Variational multiple shooting for Bayesian ODEs with Gaussian processes (Supplementary material)

Pashupati Hegde[1] **Çağatay Yıldız**[1] **Harri Lähdesmäki**[1] **Samuel Kaski**[1] **Markus Heinonen**[1]

[1] Department of Computer Science, Aalto University, Finland

1 DETAILED DERIVATIONS

1.1 INFERENCE FOR THE VANILLA GPODE MODEL

The model. We consider the problem of inferring an ODE system

$$\mathbf{y}(t) = \mathbf{x}(t) + \boldsymbol{\epsilon} \tag{1}$$

$$\mathbf{x}(t) = \mathbf{x}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\tau))d\tau \tag{2}$$

from some noisy observations $\mathbf{y}(t)$ of the true system state $\mathbf{x}(t) \in \mathbb{R}^D$, whose evolution over time $t \in \mathbb{R}_+$ follows a differential equation vector field

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt} := \mathbf{f}(\mathbf{x}(t)), \qquad \mathbf{f} : \mathbb{R}^D \mapsto \mathbb{R}^D \tag{3}$$

starting from an initial state $\mathbf{x}_0 \in \mathbb{R}^D$. Our goal is to learn the underlying ODE vector field $\mathbf{f}$.

We propose a Gaussian process prior to the differential function

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')). \tag{4}$$

Following Titsias (2009) for sparse inference of GPs using inducing variables, we augment the full model with inducing values $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_M)^T \in R^{M \times D}$ and inducing locations $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)^T \in R^{M \times D}$, which results in a low-rank GP

$$p(\mathbf{U}) = \mathcal{N}(\mathbf{U}|\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}}) \tag{5}$$

$$p(\mathbf{f}|\mathbf{U}) = \mathcal{N}(\mathbf{f}|\mathbf{A}\text{vec}(\mathbf{U}), \mathbf{K}_{\mathbf{XX}} - \mathbf{A}\mathbf{K}_{\mathbf{ZZ}}\mathbf{A}^T), \tag{6}$$

where $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{N'})^T \in \mathbb{R}^{N' \times D}$ collects all the intermediate state evaluations $\mathbf{x}(t_i)$ encountered along numerical approximation of the true continuous ODE integral (2), $\mathbf{f} = (\mathbf{f}(\mathbf{x}_1)^T, \ldots, \mathbf{f}(\mathbf{x}_{N'})^T)^T \in \mathbb{R}^{N'D \times 1}$, $\mathbf{K}_{\mathbf{XX}}$ is a block-partitioned matrix of size $N'D \times N'D$ with $D \times D$ blocks, so that block $(\mathbf{K}_{\mathbf{XX}})_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$, and $\mathbf{A} = \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}$.

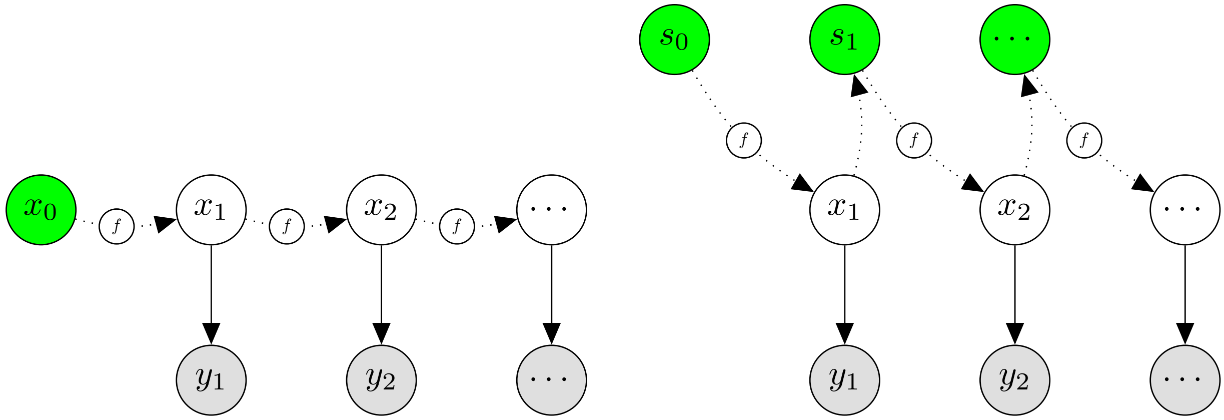
The joint model The joint probability of the model is

$$p(\mathbf{Y}, \mathbf{f}, \mathbf{U}, \mathbf{x}_0) = \prod_{i=1}^N p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0)p(\mathbf{f}|\mathbf{U})p(\mathbf{U})p(\mathbf{x}_0) \tag{7}$$

$$= \prod_{i=1}^N \underbrace{p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0)}_{\text{likelihood}} \underbrace{p(\mathbf{f}, \mathbf{U})}_{\text{GP prior}} \underbrace{p(\mathbf{x}_0)}_{\text{initial state prior}}, \tag{8}$$

where we assume a standard Gaussian prior $p(\mathbf{x}_0) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ for the unknown initial state $\mathbf{x}_0$.

Accepted for the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022).



(a) Vanilla GPODE model

(b) Shooting GPODE model

Figure 1: Plate diagrams: latent random variables that are considered during model inference are shaded in green. The intermediate variables $\mathbf{x}_i$ (unshaded) are defined as deterministic transformations of the inferred variables (conditioned on the vectorfield). In the vanilla GPODE formulation (a), the initial state distribution $\mathbf{x}_0$ is integrated forward in time to match all the observations $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_N\}$ forming a full trajectory. The shooting version (b) splits the full trajectory into multiple subintervals. Every subinterval i starts with an approximated state distribution $\mathbf{s}_i$, which is integrated forward to match the next observation $\mathbf{y}_{i+1}$. In addition, the state evolution from the previous shooting variable is matched to the variational shooting approximation at the current state.

