. (2013) use linearisation of terms in non-linear models, which greatly simplifies the mechanistic aspect of the model, while Hartikainen et al. (2012) use non-linear filtering approaches for approximate inference. In the latter case, the results are effective for known parameters, but the sequential nature of the inference scheme leads to challenges in joint parameter estimation; such challenges are the inspiration for the work by Durrande et al. (2019). Problems where the dynamics are partially known, but where one or more terms are non-linear, for example if α_1 in

*Corresponding author [e: w.ward@sheffield.ac.uk]. [1]Department of Computer Science, The University of Sheffield, UK; [2]School of Computing, Newcastle University, UK; [3]School of Mathematics, Statistics and Physics, Newcastle University, UK

(1) were dependent on x or u, occur in many areas, inclduing biomechanics (Barenco et al., 2006); population modelling (Liu, 2010), structural health monitoring (Worden et al., 2018) and control systems (Conte et al., 1999). In the case where parameters are also unknown, sequential methods can struggle to perform joint inference effectively.

In this work, we seek to build a method that can jointly infer a system with particually known dynamics and some unknown forcing term; and model parameters, given noisy observations of only part of the model. We build a flexible approach using Bayesian variational inference to optimise a surrogate estimation of our posterior state. As in Ryder et al. (2018a), we use black-box variational inference and construct a neural network-based representation of the joint solution. To give our solution flexibility, we construct the variational distribution using normalising flows; designing a new architecture designed to model multivariate systems where high proportions of the state is unobserved.

The main contributions of this paper include building a simulation-based variational estimation for non-linear latent force models, using a sigma point method to propagate uncertainty in the loss term. We integrate Bayesian parameter estimation into the inference method, which includes marginal approximations of Gaussian process hyperparameters. The multivariate extension of the normalising flow is a novel contribution, designed to encode dependencies between latent dimensions based on the Markov properties of the underlying model. The approach is applied to a Gaussian process regression to demonstrate the ability of the normalising flow to approximate samples and apply model criticism, comparing the estimate with the known posterior to quantify the effectiveness of the approximation. We then apply the approach to simulated and real non-linear forced models to demonstrate the proposed approach for jointly estimating state, forcing term and parameters, and show how this can be extended to multivariate outputs. Further, we demonstrate that the method can be easily extended to problems with non-Gaussian likelihoods.

2 BACKGROUND

This section briefly reviews the related background of this work: discretising moments for Gaussian filtering of non-linear stochastic differential equations; and normalising flows in the context of this work.

2.1 Continuous-Time Filtering

Given a (non-linear) stochastic differential equation, describing some state variable, $x(t)$ driven by dynamics with coefficients $\alpha_i(x,t)$ and white noise, $w(t)$,

$$\alpha_0(x,t)x(t) + \sum_{i=1}^{n} \alpha_i(x,t)\frac{\mathrm{d}^i}{\mathrm{d}t^i}x(t) = w(t), \tag{2}$$

we desire to regress the state on some observations, often noisly observed, e.g. $y \,|\, x \sim \mathcal{N}(x, \sigma^2)$. One such method is to use Bayesian filtering to calculate the posterior $p(\boldsymbol{x} \,|\, \mathbf{y})$ by forward propagating state-estimates and updating the approximation with observations.

Kalman-Bucy Filter The Kalman filter is a widely recognised approach to solving state-space models conditioned on a set of observations (Särkkä, 2013). The Kalman filter relies on a discrete-time update, so the SDE must be discretised. If the system is linear and time-invariant, this can be performed exactly and the underlying state-space model is defined $\boldsymbol{x}_{k+1} = \exp((t_{k+1}-t_k)\mathbf{D})\boldsymbol{x}_k + \mathbf{Q}\boldsymbol{\varepsilon}_k$, where $\mathbf{D}$ is the companion matrix representing the n-order system as a first-order SDE, $\mathbf{Q}$ is derived

from the steady-state covariance of the system, and $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Inference can be performed using standard Kalman filtering techniques (Särkkä, 2013).

State-Space Gaussian Processes The state-space model interpretation of a Gaussian process (GP) with finitely-differentiable covariance treats the regression as a continuous-time SDE that can be described in the form of (2). The coefficients of the SDE, α_i, are scalar and derived from the covariance function of a GP prior.

Given that the SDE representation of the Gaussian process is linear and time-invariant, due to the constant coefficients, it can be solved in the forward-direction exactly using the Kalman-Bucy filter. To calculate the full posterior given the data, a backwards pass using, e.g. the Rauch-Tung-Streibel smoother can be performed (Hartikainen and Särkkä, 2010).

The state-space representation of Gaussian processes also allows latent force models to be expressed as a joint companion system, combining the underlying system dynamics with the dynamics of the Gaussian process placed over the prior (Hartikainen et al., 2012; Sarkka et al., 2018).

Unscented Kalman-Bucy Filter Where the state to be inferred is subject to *non-linear* dynamics, such that α_i in (2) are dependent on x, the posterior cannot be calculated exactly. There are, however, several approaches to state-space modelling of non-linear systems, including the extended Kalman filter, and sequential Monte Carlo filters (Särkkä, 2013). The unscented Kalman filter (UKF) is another such approach, using a set of so-called *sigma points* to characterise the moments of the state estimate. Sigma points are individually propagated through the dynamics and can be combined by means of weighted sums to obtain approximations of the mean and covariance (Julier and Uhlmann, 1997).

A continuous-time extension of the UKF was introduced in Sarkka (2007), defining continuous-time dynamics for predicting the moments, $\boldsymbol{m}(t)$ and $\boldsymbol{P}(t)$. For a given system, with dynamics $\boldsymbol{D}(\boldsymbol{x}, t)$, diffusion term, $\boldsymbol{\Sigma}(t)$, and sigma points $\boldsymbol{\chi}$, the moments such that $\boldsymbol{x}(t) \sim \mathcal{N}(\boldsymbol{m}(t), \boldsymbol{P}(t))$ are defined

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{m}(t) &= \boldsymbol{D}(\boldsymbol{\chi}, t)\boldsymbol{w}^{(m)} \\ \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{P}(t) &= \boldsymbol{\chi}(t)\mathbf{W}\boldsymbol{D}(\boldsymbol{\chi}, t)^\top + \boldsymbol{D}(\boldsymbol{\chi}, t)\mathbf{W}\boldsymbol{\chi}(t)^\top + \boldsymbol{\Sigma}(t). \end{aligned} \tag{3}$$

Weights $\boldsymbol{w}^{(m)}$ and $\mathbf{W}$ derive from the unscented transform and are defined in the construction of $\boldsymbol{\chi}$. The moments can be forward-solved using an ODE solver, such as a Runge-Kutta scheme (Sarkka, 2007). At observation times, the update-step of the discrete-time UKF can be used to update the estimate of $\boldsymbol{x}$, which can be used as an initial value for further prediction.

