Identification of Gaussian Process State Space Models

Stefanos Eleftheriadis[*]
stefanos@prowler.io

Thomas F.W. Nicholson[*]
tom@prowler.io

Marc Peter Deisenroth[*†]
marc@prowler.io

James Hensman[*]
james@prowler.io

Abstract

The Gaussian process state space model (GPSSM) is a non-linear dynamical system, where unknown transition and/or measurement mappings are described by GPs. Most research in GPSSMs has focussed on the state estimation problem. However, the key challenge in GPSSMs has not been satisfactorily addressed yet: system identification. To address this challenge, we impose a structured Gaussian variational posterior distribution over the latent states, which is parameterised by a recognition model in the form of a bi-directional recurrent neural network. Inference with this structure allows us to recover a posterior smoothed over the entire sequence(s) of data. We provide a practical algorithm for efficiently computing a lower bound on the marginal likelihood using the reparameterisation trick. This additionally allows arbitrary kernels to be used within the GPSSM. We demonstrate that we can efficiently generate plausible future trajectories of the system we seek to model with the GPSSM, requiring only a small number of interactions with the true system.

1 Introduction

State space models can effectively address the problem of learning patterns and predicting behaviour in sequential data. Due to their modelling power they have a vast applicability in various domains of science and engineering such as robotics, finance, neuroscience [Brown et al., 1998].

Most research and applications have focussed on linear state space models for which solutions for inference (state estimation) and learning (system identification) are well established [Kalman, 1960, Ljung, 1999]. In this work, we are interested in a non-linear flavour of state space models. In particular, we consider the case where a Gaussian process is responsible for modelling the underlying dynamics. The latter is widely known as the Gaussian process state space model (GPSSM) and possesses some desired properties: firstly, due to the non-parametric nature of the GPs, the GPSSM has the promise to be effective in learning from small datasets. Hence, in many situations it can be advantageous over well-known parametric models (e.g., recurrent neural networks – RNN), as in various reinforcement learning tasks where we are interested in efficiently learning a controller from observations of multiple short episodes. In such settings, we naturally need to deal with lack of data initially. Classical system identification methods [Ljung, 1999] require many data points to find the underlying model. This is where the probabilistic nature of the GPs can be proven critical. By using a GP for the latent transitions, we can get away with an approximate model and learn a distribution over functions. Hence, we can potentially account for model errors whilst quantifying uncertainty, as discussed and empirically shown by Schneider [1997], Deisenroth et al. [2015]. The latter ensures that the system will not become overconfident in regions of the space where data are not available.

Although this seems attractive, proper system identification with the GPSSM is a challenging task. This is due to the un-identifiability problems that are caused from the fact that both states and transition

[*] PROWLER.io, Cambridge, UK.
[†] Department of Computing, Imperial College London, UK.

functions are unknown. Hence, most work has focused only on state estimation of the GPSSM. In this paper, we focus on addressing the challenge of system identification and based on recent work by Frigola et al. [2014] we propose a novel inference method for learning the GPSSM. Specifically, we approximate the entire process of the state transition function by employing the framework of variational inference. Moreover, we assume a Markov-structured Gaussian posterior distribution over the latent states. The variational posterior can be naturally combined with a recognition model based on bi-directional recurrent neural networks, which facilitates smoothing of the posterior over the entire sequence of data. We present an efficient algorithm based on the reparameterisation trick for computing the lower bound on the marginal likelihood. This significantly accelerates learning of the model and allows for the use of arbitrary non-smooth kernel functions.

2 Gaussian process state space models

In the following, we present the GPSSM, a dynamical system whose building blocks are Gaussian processes. We consider a dynamical system

$$\boldsymbol{x}_t = f(\boldsymbol{x}_{t-1}, \boldsymbol{a}_{t-1}) + \boldsymbol{\epsilon}_f, \quad \boldsymbol{y}_t = g(\boldsymbol{x}_t) + \boldsymbol{\epsilon}_g, \tag{1}$$

where t indexes time, $\boldsymbol{x} \in \mathbb{R}^D$ is a latent state, $\boldsymbol{a} \in \mathbb{R}^P$ are control signals (actions) and $\boldsymbol{y} \in \mathbb{R}^O$ are measurements/observations. Furthermore, we assume i.i.d. Gaussian system/measurement noise $\boldsymbol{\epsilon}_{(\cdot)} \sim \mathcal{N}\big(\boldsymbol{0}, \sigma^2_{(\cdot)}\boldsymbol{I}\big)$. The state-space model eq. (1) can be fully described by the measurement and transition functions, g and f, respectively.

The key idea of a GPSSM is to model the transition function f and/or the measurement function g in eq. (1) using Gaussian processes. A Gaussian process (GP) is a distribution over functions and is fully specified by a mean $\eta(\cdot)$ and a covariance function $k(\cdot,\cdot)$ [see e.g. Rasmussen and Williams, 2006]. The covariance function allows us to encode basic structural assumptions of the class of functions we want to model, e.g., smoothness, periodicity or stationarity.

A common choice for a covariance function is the radial basis function (RBF) $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma_f^2 \exp\big(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}')^\top \boldsymbol{\Lambda}^{-1}(\boldsymbol{x} - \boldsymbol{x}')\big)$, which encodes the assumption that the underlying function is infinitely differentiable. Here, σ_f is a parameter controlling the amplitude of the function, and $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1^2, \ldots, \lambda_D^2)$ is a diagonal matrix of squared length-scales λ_i.

In this work we make use of two important properties of GPs. Let $f(\cdot)$ denote a Gaussian process random function, and $\boldsymbol{X} = [\boldsymbol{x}_i]_{i=1}^N$ be a series of points in the domain of that function. Then, any finite subset of function evaluations are jointly Gaussian distributed, if $\boldsymbol{f} = [f(\boldsymbol{x}_i)]_{i=1}^N$ then

$$p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}\big(\boldsymbol{f} \,|\, \boldsymbol{\eta}(\boldsymbol{X}),\, \boldsymbol{K}_{XX}\big)\,, \tag{2}$$

where the matrix $\boldsymbol{K}_{XX}$ contains evaluations of the kernel function at all pairs of datapoints in $\mathbf{X}$, and $\boldsymbol{\eta}(\boldsymbol{X})$ is vector containing evaluations of the prior mean function at all points. This property leads to the widely used GP regression model: if additive Gaussian noise is assumed, the marginal likelihood can be computed in closed form, enabling learning of the kernel parameters.

The second important GP property is that the conditional distribution of a Gaussian process is another Gaussian process. If we are to observe the values $\boldsymbol{f}$ at the input locations $\boldsymbol{X}$, then we predict the values elsewhere on the GP at $\boldsymbol{x}_\star$ using the conditional

$$f(\boldsymbol{x}_\star) \,|\, \boldsymbol{f} \sim \mathcal{GP}\big(\eta(\boldsymbol{x}_\star) + k(\boldsymbol{x}_\star, \boldsymbol{X})\boldsymbol{K}_{XX}^{-1}(\boldsymbol{f} - \boldsymbol{\eta}(\boldsymbol{X})),\, k(\boldsymbol{x}_\star, \boldsymbol{x}_\star) - k(\boldsymbol{x}_\star, \boldsymbol{X})\boldsymbol{K}_{XX}^{-1}k(\boldsymbol{X}, \boldsymbol{x}_\star)\big)\,. \tag{3}$$

In the GPSSM, we are presented with neither values of the function on which to condition, nor on *inputs* to the function, since the hidden states $\boldsymbol{x}_t$ are latent. The challenge of inference in the GPSSM lies in dually inferring the latent variables $\boldsymbol{x}$ and in fitting the Gaussian process dynamics $f(\boldsymbol{x})$.

In the GPSSM, we place independent GP priors on the transition function f in the state-space model from eq. (1) for each output dimension of $\boldsymbol{x}_{t+1}$, and collect realisations of those functions in the random variables $\boldsymbol{f}$, such that

$$f_d(\cdot) \sim \mathcal{GP}\big(\eta_d(\cdot),\, k_d(\cdot,\cdot)\big), \quad \boldsymbol{f}_t = [f_d(\tilde{\boldsymbol{x}}_{t-1})]_{d=1}^D \quad \text{and} \quad p(\boldsymbol{x}_t|\boldsymbol{f}_t) = \mathcal{N}(\boldsymbol{x}_t|\boldsymbol{f}_t, \sigma_f^2\boldsymbol{I}), \tag{4}$$

where we used the short-hand notation $\tilde{\boldsymbol{x}}_t = [\boldsymbol{x}_t, \boldsymbol{a}_t]$ to collect the state-action pair at time t. In this work we use a mean function that keeps the state constant, so $\eta_d(\tilde{\boldsymbol{x}}_t) = \boldsymbol{x}_t^{(d)}$.

To reduce some of the un-identifiability problems of GPSSMs, we assume a linear measurement mapping g so that the data conditional is

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t|\boldsymbol{W}_g\boldsymbol{x}_t + \boldsymbol{b}_g, \sigma_g^2\boldsymbol{I})\,. \tag{5}$$

The linear observation model $g(\boldsymbol{x}) = \boldsymbol{W}_g\boldsymbol{x} + \boldsymbol{b}_g + \boldsymbol{\epsilon}_g$ is not limiting, since a non-linear g could be replaced by additional dimensions in the state space [Frigola-Alcade, 2015].

2.1 Related work

State estimation in GPSSMs has been proposed by [Ko and Fox, 2009a, Deisenroth et al., 2009] for filtering and by [Deisenroth et al., 2012, Deisenroth and Mohamed, 2012] for smoothing using both deterministic (e.g., linearisation) and stochastic (e.g., particles) approximations. These approaches did not focus on system identification (parameter learning) but on inference in learned GPSSMs. This can be attributed to the fact that learning of the state transition function f without observing the system's true state $\boldsymbol{x}$ is challenging.