Inference. Our primary goal is to learn the vector field $\mathbf{f}$ by inferring the model posterior $p(\mathbf{f}, \mathbf{U}, \mathbf{x}_0|\mathbf{Y})$, which is intractable. We resort to stochastic variational inference Hensman et al. (2013), and introduce a factorized Gaussian posterior approximation for the inducing variables across state dimensions

$$q(\mathbf{U}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_d|\mathbf{m}_d, \mathbf{Q}_d), \tag{9}$$

where, $\mathbf{u}_d \in \mathbb{R}^M$ and $\mathbf{m}_d \in \mathbb{R}^M, \mathbf{Q}_d \in \mathbb{R}^{M \times M}$ are the mean and covariance parameters of the variational Gaussian posterior approximation for the inducing variables. The Gaussian process posterior process with an inducing approximation can be written as

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) d\mathbf{U} \tag{10}$$

$$= \int \mathcal{N}\left(\mathbf{f}|\mathbf{A}\text{vec}(\mathbf{U}), \mathbf{K}_{\mathbf{XX}} - \mathbf{A}\mathbf{K}_{\mathbf{ZZ}}\mathbf{A}^T\right) q(\mathbf{U}) d\mathbf{U}. \tag{11}$$

We also introduce posterior approximation for the initial state variable $\mathbf{x}_0$,

$$q(\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_0|\mathbf{m}_0, \mathbf{S}_0). \tag{12}$$

This results in a variational joint posterior approximation

$$q(\mathbf{f}, \mathbf{U}, \mathbf{x}_0) = q(\mathbf{f}, \mathbf{U}) q(\mathbf{x}_0) \tag{13}$$

$$= p(\mathbf{f}|\mathbf{U}) q(\mathbf{U}) q(\mathbf{x}_0), \tag{14}$$

ELBO. With the above model specification, under variational inference of the posterior approximations, the evidence lower bound (ELBO) $\log p(\mathbf{Y}) \geq \mathcal{L}$ can be written as,

$$\mathcal{L} = \iiint q(\mathbf{f}, \mathbf{U}, \mathbf{x}_0) \log \frac{p(\mathbf{Y}, \mathbf{f}, \mathbf{U}, \mathbf{x}_0)}{q(\mathbf{f}, \mathbf{U}, \mathbf{x}_0)} d\mathbf{f} d\mathbf{U} d\mathbf{x}_0 \tag{15}$$

$$= \iiint q(\mathbf{f}, \mathbf{U}, \mathbf{x}_0) \log \prod_{i=1}^{N} \underbrace{p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0)}_{\mathcal{L}_y} \frac{p(\mathbf{f}|\mathbf{U})}{p(\mathbf{f}|\mathbf{U})} \underbrace{\frac{p(\mathbf{U})}{q(\mathbf{U})}}_{\mathcal{L}_u} \underbrace{\frac{p(\mathbf{x}_0)}{q(\mathbf{x}_0)}}_{\mathcal{L}_{\mathbf{x}_0}} d\mathbf{f} d\mathbf{U} d\mathbf{x}_0. \tag{16}$$

Hence the ELBO decomposes into three additive terms

$$\mathcal{L} = \mathcal{L}_y + \mathcal{L}_u + \mathcal{L}_{\mathbf{x}_0}, \tag{17}$$

where each term contains the (relevant parts of) expectation over $q(\mathbf{f}, \mathbf{U}, \mathbf{x}_0)$.

Likelihood term. The variational likelihood term $\mathcal{L}_y$ is an expectation of the likelihood wrt the variationally marginalized vectorfield posterior $q(\mathbf{f})$, and the initial state distribution $q(\mathbf{x}_0)$,

$$\mathcal{L}_y = \iint q(\mathbf{f}, \mathbf{x}_0) \log p(\mathbf{y}|\mathbf{f}, \mathbf{x}_0) d\mathbf{f} d\mathbf{x}_0 \tag{18}$$

$$= \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f},\mathbf{x}_0)} \log p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0). \tag{19}$$

This term computes the likelihood $p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0) = p(\mathbf{y}_i|\mathbf{x}_i)$ over ODE state solutions $\mathbf{x}_i = \mathbf{x}_0 + \int_0^{t_i} \mathbf{f}(\mathbf{x}(\tau))d\tau$ for a single realization of the vector field $\mathbf{f} \sim p(\mathbf{f})$ and the initial state $\mathbf{x}_0 \sim p(\mathbf{x}_0)$. Because of the non-linear integration $\mathbf{x}_0 \mapsto \mathbf{x}(t)$, we cannot solve this integral analytically. Instead, we resort to Monte Carlo integration by sampling ODE trajectories over different vector field realizations $\mathbf{f} \sim q(\mathbf{f})$ and initial states $\mathbf{x}_0 \sim q(\mathbf{x}_0)$. In practice, this term can be approximated as

$$\mathcal{L}_y \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N} \log p(\mathbf{y}_i|\mathbf{f}^{(s)}, \mathbf{x}_0^{(s)}) \tag{20}$$

where we sum over S reparameterized samples $\mathbf{f}^{(s)} \sim q(\mathbf{f})$ and $\mathbf{x}_0^{(s)} \sim q(\mathbf{x}_0)$.

Inducing KL. This term corresponds to the KL divergence between variational posterior and the prior distribution of inducing values. This term can be derived analytically as the KL between multivariate Gaussians.

$$\mathcal{L}_u = \int q(\mathbf{U}) \log \frac{p(\mathbf{U})}{q(\mathbf{U})} d\mathbf{U} \tag{21}$$

$$= \sum_{d=1}^{D} \int q(\mathbf{u}_d) \log \frac{p(\mathbf{u}_d)}{q(\mathbf{u}_d)} d\mathbf{u} \tag{22}$$

$$= -\sum_{d=1}^{D} \mathrm{KL}\left[q(\mathbf{u}_d)||p(\mathbf{u}_d)\right] \tag{23}$$

Initial state KL. This term corresponds to the KL divergence between variational posterior and the prior distribution of the initial state. With an assumption of Gaussian prior and variational posterior, this term can also be derived analytically,

$$\mathcal{L}_{\mathbf{x}_0} = \int q(\mathbf{x}_0) \log \frac{p(\mathbf{x}_0)}{q(\mathbf{x}_0)} d\mathbf{x}_0 \tag{24}$$

$$= -\mathrm{KL}\left[q(\mathbf{x}_0)||p(\mathbf{x}_0)\right] \tag{25}$$

Complete ELBO. The full ELBO is then

$$\mathcal{L} = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{f},\mathbf{x}_0)} \log p(\mathbf{y}_i|\mathbf{f}, \mathbf{x}_0) - \mathrm{KL}[q(\mathbf{U}) \,||\, p(\mathbf{U})] - \mathrm{KL}[q(\mathbf{x}_0) \,||\, p(\mathbf{x}_0)] \tag{26}$$