2.2 Autoregressive Flows

Normalising flows can be used to represent a probability distribution $q(\boldsymbol{x})$ as a differential transformation of some base density, e.g. $\boldsymbol{g} : \boldsymbol{z} \mapsto \boldsymbol{x}$, $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ (Rezende and Mohamed, 2015). The mapping, $\boldsymbol{g}$ must be invertible and differentiable, such that

$$q(\boldsymbol{x}) = p(\boldsymbol{g}^{-1}(\boldsymbol{x}))\big|\boldsymbol{G}_{-1}(\boldsymbol{x})\big|,$$

where $\boldsymbol{G}_{-1}$ is the Jacobian of $\boldsymbol{g}^{-1}$.

The tractability of $|\boldsymbol{G}_{-1}|$ is a restriction on the choice of mapping; enforcing an autoregressive expression, $x_i = \boldsymbol{g}(z_1, \ldots, z_i)$ results in a Jacobian that is triangular and therefore the determinant

is the product of its diagonal elements. This form of $\boldsymbol{g}$ gives rise to the concept of autoregressive flows (Kingma et al., 2016).

Inverse Autoregressive Flows Kingma et al. (2016) describe the mapping by reparametrising its inverse, transforming variables sampled from the base density by functions that represent the dependencies between dimensions. An inverse autoregressive flow (IAF) can be defined in terms of two transformations, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$:

$$x_i = \boldsymbol{\sigma}(z_1, \ldots, z_i) \cdot z_i + \boldsymbol{\mu}(z_1, \ldots, z_i) \tag{4}$$

The log probability is therefore

$$\log q(\boldsymbol{x}) = \log p(\boldsymbol{z}) - \sum \log \sigma_i, \tag{5}$$

where $\sigma_i = \boldsymbol{\sigma}(z_1, \ldots, z_i)$.

The shift and scale functions, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are often defined as outputs of a neural network encoding the autoregressive requirements of the system. For example, in Kingma et al. (2016) the authors make use of convolutional layers, such as ResNet (He et al., 2016).

Local IAFs An issue with IAFs as described is that they can become computationally expensive to compute for high dimensional $\boldsymbol{x}$ due to the large number of inputs to the shift and scale functions. Ryder et al. (2018b) introduce a local version of the IAF for approximating state-space models, in which the shift and scale depend only on the immediately preceding r entries of the vector, termed the receptive field: $x_i = f(z_{i-r}, \ldots, z_i)$. This approach is similar to that used in PixelCNN and WaveNet (Van den Oord et al., 2016; van den Oord et al., 2016), using causal convolutions with restricted kernel width.

3 VARIATIONAL INFERENCE FOR NON-LINEAR LATENT FORCE MODELS

We consider the approximate inference of the joint state posterior, $p(\boldsymbol{x}(t), u(t), \boldsymbol{\theta} \,|\, \mathbf{y})$ of a non-linear latent force model of the form

$$\alpha_0(\boldsymbol{x}, u, t; \boldsymbol{\theta})\boldsymbol{x}(t) + \sum_{i=1}^{n} \alpha_i(\boldsymbol{x}, u, t; \boldsymbol{\theta}) \frac{\mathrm{d}^i}{\mathrm{d}t^i} \boldsymbol{x}(t) = u(t), \tag{6}$$

where $u(t) \sim \mathcal{GP}(0, k(t, t'))$ is the prior placed over unknown force, and $\mathbf{y}$ are observations at times τ_j, $j = 1, \ldots, t$, such that $\boldsymbol{y}_j \sim \pi(\boldsymbol{h}(\boldsymbol{x}(\tau_j); \boldsymbol{\theta}))$, some likelihood conditional on an emission model $\boldsymbol{h}(\boldsymbol{x})$.

Given its intractability, we use variational Bayes to approximate the conditional posterior. We define a joint state vector $\boldsymbol{f}(t) = [\boldsymbol{x}(t), \mathrm{d}\boldsymbol{x}/\mathrm{d}t, \ldots, u(t_k), \mathrm{d}u/\mathrm{d}t, \ldots]^\top$ and construct a first-order companion SDE for (6) such that it can be written

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}(t) = \boldsymbol{D}(\boldsymbol{f}, t; \boldsymbol{\theta}) + \mathbf{L}\varsigma w(t), \tag{7}$$

where $\boldsymbol{D}$ is the companion form dynamic, $\mathbf{L}$ is a column vector of zeros in all but the final row, which equals 1; and $w(t)$ is a unit white noise process, and ς^2 is the variance as derived from the state-space form of the GP prior.

For some finite-time mesh, $t_0, \ldots, t_T$, and observation times $\tau_1 \ldots, \tau_N$, which for simplicity are contained within the mesh, the joint posterior

$$p(\boldsymbol{x}_{0:T}, u_{0:T}, \boldsymbol{\theta} \,|\, \boldsymbol{y}) \propto \\ p(\boldsymbol{\theta}) p(\boldsymbol{f}_0 \,|\, \boldsymbol{\theta}) \prod_{k=0}^{T-1} p(\boldsymbol{f}_{k+1} \,|\, \boldsymbol{f}_k, \boldsymbol{\theta}) \prod_{j=1}^{N} p(\boldsymbol{y}_j \,|\, \boldsymbol{f}(\tau_j), \boldsymbol{\theta}), \tag{8}$$

where subscripts denote discrete-time evaulations, e.g. $\boldsymbol{f}_k \triangleq \boldsymbol{f}(t_k)$.

3.1 Variational Bounds

Given the intractibility of (8), we construct a variational approximation of the posterior to jointly estimate the system state and parameters, in the form of $q(\boldsymbol{f}, \boldsymbol{\theta}) = q(\boldsymbol{\theta}) q(\boldsymbol{f} \,|\, \boldsymbol{\theta})$. To obtain the optimal approximation, we seek to find $q^* \in \mathcal{Q}$ to minimise the KL-divergence between our estimate and the posterior:

$$\mathcal{KL}[q^* \,\|\, p] = \mathbb{E}_{\boldsymbol{f}, \boldsymbol{\theta} \sim q}[\log q(\boldsymbol{f}, \boldsymbol{\theta}) - \log p(\boldsymbol{x}, u, \boldsymbol{\theta} \,|\, \mathbf{y})] \,. \tag{9}$$

Noting that $p(\boldsymbol{x}, u, \boldsymbol{\theta} \,|\, \mathbf{y}) = p(\boldsymbol{x}, u, \boldsymbol{\theta}, \mathbf{y}) / p(\mathbf{y})$ and that the evidence $p(\mathbf{y})$ does not depend on q, is equivalent to maximising

$$\mathcal{L}(q) = \mathbb{E}_{\boldsymbol{f}, \boldsymbol{\theta} \sim q}[p(\boldsymbol{x}, u, \boldsymbol{\theta}, \mathbf{y}) - \log q(\boldsymbol{f}, \boldsymbol{\theta})] \,. \tag{10}$$

To allow straightforward unbiased estimation of (10), we follow the example of Kingma and Ba (2014), Rezende and Mohamed (2015), and Titsias and Lázaro-Gredilla (2014) by reparameterising q and taking $\boldsymbol{f} = \boldsymbol{m}_f(\boldsymbol{\varepsilon}_f, \boldsymbol{\theta}; \boldsymbol{\phi}_f)$ and $\boldsymbol{\theta} = \boldsymbol{m}_\theta(\boldsymbol{\varepsilon}_\theta; \boldsymbol{\phi}_\theta)$. The functions $\boldsymbol{m}_f$ and $\boldsymbol{m}_\theta$ should be defined such that they are invertible mappings of random variables, $\boldsymbol{\varepsilon}_f, \boldsymbol{\varepsilon}_\theta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and be parameterised by $\boldsymbol{\phi}_f$ and $\boldsymbol{\phi}_\theta$ respectively. Thus, the family of functions, $\boldsymbol{Q}$ representing q, are the functions $\boldsymbol{m}_f$ and $\boldsymbol{m}_\phi$, parameterised by variational parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}_f, \boldsymbol{\phi}_\theta\}$.

