Overcoming Mean-Field Approximations
in Recurrent Gaussian Process Models

Alessandro Davide Ialongo [1 2] **Mark van der Wilk** [3] **James Hensman** [3] **Carl Edward Rasmussen** [1 3]

Abstract

We identify a new variational inference scheme for dynamical systems whose transition function is modelled by a Gaussian process. Inference in this setting has either employed computationally intensive MCMC methods, or relied on factorisations of the variational posterior. As we demonstrate in our experiments, the factorisation between latent system states and transition function can lead to a miscalibrated posterior and to learning unnecessarily large noise terms. We eliminate this factorisation by explicitly modelling the dependence between state trajectories and the low-rank representation of our Gaussian process posterior. Samples of the latent states can then be tractably generated by conditioning on this representation. The method we obtain gives better predictive performance and more calibrated estimates of the transition function, yet maintains the same time and space complexities as mean-field methods[1].

1 Introduction

Many time series are well explained by assuming their progression is determined by an underlying dynamical system. A model of this dynamical system can be used to make strong predictions of the future behaviour of the time series. Many fields require predictions of time series with unknown dynamics, such as economic models of financial markets, or models of physical systems that are to be controlled with model predictive control or model-based reinforcement learning. Particularly in low-data regimes, estimates of the uncertainty in the predictions are crucial for making robust decisions, as the expected utility of an action can depend

strongly on the distribution of the outcome (von Neumann et al., 1944). A striking example of this is model based policy search (Deisenroth & Rasmussen, 2011).

There are many approaches to time series prediction. Within the machine learning community, autoregressive (AR) (Billings, 2013) and state-space (SSMs) models are popular, in part due to the relative ease with which predictions can be made. AR models model the next observation by learning a mapping from the H last observations, while SSMs model the underlying dynamical system directly by learning a *transition function* which maps a *state* forward in time. At each point in time, the state contains sufficient information for predicting both future states and observations. While AR models can be easier to train, SSMs have the potential to be more data efficient, due to their minimal state representation.

We aim to learn complex, possibly stochastic, non-linear dynamical systems from noisy data, following the Bayesian paradigm in order to capture model uncertainty. We use a Gaussian process (GP) prior (Rasmussen & Williams, 2005) for the transition function, giving the Gaussian Process State Space Model (GPSSM) (Frigola et al., 2013). Aside from capturing uncertainty, GPs are non-parametric, guaranteeing that our model complexity will not saturate as we observe more data.

Despite the challenge of performing accurate approximate inference in the GPSSM, an impressive amount of progress has been made (Frigola et al., 2014; McHutchon, 2014; Eleftheriadis et al., 2017; Ialongo et al., 2017; Doerr et al., 2018), helped along by the development of elegant variational inference techniques (Titsias, 2009; Hensman et al., 2013) which retain the key non-parametric property of GPs. In this work, we improve variational approximations by critically examining existing methods and their failure modes. We propose a family of non-factorised variational posteriors which alleviate crucial problems with earlier approaches.

2 Background

State-space models are common in machine learning, and appear in many forms. At the most basic level, linear-Gaussian state-space models can be learned by maximum likelihood, combining Kalman smoothing and EM for in-

---

[1]Computational and Biological Learning Group, University of Cambridge [2]Max Planck Institute for Intelligent Systems, Tübingen [3]PROWLER.io. Correspondence to: Alessandro Davide Ialongo <adi24@cam.ac.uk>.

[1]Code: https://github.com/ialong/GPt.

stance (Roweis & Ghahramani, 1999). Extensions have been developed for deterministic non-linear transitions (Ghahramani & Roweis, 1999). Recurrent neural networks, like the popular LSTM (Hochreiter & Schmidhuber, 1996) learn deterministic mappings with state space structure for sequence prediction, which have seen successful recent use in a wide range of tasks including translation (Sutskever et al., 2014).

2.1 Bayesian non-parametric approaches

We are particularly interested in prediction tasks which require good estimates of uncertainty, for example, for use in model-based reinforcement learning systems (Deisenroth & Rasmussen, 2011). These applications distinguish themselves by requiring 1) continual learning, as datasets are incrementally gathered, and 2) uncertainty estimates to ensure that policies can be learned which are robust to the many dynamics that are consistent with the small dataset. Bayesian non-parametric models provide a unified and elegant way of solving both these problems: model complexity scales with the size of the dataset and is controlled by the Bayesian *Occam's razor* (Rasmussen & Ghahramani, 2000). In this work, we focus on approximations (Titsias, 2009; Hensman et al., 2013; Matthews et al., 2016) which offer improved computational scaling with the ability to fully recover the original non-parametric model.