1.2 DECOUPLED SAMPLING OF GPODES

In this section, we provide details for simulating valid ODE trajectories from a GP vector field posterior of the form

$$q(\mathbf{u}) = \mathcal{N}(\mathbf{m}, \mathbf{Q}), \tag{27}$$

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{u})q(\mathbf{u})d\mathbf{u} \tag{28}$$

$$= \int \mathcal{N}\left(\mathbf{f}|\mathbf{A}\mathbf{u}, \mathbf{K}_{\mathbf{X}\mathbf{X}} - \mathbf{A}\mathbf{K}_{\mathbf{Z}\mathbf{Z}}\mathbf{A}^T\right) q(\mathbf{u})d\mathbf{u}, \tag{29}$$

where $\mathbf{A} = \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}$ and $\mathbf{m} \in \mathbb{R}^M, \mathbf{Q} \in \mathbb{R}^{M \times M}$ are the variational mean and covariance parameters of the Gaussian posterior approximation for inducing variables. For simplicity, we consider a scalar valued GP, but it is straightforward to extend this approach to vector-valued GPs.

A sparse GP posterior of the form (29) can be decomposed into two parts using Matheron's rule (Corollary 2 Wilson et al. (2020)),

$$\underbrace{f(\mathbf{x})|\mathbf{u}}_{\texttt{posterior}} = \underbrace{f(\mathbf{x})}_{\texttt{prior}} + \underbrace{k(\mathbf{x},\mathbf{Z})K(\mathbf{Z},\mathbf{Z})^{-1}(\mathbf{u}-\mathbf{f}_{\mathbf{Z}})}_{\texttt{update}}. \tag{30}$$

Wilson et al. (2020) propose a decoupled sampling from the `posterior` by using different bases for the `prior` and `update` terms. In particular, they propose Fourier basis functions for the `prior` term and canonical basis for the `update` term respectively

$$\underbrace{f(\mathbf{x})|\mathbf{u}}_{\texttt{posterior}} \approx \underbrace{\sum_{i=1}^{F} w_i \phi_i(\mathbf{x})}_{\texttt{prior}} + \underbrace{\sum_{j=1}^{M} \nu_j K(\mathbf{x}, \mathbf{z}_j)}_{\texttt{update}}, \tag{31}$$

where we use F Fourier bases $\phi_i(\cdot)$ with $w_i \sim \mathcal{N}(0,1)$ (Rahimi and Recht, 2007) to represent the stationary prior, and function basis $K(\cdot, \mathbf{z}_j)$ for the posterior update with $\boldsymbol{\nu} = K(\mathbf{Z},\mathbf{Z})^{-1}(\mathbf{u} - \mathbf{\Phi w})$, $\mathbf{\Phi} = \boldsymbol{\phi}(\mathbf{Z}) \in \mathbb{R}^{M \times F}, \mathbf{w} \in \mathbb{R}^F$. We can evaluate functions from the posterior (29) in linear time at arbitrary locations.

For the experimental results presented in the paper, we use a squared exponential kernel for which we can compute the feature maps $\boldsymbol{\phi}_i(\mathbf{x}) = \sqrt{\frac{\sigma_f^2}{F}}(\cos \mathbf{x}^T \boldsymbol{\omega}_i, \sin \mathbf{x}^T \boldsymbol{\omega}_i)$ where $\boldsymbol{\omega}_i$ is sampled proportional to the spectral density of the squared exponential kernel $\boldsymbol{\omega}_i \sim \mathcal{N}(\mathbf{0}, \Lambda^{-1})$, Λ is a diagonal matrix collecting lengthscale parameters of the kernel $\Lambda = \text{diag}(l_1^2, l_2^2, \ldots, l_D^2)$ and σ_f^2 is the signal variance parameter. In the case of the squared exponential kernel, this results in $2F$ feature maps $\boldsymbol{\phi}(\mathbf{x}) \in \mathbb{R}^{2F}$, for which we sample weights $\mathbf{w} \in \mathbb{R}^{2F}$ from the standard Normal $w_i \sim \mathcal{N}(0,1)$. By fixing random samples of feature maps $\boldsymbol{\phi}(\cdot)$, corresponding weights $\mathbf{w}$ and inducing values $\mathbf{u}$ for an ODE integration call, we can sample a unique ODE trajectory from a posterior vector field of the form (29).

1.3 PROBABILISTIC SHOOTING FORMULATION FOR GPODES

The model. We consider the problem of inferring an ODE system

$$\mathbf{y}(t) = \mathbf{x}(t) + \boldsymbol{\epsilon} \tag{32}$$

$$\mathbf{x}(t) = \mathbf{s}_0 + \int_0^t \mathbf{f}(\mathbf{x}(\tau))d\tau, \tag{33}$$

from some noisy observations $\mathbf{y}(t)$ of the true system state $\mathbf{x}(t) \in \mathbb{R}^D$, whose evolution over time $t \in \mathbb{R}_+$ follows a differential equation

$$\dot{\mathbf{x}}(t) = \frac{d\mathbf{x}(t)}{dt} := \mathbf{f}(\mathbf{x}(t)), \qquad \mathbf{f} : \mathbb{R}^D \mapsto \mathbb{R}^D \tag{34}$$

starting from the initial state $\mathbf{s}_0 \in \mathbb{R}^D$. Our goal is to learn the underlying ODE vector field $\mathbf{f}$.

Shooting augmentation. We propose an augmented 'shooting' ODE system

$$\mathbf{y}_i = \mathbf{x}(t_i; \mathbf{s}_{i-1}) + \boldsymbol{\epsilon} \tag{35}$$

$$\mathbf{x}(t_i; \mathbf{s}_{i-1}) = \mathbf{s}_{i-1} + \int_{t_{i-1}}^{t_i} \mathbf{f}(\mathbf{x}(\tau))d\tau \tag{36}$$

$$\mathbf{s}_i = \mathbf{x}(t_i; \mathbf{s}_{i-1}) + \boldsymbol{\xi}, \tag{37}$$

where we divide the state function $\mathbf{x}(t)$ into N short segments, with the end state of i^{th} segment $\mathbf{x}(t_i; \mathbf{s}_{i-1})$ defining solutions to initial value problems (36) starting from the corresponding shooting variables $\mathbf{s}_{i-1}$. These short shooting segments follow the same differential $\mathbf{f}$ as the original model. The augmented system is equivalent to the original ODE system, in the limit when the tolerance parameter $\boldsymbol{\xi} \to \mathbf{0}$.