An unbiased estimate of $\mathcal{L}(q)$ can be obtained using M Monte Carlo samples,

$$\mathcal{L}(q) \approx \\ \frac{1}{M} \sum_{i=1}^{M} \log \left(p(\boldsymbol{\theta}^{(i)} p(\boldsymbol{f}^{(i)} \,|\, \boldsymbol{\theta}^{(i)}) p(\mathbf{y} \,|\, \boldsymbol{f}^{(i)}, \boldsymbol{\theta}^{(i)}) \right) \\ - \log \left(q(\boldsymbol{\theta}^{(i)}) q(\boldsymbol{f}^{(i)} \,|\, \boldsymbol{\theta}^{(i)}) \right), \tag{11}$$

where $\boldsymbol{f}^{(i)} = \boldsymbol{m}_f(\boldsymbol{\varepsilon}_f^{(i)}, \boldsymbol{\theta}^{(i)}; \boldsymbol{\phi}_f)$ and $\boldsymbol{\theta}^{(i)} = \boldsymbol{m}_\theta(\boldsymbol{\varepsilon}_\theta^{(i)}; \boldsymbol{\phi}_\theta)$ for independent samples $\boldsymbol{\varepsilon}_f^{(i)}$, $\boldsymbol{\varepsilon}_\theta^{(i)}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

To maximise $\mathcal{L}(q)$ with respect to $\boldsymbol{\phi}$, we use a stochastic gradient optimisation algorithm, approximating $\nabla_\phi \mathcal{L}$ with an unbiased estimate of sample gradients, calculated using automatic differentiation of (11) (Ranganath et al., 2014; Ryder et al., 2018a).

3.2 Discretisation

In calculating $\mathcal{L}$, we must approximate the marginal of our joint system state $p(\boldsymbol{f} \,|\, \boldsymbol{\theta}) = \prod p(\boldsymbol{f}_{k+1} \,|\, \boldsymbol{f}_k, \boldsymbol{\theta})$. To do so, we assume that transition is (approximately) normal, and build moments using the prediction dynamics of the continuous-time unscented Kalman filter, described by (3),

$$\boldsymbol{f}_{k+1} \,|\, \boldsymbol{f}_k, \boldsymbol{\theta} \sim \mathcal{N}(\boldsymbol{m}_{k+1}, \boldsymbol{P}_{k+1}). \tag{12}$$

We construct the discrete-time predictions using Euler's method, with step size $\Delta_t = t_{k+1} - t_k$ (Griffiths and Higham, 2010). The update steps for the mean and covariance of the transition density are defined

$$\boldsymbol{m}_{k+1} = \boldsymbol{f}_k + \Delta_t \boldsymbol{D}(\boldsymbol{\chi}_k, t; \boldsymbol{\theta})\boldsymbol{\omega}^{(m)} \tag{13}$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{\Sigma} + \Delta_t \boldsymbol{U}(\boldsymbol{\chi}_k, t; \boldsymbol{\theta}), \tag{14}$$

where

$$\boldsymbol{U}(\boldsymbol{\chi}, t; \boldsymbol{\theta}) = \boldsymbol{\chi}\mathbf{W}\boldsymbol{D}(\boldsymbol{\chi}, t; \boldsymbol{\theta})^\top + \boldsymbol{D}(\boldsymbol{\chi}, t; \boldsymbol{\theta})\mathbf{W}\boldsymbol{\chi}^\top + \boldsymbol{\Sigma}, \tag{15}$$

and $\boldsymbol{\Sigma} = \tilde{\boldsymbol{L}}\boldsymbol{Q}_c\tilde{\boldsymbol{L}}^\top$. $\boldsymbol{Q}_c$ represents the $m \times m$ steady-state covariance of the state-space GP prior of u, and $\tilde{\boldsymbol{L}}$ is a $d \times m$ masking term to map $\boldsymbol{Q}_c$ to $d \times d$, where d is the dimension of the joint state.

Unscented transform sigma points $\boldsymbol{\chi}_k$ and corresponding weight terms $\boldsymbol{\omega}^{(m)}$ and $\boldsymbol{W}$ are defined in matrix-form and built from base density $\mathcal{N}(\boldsymbol{f}_k, \boldsymbol{\Sigma})$ (Sarkka, 2007). Sigma points are defined

$$\boldsymbol{\chi}_k = [\boldsymbol{f}_k \quad \dots \quad \boldsymbol{f}_k] + \begin{bmatrix}\mathbf{0} & \sqrt{(d+\eta)\boldsymbol{\Sigma}} & -\sqrt{(d+\eta)\boldsymbol{\Sigma}}\end{bmatrix}, \tag{16}$$

with weight terms defined

$$\begin{aligned}
\omega_0^{(m)} &= \eta(d+\eta)^{-1} \\
\omega_0^{(c)} &= \eta(d+\eta+1-\alpha_\chi^2+\beta_\chi)^{-1} \\
\omega_i^{(m)} &= \omega_i^{(c)} = (2d+2\eta)^{-1}
\end{aligned} \tag{17}$$

$$\begin{aligned}
\boldsymbol{\omega}_{-1} &= \mathbf{I} - \begin{bmatrix}\boldsymbol{\omega}^{(m)} & \dots & \boldsymbol{\omega}^{(m)}\end{bmatrix} \\
\mathbf{W} &= \boldsymbol{\omega}_{-1}\mathrm{diag}[\boldsymbol{\omega}^{(c)}]\boldsymbol{\omega}_{-1}^\top,
\end{aligned} \tag{18}$$

where $\eta = \alpha_\chi^2(d + \kappa_\chi) - \kappa_\chi$ is the unscented scaling term and α_χ, β_χ, and κ_χ are hyperparameters of the transform (Julier and Uhlmann, 1997).

3.3 Multivariate Masking of Local IAF

As in both Kingma et al. (2016) and Ryder et al. (2018b), we build $q(\boldsymbol{f} \mid \boldsymbol{\theta})$ as a hierarchy of inverse autoregressive flows to create a flexible approximation density. Because we are dealing with multivariate states, we design a novel variant for vector-valued temporal states.

First, the system state is flattened such that each entry corresponds to a single dimension, and the hierarchical flow transformation is constructed such that each layer acts only on a single dimension in the state. Successive layers alternate updates for each dimension, as motivated in Dinh et al. (2016). A receptive field mask is used to enforce locally temporal dependencies, and each flow layer takes in the output of the previous layer, the model parameters, $\boldsymbol{\theta}$, and a local feature vector, $\mathcal{D}$.