Towards this approach, Wang et al. [2008], Ko and Fox [2009b], Turner et al. [2010] proposed methods for learning GPSSMs based on maximum likelihood estimation. Frigola et al. [2013] followed a fully Bayesian treatment to the problem and proposed an inference mechanism based on particle Markov chain Monte Carlo. Specifically, they first obtain sample trajectories from the smoothing distribution that could be used to define a predictive density via Monte Carlo integration. Then, conditioned on this trajectory they sample the model's hyper-parameters. The downside of this approach is the expensive inference, whose computational cost scales proportionally to the length of the time series and the number of the particles. In order to tackle this inefficiency, Frigola et al. [2014] suggested to follow a hybrid inference approach combining variational inference and sequential Monte Carlo. Using the sparse variational framework from [Titsias, 2009] to approximate the GP led to a tractable distribution over the state transition function that is independent of the length t.

An alternative to learning a state-space model is to follow an autoregressive strategy as in Murray-Smith and Girard [2001], Likar and Kocijan [2007], Turner [2011], Roberts et al. [2013], Kocijan [2016], in order to model the function as a direct mapping from previous to current observations. However, such an autoregressive structure can be problematic: as it is learned over the true observations and not in a latent space, noise is propagated through the system during inference. To alleviate this, Mattos et al. [2015] proposed the recurrent GP, a non-linear dynamical model that resembles a deep GP mapping from observed inputs to observed outputs, with an autoregressive structure on the intermediate latent states. They further followed the idea from [Dai et al., 2015] and introduced a recognition model to approximate the true posterior of the latent state. More specifically, they suggested to use an RNN to model the means of the variational approximate distribution at each layer as a function of past latent states from previous layers. A downside to this approach is the need to update the approximation to the posterior after observing new data. Hence, performing inference for future points under the recurrent GP requires to feed forward the future actions into the RNN, in order to propagate uncertainty towards the outputs. Another issue stems from the model's inefficiency in analytically computing expectations of the kernel functions under the approximate posterior when dealing with high-dimensional latent states. Recently, Al-Shedivat et al. [2016], inspired by the promising results of the recurrent GPs [Mattos et al., 2015], introduced a recurrent structure to the manifold GP [Calandra et al., 2016]: They proposed to use an LSTM in order to map the observed inputs onto a non-linear manifold, which is the space that the GP actually operates on. For efficient inference they follow an approximate inference scheme based on Kronecker products over Toeplitz-structured kernels.

A common theme in the above work is their attempt to fit models to noisy sequences. This causes uncertainty to appear on both inputs and outputs of the GP. McHutchon and Rasmussen [2011] proposed to cope with this by taking a local linearisation of the function and using it to propagate uncertainty from the inputs to the output of the GP, where it can be dealt with more naturally.

3 Inference

Our inference scheme uses varitational Bayes [see e.g. Beal, 2003, Blei et al., 2017]. We first define the form of the approximation to the posterior (denoted $q(...)$), and then derive the evidence lower bound (ELBO), with respect to which the posterior approximation is optimized in order to minimize the Kullback-Leibler divergence between the approximate and true posteriors. We detail how the ELBO is estimated in a stochastic fashion and optimized using gradient-based methods, and describe how the form of the approximate posterior is gived by a recurrent neural network. The graphical models of the GPSSM and our proposed approximation are shown in Figure 1.

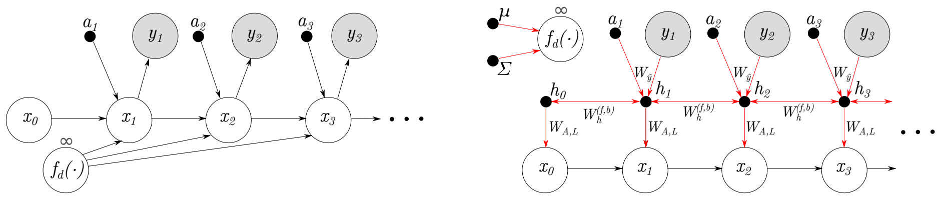

Figure 1: The GPSSM with the GP state transition functions (left), and the proposed approximation with the recognition model in the form of a bi-RNN (right). Black arrows show conditional dependencies of the model, red arrows show the data-flow in the recognition.

3.1 Posterior approximation

Following [Frigola et al., 2014], we adopt a variational approximation to the posterior, assuming factorisation between the latent functions $f(\cdot)$ and the state trajectories $\boldsymbol{X}$. However, unlike Frigola et al.'s work, we do not run particle MCMC to approximate the state trajectories, but instead assume that the posterior over states is given by a Markov-structured Gaussian distribution parameterised by a recognition model (see section 3.3). In concordance with [Frigola et al., 2014], we adopt a sparse variational framework to approximate the GP. The sparse approximation allows us to deal with both (a) the unobserved nature of the GP inducing inputs and (b) any potential computational scaling issues with the GP by controlling the number of inducing points in the approximation.

The variational approximation to the GP posterior is formed as follows: Let $\boldsymbol{Z} = [\boldsymbol{z}_1, \dots, \boldsymbol{z}_M]$ be some selected points in the same domain as $\tilde{\boldsymbol{x}}$. For each Gaussian process $f_d(\cdot)$, we collect evaluations of the function at $\boldsymbol{Z}$ into the inducing variables $\boldsymbol{u}_d = [f_d(\boldsymbol{z}_m)]_{m=1}^M$, meaning that the density of $\boldsymbol{u}_d$ under the GP prior is $\mathcal{N}(\boldsymbol{\eta}_d(\boldsymbol{Z}), \boldsymbol{K}_{zz})$. We make a mean-field variational approximation to the posterior for $\boldsymbol{U}$, taking the form $q(\boldsymbol{U}) = \prod_{d=1}^D \mathcal{N}(\boldsymbol{u}_d \,|\, \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d)$. The variational posterior of the *rest* of the points on the GP is assumed to be given by the same conditional distribution as the prior:

$$f_d(\cdot) \,|\, \boldsymbol{u}_d \sim \mathcal{GP}\big(\eta_d(\cdot) + k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}(\boldsymbol{u}_d - \boldsymbol{\eta}_d(\boldsymbol{Z})), \quad k(\cdot,\cdot) - k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}k(\boldsymbol{Z}, \cdot)\big)\,. \tag{6}$$

Integrating this expression with respect to the prior distribution $p(\boldsymbol{u}_d) = \mathcal{N}(\boldsymbol{\eta}_d(\boldsymbol{Z}), \boldsymbol{K}_{ZZ})$ gives the GP prior in eq. (4). Integrating with respect to the variational distribution $q(\boldsymbol{U})$ gives our approximation to the posterior process $f_d(\cdot) \sim \mathcal{GP}\big(\mu_d(\cdot), v_d(\cdot,\cdot)\big)$, with

$$\mu_d(\cdot) = \eta_d(\cdot) + k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}(\boldsymbol{\mu}_d - \boldsymbol{\eta}_d(\boldsymbol{Z})), v_d(\cdot,\cdot) = k(\cdot,\cdot) - k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}[\boldsymbol{K}_{zz} - \boldsymbol{\Sigma}_d]\boldsymbol{K}_{zz}^{-1}k(\boldsymbol{Z}, \cdot)\,. \tag{7}$$

The approximation to the state trajectory posterior is assumed to have a Gauss-Markov structure:

$$q(\boldsymbol{x}_0) = \mathcal{N}\big(\boldsymbol{x}_0 \,|\, \boldsymbol{m}_0, \boldsymbol{L}_0\boldsymbol{L}_0^\top\big), \quad q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}) = \mathcal{N}\big(\boldsymbol{x}_t \,|\, \boldsymbol{A}_t\boldsymbol{x}_{t-1}, \boldsymbol{L}_t\boldsymbol{L}_t^\top\big)\,. \tag{8}$$

This distribution is specified through a single mean vector $\boldsymbol{m}_0$, a series of square matrices $\boldsymbol{A}_t$, and a series of lower-triangular matrices $\boldsymbol{L}_t$. Our assumption that the state-trajectory can be modelled as Gaussian is not true in general, though our experiments suggest that the approximation is successful.

With the approximating distributions for the variational posterior defined in eq. (7) and (8), we are ready to derive the evidence lower bound (ELBO) on the model's true likelihood. Following [Frigola-Alcade, 2015] equation 5.10, the ELBO is given by

$$\begin{aligned} \text{ELBO} = \;& \mathbb{E}_{q(\boldsymbol{x}_0)}[\log p(\boldsymbol{x}_0)] + \text{H}[q(\boldsymbol{X})] - \text{KL}[q(\boldsymbol{U}) \,||\, p(\boldsymbol{U})] \\ & + \mathbb{E}_{q(\boldsymbol{X})}\Big[\sum_{t=1}^T \sum_{d=1}^D -\frac{1}{2\sigma_f^2} v_d(\tilde{\boldsymbol{x}}_{t-1}, \tilde{\boldsymbol{x}}_{t-1}) + \log \mathcal{N}\big(x_t^{(d)} \,|\, \mu_d(\tilde{\boldsymbol{x}}_{t-1}), \sigma_f^2\big)\Big] \\ & + \mathbb{E}_{q(\boldsymbol{X})}\Big[\sum_{t=1}^T \log \mathcal{N}\big(\boldsymbol{y}_t \,|\, g(\boldsymbol{x}_t), \sigma_g^2 \boldsymbol{I}_O\big)\Big]\,, \end{aligned} \tag{9}$$

where KL[·||·] is the Kullback-Leibler divergence between two distributions, and H[·] denotes the entropy of a distribution. Note that with the above formulation we can naturally deal with multiple episodic data since the ELBO can be factorised across each independent episode. We can now learn the GPSSM by optimising the ELBO w.r.t. the parameters of the model and the variational parameters. A full derivation is provided in the supplementary material.

The form of the ELBO justifies the Markov-structure that we have assumed for the variational distribution $q(\boldsymbol{X})$: we see that the latent states only interact over pair-wise time steps $\boldsymbol{x}_t$ and $\boldsymbol{x}_{t-1}$, so adding further structure to $q(\boldsymbol{X})$ is unnecessary.