We assume Gaussian distributions on both observation noise and tolerance parameters, resulting in the following distributions,

$$p(\mathbf{y}_i|\mathbf{s}_{i-1}) = \mathcal{N}(\mathbf{y}_i|\mathbf{x}(t_i;\mathbf{s}_{i-1}), \sigma_y^2\mathbf{I}); \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_y^2\mathbf{I}), \tag{38}$$

$$p(\mathbf{s}_i|\mathbf{s}_{i-1}) = \mathcal{N}(\mathbf{s}_i|\mathbf{x}(t_i;\mathbf{s}_{i-1}), \sigma_\xi^2\mathbf{I}); \qquad \boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma_\xi^2\mathbf{I}). \tag{39}$$

Gaussian process ODE. We propose a Gaussian process prior for the differential function

$$\mathbf{f}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, k(\mathbf{x}, \mathbf{x}')) \tag{40}$$

In addition, we augment the full model with inducing values $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_M)^T \in R^{M\times D}$ and inducing locations $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_M)^T \in R^{M\times D}$, which results in a low-rank GP

$$p(\mathbf{U}) = \mathcal{N}(\mathbf{U}|\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}}) \tag{41}$$

$$p(\mathbf{f}|\mathbf{U}) = \mathcal{N}(\mathbf{f}|\mathbf{A}\mathrm{vec}(\mathbf{U}), \mathbf{K}_{\mathbf{XX}} - \mathbf{A}\mathbf{K}_{\mathbf{ZZ}}\mathbf{A}^T), \tag{42}$$

where $\mathbf{A} = \mathbf{K}_{\mathbf{XZ}}\mathbf{K}_{\mathbf{ZZ}}^{-1}$.

The joint model. The joint probability of the model is

$$p(\mathbf{Y}, \mathbf{S}, \mathbf{f}, \mathbf{U}) = p(\mathbf{Y}|\mathbf{S}, \mathbf{f})p(\mathbf{S}|\mathbf{f})p(\mathbf{f}|\mathbf{U})p(\mathbf{U}) \tag{43}$$

$$= \prod_{i=1}^{N} \underbrace{p(\mathbf{y}_i|\mathbf{s}_{i-1}, \mathbf{f})}_{\text{likelihood}} \prod_{i=1}^{N-1} \underbrace{p(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f})}_{\text{shooting prior}} \underbrace{p(\mathbf{s}_0)}_{\text{initial state}} \underbrace{p(\mathbf{f}|\mathbf{U})p(\mathbf{U})}_{\text{GP prior}}, \tag{44}$$

where $\mathbf{S} = (\mathbf{s}_0, \mathbf{s}_1, \ldots \mathbf{s}_{N-1})^T \in \mathbb{R}^{N\times D}$ collects all shooting variables.

We also note that observations are at indices $1, \ldots, N$, while the shooting variables are always one behind the observations at $0, \ldots, N-1$ (see plate diagram 1 (b)).

Inference. Our primary goal is to learn the vector field $\mathbf{f}$ by inferring the model posterior $p(\mathbf{S}, \mathbf{f}, \mathbf{U}|\mathbf{Y})$, which is intractable. Similar to non-shooting GPODEs, we introduce a factorized Gaussian posterior approximation for the inducing variables across state dimensions

$$q(\mathbf{U}) = \prod_{d=1}^{D} \mathcal{N}(\mathbf{u}_d|\mathbf{m}_d, \mathbf{Q}_d), \tag{45}$$

where, $\mathbf{u}_d \in \mathbb{R}^M$ and $\mathbf{m}_d \in \mathbb{R}^M, \mathbf{Q}_d \in \mathbb{R}^{M\times M}$ are the mean and covariance parameters of the variational Gaussian posterior approximation for the inducing variables.

The Gaussian process posterior process with an inducing approximation can be written as

$$q(\mathbf{f}) = \int p(\mathbf{f}|\mathbf{U})q(\mathbf{U})d\mathbf{U} \tag{46}$$

$$= \int \mathcal{N}\left(\mathbf{f}|\mathbf{A}\mathrm{vec}(\mathbf{U}), \mathbf{K}_{\mathbf{XX}} - \mathbf{A}\mathbf{K}_{\mathbf{ZZ}}\mathbf{A}^T\right) q(\mathbf{U})d\mathbf{U}. \tag{47}$$

Next, we introduce a factorized Gaussian posterior approximations for the shooting variables $\mathbf{S}$ as well,

$$q(\mathbf{S}) = \prod_{i=0}^{N-1} q(\mathbf{s}_i) = \prod_{i=0}^{N-1} \mathcal{N}(\mathbf{s}_i|\mathbf{a}_i, \Sigma_i). \tag{48}$$

where, $\mathbf{a}_i \in \mathbb{R}^D$ and $\Sigma_i \in \mathbb{R}^{D\times D}$ are the mean and covariance parameters of the variational Gaussian posterior approximation for the shooting variables.

This results in a variational joint posterior approximation

$$q(\mathbf{S}, \mathbf{f}, \mathbf{U}) = q(\mathbf{S})q(\mathbf{f}, \mathbf{U}) \tag{49}$$

$$= \prod_{i=0}^{N-1} q(\mathbf{s}_i)p(\mathbf{f}|\mathbf{U})q(\mathbf{U}). \tag{50}$$

ELBO. Under variational inference the posterior approximations q are optimized to match the true posterior in the KL sense,

$$\arg\min_q \ \mathrm{KL}\left[q(\mathbf{S},\mathbf{f},\mathbf{U}) \,||\, p(\mathbf{S},\mathbf{f},\mathbf{U}|\mathbf{Y})\right]. \tag{51}$$