A demonstration of the flow dependencies is shown in Figure 1. Formally, the flow hierarchy can be defined for each element of the flattened joint state vector rolled out over time as such:

$$\begin{aligned}
z_i^{(0)} &= \varepsilon_i \\
z_i^{(l)} &= (\delta_i^l \sigma_i + (1 - \delta_i^l)) z_i^{(l-1)} + \delta_i^l \mu_i,
\end{aligned} \tag{19}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\delta_i^l = 1$ if $(i \bmod l = 0)$ else 0 and

$$[\mu_i, \sigma_i] = \textsc{AutoregressiveNN}(\mathbf{r}_i^{(l)} \odot \boldsymbol{z}^{(l-1)}, \boldsymbol{\theta}, \mathcal{D}). \tag{20}$$

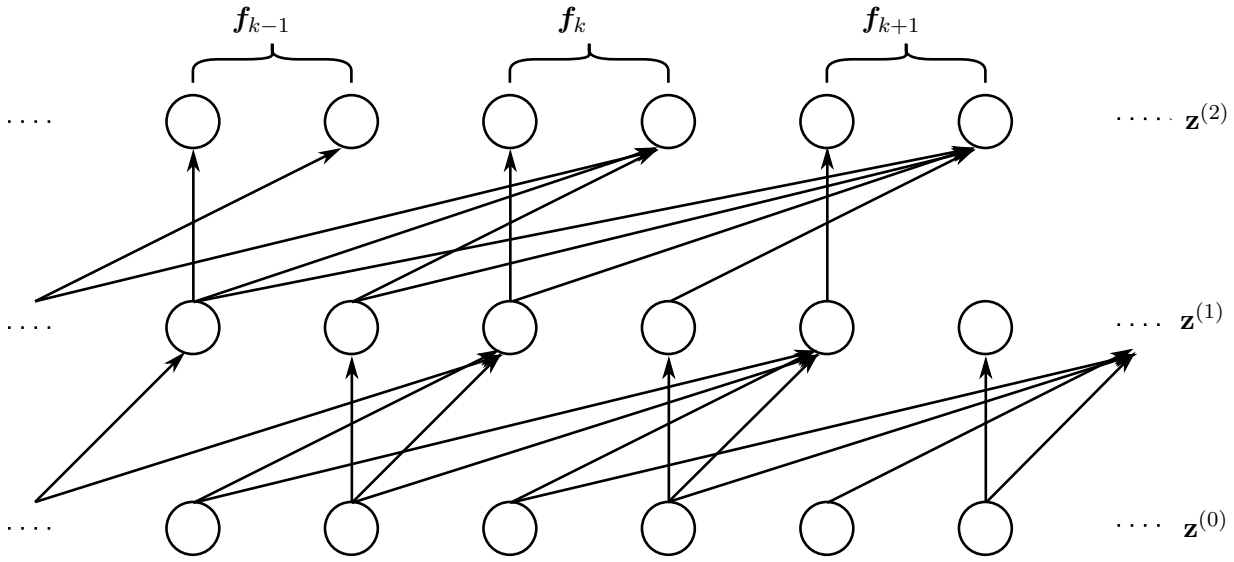

Figure 1: Example architecture of multivariate masking of local inverse autoregressive flows, with state dimension $d = 2$ and receptive field width $r = 2$

The autoregressive neural network here encodes the conditional depencencies of the system. The receptive mask, $\mathbf{r}_i^{(l)}$ is a binary vector that masks the local receptive field. In practise, the receptive mask is enforced using causal 1-D convolutions with kernel width equal to $r \cdot d$, where d is the state dimension and r is the receptive field's width. The autoregressive neural network is constructed from multiple layers of convolutions with batch normalisation between layers. Details of the architecture are provided in Algorithm 1.

The final flow, $z^{(N)}$ is used to generate an estimate for $\boldsymbol{f}$, with $\boldsymbol{f} = \text{flatten}^{-1}(\boldsymbol{h}(\boldsymbol{z}^{(N)}))$, where $\text{flatten}^{-1} : \mathbb{R}^{dT \times 1} \to \mathbb{R}^{d \times T}$ is the inverse transformation to map the flattened vector back to its multivariate form. We also define $\boldsymbol{h}$ to be some optional bijector to enforce constraints on $\boldsymbol{f}$, for example $\boldsymbol{h}(\cdot) \triangleq \log(\exp(\cdot) + 1)$, the softplus operator, can be used to enforce positivity of $\boldsymbol{f}$.

The log probability of the multivariate extension to local IAF is thus

$$\begin{aligned} \log q(\boldsymbol{f}) = \frac{T}{2}(\boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon} + \log 2\pi) + \sum_{i=1}^{T} \sum_{l=1}^{N} \delta_i^l \sigma_i + (1 - \delta_i^l) \\ + \log |\boldsymbol{H}_{-1}(\boldsymbol{f})|, \end{aligned} \tag{21}$$

where $\boldsymbol{H}_{-1}$ is the Jacobian of the inverse of $\boldsymbol{h}$.

Local Features The features for the inverse autoregressive flow represent the additional input data to the model, including parameters and observation data. The flattening of the latent dimension is mirrored in the flattening of the observation data, with latent dimensions receiving 0 as input.

For each element of a given flow, $\boldsymbol{z}_i^{(l)}$, the feature vector, $\mathcal{D}$ consists of: the current discrete-time point, t_k; the time until the next observation, $\tau_j - t_k$, such that $\tau_{j-1} < t_k \leq \tau_j$; the next observation, $\boldsymbol{y}_j$; and a binary mask indicating with value 1 that there is an observation at τ_k *and* that the current corresponding state dimension is not latent.

The feature vector is concatenated with the base sample of the flow before passing to the autoregressive network used to generate shift and scale terms, $\boldsymbol{\mu}^{(i)}$ and $\boldsymbol{\sigma}^{(i)}$. A sample from the

parameter distribution is encoded with a densely-connected multilayer perceptron and the output is added to the first layer of the autoregressive network.

3.4 Joint Parameter Estimation

For the estimation of model parameters, we use a separate variational distribution $q(\boldsymbol{\theta})$, to be optimised as part of the maximisation of (11). A simple family of distributions for q is the family of mean-field Gaussian approximations. Here, we sample each parameter from an independent multivariate Gaussian, parameterised by $\boldsymbol{\phi}_\theta = \{\boldsymbol{\mu}_\theta, \boldsymbol{s}_\theta\}$. In this case, $\boldsymbol{m}_\theta(\boldsymbol{\varepsilon}_\theta^{(i)}; \boldsymbol{\phi}_\theta)$ is defined such that $\boldsymbol{\theta}^{(i)} = \text{diag}[\boldsymbol{s}_\theta]\boldsymbol{\varepsilon}_\theta^{(i)} + \boldsymbol{\mu}_\theta$. Constraints on elements of $\boldsymbol{\theta}$ can also be applied, for example by placing the variational distribution of $\log \theta_i$ if it is constrained to be positive.