3.2 Efficient computation of the ELBO

To compute the ELBO in eq. (9), we need to compute expectations w.r.t. $q(\boldsymbol{X})$. Frigola et al. [2014] showed that for the RBF kernel the relevant expectations can be computed in closed form in a similar way to Titsias and Lawrence [2010]. To allow for general kernels, we propose to use the reparameterisation trick [Kingma and Welling, 2013] instead: by sampling a single trajectory from $q(\boldsymbol{X})$ and evaluating the integrand in eq. (9), we obtain an unbiased estimate of the ELBO. To draw a sample from the Gauss-Markov structure in eq. (8), we first sample $\boldsymbol{\epsilon}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}),\ t = 0 \ldots T$, and then apply the recursion

$$\boldsymbol{x}_0 = \boldsymbol{m}_0 + \boldsymbol{L}_0\boldsymbol{\epsilon}_0, \quad \boldsymbol{x}_t = \boldsymbol{A}_t\boldsymbol{x}_{t-1} + \boldsymbol{L}_t\boldsymbol{\epsilon}_t\,. \tag{10}$$

This simple estimator of the ELBO can then be used in optimisation using stochastic gradient methods; we have used the Adam optimizer [Kingma and Ba, 2014]. It may seem initially counter-intuitive to use a stochastic estimate of the ELBO where one is available in closed form, but this approach offers two distinct advantages. First, computation is dramatically reduced: our scheme requires $\mathcal{O}(TD)$ storage in order to evaluate the integrand in (9) at a single sample from $q(\boldsymbol{X})$. A scheme which computes the integral in closed form requires $\mathcal{O}(TM^2)$ (where M is the number of inducing variables in the sparse GP) storage in order to store the sufficient statistics of the kernel evaluations. The second advantage is that we are no longer restricted to the RBF kernel, but can use any valid kernel for inference and learning in GPSSMs. The reparameterisation trick also allows us to perform batched updates of the model parameters, amounting to doubly stochastic variational inference [Titsias and Lázaro-Gredilla, 2014], which we experimentally found to improve run-time and sample-efficiency.

Some of the elements of the ELBO in eq. (9) are still available in closed-form. To reduce the variance of the estimate of the ELBO we exploit this where possible: The entropy of the Gauss-Markov structure is: $\mathrm{H}[q(\boldsymbol{X})] = -\frac{TD}{2}\log(2\pi) - \frac{TD}{2} - \sum_{t=0}^{T}\log(\det(L_t))$; the expected data likelihood (last term in eq. (9)) can be computed easily given the marginals of $q(\boldsymbol{X})$, which are given by

$$q(\boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{m}_t, \boldsymbol{\Sigma}_t), \quad \boldsymbol{m}_t = \boldsymbol{A}\boldsymbol{m}_{t-1}, \quad \boldsymbol{\Sigma}_t = \boldsymbol{A}_t\boldsymbol{\Sigma}_{t-1}\boldsymbol{A}_t^\top + \boldsymbol{L}_t\boldsymbol{L}_t^\top\,, \tag{11}$$

and the necessary Kullback-Leibler divergences can be computed analytically also: we use the implementations from GPflow [Matthews et al., 2017].

3.3 A recurrent recognition model

The variational distribution of the latent trajectories in eq. (8) has a large number of parameters $(\boldsymbol{A}_t, \boldsymbol{L}_t)$ that grows with the length of the dataset. Further, if we wish to train a model on multiple episodes (independent data sequences sharing the same dynamics), then the number of parameters grows still. To alleviate this, we propose to use a recognition model in the form of a bi-directional recurrent neural network (bi-RNN).

A bi-RNN is a combination of two independent RNNs operating on opposite directions of the sequence. Each network is specified by two weight matrices $\boldsymbol{W}$ acting on a hidden state $\boldsymbol{h}$:

$$\boldsymbol{h}_{t+1}^{(f)} = \phi(\boldsymbol{W}_h^{(f)}\boldsymbol{h}_t^{(f)} + \boldsymbol{W}_{\tilde{y}}^{(f)}\tilde{\boldsymbol{y}}_t + \boldsymbol{b}_h^{(f)})\,, \tag{12}$$

$$\boldsymbol{h}_{t-1}^{(b)} = \phi(\boldsymbol{W}_h^{(b)}\boldsymbol{h}_t^{(b)} + \boldsymbol{W}_{\tilde{y}}^{(b)}\tilde{\boldsymbol{y}}_t + \boldsymbol{b}_h^{(b)})\,, \tag{13}$$

where $\tilde{\boldsymbol{y}}_t$ denotes the concatenation of the observed data and control actions $[\boldsymbol{y}_t, \boldsymbol{a}_t]$ and the superscripts denote the direction of the RNN. The activation function ϕ acts on each element of its argument separately, we use the $\tanh$ function. In our experiments we found that using gated recurrent units [Cho et al., 2014] improved performance of our model. We now make the parameters of the Gauss-Markov structure dependent on the sequences $h^{(f)}, h^{(b)}$, so that

$$\boldsymbol{A}_t = \text{reshape}(\boldsymbol{W}_A[\boldsymbol{h}_t^{(f)}; \boldsymbol{h}_t^{(b)}] + \boldsymbol{b}_A), \quad \boldsymbol{L}_t = \text{reshape}(\boldsymbol{W}_L[\boldsymbol{h}_t^{(f)}; \boldsymbol{h}_t^{(b)}] + \boldsymbol{b}_L)\,. \tag{14}$$

The parameters of the Gauss-Markov structure $q(\boldsymbol{X})$ are now almost completely encapsulated in the recurrent recognition model as $\boldsymbol{W}_h^{(f,b)}, \boldsymbol{W}_{\tilde{y}}^{(f,b)}, \boldsymbol{W}_A, \boldsymbol{W}_L, \boldsymbol{b}_h^{(f,b)}, \boldsymbol{b}_A, \boldsymbol{b}_L$. We only need to infer the parameters of the initial state, $\boldsymbol{m}_0, \boldsymbol{L}_0$; this is where we utilise the functionality of the bi-RNN structure. Instead of directly learning the initial state $q(\boldsymbol{x}_0)$, we can now obtain it indirectly via the output of the backward RNN. Another nice property of the proposed recognition model is that now $q(\boldsymbol{X})$ is recognised from both future and past observations, since the proposed bi-RNN recognition model can be regarded as a forward and backward sequential smoother of our variational posterior.

4 Experiments

In this section, we benchmark the proposed GPSSM approach on data from one illustrative example and two challenging non-linear data sets. Our aim is to explicitly demonstrate that we can: (i) cheaply and effortlessly benefit from the use of non-smooth kernels with our approximate inference and accurately model non-smooth transition functions; (ii) successfully learn non-linear dynamical systems even when we do not have access to the true state (partially observed inputs); (iii) sample future trajectories and generate plausible future states of the system even when trained with a small number of episodes of fully and partially observed inputs.

4.1 Non-linear System identification

We first apply our approach to a synthetic dataset generated broadly according to [Frigola et al., 2014]. The data is created using a non-linear, non-smooth transition function with additive state and observation noise according to: $p(x_{t+1}|x_t) = \mathcal{N}(f(x_t), \sigma_f^2)$, and $p(y_t|x_t) = \mathcal{N}(x_t, \sigma_g^2)$, where

$$f(x) = x_t + 1, \quad \text{if } x < 4, \qquad 13 - 2x, \quad \text{otherwise} \tag{15}$$

In our experiments we set the noise variances to $\sigma_f^2 = 0.01$ and $\sigma_g^2 = 0.1$, and generate 200 sequences (episodes) of length 10 that were used as the observed data for training the GPSSM. We used 2 inducing points (initialised uniformly across the range of the input data) for approximating the GP and 20 hidden units for the recurrent recognition model.

In this experiment, we demonstrate the ability to perform inference under the proposed model with arbitrary non-smooth kernel functions.[3] Here we compare with the following kernels: RBF, additive composition of the RBF (initial $\ell = 10$) and Matern ($\nu = \frac{1}{2}$, initial $\ell = 0.1$) [Rasmussen and Williams, 2006], arc-cosine (order 0) [Cho and Saul, 2009], and the MGP kernel [Calandra et al., 2016] (depth 5, hidden dimensions $[3, 2, 3, 2, 3]$, $\tanh$ activation, Matern ($\nu = \frac{1}{2}$) compound kernel). All kernels are additively composed with the constant kernel.

The learnt GP state transition functions are shown in Figure 2. As we can see, by using the non-smooth kernels we are able to learn accurate transitions and model the instantaneous dynamical change, as opposed to the smooth transition learnt with the RBF. Note that all non-smooth kernels place inducing points directly on the peak (at $x_t = 4$) to model the kink, whereas the RBF kernel explains this behaviour as a longer-scale wiggliness of the posterior process. An interesting finding is that when using a kernel without the RBF component the GP posterior quickly reverts to the mean function ($m(x) = x$) as we move away from the data: the short length-scales that enable them to model the instantaneous change prevent them from extrapolating downwards in the transition function. The composition of the RBF and Matern kernel benefits from long and short length scales. Therefore, it can learn the instantaneous, while also extrapolating to unseen regions of the function's domain. The posteriors can be viewed across a longer range of the function space in the supplementary material.

4.2 Modelling cart-pole dynamics

In this experiment, we demonstrate the efficacy of the proposed GPSSM on learning the non-linear dynamics of the cart-pole system from [Deisenroth and Rasmussen, 2011]. The system is composed

[3] This is the benefit of using the reparameterisation trick for approximating the ELBO (see section 3.2).