This is equivalent to maximizing the evidence lower bound (ELBO) $\log p(\mathbf{Y}) \geq \mathcal{L}$,

$$\mathcal{L} = \iiint q(\mathbf{S},\mathbf{f},\mathbf{U}) \log \left[\frac{p(\mathbf{Y},\mathbf{S},\mathbf{f},\mathbf{U})}{q(\mathbf{S},\mathbf{f},\mathbf{U})}\right] d\mathbf{S} d\mathbf{f} d\mathbf{U} \tag{52}$$

$$= \iiint q(\mathbf{S},\mathbf{f},\mathbf{U}) \log \left[\prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) \cdot \prod_{i=1}^{N-1} \frac{p(\mathbf{s}_i|\mathbf{s}_{i-1},\mathbf{f})}{q(\mathbf{s}_i)} \cdot \frac{p(\mathbf{s}_0)}{q(\mathbf{s}_0)} \cdot \frac{p(\mathbf{f},\mathbf{U})}{q(\mathbf{f},\mathbf{U})}\right] d\mathbf{S} d\mathbf{f} d\mathbf{U} \tag{53}$$

$$= \underbrace{\iint q(\mathbf{S})q(\mathbf{f}) \log \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) d\mathbf{S} d\mathbf{f}}_{\mathcal{L}_y} + \underbrace{\iint q(\mathbf{S})q(\mathbf{f}) \log \prod_{i=1}^{N-1} p(\mathbf{s}_i|\mathbf{s}_{i-1},\mathbf{f}) d\mathbf{S} d\mathbf{f}}_{\mathcal{L}_{sc}}$$

$$\underbrace{- \int q(\mathbf{S}) \log \prod_{i=1}^{N-1} q(\mathbf{s}_i) d\mathbf{S}}_{\mathcal{L}_{se}} + \underbrace{\int q(\mathbf{s}_0) \log \frac{p(\mathbf{s}_0)}{q(\mathbf{s}_0)} d\mathbf{s}_0}_{\mathcal{L}_0} + \underbrace{\int q(\mathbf{U}) \log \frac{p(\mathbf{U})}{q(\mathbf{U})} d\mathbf{U}}_{\mathcal{L}_u} \tag{54}$$

$$\tag{55}$$

which results in the ELBO decomposing into four additive terms

$$\mathcal{L} = \mathcal{L}_y + \mathcal{L}_{sc} + \mathcal{L}_{se} + \mathcal{L}_0 + \mathcal{L}_u, \tag{56}$$

where each term contains the (relevant parts of) expectation over $q(\mathbf{S},\mathbf{f},\mathbf{U})$.

Likelihood term. The variational likelihood term $\mathcal{L}_y$ is an expectation of the likelihood under the posteriors of shooting variables $q(\mathbf{S})$ and the posterior vectorfield $q(\mathbf{f})$,

$$\mathcal{L}_y = \iint q(\mathbf{S})q(\mathbf{f}) \log \prod_{i=1}^{N} p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) d\mathbf{S} d\mathbf{f} \tag{57}$$

$$= \sum_{i=1}^{N} \iint q(\mathbf{s}_{i-1})q(\mathbf{f}) \log p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) d\mathbf{s}_{i-1} d\mathbf{f} \tag{58}$$

$$= \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{s}_{i-1})q(\mathbf{f})} \Big[\log p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) \Big]. \tag{59}$$

We can evaluate this term with Monte Carlo integration by taking reparameterized samples from the posteriors $\mathbf{f}^{(s)} \sim q(\mathbf{f})$ and $\mathbf{s}_{i-1}^{(s)} \sim q(\mathbf{s}_{i-1})$ as below

$$\mathcal{L}_y = \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{s}_{i-1},\mathbf{f})} \Big[\log p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) \Big] \tag{60}$$

$$\mathcal{L}_y \approx \frac{1}{S} \sum_{s=1}^{S} \sum_{i=1}^{N} \Big[\log p(\mathbf{y}_i|\mathbf{x}_i^{(s)}) \Big], \tag{61}$$

where $\mathbf{x}_i^{(s)}$ is defined as solution to the following initial value problem,

$$\mathbf{x}_i^{(s)} := \mathbf{x}^{(s)}(t_i; \mathbf{s}_{i-1}) = \mathbf{s}_{i-1}^{(s)} + \int_{t_{i-1}}^{t_i} \mathbf{f}^{(s)}(\mathbf{x}(\tau)) d\tau. \tag{62}$$

Shooting cross-entropy term. This term computes the cross-entropy between the prior specification for the shooting variables under the ODE evolution $p(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f})$, and the point-wise approximations $q(\mathbf{s}_i)$,

$$\mathcal{L}_{se} = \iint q(\mathbf{S})q(\mathbf{f})\Big[\log \prod_{i=1}^{N-1} p(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f})\Big] d\mathbf{S}d\mathbf{f} \tag{63}$$

$$= \iint q(\mathbf{s}_{N-1})\cdots q(\mathbf{s}_1)q(\mathbf{s}_0)q(\mathbf{f})\Big[\log p(\mathbf{s}_{N-1}|\mathbf{s}_{N-2}, \mathbf{f})\cdots p(\mathbf{s}_1|\mathbf{s}_0, \mathbf{f})\Big] d\mathbf{S}d\mathbf{f} \tag{64}$$

$$= \sum_{i=1}^{N-1} \iint q(\mathbf{f})q(\mathbf{s}_i)q(\mathbf{s}_{i-1})\Big[\log p(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f})\Big] d\mathbf{s}_{i-1}d\mathbf{s}_i d\mathbf{f} \tag{65}$$

$$= \sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i, \mathbf{s}_{i-1}, \mathbf{f})}\Big[\log p\left(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f}\right)\Big]. \tag{66}$$

This term can also be numerically estimated with Monte Carlo integration using posterior samples $\mathbf{f}^{(s)} \sim q(\mathbf{f})$, $\mathbf{s}_{i-1}^{(s)} \sim q(\mathbf{s}_{i-1})$ and $\mathbf{s}_i^{(s)} \sim q(\mathbf{s}_i)$

$$\mathcal{L}_{se} = \sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i, \mathbf{s}_{i-1}, \mathbf{f})}\Big[\log p\left(\mathbf{s}_i|\mathbf{s}_{i-1}, \mathbf{f}\right)\Big] \tag{67}$$

$$\approx \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N-1} \log p\left(\mathbf{s}_i^{(s)}|\mathbf{x}_i^{(s)}\right), \tag{68}$$

$$\mathbf{x}_i^{(s)} := \mathbf{x}^{(s)}(t_i; \mathbf{s}_{i-1}) = \mathbf{s}_{i-1}^{(s)} + \int_{t_{i-1}}^{t_i} \mathbf{f}^{(s)}(\mathbf{x}(\tau))d\tau. \tag{69}$$

Shooting entropy term. This term computes the entropy of the posterior approximations for shooting variables $q(\mathbf{s}_i)$. Since we assume factorized Gaussian approximations, this term can be simplified analytically as the sum of Gaussian entropy.