For the models presented in this paper, empical results demonstrated that the mean-field approach is sufficient for parameter estimation, particularly when there is known independence, such as between parameters of the input GP and of the system dynamics. However, such an approach can often be a poor choice for more complex dependencies (Blei et al., 2017), so alternative approaches may be used, e.g. masked autoregressive density estimation (Germain et al., 2015).

4 RELATED WORK

Physically-inspired inference of unknown systems with Gaussian processes can be considered in the process convolution interpretation of multi-output GPs (Álvarez and Lawrence, 2011), which consider the system as an integral problem with a shared latent GP describing the dependencies. This approach has been used for latent force models (Alvarez et al., 2013), and introduced to non-linear dynamics in Lawrence et al. (2007) and Titsias and Lawrence (2009), more recently being generalised to non-linear systems using a series approximation (Álvarez et al., 2019). The interpretation of latent force models as state-space models has been applied to non-linear problems in Hartikainen and Särkkä (2010) using the unscented Kalman filter, but does not apply joint parameter estimation. Incorporating non-Gaussian likelihoods in the state-space approach to GP regression has been discussed in Nickisch et al. (2018).

Approximation methods for ODEs and SDEs with fully unknown dynamics that use GPs to approximate some part of the system include gradient-matching GP regression fits, as in Wenk et al. (2018), or approximating the phase and diffusion matrix of non-linear oscillators with GPs given only observations (Heinonen et al., 2018; Yildiz et al., 2018). Recent works investigating parameter estimation in stochastic systems with known dynamics includes approaches using variationa; inference (Ryder et al., 2018a; Bińkowski et al., 2017), and MCMC (Abbati et al., 2019).

Algorithm 1 l^{th} AUTOREGRESSIVENN$(\boldsymbol{z}^{(l-1)}, \boldsymbol{\theta}, \mathcal{D})$

$\xi^{(0a)} \leftarrow \text{CONV1D}(\boldsymbol{z}^{(l-1)}, \mathcal{D})$
$\xi^{(0b)} \leftarrow \text{DENSE}(\boldsymbol{\theta})$
$\xi^{(1)} \leftarrow \text{ELU}(\xi^{(0a)} + \xi^{(0b)})$
for $i = 2 \dots n_\ell$ **do**
  $\xi^{(i)} \leftarrow \text{BATCHNORM}(\text{CONV1D}(\text{ELU}(\xi^{(i-1)})))$
end for
$[\boldsymbol{\mu}, \boldsymbol{s}] \leftarrow \text{CONV1D}(\xi^{(n_\ell)})$
$\boldsymbol{\sigma} \leftarrow \text{SOFTPLUS}(\boldsymbol{s})$
return $[\boldsymbol{\mu}, \boldsymbol{\sigma}]$

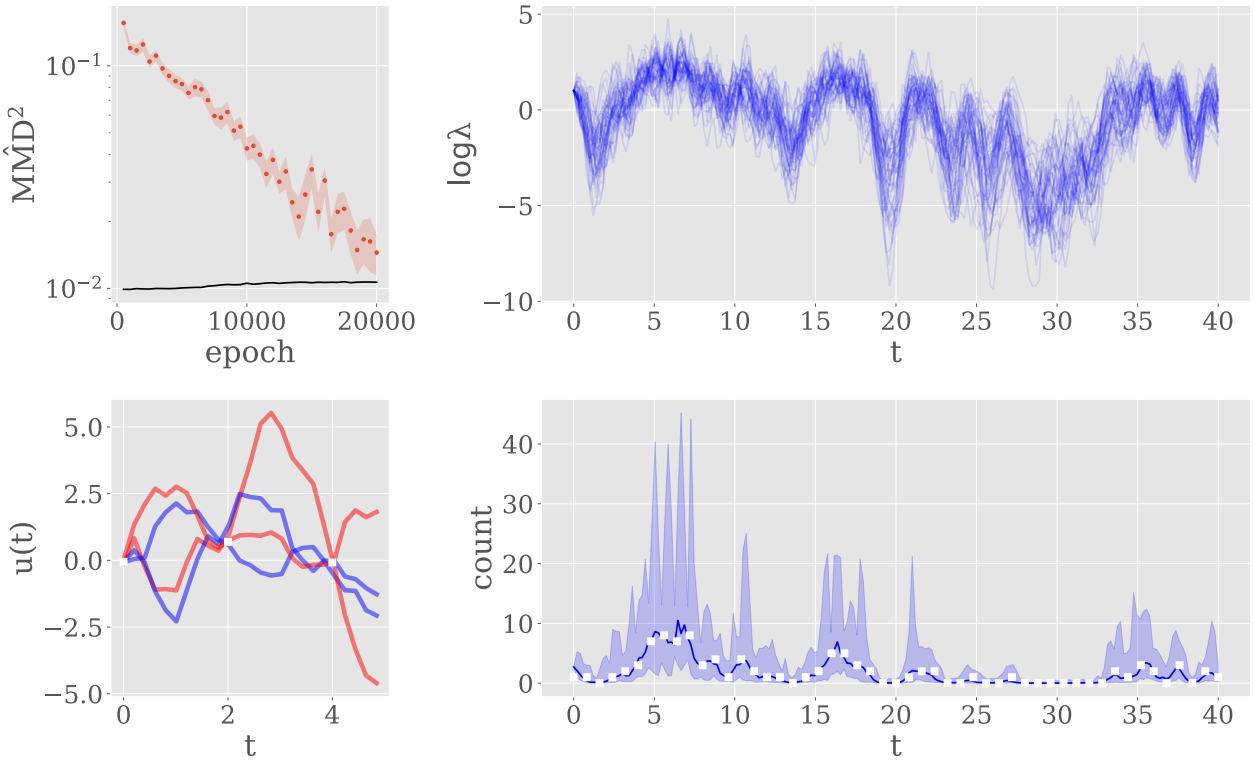

Figure 2: Maximum mean discrepancy (MMD^2) scores comparing samples from the variational approximation of GP with Matérn-3/2 covariance with samples the true posterior, plotted against training epoch [top left]. The black line indicates the threshold under which the null hypothesis can be accepted with 95% confidence. Two samples from the true posterior (red) and variational approximation (blue) conditioned on some observations (white) [bottom left]. Samples form a variational approximation of a latent GP conditioned on count data [top right], with predictive density plotted with 95% confidence intervals (shaded)

This paper addresses the combined problem of partially known dynamics with unknown parameters using autoregressive neural networks. Similar simulation-based inference methods for sequential data include the sequential neural likelihood (Papamakarios et al., 2019), which presents a likelihood-free inference model with masked autoregressive flows (Papamakarios et al., 2017). A related approach for low-dimensional state-space models was introduced in Ryder et al. (2018b).

5 MODEL CRITICISM

A special case of the model defined in (6) is an SDE with the deterministic dynamics of a Matérn GP, and the forcing term has a GP prior with white-noise covariance. The resulting system would be equivalent to a state-space GP. Thus, we can infer the posterior exactly for evaluation of the proposed approximate inference approach.