2.2 Gaussian process State Space Models

Over the years, recurrent models with GP transitions (GPSSM) have been proposed in a variety of ways, each with its own inference method (Wang et al., 2005; Ko & Fox, 2009; Turner et al., 2010; Frigola et al., 2013). Early methods relied on *maximum a posteriori* inference for either the latent states (Wang et al., 2005) or the transition function (Wang et al., 2008; Turner et al., 2010). Frigola et al. (2013) presented the first fully Bayesian treatment with a particle MCMC method that sampled over the latent states and the GP transition function. Due to issues with computational complexity and sampling in higher dimensional latent spaces, later attention turned to variational methods. All variational inference schemes have so far relied on independence assumptions between latent states and transition function (Frigola et al., 2014; McHutchon, 2014; Ialongo et al., 2017), sometimes even factorising the state distribution over time (Mattos et al., 2016). Eleftheriadis et al. (2017) introduced a recognition model to help with optimisation of the variational distribution, while keeping the mean-field assumption. Recently, Doerr et al. (2018) introduced the first method to account for the dependence between transition function and latent states by using a doubly stochastic inference scheme similar to Salimbeni & Deisenroth (2017). However, their approach has severe limitations which we will discuss and highlight experimentally. Bui et al. (2016) investigated Power Expectation Propagation as an alternative approach for fitting factorised approximate posteriors.

2.3 Variational inference approaches

Variational methods often make simplifying assumptions about the form of the approximate posterior to improve computational tractability. This may result in significantly biased solutions. The bias is particularly severe if independence assumptions are made where strong correlations are actually present (Turner & Sahani, 2011). Clearly, in dynamical systems, the path the latent states take depends strongly on the dynamics. Hence, we focus on improving existing variational inference schemes by removing the independence assumption between latent states and transition function. We identify a general method that performs well across varying noise scales.

3 The Model

3.1 State-space model structure

Conceptually, a GPSSM is like other state-space models. We model discrete-time sequences of observations $Y = \{\mathbf{y}_t\}_{t=1}^T$, where $\mathbf{y}_t \in \mathbb{R}^E$, by positing that each data point is generated by a corresponding latent variable $\mathbf{x}_t \in \mathbb{R}^D$ (the system's "state"). These latent variables form a Markov chain, implying that, for any time-step t, we can generate $\mathbf{x}_{t+1}$ by conditioning only on $\mathbf{x}_t$ and the transition function f. We take the system to evolve according to a single, time-invariant transition function, which we would like to infer. While our inference scheme provides the freedom to choose the transition $p(\mathbf{x}_{t+1}|f, \mathbf{x}_t)$ and observation $p(\mathbf{y}_t|\mathbf{x}_t)$ distributions to any density we can evaluate, we use Gaussian distributions here. We also assume a linear mapping between $\mathbf{x}_t$ and the mean of $\mathbf{y}_t \,|\, \mathbf{x}_t$. This does not limit the range of systems that can be modelled, though it may require us to choose a higher dimensional latent state (Frigola, 2015). We make this choice to reduce the non-identifiabilities between transitions and emissions. To reduce them further we may also impose constraints on the scale of f or of the linear mapping $\mathbf{C}$. This allows the model to be expressed by the following equations:

$$f \sim \mathcal{GP}(m(\cdot), k(\cdot, \cdot)) \tag{1}$$

$$\mathbf{x}_1 \sim \mathcal{N}\left(\boldsymbol{\mu}_{p_1}, \boldsymbol{\Sigma}_{p_1}\right) \tag{2}$$

$$\mathbf{x}_{t+1} \,|\, f, \mathbf{x}_t \sim \mathcal{N}(f(\mathbf{x}_t), \mathbf{Q}) \tag{3}$$

$$\mathbf{y}_t \,|\, \mathbf{x}_t \sim \mathcal{N}(\mathbf{C}\mathbf{x}_t + \mathbf{d}, \mathbf{R}) \tag{4}$$

where we take the "process noise" covariance matrix $\mathbf{Q}$ to be diagonal, encouraging the transition function to account for all correlations between latent dimensions. The crucial difference from a standard state-space model is that we have placed a GP prior over the transition function. Thus, for a given mean function $m(\cdot)$ and positive-definite covariance function $k(\cdot, \cdot)$, any finite collection of function evaluations, at arbitrary inputs $Z = \{\mathbf{z}_n\}_{n=1}^N, \mathbf{z}_n \in \mathbb{R}^D$, will have a