$$\mathcal{L}_{se} = -\int q(\mathbf{S}) \log \prod_{i=1}^{N-1} q(\mathbf{s}_i) d\mathbf{S} \tag{70}$$

$$= -\sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i)}\Big[\log q(\mathbf{s}_i)\Big]. \tag{71}$$

Initial state KL term. This term corresponds to the KL divergence between variational posterior and the prior distribution of the initial state. With the assumption of Gaussian prior and variational posterior, this term can also be derived analytically,

$$\mathcal{L}_0 = \int q(\mathbf{s}_0) \log \frac{p(\mathbf{s}_0)}{q(\mathbf{s}_0)} d\mathbf{s}_0 \tag{72}$$

$$= -\operatorname{KL}\left[q(\mathbf{s}_0)||p(\mathbf{s}_0)\right]. \tag{73}$$

Inducing KL term. This term corresponds to the KL divergence between variational posterior and prior distribution of inducing values. This term can also be derived analytically as the KL between multivariate Gaussians.

$$\mathcal{L}_u = \int q(\mathbf{U}) \log \frac{p(\mathbf{U})}{q(\mathbf{U})} d\mathbf{U} \tag{74}$$

$$= \sum_{d=1}^{D} \int q(\mathbf{u}_d) \log \frac{p(\mathbf{u}_d)}{q(\mathbf{u}_d)} d\mathbf{u} \tag{75}$$

$$= -\sum_{d=1}^{D} \operatorname{KL}\left[q(\mathbf{u}_d)||p(\mathbf{u}_d)\right]. \tag{76}$$

Complete ELBO. The full ELBO is then

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_y + \mathcal{L}_{sc} + \mathcal{L}_{se} + \mathcal{L}_0 + \mathcal{L}_u && (77) \\
&= \sum_{i=1}^{N} \mathbb{E}_{q(\mathbf{s}_{i-1},\mathbf{f})} \Big[\log p(\mathbf{y}_i|\mathbf{s}_{i-1},\mathbf{f}) \Big] + \sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i,\mathbf{s}_{i-1},\mathbf{f})} \Big[\log p\left(\mathbf{s}_i|\mathbf{s}_{i-1},\mathbf{f}\right) \Big] \\
&\quad - \sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i)} \Big[\log q(\mathbf{s}_i) \Big] - \mathrm{KL}[q(\mathbf{s}_0)\,||\,p(\mathbf{s}_0)] - \mathrm{KL}[q(\mathbf{U})\,||\,p(\mathbf{U})] && (78)
\end{aligned}
$$

which in practice is numerically estimated with Monte Carlo integration

$$
\begin{aligned}
\mathcal{L} &\approx \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N} \Big[\log p(\mathbf{y}_i|\mathbf{x}_i^{(s)}) \Big] + \frac{1}{S}\sum_{s=1}^{S}\sum_{i=1}^{N-1} \log p\left(\mathbf{s}_i^{(s)}|\mathbf{x}_i^{(s)}\right) \\
&\quad - \sum_{i=1}^{N-1} \mathbb{E}_{q(\mathbf{s}_i)} \Big[\log q(\mathbf{s}_i) \Big] - \mathrm{KL}[q(\mathbf{s}_0)\,||\,p(\mathbf{s}_0)] - \mathrm{KL}[q(\mathbf{U})\,||\,p(\mathbf{U})] && (79)
\end{aligned}
$$

where $\mathbf{f}^{(s)} \sim q(\mathbf{f})$, $\mathbf{s}_{i-1}^{(s)} \sim q(\mathbf{s}_{i-1})$, $\mathbf{s}_i^{(s)} \sim q(\mathbf{s}_i)$ and

$$
\mathbf{x}_i^{(s)} := \mathbf{x}^{(s)}(t_i;\mathbf{s}_{i-1}) = \mathbf{s}_{i-1}^{(s)} + \int_{t_{i-1}}^{t_i} \mathbf{f}^{(s)}(\mathbf{x}(\tau))d\tau. \qquad (80)
$$

2 EXPERIMENTAL DETAILS

Algorithm 1 GPODEs : Bayesian inference of ODEs using Gaussian processes

```
Inputs:
    - Observed states $\mathbf{Y}$, observation time sequence $\mathbf{t}$.
Initialize hyperparameters:
    - Kernel parameters $\theta$, likelihood parameters, inducing locations $\mathbf{Z}$.
Initialize variational parameters:
    - Parameters of $q(\mathbf{U}) = \mathcal{N}(\mathbf{m}, \mathbf{Q})$.
    - Parameters of $q(\mathbf{x_0}) = \mathcal{N}(\mathbf{a_0}, \mathbf{\Sigma_0})$.
Optimization:
for every optimization step do
  (1) Sample a function $\mathbf{f}$ from the ODE posterior in (11) by taking following samples:
      - Parameters of Fourier bases $\omega_\theta$ proportional to the spectral density of GP kernel,
      - Weights $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,
      - Sample from the inducing posterior $\mathbf{U} \sim \mathcal{N}(\mathbf{m}, \mathbf{Q})$.
  (2) Sample initial state $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{a_0}, \mathbf{\Sigma_0})$.
  (3) Compute predicted states $\hat{\mathbf{Y}} = \mathrm{ODEsolve}(\mathbf{f}, \mathbf{x_0}, \mathbf{t})$.
  (4) Compute ELBO from (26) : likelihood$(\mathbf{Y}, \hat{\mathbf{Y}})$, KL$[q(\mathbf{U})||p(\mathbf{U})]$, KL$[q(\mathbf{x_0})||p(\mathbf{x_0})]$.
  (5) Update all parameters with stochastic gradients of ELBO.
end for
```

2.1 OPTIMIZATION SETUP

We use Adam (Kingma and Ba, 2014) optimizer and jointly train all the variational parameters and hyperparameters. The complete list of optimized parameters, along with additional method-specific details, are given below.

Vanilla GPODE model. We use 'whitened' representation for the inducing variables and optimize following parameters against the evidence lowerbound (see algorithm 1).

- Variational parameters:

  - Inducing variables $q(\mathbf{U})$, initial states $q(\mathbf{x}_0)$
- Hyperparameters:
  - Inducing locations $\mathbf{Z}$
  - Likelihood parameters: scale parameter for the Gaussian likelihood
  - Kernel parameters: length scales and signal variance parameters in case of squared exponential kernel

Shooting GPODE model. We use 'whitened' representation for inducing variables and optimize the following parameters against the evidence lower bound.