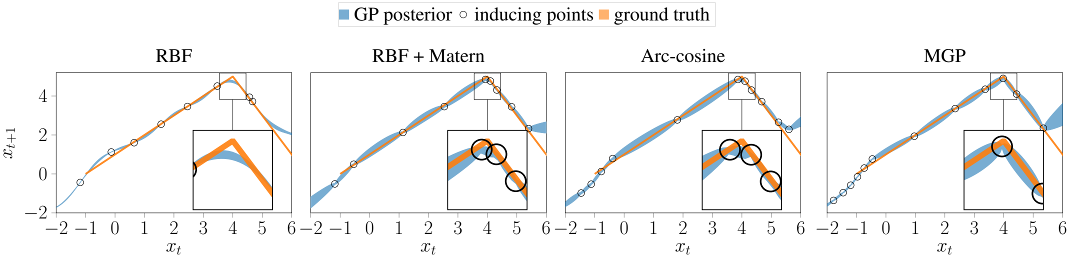

Figure 2: Visualisation of the GP state transition function learnt with different kernels. The true underlying function is given by eq. (15).

of a cart running on a track, with a freely swinging pendulum attached to it. The state of the system consists of the cart's position and velocity, and the pendulum's angle and angular velocity, while a horizontal force (action) $a \in [-10, 10]N$ can be applied to the cart. We used the data-efficient reinforcement learning algorithm from [Deisenroth and Rasmussen, 2011] to learn a feedback controller that swings the pendulum up and to balances it in the inverted position in the middle of the track. We collected trajectory data from 16 trials during learning; each trajectory/episode was $4\,\mathrm{s}$ (40 time steps) long. The 16th episode serves as the test data.

When training the GPSSM for the cart-pole system we used data up to the first 15 episodes. We used 100 inducing points to approximate the GP function with a Matern $\nu = \frac{1}{2}$ and 50 hidden units for the recurrent recognition model. The learning rate for the Adam optimiser was set to 10^{-3}. We qualitatively assess the performance of our model by feeding the control sequence of the last episode to the GPSSM in order to generate future responses.

In Figure 3, we demonstrate the ability of the proposed GPSSM to learn the underlying dynamics of the system from a different number of episodes with fully and partially observed data. In the top row, the GPSSM observes the full 4D state, while in the bottom row we train the GPSSM with only the cart's position and the pendulum's angle observed (i.e., velocities are hidden). In both cases, sampling long-term trajectories based on only 2 episodes for training does not result in plausible future trajectories. However, we could model part of the dynamics after training with only 8 episodes (320 time steps interaction with the system), while training with 15 episodes (600 time steps in total) allowed the GPSSM to produce trajectories similar to the ground truth. It is worth emphasising the fact that the GPSSM could recover the unobserved velocities in the latent states, which resulted in smooth transitions of the cart and swinging of the pendulum. Hence, the simulated behaviour was close to the ground truth. Detailed fittings for each episode and learnt latent states with observed and hidden velocities are provided in the supplementary material.

We also ran experiments using lagged actions where the current partially observed state is affected by the action two time-steps previous. The results (provided in the supplementary material) show that we are able to sample future trajectories with an accuracy similar to time-aligned actions. This indicates that our model is able to learn a compressed representation of the full state and previous inputs, essentially 'remembering' the lagged actions.

4.3 Modelling double-pendulum dynamics

Similarly to the previous experiment, in this section, we learn and model the dynamics of the double pendulum system from [Deisenroth et al., 2015]. The double pendulum is a two-link robot arm with two actuators. The state of the system consists of the angles and the corresponding angular velocities of the inner and outer link, respectively, while different torques $a_1, a_2 \in [-2, 2]$ Nm can be applied to the two actuators. The task of swinging the double pendulum and balancing it in the upwards position is extremely challenging. First, it requires the interplay of two correlated control signals (i.e., the torques). Second, the behaviour of the system, when operating at free will, is chaotic.

We follow a similar approach as with the cart-pole dataset and learn the underlying dynamics from episodic data (15 episodes, 30 time steps long each). Training of the GPSSM was performed with data up to 14 episodes, while always demonstrating the learnt underlying dynamics on the last episode, which serves as the test set. We used 200 inducing points to approximate the GP function with a

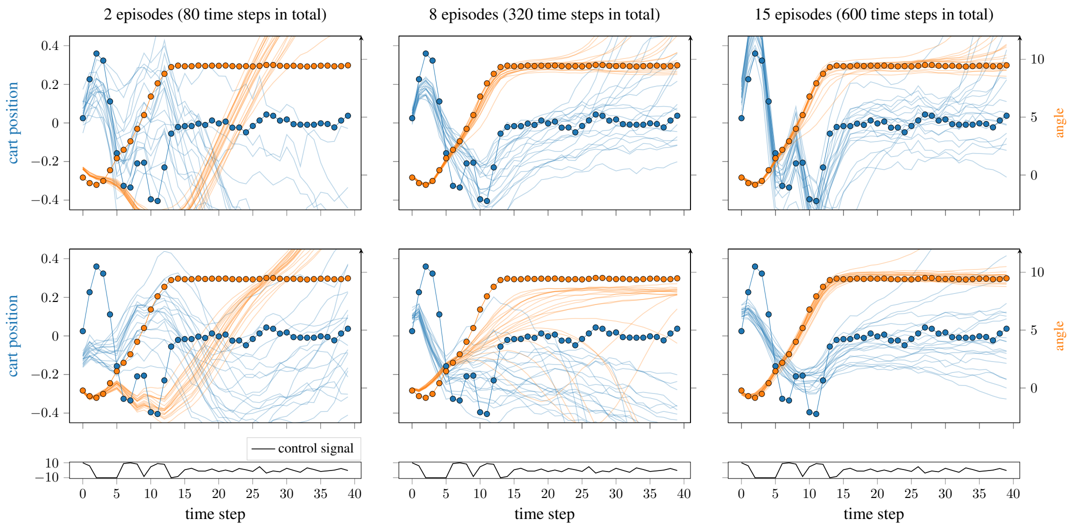

Figure 3: Predicting the cart's position and pendulum's angle behaviour from the cart-pole dataset by applying the control signal of the testing episode to sampled future trajectories from the proposed GPSSM. Learning of the dynamics is demonstrated with *observed* (upper row) and *hidden* (lower row) velocities and with increasing number of training episodes. Ground truth is denoted with the marked lines.

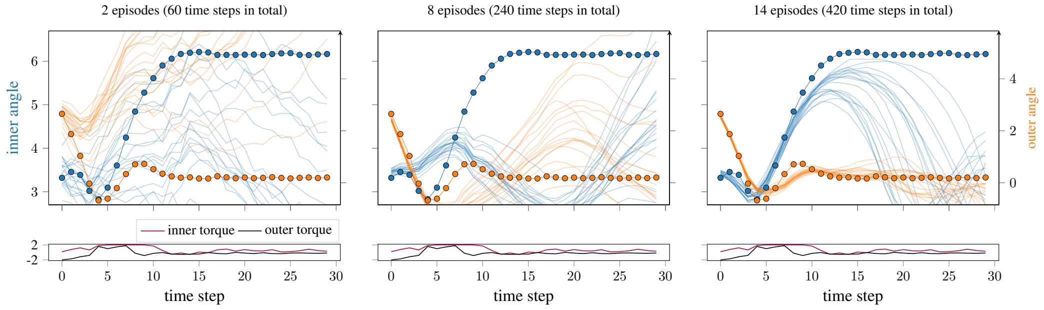

Figure 4: Predicting the behaviour of the inner and outer pendulum's angle from the double-pendulum dataset by applying the control signals of the testing episode to sampled future trajectories from the proposed GPSSM. Learning of the dynamics is demonstrated with *hidden* angular velocities (see supplementary material for observed velocities) and with increasing number of training episodes. Ground truth is denoted with the marked lines.

Matern $\nu = \frac{1}{2}$ and 80 hidden units for the recurrent recognition model. The learning rate for the Adam optimiser was set to 10^{-3}. The difficulty of the task is evident in Figure 4, where we can see that even after observing 14 episodes we cannot accurately predict the system's future behaviour for more than 15 time steps (i.e., 1.5 s). It is worth noting again we can generate reliable simulation even though we observe only the pendulums' angles.

5 Conclusion

We have proposed a novel inference mechanism for the GPSSM in order to address the challenge of non-linear system identification. We derived a variational lower bound for approximating the entire process and introduced a Gaussian posterior distribution over the latent states, which induces a Markov structure. By exploiting the reparameterisation trick in our inference we achieve computational efficiency during training, while benefiting from learning from non-smooth kernel functions. We have provided experimental evidence that our approach could identify latent dynamics, even from partial observations, while requiring only small data sets for this challenging task.

References

Maruan Al-Shedivat, Andrew G. Wilson, Yunus Saatchi, Zhiting Hu, and Eric P. Xing. Learning scalable deep kernels with recurrent structure. *arXiv preprint arXiv:1610.08936*, 2016.

Matthew J. Beal. *Variational algorithms for approximate Bayesian inference*. PhD thesis, University of London, London, UK, 2003.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, (accepted), 2017.

Emery N. Brown, Loren M. Frank, Dengda Tang, Michael C. Quirk, and Matthew A. Wilson. A statistical paradigm for neural spike train decoding applied to position prediction from ensemble firing patterns of rat hippocampal place cells. *Journal of Neuroscience*, 18(18):7411–7425, 1998.

Roberto Calandra, Jan Peters, Carl E. Rasmussen, and Marc P. Deisenroth. Manifold Gaussian processes for regression. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, 2016.

KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *CoRR*, abs/1409.1259, 2014.

Youngmin Cho and Lawrence K. Saul. Kernel methods for deep learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 342–350. 2009.

Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2015.

Marc P. Deisenroth and Shakir Mohamed. Expectation propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2618–2626, 2012.

Marc P. Deisenroth and Carl E. Rasmussen. PILCO: A model-based and data-efficient approach to policy search. In *International Conference on Machine Learning (ICML)*, pages 465–472, 2011.

Marc P. Deisenroth, Marco F. Huber, and Uwe D. Hanebeck. Analytic Moment-based Gaussian process filtering. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 225–232, June 2009.

Marc P. Deisenroth, Ryan D. Turner, Marco Huber, Uwe D. Hanebeck, and Carl E. Rasmussen. Robust filtering and smoothing with Gaussian processes. *IEEE Transactions on Automatic Control*, 57(7):1865–1871, 2012.

Marc P. Deisenroth, Dieter Fox, and Carl E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2): 408–423, 2015.