Gaussian joint distribution:

$$p(f(Z)) \triangleq p(f(\mathbf{z}_1), \ldots, f(\mathbf{z}_N)) = \mathcal{N}(\boldsymbol{m}_Z, K_{Z,Z}) \quad (5)$$

where $[\boldsymbol{m}_Z]_{n,1} = m(\mathbf{z}_n)$ and $[K_{Z,Z}]_{m,n} = k(\mathbf{z}_m, \mathbf{z}_n)$. This gives the joint density of our model:

$$p(X, Y, f(X)) =$$
$$p(x_1)p(f(X)) \prod_{t=1}^{T-1} p(\mathbf{x}_{t+1} \mid f, \mathbf{x}_t) \prod_{t=1}^{T} p(\mathbf{y}_t \mid \mathbf{x}_t). \quad (6)$$

In order to generate samples from this model, the function value $f(\mathbf{x}_{t+1})$ for input $\mathbf{x}_{t+1}$ has to be sampled by conditioning on all the previously sampled function values:

$$p(f(\mathbf{x}_{t+1}) \mid f(\mathbf{x}_{1:t})) = \mathcal{N}\left(f(\mathbf{x}_{t+1}) \mid \boldsymbol{\mu}_{f_{t+1}}, \boldsymbol{\Sigma}_{f_{t+1}}\right) \quad (7)$$

$$f(\mathbf{x}_{1:t}) \triangleq [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_t)]^\mathsf{T} \quad (8)$$

$$\boldsymbol{\mu}_{f_{t+1}} \triangleq m(\mathbf{x}_{t+1}) + K_{\mathbf{x}_{t+1}, \mathbf{x}_{1:t}} K_{\mathbf{x}_{1:t}, \mathbf{x}_{1:t}}^{-1} (f(\mathbf{x}_{1:t}) - \boldsymbol{m}_{\mathbf{x}_{1:t}}) \quad (9)$$

$$\boldsymbol{\Sigma}_{f_{t+1}} \triangleq K_{\mathbf{x}_{t+1}, \mathbf{x}_{t+1}} - K_{\mathbf{x}_{t+1}, \mathbf{x}_{1:t}} K_{\mathbf{x}_{1:t}, \mathbf{x}_{1:t}}^{-1} K_{\mathbf{x}_{1:t}, \mathbf{x}_{t+1}} \quad (10)$$

By multiplying the Gaussian conditionals, we can return to our GP prior as in equation 5. Although the function values are jointly Gaussian, the states X are not necessarily so, since the GP's mean and kernel functions specify non-linear relations between inputs $\mathbf{x}_t$ and outputs $f(\mathbf{x}_t)$.

3.2 Multivariate latent states

If our state has more than one dimension (i.e. $D > 1$), we model the system's transitions $f : \mathbb{R}^D \to \mathbb{R}^D$ by placing a GP prior on each of the $f_d : \mathbb{R}^D \to \mathbb{R}$ dimension-specific functions, and we write:

$$f(\mathbf{x}_t) \triangleq \{f_d(\mathbf{x}_t)\}_{d=1}^D, \quad p(f(X)) = \prod_{d=1}^{D} p(f_d(X)) \quad (11)$$

where each independent GP has its own mean and kernel functions: $m_d(\cdot), k_d(\cdot, \cdot)$. For ease of notation, we drop subscripts indexing state dimensions d. Since the GP's are independent from each other, each process only has to condition on its own function evaluations. Thus, the multivariate conditional $p(f(\mathbf{x}_{t+1}) \mid f(\mathbf{x}_{1:t}))$ is a diagonal Gaussian where the mean and variance of dimension d are as in 9 and 10, using as mean function $m_d(\cdot)$ and as kernel function $k_d(\cdot, \cdot)$. We use $\boldsymbol{\mu}_{f_{t+1}}, \boldsymbol{\Sigma}_{f_{t+1}}$ to refer to the whole mean vector and the diagonal covariance matrix respectively.