- Variational parameters:
  - Inducing variables $q(\mathbf{U})$, shooting states $q(\mathbf{S})$
- Hyperparameters:
  - Inducing locations $\mathbf{Z}$
  - Likelihood parameters: scale parameter for the Gaussian likelihood
  - Kernel parameters: length scales and signal variance parameters in case of squared exponential kernel

npODE model. We use 'whitened' representation for inducing variables, maximum a posteriori (MAP) objective, and optimize following parameters:

- Inducing values $\mathbf{U}$ and locations $\mathbf{Z}$.
- Likelihood parameters: scale parameter for the Gaussian likelihood.
- Kernel parameters: length scales and signal variance parameters in case of the squared exponential kernel.

NeuralODE model. We use `tanh` activation and a fully connected block with one hidden layer having 32 units in Van der Pol/ Fitz-Hugh Nagumo experiments. In MoCap experiments, we try one/two hidden layers with 64/128 hidden units, and report the best results. All the network parameters were optimized against `MSE` loss.

Bayesian NeuralODE model. We utilized the codebase [1] provided by Dandekar et al. (2020) for training Bayesian version of NeuralODEs. We used networks with one hidden layer and 32 units VDP/FHN experiments and performed posterior sampling with HMC. In case of experiments with long sequences (shooting illustration on VDP and MocCap) the HMC sampling had convergence issues, hence we performed variational inference instead. In case of MoCap experiments, we tried networks with two hidden layers and 64/128 hidden units, and performed mean-field variational inference.

2.2 ADDITIONAL DETAILS ON THE INDUCING VARIABLES

'Whitening' the inducing variables. While performing sparse inference for GPs using inducing variables, it is a common practice to use noncental parameterization $\tilde{\mathbf{U}} = \mathbf{L}_\theta \mathbf{U}$ where $\mathbf{L}_\theta {\mathbf{L}_\theta}^T = \mathbf{K}_\theta(\mathbf{Z}, \mathbf{Z})$ (Hensman et al., 2015). Such a reparametrization turns the inference for $\mathbf{U}$ with prior $\mathcal{N}(\mathbf{0}, \mathbf{K}_{\mathbf{ZZ}})$ into inference for $\tilde{\mathbf{U}}$ with isotropic Gaussian prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This generally improves the optimization performance by decorrelating the latent parameters from each other.

Initializing inducing variables using data gradients. In case of sparse Gaussian process model with inducing variables, we initialize the vector field with empirical gradients from the observed data. We first initialize inducing locations $\mathbf{Z}$ as `kmeans` cluster centers of observations $\mathbf{Y}$. Next we compute empirical gradient estimates, $\dot{\mathbf{Y}} = (\mathbf{y}_2 - \mathbf{y}_1, \mathbf{y}_3 - \mathbf{y}_2, \dots, \mathbf{y}_N - \mathbf{y}_{N-1})$ at locations $\tilde{\mathbf{Y}} = (\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{N-1})$ and initialize inducing values $\mathbf{U}$ as the GP mean interpolation of empirical gradients at inducing locations.

$$\mathbf{U} = \Delta t \cdot K(\mathbf{Z}, \tilde{\mathbf{Y}}) K(\tilde{\mathbf{Y}}, \tilde{\mathbf{Y}})^{-1} \dot{\mathbf{Y}}, \tag{81}$$

where Δt is the time difference between two consecutive observations in the dataset.

2.3 ADDITIONAL DETAILS ON THE CMU MOCAP EXPERIMENT

Details on the dataset. The dataset used in this experiment was obtained from `http://mocap.cs.cmu.edu/`. The database consists of sensor recordings of multiple activities for different subjects in `.amc` files. We selected three subjects

[1] `https://github.com/RajDandekar/MSML21_BayesianNODE`

Table 1: For each subject (a), we report the activity considered for the experiment (b), the data split train/validation/test (c), the number of sequences considered for the corresponding split (d), and the files used in the corresponding split (e).

(a) subject	(b) activity	(c) split	(d) # sequences	(e) files
subject 09	running	train	6	05.amc, 06.amc, 07.amc, 08.amc, 09.amc, 11.amc
		validation	2	01.amc, 02.amc
		test	2	03.amc, 04.amc
subject 35	walking	train	16	01.amc, 02.amc, 03.amc, 04.amc, 05.amc, 06.amc, 07.amc, 08.amc, 09.amc, 10.amc, 11.amc, 12.amc, 13.amc, 14.amc, 15.amc, 16.amc
		validation	3	28.amc, 29.amc, 30.amc
		test	4	31.amc, 32.amc, 33.amc, 34.amc
subject 39	walking	train	6	01.amc, 02.amc, 07.amc, 08.amc, 09.amc, 10.amc
		validation	2	03.amc, 04.amc
		test	2	05.amc, 06.amc

with the most number of walking or running sequences: subjects 09, 35, and 39. The .amc files considered for train, validation and test purposes are given in table 1. The training sequences and their lengths were selected to include at least one full cycle of the dynamics while learning the model. The observation sequence lengths for training/test/validation splits are reported in table 2.

Details on the PCA In the CMU MoCap experiment, we project the data from D dimensional observation-space to $K < D$ dimensional latent-space using eigenvectors corresponding to top-K eigenvalues. The ODE model is then learnt in the latent-space and model predictions are projected back into the observation-space using K eigenvectors. We refer to this as 'inverting the PCA' in the main text.

Table 2: For each subject (a), we report the experiment type (b), the data split train/validation/test (c), and the number of observations considered for the corresponding split.