Roger Frigola, Fredrik Lindsten, Thomas B. Schön, and Carl E. Rasmussen. Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3156–3164, 2013.

Roger Frigola, Yutian Chen, and Carl E. Rasmussen. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3680–3688, 2014.

Roger Frigola-Alcade. *Bayesian time series learning with Gaussian processes*. PhD thesis, University of Cambridge, 2015.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME — Journal of Basic Engineering*, 82(Series D):35–45, 1960.

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

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2013.

Jonathan Ko and Dieter Fox. GP-BayesFilters: Bayesian filtering using Gaussian process prediction and observation models. *Autonomous Robots*, 27(1):75–90, July 2009a.

Jonathan Ko and Dieter Fox. Learning GP-BayesFilters via Gaussian process latent variable models. In *Proceedings of Robotics: Science and Systems*, June 2009b.

Juš Kocijan. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

Bojan Likar and Juš Kocijan. Predictive control of a gas-liquid separation plant based on a Gaussian process model. *Computers & chemical engineering*, 31(3):142–152, 2007.

Lennart Ljung. *System identification: Theory for the user*. Prentice Hall, 1999.

Alexander G. de G. Matthews. *Scalable Gaussian process inference using variational methods*. PhD thesis, Cambridge University, 2017.

Alexander G. de G. Matthews, James Hensman, Richard E. Turner, and Zoubin Ghahramani. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *The 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 231–239. JMLR Workshop and Conference Proceedings, 2016.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 18(40):1–6, 2017.

César Lincoln C. Mattos, Zhenwen Dai, Andreas Damianou, Jeremy Forth, Guilherme A. Barreto, and Neil D. Lawrence. Recurrent Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2015.

Andrew McHutchon and Carl E. Rasmussen. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems (NIPS)*. The MIT Press, 2011.

Roderick Murray-Smith and Agathe Girard. Gaussian process priors with ARMA noise models. In *Irish Signals and Systems Conference*, pages 147–152, 2001.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. The MIT Press, Cambridge, MA, USA, 2006.

Stephen Roberts, Michael Osborne, Mark Ebden, Steven Reece, Neale Gibson, and Suzanne Aigrain. Gaussian processes for time-series modelling. *Philosophical Transactions of the Royal Society A*, 371(1984):20110550, 2013.

Jeff G. Schneider. Exploiting model uncertainty estimates for safe dynamic control learning. In *Advances in Neural Information Processing Systems (NIPS)*. 1997.

Michalis K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, pages 567–574, 2009.

Michalis K. Titsias and Neil D. Lawrence. Bayesian Gaussian process latent variable model. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9 of *JMLR W&CP*, pages 844–851, 2010.

Michalis K. Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1971–1979, 2014.

Ryan D. Turner. *Gaussian processes for state space models and change point detection*. PhD thesis, University of Cambridge, Cambridge, UK, 2011.

Ryan D. Turner, Marc P. Deisenroth, and Carl E. Rasmussen. State-space inference and learning with Gaussian processes. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume JMLR: W&CP 9, pages 868–875, 2010.

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, 2008.

A Derivation of the ELBO

This appendix contains three parts: we first explicate the joint distribution of the model and data $p(\boldsymbol{X}, \boldsymbol{f}(\cdot), \boldsymbol{Y})$; then we describe the variational approximation to the model posterior $q(\boldsymbol{X}, \boldsymbol{f}(\cdot))$; then we show how they combine to produce the ELBO. Table 1 provides some nomenclature.

Table 1: Nomenclature used in this derivation

$t = 0 \dots T$	time steps indexed t
$d = 1 \dots D$	dimension of hidden states $\boldsymbol{x}_t$ indexed d
O	dimension of the observed data
$m = 1 \dots M$	number of inducing variables indexed m
$\boldsymbol{x}_t$	hidden state at time t, $\boldsymbol{x}_t \in \mathbb{R}^D$
$\boldsymbol{a}_t$	control input (action) at time t, $\boldsymbol{a}_t \in \mathbb{R}^P$
$\tilde{\boldsymbol{x}}_t$	concatenation of control input and state at t
$\boldsymbol{y}_t$	observation at time t, $\boldsymbol{y}_t \in \mathbb{R}^O$
$\tilde{\boldsymbol{y}}_t$	concatenation of control input and observation at t
$\boldsymbol{X}$	collection of hidden states, $= [\boldsymbol{x}_t]_{t=0}^T$.
$\boldsymbol{Y}$	collection of observations, $= [\boldsymbol{y}_t]_{t=0}^T$.
σ_f^2	variance of state transition noise
σ_n^2	variance of observation nosie
$f_d(\cdot)$	the d^{th} Gaussian process (GP)
$\boldsymbol{f}(\cdot)$	collection of GPs, $= [f_d(\cdot)]_{d=1}^D$
$\eta_d(\cdot)$	prior mean function of the d^{th} GP
$k_d(\cdot\,,\cdot)$	prior covariance function of the d^{th} GP
$\mu_d(\cdot)$	posterior mean function of the d^{th} GP
$v_d(\cdot,\cdot)$	posterior covariance function of the d^{th} GP
$\boldsymbol{Z}$	Locations of variational pseudo-inputs
$\boldsymbol{u}_d$	evaluations of the d^{th} GP at the pseudo-inputs: $\boldsymbol{u}_d = [f_d(\boldsymbol{z}_m]_{m=1}^M$.
$\boldsymbol{U}$	collection: $\boldsymbol{U} = [\boldsymbol{u}_d]_{d=1}^D$
$\boldsymbol{\mu}_d$	variational posterior mean of $\boldsymbol{u}_d$
$\boldsymbol{\Sigma}_d$	variational posterior covariance of $\boldsymbol{u}_d$
$\boldsymbol{A}_t$	variational transition matrix of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$
$\boldsymbol{L}_t$	triangular-square-root of variational covariance of $q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1})$
$\boldsymbol{m}_t$	variational mean of the marginal $q(\boldsymbol{x}_t)$
$\boldsymbol{S}_t$	variational covariance of the marginal $q(\boldsymbol{x}_t)$

A.1 Model joint distribution

Here we define the joint distribution of the the Gaussian processes f, the latent states $\boldsymbol{x}$ and the data $\boldsymbol{y}$.

The Gaussian processes have prior mean $\eta(\cdot)$ and prior covariances $k(\cdot,\,\cdot)$:

$$p(f_d(\cdot)) = \mathcal{GP}\big(\eta_d(\cdot),\, k_d(\cdot,\cdot)\big) \quad d = 1 \dots D\,. \tag{16}$$

We note that placing a measure p on the function f causes some measure-theoretic discrepancies. Nonetheless, the derivation holds following a more theoretical consideration of the problem [Matthews et al., 2016], and the intuition given by our derivation is correct.

The initial state is assumed to be drawn from a standard normal distribution

$$p(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{0},\, \boldsymbol{I}_D)\,. \tag{17}$$

The state transition depends on the Gaussian processes:

$$p(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{f}(\cdot)) = \mathcal{N}\big(\boldsymbol{x}_t \mid \boldsymbol{f}(\tilde{\boldsymbol{x}}_{t-1}),\, \sigma_f^2 \boldsymbol{I}_D\big)\,, \tag{18}$$

We assume a linear-Gaussian observation model:

$$p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t) = \mathcal{N}\big(\boldsymbol{y}_t \,|\, \boldsymbol{W}_g \boldsymbol{x}_t + \boldsymbol{b}_g, \sigma_n^2 \boldsymbol{I}_O\big) \tag{19}$$

The joint density is then

$$p(\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{Y}) = \prod_{d=1}^{D} p(f_d(\cdot))\, p(\boldsymbol{x}_0) \prod_{t=1}^{T} p(\boldsymbol{y}_t \,|\, \boldsymbol{x}_t) \prod_{t=1}^{T} p(\boldsymbol{x}_t \,|\, \boldsymbol{f}, \boldsymbol{x}_{t-1}) \tag{20}$$

A.2 Approximate posterior distribution

We will use variational Bayes to approximate the posterior distribution over $\boldsymbol{f}$ and $\boldsymbol{X}$, whilst simultaneously obtaining a bound on the marginal likelihood (the ELBO) which will be used to train the parameters of the model, including covariance function parameters, noise variances and the parameters $\boldsymbol{W}_g, \boldsymbol{b}_g$ of the linear output mapping.

The posterior over Gaussian processes takes the form of a sparse GP. We introduce a series of M variational inducing points $\boldsymbol{Z} = [\boldsymbol{z}_m]_{m=1}^M$ which lie in the same domain at $\tilde{\boldsymbol{x}}$. Following convention, the values of the d^{th} function at those points are denoted $\boldsymbol{u}_d = [f_d(\boldsymbol{z}_m)]_{m=1}^M$. Note that the variables $\boldsymbol{u}$ are not *auxiliary* variables, but part of the original model specification, being part of the GP. We assume a variational posterior of the form

$$q(\boldsymbol{U}) = \prod_{d=1}^{D} \mathcal{N}\big(\boldsymbol{u}_d \,|\, \boldsymbol{\mu}_d,\, \boldsymbol{\Sigma}_d\big)\,. \tag{21}$$

The remainder of the GPs conditioned on $\boldsymbol{u}$ are assumed to take the same form as the GP prior conditional. That is

$$q(f_d(\cdot) \,|\, \boldsymbol{u}_d) = p(f_d(\cdot) \,|\, \boldsymbol{u}_d) = \mathcal{G}P\big(\eta_d(\cdot) + k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}(\boldsymbol{u}_d - \boldsymbol{\eta}_d(\boldsymbol{Z})),\, k(\cdot,\cdot) - k(\cdot,\boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}k(\boldsymbol{Z},\cdot)\big)\,. \tag{22}$$