3.3 Control inputs

A control input $\mathbf{c}_t \in \mathbb{R}^{D_\mathbf{c}}$ influences the transition in a Markovian manner. We can therefore view it as being part of an "augmented" latent state. This makes our transition function $f : \mathbb{R}^{D_\mathbf{x} + D_\mathbf{c}} \to \mathbb{R}^{D_\mathbf{x}}$, and requires only a change to f's kernel and mean functions to be defined over $D_\mathbf{x} + D_\mathbf{c}$ input dimensions. Our transition probability is now also conditioned on $\mathbf{c}_t$: $p(\mathbf{x}_{t+1} \mid f, \mathbf{x}_t, \mathbf{c}_t)$, although we drop the

explicit conditioning on $\mathbf{c}_t$ from the notation, as it is not a random variable and we always condition on it, both in training and testing.

3.4 Computational costs of sampling

Despite the conceptual similarity between the non-parametric GPSSM, and parametric non-linear state space models, where the transition function is directly parameterised, it is considerably more expensive to sample from the GPSSM prior. The main reason for this, can be seen in eqs. (7) to (10), where to sample $\mathbf{x}_T$ we have to condition on the $T - 1$ previous points. This requires incrementally building up a $T \times T$ matrix inverse, which costs $\mathcal{O}(T^3)$ in time. Since our non-parametric transition function is implicitly defined by the function values $f(\mathbf{x}_{1:t})$ sampled so far, it is necessary to condition on all previous samples to ensure our overall function is self-consistent over the whole time-series.

This poses a problem for MCMC based inference methods, which rely on drawing samples from distributions closely related to the prior. Prediction from a GP posterior also has cubic cost. One way to avoid this cost is to sample from an adapted GP prior (e.g. FITC (Snelson & Ghahramani, 2005)), which was suggested by Frigola et al. (2013).

Modifying the model in this way can have unintended consequences (Bauer et al., 2016), so we focus on performing inference without assumptions on the structure of the GP or its kernel. Our inference scheme should approximate the correct model, while avoiding the $\mathcal{O}(T^3)$ cost at training time.

4 Variational inference

4.1 General variational bound

We are interested in approximating the posterior $p(X, f \mid Y)$[2] using variational inference. The general procedure is to construct a variational lower bound $\mathcal{L}$ to the log marginal likelihood, that has the KL divergence from an approximation to the true posterior as its slack: $\log p(Y) - \mathcal{L} = \mathrm{KL}[q(X, f) \| p(X, f \mid Y)]$. By maximising $\mathcal{L}$ w.r.t. q, we improve the quality of the approximation.

Practical stochastic variational inference places two requirements on approximating distributions. We firstly need to be able to generate samples from them, and secondly, we need to be able to evaluate their density. We start by deriving the lower bound for a general joint approximate posterior $q(X, f)$, to which we will later add constraints to produce different tractability-accuracy trade-offs. Following the usual derivation using Jensen's inequality (Blei et al.,

---

[2]We abuse notation by directly denoting a density over the infinite-dimensional f, even though such a density does not exist. We only ever concern ourselves with finite subsets of f.

2017), we obtain

$$\log p(Y) \geq \int \mathrm{d}f \mathrm{d}X q(f, X) \left[\log \prod_{t=1}^{T} p(\mathbf{y}_t | \mathbf{x}_t) + \right.$$

$$\left. \log \frac{p(\mathbf{x}_1) p(f) \prod_{t=1}^{T-1} p(\mathbf{x}_{t+1} | f, \mathbf{x}_t)}{q(f, X)} \right], \quad (12)$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t)}[\log p(\mathbf{y}_t | \mathbf{x}_t)] - \mathrm{KL}[q(f) \| p(f)] +$$

$$\mathbb{E}_{q(f, X)} \left[\log \frac{p(\mathbf{x}_1) \prod_{t=1}^{T-1} p(\mathbf{x}_{t+1} | f, \mathbf{x}_t)}{q(X | f)} \right]. \quad (13)$$

We constrain the approximate posteriors that we consider to use a sparse Gaussian process for f (Titsias, 2009; Hensman et al., 2013; Matthews, 2016). This constraint is ubiquitous in models where the inputs to a GP are random variables (Damianou et al., 2016), such as the GPLVM (Titsias & Lawrence, 2010), or deep GP (Damianou & Lawrence, 2013), as it provides analytical as well as computational tractability. A sparse GP approximation specifies a free Gaussian density on the function values $\mathbf{u}$ at M locations $q(\mathbf{u}) = \mathcal{N}(\mathbf{u}; \boldsymbol{\mu}_\mathbf{u}, \boldsymbol{\Sigma}_\mathbf{u})$, and uses the prior conditional for the rest of the Gaussian process, which we denote $f^{\neq \mathbf{u}}$. At this time, we let X depend on the entire process f for generality:

$$q(f^{\neq \mathbf{u}}, \mathbf{u}, X) = p(f^{\neq \mathbf{u}} | \mathbf{u}) q(\mathbf{u}) q(X | f^{\neq \mathbf{u}}, \mathbf{u}). \quad (14)$$