(a) subject	(b) experiment	(c) split	(d) sequence length
subject 09	short	train	50
		validation	120
		test	120
	long	train	100
		validation	120
		test	120
subject 35	short	train	50
		validation	300
		test	300
	long	train	250
		validation	300
		test	300
subject 39	short	train	100
		validation	300
		test	300
	long	train	250
		validation	300
		test	300

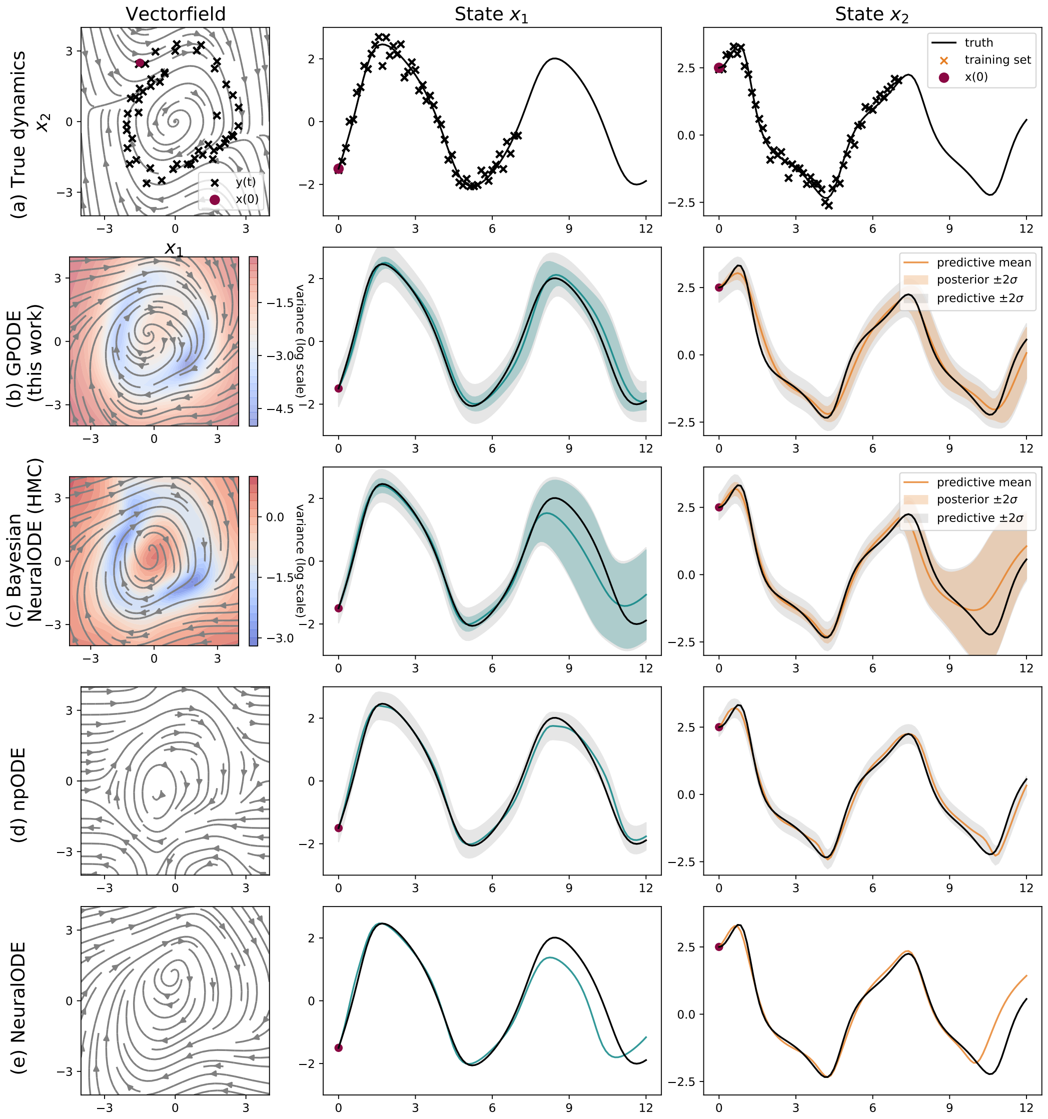


Figure 2: Learning the 2D Van der Pol dynamics on irregularly sampled observations **(a)** with alternative methods **(b-d)**. Column 1 shows the vector fields while columns 2 and 3 show the state trajectories $x_1(t)$ and $x_2(t)$. GPODE learns the posterior accurately.

References

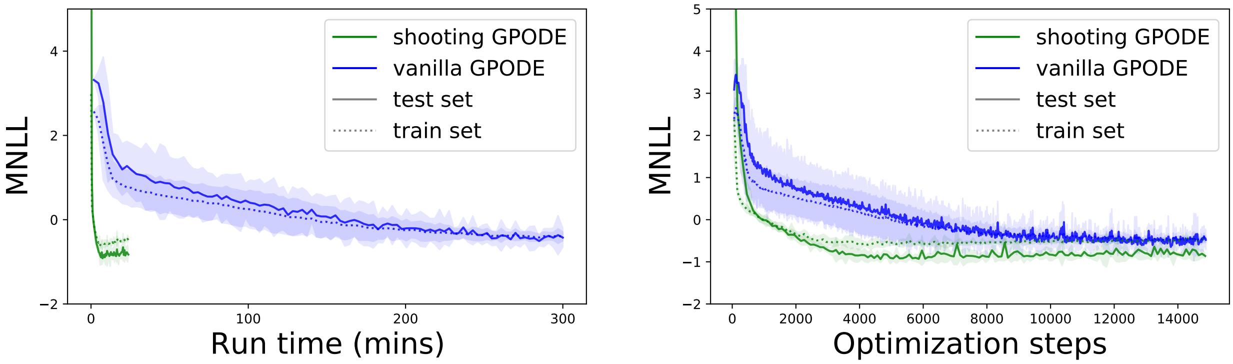
Raj Dandekar, Karen Chung, Vaibhav Dixit, Mohamed Tarek, Aslan Garcia-Valadez, Krishna Vishal Vemula, and Chris Rackauckas. Bayesian neural ordinary differential equations. *arXiv preprint arXiv:2012.07244*, 2020.

James Hensman, Nicolò Fusi, and Neil Lawrence. Gaussian processes for big data. In *Uncertainty in Artificial Intelligence*, pages 282–290, 2013.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. *Advances in Neural Information Processing Systems*, 28:1648–1656, 2015.

Diederik P Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference on Learning Representations*, 2014.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, pages 1177–1184, 2007.



(a) Convergence across wall-clock time

(b) Convergence across gradient steps during optimization

Figure 3: Optimization efficiency with GPODE models.

Michalis Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Artificial Intelligence and Statistics*, pages 567–574, 2009.

James Wilson, Viacheslav Borovitskiy, Alexander Terenin, Peter Mostowsky, and Marc Deisenroth. Efficiently sampling functions from Gaussian process posteriors. In *International Conference on Machine Learning*, pages 10292–10302, 2020.