Marginalising with respect to $\boldsymbol{u}_d$ leads to our approximation to the GP:

$$q(f_d(\cdot)) = \mathcal{GP}\big(\mu_d(\cdot), v_d(\cdot,\cdot)\big)\,, \tag{23}$$

with

$$\mu_d(\cdot) = \eta_d(\cdot) + k(\cdot, \boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}(\boldsymbol{\mu}_d - \boldsymbol{\eta}_d(\boldsymbol{Z}))\,, \tag{24}$$

$$v_d(\cdot,\cdot) = k(\cdot,\cdot) - k(\cdot,\boldsymbol{Z})\boldsymbol{K}_{zz}^{-1}[\boldsymbol{K}_{zz} - \boldsymbol{\Sigma}_d]\boldsymbol{K}_{zz}^{-1}k(\boldsymbol{Z},\cdot)\,. \tag{25}$$

The approximation to the posterior over state trajectories is given a Gauss-Markov structure of the form

$$q(\boldsymbol{X}) = q(\boldsymbol{x}_0) \prod_{t=1}^{T} q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1})\,, \tag{26}$$

where

$$q(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0 \,|\, \boldsymbol{m}_0,\, \boldsymbol{L}_0\boldsymbol{L}_0^\top) \tag{27}$$

$$q(\boldsymbol{x}_t \,|\, \boldsymbol{x}_{t-1}) = \mathcal{N}(\boldsymbol{x}_t \,|\, \boldsymbol{A}_t\boldsymbol{x}_{t-1},\, \boldsymbol{L}_t\boldsymbol{L}_t^\top)\,. \tag{28}$$

The complete set of variational parameters is then $\boldsymbol{Z}, \{\boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d\}_{d=1}^D, \boldsymbol{m}_0, \boldsymbol{L}_0, \{\boldsymbol{A}_t, \boldsymbol{L}_t\}_{t=1}^T$. The parameters of $q(\boldsymbol{X})$ are reconfigured to be the output of an RNN recognition model (see main text), whilst we optimise the parameters controlling $\boldsymbol{f}(\cdot)$ directly.

The joint posterior then factors as

$$q(\boldsymbol{f}(\cdot), \boldsymbol{X}) = \prod_{d=1}^{D} q(f_d(\cdot)) q(\boldsymbol{X})\,. \tag{29}$$

A.3 The ELBO

Having specified the forms of the model and the approximate posterior, we are ready to derive the ELBO. Following the standard variational Bayes methods, we write

$$\mathrm{ELBO} = \mathbb{E}_{q(\boldsymbol{X})q(\boldsymbol{f}(\cdot))}\left[\log \frac{p(\boldsymbol{Y} \mid \boldsymbol{X})p(\boldsymbol{X} \mid \boldsymbol{f}(\cdot))}{q(\boldsymbol{X})} \frac{p(\boldsymbol{f}(\cdot))}{q(\boldsymbol{f}(\cdot))}\right] . \tag{30}$$

We will split the ELBO into four parts, dealing with each in turn:

$$\mathrm{ELBO} = \underbrace{\mathbb{E}_{q(\boldsymbol{X})}\big[\log p(\boldsymbol{Y} \mid \boldsymbol{X})\big]}_{\text{part 1}} + \underbrace{\mathbb{E}_{q(\boldsymbol{X})q(\boldsymbol{f}(\cdot))}\big[\log p(\boldsymbol{X} \mid \boldsymbol{f}(\cdot))\big]}_{\text{part 2}} - \underbrace{\mathbb{E}_{q(\boldsymbol{X})}\big[\log q(\boldsymbol{X})\big]}_{\text{part 3}} + \underbrace{\mathbb{E}_{q(\boldsymbol{f}(\cdot))}\big[\log \frac{p(\boldsymbol{f}(\cdot))}{q(\boldsymbol{f}(\cdot))}\big]}_{\text{part 4}} . \tag{31}$$

Part 1 This expression can be computed straight-forwardly in closed form due to our choice of a linear-Gaussian emission $g(\boldsymbol{x})$. Let $\boldsymbol{m}_t, \boldsymbol{\Sigma}_t$ be the marginals of $q(\boldsymbol{x}_t)$ computed via the recursion, and recall the form of the linear emission function $g(\boldsymbol{x}_t) = \boldsymbol{W}_g\boldsymbol{x}_t + \boldsymbol{b}_g$

$$\mathbb{E}_{q(\boldsymbol{X})}\big[\log p(\boldsymbol{Y} \mid \boldsymbol{X})\big] = \mathbb{E}_{q(\boldsymbol{X})}\Big[\sum_{t=1}^{T} \log \mathcal{N}\big(\boldsymbol{y}_t \mid g(\boldsymbol{x}_t), \sigma_g^2\big)\Big] \tag{32}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t)}\Big[\log \mathcal{N}\big(\boldsymbol{y}_t \mid \boldsymbol{W}_g\boldsymbol{x}_t + \boldsymbol{b}_g, \sigma_g^2\big)\Big] \tag{33}$$

$$= \sum_{t=1}^{T} \log \mathcal{N}\big(\boldsymbol{y}_t \mid \boldsymbol{W}_g\boldsymbol{m}_t + \boldsymbol{b}_g, \sigma_g^2\big) - \tfrac{1}{2\sigma_n^2}\mathrm{tr}(\boldsymbol{W}_g^\top\boldsymbol{W}_g\boldsymbol{\Sigma}_t) . \tag{34}$$

In practise we defer this simple computation to the `variational_expectations` functionality in GPflow [Matthews et al., 2017].

Part 2 This expression cannot be computed in closed form without restriction to the RBF kernel as in [Frigola-Alcade, 2015]. We eliminate the integral with respect to $\boldsymbol{f}$ here, and then use the reparameterisation trick to estimate the integral with respect to $\boldsymbol{X}$ (see main text).

$$\text{part 2} = \mathbb{E}_{q(\boldsymbol{X})q(\boldsymbol{f}(\cdot))}\big[\log p(\boldsymbol{X} \mid \boldsymbol{f}(\cdot))\big] \tag{35}$$

$$= \mathbb{E}_{q(\boldsymbol{X})q(\boldsymbol{f}(\cdot))}\big[\log p(\boldsymbol{x}_0)\prod_{t=1}^{T}\mathcal{N}\big(\boldsymbol{x}_t \mid \boldsymbol{f}(\tilde{\boldsymbol{x}}_{t-1}),\, \sigma_f^2\boldsymbol{I}_D\big)\big] \tag{36}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_0)}\big[\log p(\boldsymbol{x}_0)\big] + \mathbb{E}_{q(\boldsymbol{X})q(\boldsymbol{f}(\cdot))}\Big[\sum_{t=1}^{T}\sum_{d=1}^{D}\log \mathcal{N}\big(\boldsymbol{x}_t^{(d)} \mid f_d(\tilde{\boldsymbol{x}}_{t-1}),\, \sigma_f^2\big)\Big] \tag{37}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_0)}\big[\log p(\boldsymbol{x}_0)\big] + \mathbb{E}_{q(\boldsymbol{X})}\Big[\sum_{t=1}^{T}\sum_{d=1}^{D}\log \mathcal{N}\big(\boldsymbol{x}_t^{(d)} \mid \mu_d(\tilde{\boldsymbol{x}}_{t-1}),\, \sigma_f^2\big) - \tfrac{1}{2}\sigma_f^{-2} v_d(\tilde{\boldsymbol{x}}_{t-1}, \tilde{\boldsymbol{x}}_{t-1})\Big], \tag{38}$$

which matches the term in the main text.

Part 3 This corresponds to the entropy of $q(\boldsymbol{X})$. It is straightforward to derive:

$$-\mathbb{E}_{q(\boldsymbol{X})}\big[\log q(\boldsymbol{X})\big] = \mathrm{H}[q(\boldsymbol{X})] = \frac{(T+1)D}{2}\log(2\pi e) + \sum_{t=0}^{T}\log(\det(\boldsymbol{L}_t)) . \tag{39}$$

Part 4 This final part is the Kullback-Leibler divergence between the prior and (approximate) posterior GPs. We first note that it can be written as a sum across dimensions d, and then that each

GP $f_d(\cdot)$ can be factored into two parts: $p(f_d(\cdot) \,|\, \boldsymbol{u}_d)p(\boldsymbol{u}_d)$ and similarly for q. This results in

$$\mathbb{E}_{q(\boldsymbol{f}(\cdot))}\left[\log \frac{p(\boldsymbol{f}(\cdot))}{q(\boldsymbol{f}(\cdot))}\right] = \sum_{d=1}^{D} \mathbb{E}_{q(f_d(\cdot))}\left[\log \frac{p(f_d(\cdot))}{q(f_d(\cdot))}\right] \tag{40}$$

$$= \sum_{d=1}^{D} \mathbb{E}_{q(f_d(\cdot) \,|\, \boldsymbol{u}_d)q(\boldsymbol{u}_d)}\left[\log \frac{p(f_d(\cdot) \,|\, \boldsymbol{u}_d)p(\boldsymbol{u}_d)}{q(f_d(\cdot) \,|\, \boldsymbol{u}_d)q(\boldsymbol{u}_d)}\right]\, . \tag{41}$$

Since we have defined the posterior conditional $q(f_d(\cdot) \,|\, \boldsymbol{u}_d)$ to match the prior conditional, the two terms cancel, resulting in

$$\mathbb{E}_{q(\boldsymbol{f}(\cdot))}\left[\log \frac{p(\boldsymbol{f}(\cdot))}{q(\boldsymbol{f}(\cdot))}\right] = \sum_{d=1}^{D} \mathbb{E}_{q(\boldsymbol{u}_d)}\left[\log \frac{p(\boldsymbol{u}_d)}{q(\boldsymbol{u}_d)}\right] \tag{42}$$

$$= -\sum_{d=1}^{D} \mathrm{KL}\big[q(\boldsymbol{u}_d)||p(\boldsymbol{u}_d)\big]\, . \tag{43}$$

Since the result is a Kullback-Leibler divergence between two finite-dimensional normal distributions, it is computed straightforwardly.

Although this notation is somewhat sloppy (since the sets of variables $f_d(\cdot)$ and $\boldsymbol{u}_d$ overlap), the result is correct. Matthews [2017] contains a more careful and significantly more technical derivation.