Matérn covariances The Matérn family of covariances are finitely differentiable and, as such, can be represented exactly as SDEs (Särkkä and Solin, 2019). For a GP with half-integer Matérn

covariance, e.g. Matérn-3/2, the dynamics can be easily represented as a stochastic LFM:

$$\lambda^n x(t) + \sum_{i=1}^{n} \binom{n}{i} \lambda^{n-i} \frac{\mathrm{d}^i}{\mathrm{d}t^i} x(t) = u(t), \tag{22}$$

where $u(t) \sim \mathcal{GP}(0, v(2\lambda)^{2n-1}(n!)^2/(2n-2)!\delta_{tt'})$, with parameters v, the variance; and $\lambda = \ell^{-1}\sqrt{2n-2}$, where ℓ is the so-called length-scale.

Maximum Mean Discrepancy We use a kernel-based two-sample test (Gretton et al., 2012) to compare samples from the variational approximation with samples from the true posterior. Mapping the corresponding samples to a reproducing kernel Hilbert space using a Gaussian kernel, we apply the two-sample test using maximum mean discrepancy (MMD) as a similarity metric between the approximate and true posterior.

The MMD^2 value for the approximation of Matérn-3/2 GP fit is shown against the number of training epochs in Figure 2. The figure shows that as training increases, the quality of approximation increases. The black solid line along the lower part of the plot indicates the threshold at which the null hypothesis, that the two distributions are the same, can be rejected with 95% confidence. The trend observed thus indicates that with training the approximation is tending to being statistically indistinguishable in such a test. For visual reference, two samples from the true posterior and approximation, further demonstrating their similarity.

6 EXPERIMENTAL RESULTS

This section details experiments for problems with intractible posteriors, demonstrating use-cases and the flexibility of the proposed approach.

Non-Gaussian Likelihoods We apply the regression problem representing latent GP with Matérn-3/2 covariance as in the previous section, but condition on a simulated set of count data to demonstrate that the approach can be easily extended to problems with non-Gaussian likelihoods. The regression problem is defined such that $\boldsymbol{f}(t)$ is the joint state that has dynamics defined in (22) with $n = 2$

$$p(y_j \mid \boldsymbol{f}(\tau_j)) = \mathcal{P}ois(y_j \mid \exp(f_1(\tau_j)).$$

Samples from the variational distribution are shown in Figure 2, along with the approximated predictive density which shows that the approximation can capture the main features and uncertainty of the system. The negative log predictive density (NLPD) for the fit using the proposed approach is -0.12218, versus -0.16576 for a GP fit with a Laplace approximation (Rasmussen and Williams, 2006).

Toy problem In this example, we demonstrate inference of a latent input function on a non-linear ODE using the proposed method. We consider a toy example with sinusoidal dynamics of an observable state, and place a Matérn-3/2 GP prior over the unknown input:

$$\frac{\mathrm{d}}{\mathrm{d}t} x(t) = -\frac{2}{3}\sin(\omega x(t)) + u(t) \tag{23}$$

Parameters from both the latent state dynamics and input covariance, $\boldsymbol{\theta} = \{\omega, v, \lambda\}$ are jointly inferred with a mean-field variational distribution $q(\boldsymbol{\theta})$. The approximate posterior was conditioned on observations generated from a sample solution to (23), with additive Gaussian noise.

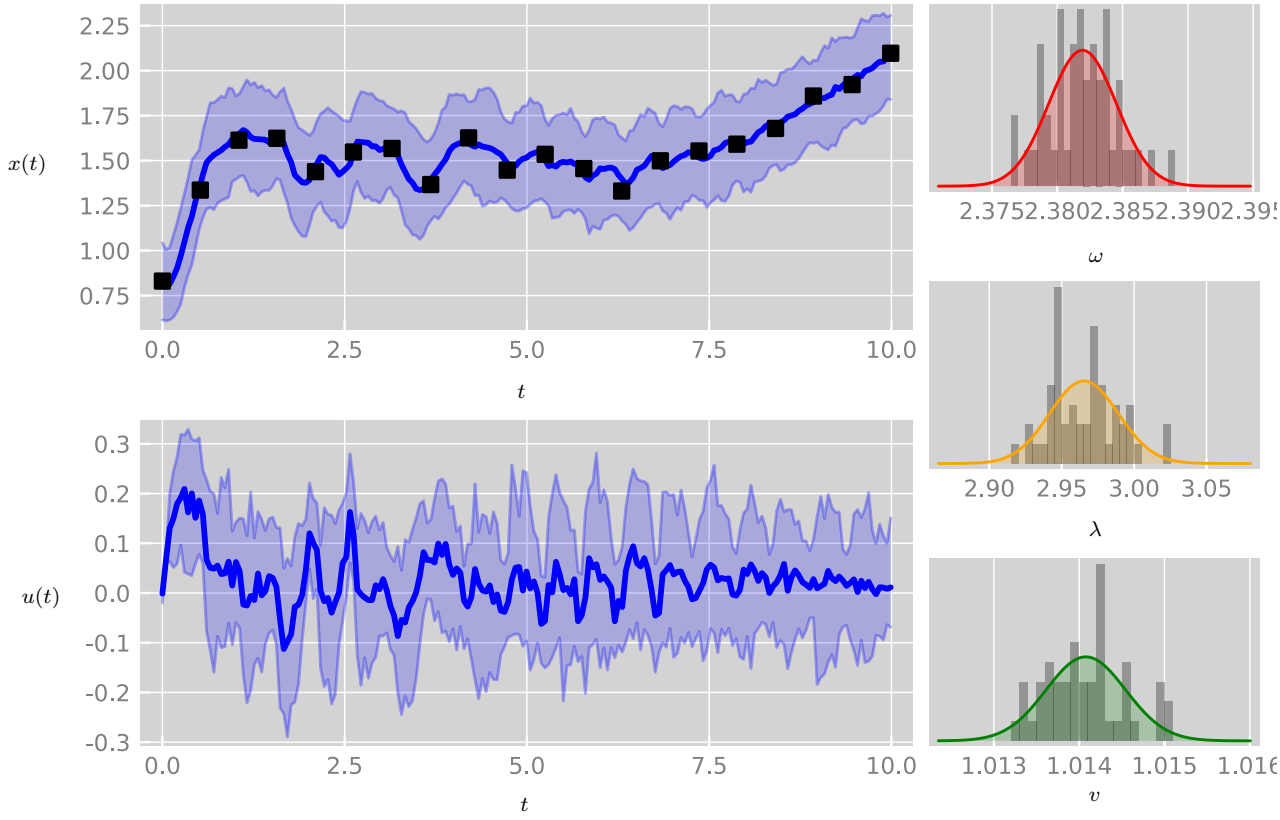

Figure 3: Latent state, $x(t)$, and forcing term, $u(t)$, estimates using the proposed approach, conditioned on noisy observations of $x(t)$ indicated by black squares. The shaded area indicates the central 95% sample quantile about the mean (solid blue). Plots on the right indicate the estimated marginals of the parameters and binned samples

Figure 3 shows the inferred posterior state and latent forcing term inferred using the variational approach. The marginals for the inferred parameters are also shown. We observe that the undelying forcing term has largest effect in the early parts of the model, $t < 5.0$, which corresponds to deviation from the fixed dynamics observed in $\mathbf{y}$.

Gene Expression In the final experiment, we consider a multi-output system using real-world data. We consider the transcriptional regulation model in Barenco et al. (2006): an ODE describing the dynamics of target gene expression that are regulated by an unobserved transcription factor, $u(t)$. For each gene in the dataset, $x_d(t)$, the dynamics can be described as

$$\frac{\mathrm{d}}{\mathrm{d}t}x_d(t) = a_d - b_d x_d(t) + s_d \frac{u(t)}{\gamma_d + u(t)}. \tag{24}$$

We assume the gene expressions are observed with some additive noise, $\sigma_y^2 = 0.25$ and place a GP prior over $\log u(t)$, as in Titsias and Lawrence (2009). Model parameters for each output include the basal transcription rate, a_d, decay rate, b_d, and sensitivity, s_d, which are all unknown. γ_d is the Michaelis constant defined for each specific gene. Additional shared parameters are those of the GP covariance function, characterising $u(t)$.

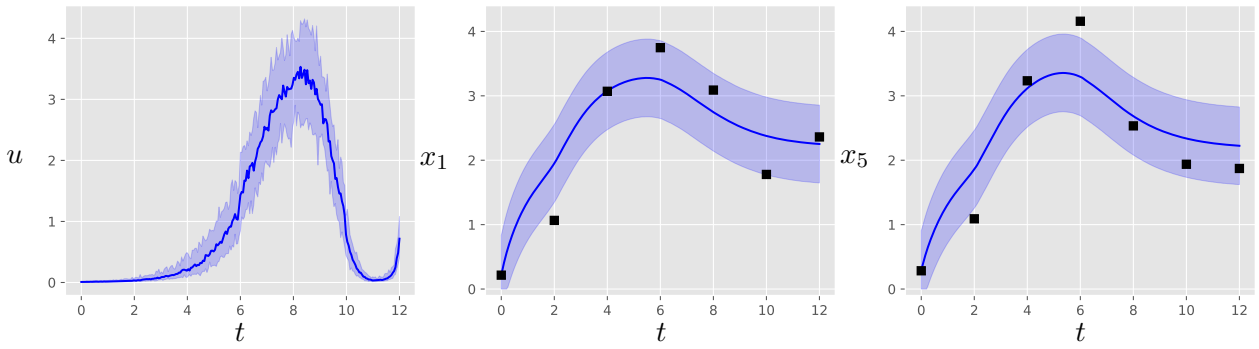

Figure 4: Inferered transcription factor concentration (left) and predicted gene expression for DDB2 and p26 sesn1 (right). Black squares indicate measured gene expression, and shaded region represented the 95^{th} sample quantile about the mean

We perform inference of the gene expression problem for 5 observed genes: DDB2, BIK, TNFRSF10b, CIp1/p21, and p26 sesn1. A Matérn-3/2 GP prior is placed over the log transcription factor term, $\log u(t)$, and a mean-field approach is used to estimate parameters for each gene $\{a_d, b_d, s_d \,|\, d = 1, \ldots, 5\}$, and GP paremters, λ and v. A softplus bijector was placed on the output paths representing x_d to enforce positivity.

Plots showing the inferred (unscaled) transcription factor concentration, $u(t)$, and two of the gene expression states: DDB2 and p26 sesn. We observe that the estimates of observable states, x_d, capture the observations and general form.

We observe that the proposed approach can easily extend to multidimensional states, and represent multi-output problems. In practice, this is similar to process convolution representation of such problems (Alvarez et al., 2013), however we are able to perform inference over non-linear models, while jointly estimating system parameters.

7 CONCLUSIONS

We present an approach for jointly inferring the parameters and state of non-linear ODEs with unknown forcing terms using a simulation-based autoregressive variational approximation. The approach can effectively simulate Gaussian process samples and infer both observed and latent states constrained to partially known dynamics systems. The method could be extended to non-linear SDEs with unknown input terms using the same proposed approach. We further demonstrate that the model can represent multi-output systems and models with non-Gaussian likelihoods.

There are some limitations to the approach, such as a tendancy to over-confidence; this can be observed in Figure 4, where we observe narrow error bars in the latent GP for lower values of t. This might be fixing certain parameters of the GP, such as the variance, or deriving some term to penalise deviation from the prior more strictly.

We have proposed a new approach to approximating non-linear latent force models as a filtering problem, using a new multivariate masking architecture of local inverse autoregressive flows to handle dependencies between observable and latent state dimensions. The joint model can effectively learn parameters in such problems with batch gradient descent; a challenge for sequential approaches such as Kalman filtering or sequential Monte Carlo due to the need to roll-out gradients through time. Further, the proposed inference method is scalable to more complex variational distributions over the parameter space; and we can in principle extend the state-dimension arbitrarily, with any

number of latent and observed terms.

Acknowledgements WOCW and MAA have been financed by the Engineering and Physical Research Council (EPSRC) Research Project EP/N014162/1. MAA has also been financed by the EPSRC Research Project EP/R034303/1. TR is supprted by the EPSRC Center for Doctoral Trainig in Cloud Computing for Big Data (EP/L015358/1).

References

G. Abbati, P. Wenk, S. Bauer, M. A. Osborne, A. Krause, and B. Schölkopf. AReS and MaRS-adversarial and MMD-minimizing regression for SDEs. *arXiv preprint arXiv:1902.08480*, 2019.

M. A. Álvarez and N. D. Lawrence. Computationally efficient convolved multiple output Gaussian processes. *Journal of Machine Learning Research*, 12(May):1459–1500, 2011.

M. A. Alvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2693–2705, 2013.

M. A. Álvarez, W. O. Ward, and C. Guarnizo. Non-linear process convolutions for multi-output Gaussian processes. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

M. Barenco, D. Tomescu, D. Brewer, R. Callard, J. Stark, and M. Hubank. Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome biology*, 7(3):R25, 2006.

M. Bińkowski, G. Marti, and P. Donnat. Autoregressive convolutional neural networks for asynchronous time series. *arXiv preprint arXiv:1703.04122*, 2017.

D. M. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

G. Conte, C. H. Moog, and A. M. Perdon. Nonlinear control systems: An algebraic setting. 1999.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803*, 2016.

N. Durrande, V. Adam, L. Bordeaux, S. Eleftheriadis, and J. Hensman. Banded matrix operators for gaussian markov models in the automatic differentiation era. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2780–2789, 2019.

M. Germain, K. Gregor, I. Murray, and H. Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889, 2015.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

D. F. Griffiths and D. J. Higham. *Numerical methods for ordinary differential equations: Initial value problems*. Springer Science & Business Media, 1 edition, 2010.

J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *2010 IEEE International Workshop on Machine Learning for Signal Processing*, pages 379–384. IEEE, 2010.

J. Hartikainen, M. Seppanen, and S. Sarkka. State-space inference for non-linear latent force models with application to satellite orbit prediction. *arXiv preprint arXiv:1206.4670*, 2012.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Heinonen, C. Yildiz, H. Mannerström, J. Intosalmi, and H. Lähdesmäki. Learning unknown ODE models with Gaussian processes. *arXiv preprint arXiv:1803.04303*, 2018.