B Full visualisation of synthetic 1D dataset

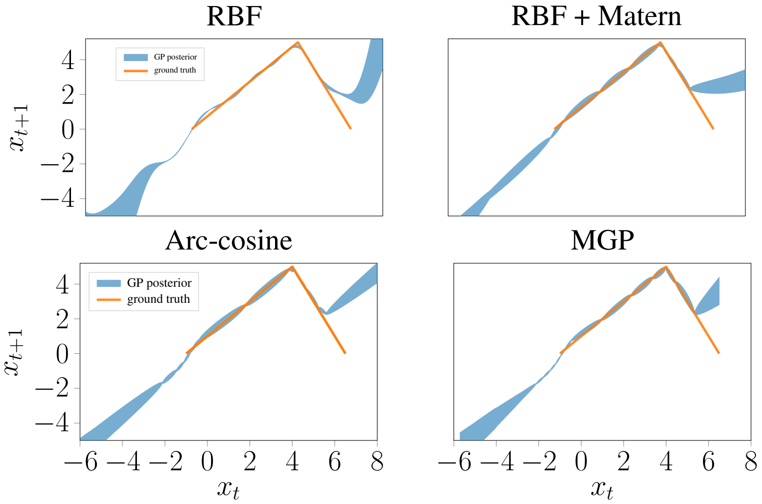

Figure 5: Visualisation of the learned GP transition functions across a greater domain of the function. It can be seen that all models revert to the mean function (defined as the identity function) away from the data. The short lengthscales of the Arc-cosine and MGP (compounded with a Matern kernel) that are used to fit the kink of the true transition function mean that they almost instantaneously revert to the mean function. The longer length scales of the RBF-containing kernels mean that we revert much more slowly to the mean.

C Modelling double-pendulum dynamics

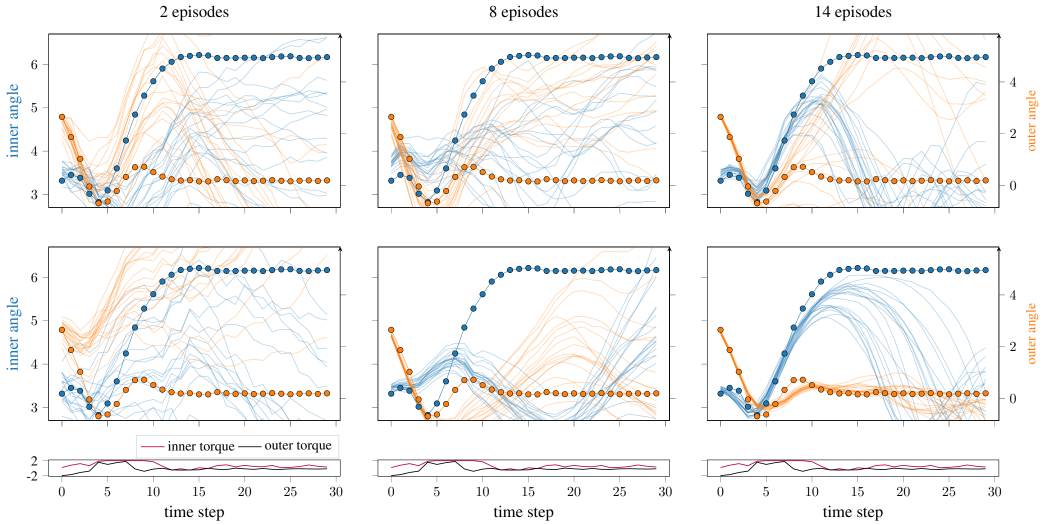

Figure 6: Predicting the chaotic behaviour of the inner and outer pendulum's angle from the double pendulum dataset by applying the control signals of the testing episode to sampled future trajectories from the proposed GPSSM. Learning of the dynamics is demonstrated with *observed* (upper row) and *hidden* (lower row) angular velocities and with increasing number of training episodes. Ground truth is denoted with the marked lines.

D Learnt latent states for cart-pole

Below we provide the learnt latent states for the cart-pole dataset with observed and hidden velocities. It is worth noting that the model has recovered similar structure for both cases.

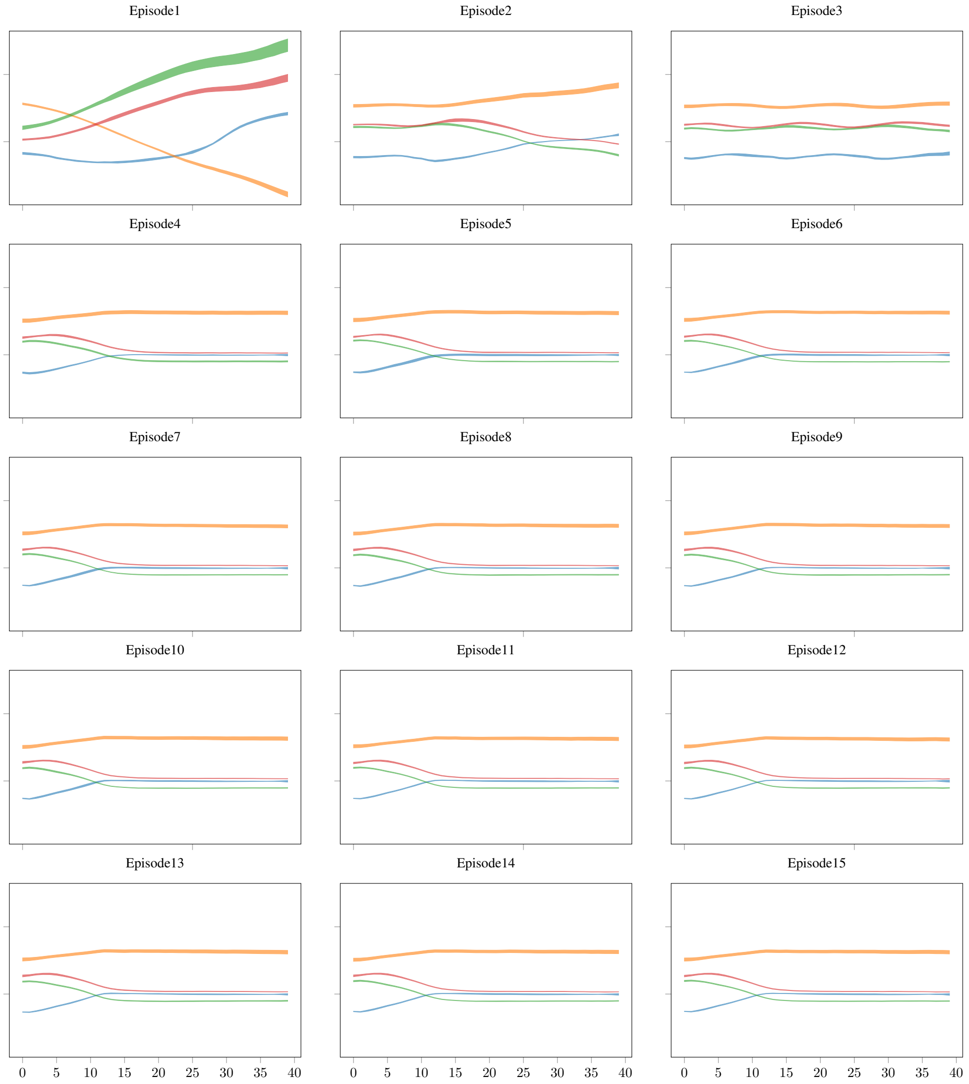

Figure 7: Learnt latent states for the cart-pole dataset with observed velocities.

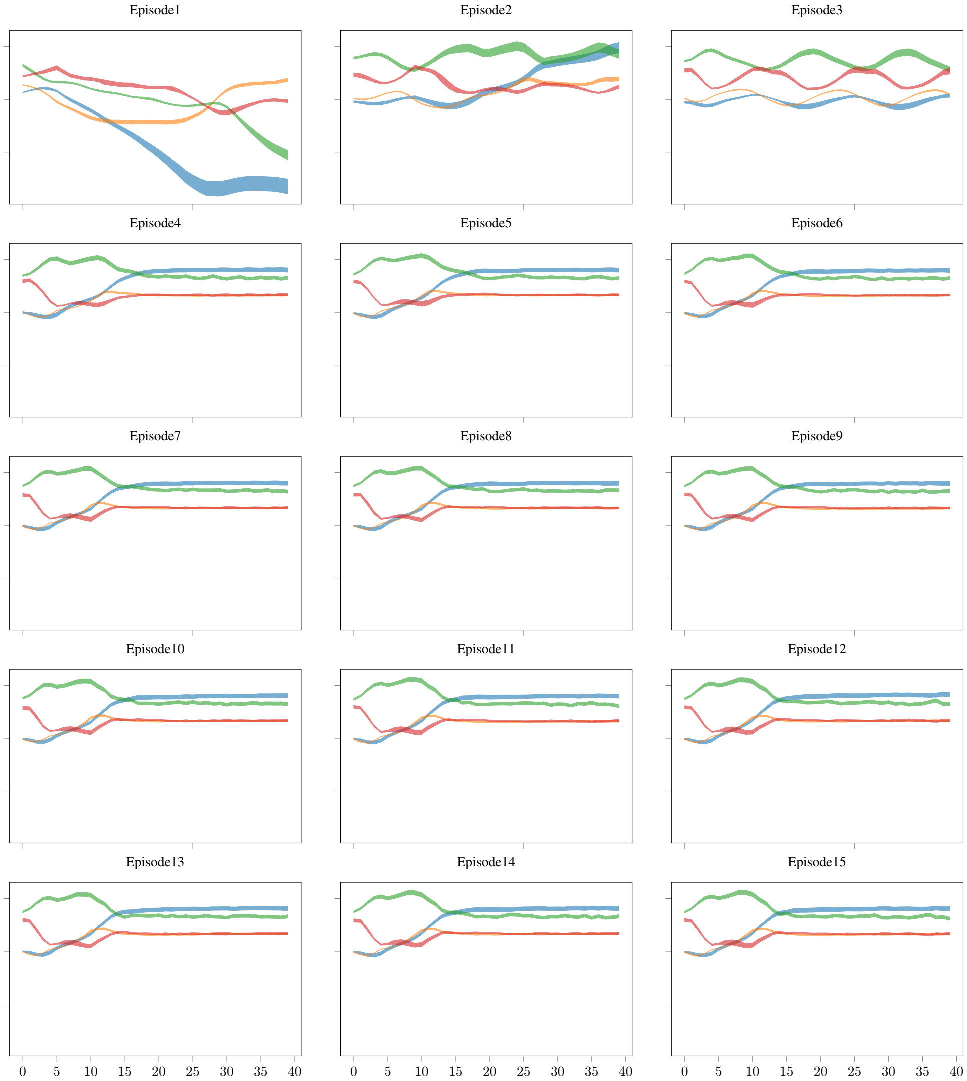

Figure 8: Learnt latent states for the cart-pole dataset with hidden velocities.

E Cart-pole training data fitting

Below we provide detailed fittings on the training episodes for the cart-pole dataset.

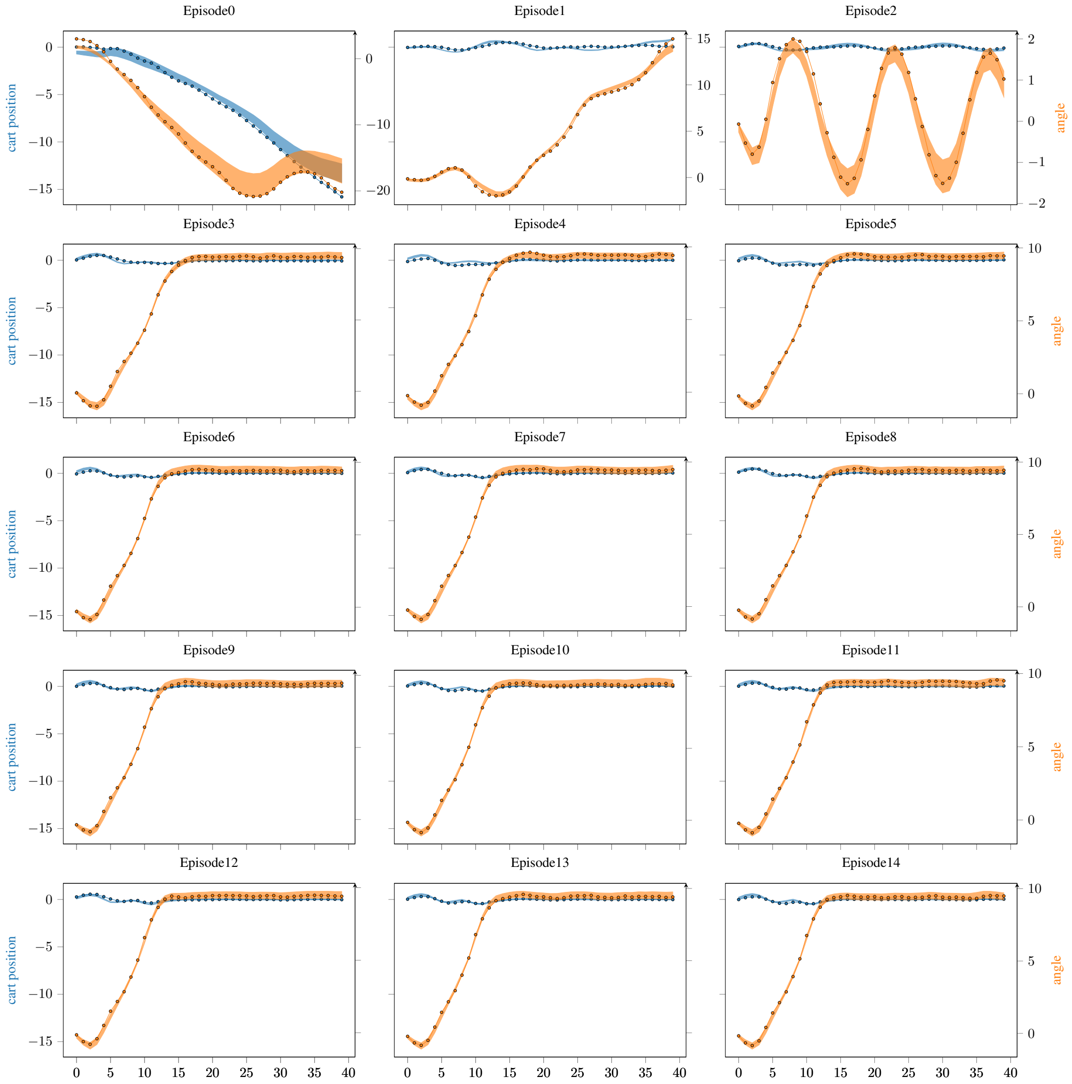

Figure 9: Detailed fittings per episode for the cart-pole dataset with observed velocities.

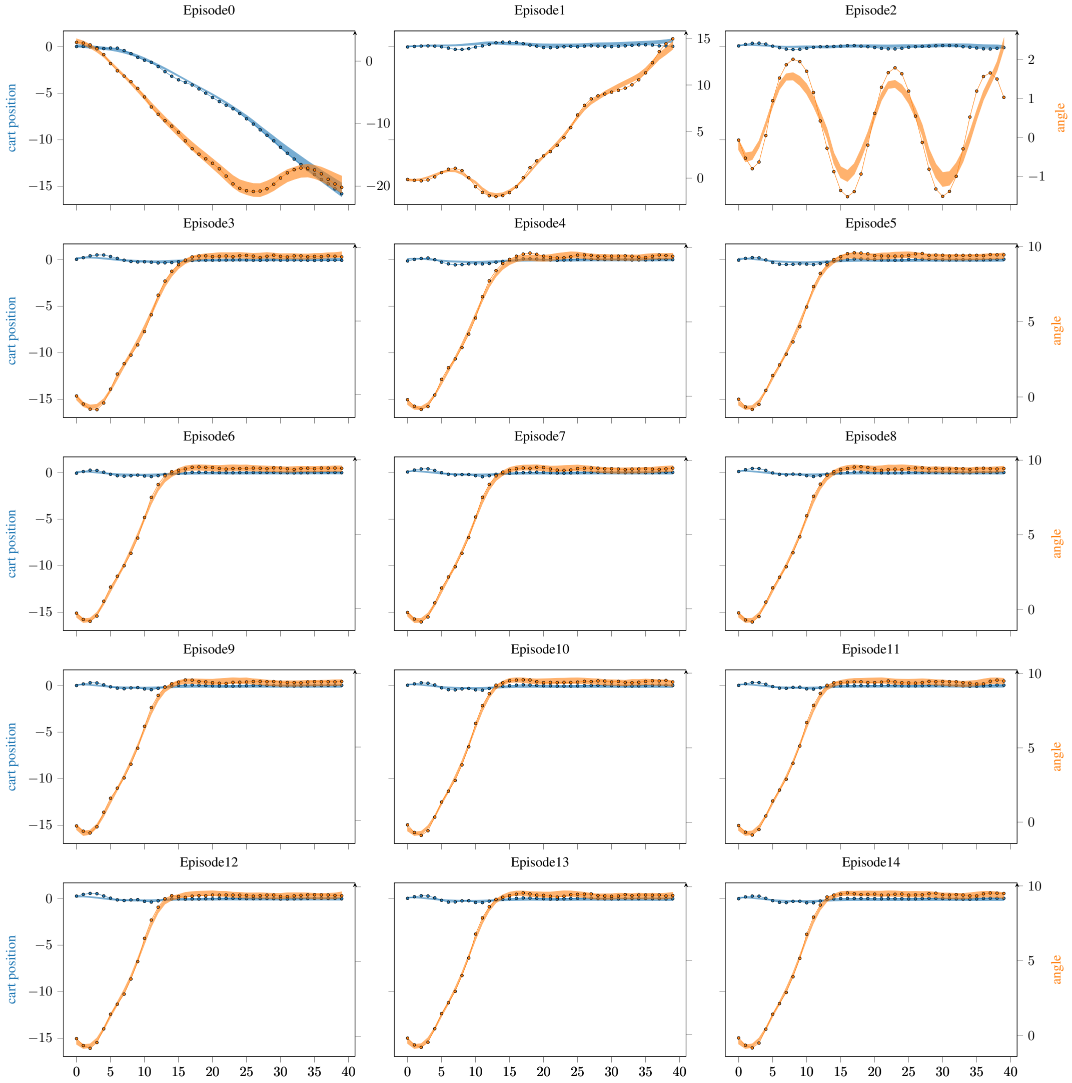

Figure 10: Detailed fittings per episode for the cart-pole dataset with hidden velocities.

F Lagged action cart-pole result

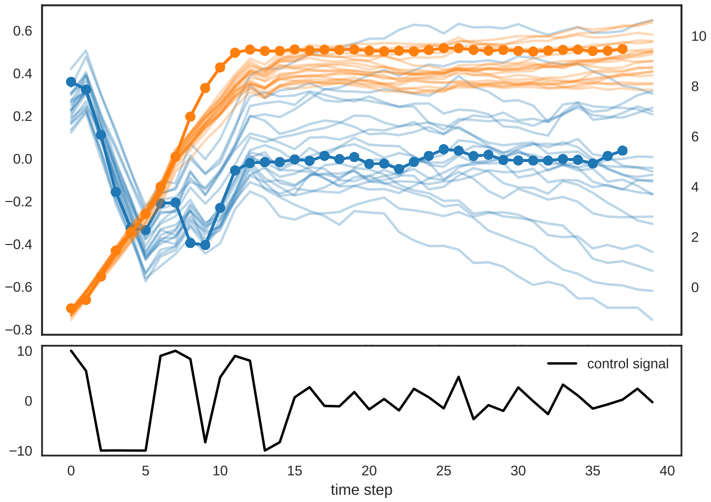

Figure 11: Results using lagged inputs on the cart-pole data