S. J. Julier and J. K. Uhlmann. New extension of the Kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–193. International Society for Optics and Photonics, 1997.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

N. D. Lawrence, G. Sanguinetti, and M. Rattray. Modelling transcriptional regulation using Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 785–792, 2007.

P.-P. Liu. An analysis of a predator–prey model with both diffusion and migration. *Mathematical and Computer Modelling*, 51(9-10):1064–1070, 2010.

H. Nickisch, A. Solin, and A. Grigorievskiy. State space Gaussian processes with non-Gaussian likelihood. *arXiv preprint arXiv:1802.04846*, 2018.

G. Papamakarios, T. Pavlakou, and I. Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, pages 2338–2347, 2017.

G. Papamakarios, D. C. Sterratt, and I. Murray. Sequential neural likelihood: Fast likelihood-free inference with autoregressive flows. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, 2019.

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

C. E. Rasmussen and C. K. Williams. *Gaussian processes for machine learning*, volume 2. MIT Press Cambridge, MA, 2006.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*, pages 1530–1538. JMLR. org, 2015.

T. Ryder, A. Golightly, A. S. McGough, and D. Prangle. Black-box variational inference for stochastic differential equations. In *International Conference on Machine Learning*, pages 4420–4429, 2018a.

T. Ryder, A. Golighty, A. S. McGough, and D. Prangle. Black-box autoregressive density estimation for state-space models. *arXiv preprint arXiv:1811.08337*, 2018b.

S. Sarkka. On unscented Kalman filtering for state estimation of continuous-time nonlinear systems. *IEEE Transactions on automatic control*, 52(9):1631–1641, 2007.

S. Särkkä. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

S. Särkkä and A. Solin. *Applied Stochastic Differential Equations*, volume 10. Cambridge University Press, 2019.

S. Sarkka, M. A. Alvarez, and N. D. Lawrence. Gaussian process latent force models for learning and stochastic control of physical systems. *IEEE Transactions on Automatic Control*, 2018.

M. Titsias and M. Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International conference on machine learning*, pages 1971–1979, 2014.

M. K. Titsias and M. Lawrence, Neil D Rattray. Efficient sampling for Gaussian process inference using control variables. In *Advances in Neural Information Processing Systems*, pages 1681–1688, 2009.

A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. WaveNet: A generative model for raw audio. In *9th ISCA Speech Synthesis Workshop*, pages 125–125, 2016.

A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pages 4790–4798, 2016.

P. Wenk, A. Gotovos, S. Bauer, N. Gorbach, A. Krause, and J. M. Buhmann. Fast Gaussian process based gradient matching for parameter identification in systems of nonlinear ODEs. *arXiv preprint arXiv:1804.04378*, 2018.

K. Worden, R. Barthorpe, E. Cross, N. Dervilis, G. Holmes, G. Manson, and T. Rogers. On evolutionary system identification with applications to nonlinear benchmarks. *Mechanical Systems and Signal Processing*, 112:194–232, 2018.

C. Yildiz, M. Heinonen, J. Intosalmi, H. Mannerström, and H. Lähdesmäki. Learning stochastic differential equations with Gaussian processes without gradient matching. In *2018 IEEE 28th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2018.

Companion Matrices for Differential Equations

We briefly describe companion matrices for turning n-order linear SDEs into first-order by representing the system as a linear operator on an augmented state variable.

Consider the 2-order system

$$a_0 x(t) + a_1 \frac{\mathrm{d}}{\mathrm{d}t} x(t) + a_2 \frac{\mathrm{d}^2}{\mathrm{d}t^2} x(t) = w(t)$$

Define a new variable, $z = \mathrm{d}x/\mathrm{d}t$, and substitute into the above equation:

$$a_0 x(t) + a_1 z(t) + a_2 \frac{\mathrm{d}}{\mathrm{d}t} z(t) = w(t)$$

This is now the 1-order system:

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t} x(t) &= z(t) \\ \frac{\mathrm{d}}{\mathrm{d}t} z(t) &= -\tilde{a}_0 x(t) - \tilde{a}_1 z(t) + a_2^{-1} w(t), \end{aligned}$$

where $\tilde{a}_0$ and $\tilde{a}_1$ are a_0/a_2 and a_1/a_2 respectively.

We can write this using a joint state, $\boldsymbol{x}(t) = \begin{bmatrix} x(t) & z(t) \end{bmatrix}^\top$:

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{x}(t) = \begin{bmatrix} 0 & 1 \\ -\tilde{a}_0 & -\tilde{a}_1 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ a_2^{-1} \end{bmatrix} w(t)$$

The companion matrix for linear n-order systems scales in a similar manner, with the final matrix consisting of a final row of scalars and off-diagonal band with value 1. The extension to non-linear systems is a straightforward extension here, simply replacing the matrix with a vector-valued function.

Unscented Transform

The unscented transform is a means for propagating a random variable, x through a non-linear functional, f, by optimally sampling about the mean and propagating each sample through f and combining the results as a weighted sum (Julier and Uhlmann, 1997). These so-called sigma points are defined as:

$$\chi_i = \begin{cases} \mathbb{E}[x] & i = 0 \\ \mathbb{E}[x] + \left[\sqrt{(n+\eta)\mathrm{cov}[x]}\right]_i & i = 1, \ldots, n \\ \mathbb{E}[x] - \left[\sqrt{(n+\eta)\mathrm{cov}[x]}\right]_{n-i} & i = n+1, \ldots, 2n \end{cases}.$$

Note that $[\cdot]_i$ indicates the i^{th} column of a matrix. n describes the dimension of the random variable x and η is a scaling parameter, defined such that $\eta = \alpha_\chi^2(n + \kappa_\chi) - n$.

The unscented transform consists of transforming each sigma point, $\gamma_i = f(\chi_i)$ and constructing a weighted sum. The approximation of $y = f(x)$ is given by $y \sim \mathcal{N}(\mu, \Sigma)$, where

$$\begin{aligned} \mu &= \sum_{i=0}^{2n} \omega_i^{(m)} \gamma_i \\ \Sigma &= \sum_{i=0}^{2n} \omega_i^{(c)} \big(\mu - \gamma_i\big)\big(\mu - \gamma_i\big)^\top. \end{aligned}$$

The weights are defined by

$$\begin{aligned}\omega_0^{(m)} &= \eta(n+\eta)^{-1} \\ \omega_0^{(c)} &= \eta(n+\eta+1-\alpha_\chi^2+\beta_\chi)^{-1} \\ \omega_i^{(m)} &= \omega_i^{(c)} = (2n+2\eta)^{-1},\end{aligned}$$

where α, β, and κ are hyperparameters controlling the spread of sigma points. There are a number of reported settings for values of these hyperparameters, one such is $\alpha_\chi = 1$, $\beta_\chi = 0$, and $\kappa_\chi = n$ (Julier and Uhlmann, 1997). These are the values used in this paper.

Implementation

The implementation was written in TensorFlow, using TensorFlow probability. All experiments were optimised using Adam with a learning rate of 5e-3. For additional numerical stability, gradient clipping was performed based on the global norm.

The unscented filtering updates were implemented using the sigma-point dynamics as described in Sarkka (2007).

The number of flows for each experiment was set to $2d$, where d is the latent state dimension.