Choosing the posterior in this way results in a finite KL between the approximate posterior and the prior, which becomes $\mathrm{KL}[q(\mathbf{u}) \| p(\mathbf{u})]$ (Matthews et al., 2016).

4.2 Approximate posterior design choices

So far, the only simplifying assumption has been reducing $q(f^{\neq \mathbf{u}} | \mathbf{u})$ to the prior conditional $p(f^{\neq \mathbf{u}} | \mathbf{u})$. We now have to specify the form of $q(X | f^{\neq \mathbf{u}}, \mathbf{u})$. Most existing approximate inference schemes focused on factorised posteriors $q(X | f^{\neq \mathbf{u}}, \mathbf{u}) = q(X)$ because of efficient sampling or even a closed form for $\mathcal{L}$. In particular, Frigola et al. (2014) find the optimal variational $q(X)$ by calculus of variations and sample from it, whereas McHutchon (2014) and Ialongo et al. (2017) constrain it to be Markov-Gaussian. The factorisation and Gaussian likelihood imply a Gaussian $q(\mathbf{u})$ as well, which, together with certain kernels, leads to a closed-form bound. Eleftheriadis et al. (2017) reduced the $\mathcal{O}(TD^2)$ memory cost associated with storing the Markov-Gaussian $q(X)$ and improved optimisation behavour by employing a recurrant neural network as a recognition model. This also parallels the work of (Krishnan et al., 2017), where recurrent neural networks perform smoothing in order to learn a deterministic model of the dynamics.

In our experiments, we show the independence that is imposed in the previous approaches between X and f does in fact encourage over-confident beliefs about the dynamics,

which is a well-known phenomenon in variational inference (Turner & Sahani, 2011). To avoid this we focus on approximate posteriors which preserve correlations between X and f in various forms. We impose one more general constraint: Markovian structure in X. This structure is required to eliminate the impractical amount of correlations that can be expressed between the large number of random variables in X for computational reasons. Furthermore, the true posterior is also Markovian in X:

$$p(X | f, Y) = p(\mathbf{x}_1 | f, Y) \prod_{t=2}^{T} p(\mathbf{x}_t | f(\mathbf{x}_{t-1}), \mathbf{y}_{t:T}). \quad (15)$$

So in comparison, we are not imposing an additional restriction, giving the general form for our approximate posterior

$$q(X | f) = q(\mathbf{x}_1) \prod_{t=1}^{T-1} q(\mathbf{x}_{t+1} | f, \mathbf{x}_t), \quad (16)$$

where we are free to specify $q(\mathbf{x}_1)$ and $q(\mathbf{x}_{t+1} | f, \mathbf{x}_t)$. Given these additional assumptions, we can simplify the ELBO from eq. (13) to

$$\mathcal{L} = \sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{x}_t)}[\log p(\mathbf{y}_t | \mathbf{x}_t)] - \mathrm{KL}[q(\mathbf{u}) \| p(\mathbf{u})] +$$

$$-\mathrm{KL}[q(\mathbf{x}_1) \| p(\mathbf{x}_1)]$$

$$- \sum_{t=1}^{T-1} \mathbb{E}_{q(f, \mathbf{x}_t)}[\mathrm{KL}[q(\mathbf{x}_{t+1} | f, \mathbf{x}_t) \| p(\mathbf{x}_{t+1} | f, \mathbf{x}_t)]]. \quad (17)$$

We now add two restrictions for all approximate posteriors: 1) we use a Gaussian $q(\mathbf{x}_{t+1} | f, \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; g_t(f, \mathbf{x}_t), h_t(f, \mathbf{x}_t))$ to obtain a tractable KL, and 2) we assume g_t and h_t are linear, and depend only on a finite number of points on $f(\cdot)$, $q(\mathbf{x}_{t+1} | f, \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t+1}; \mathbf{A}_t \mathbf{f}_t + \mathbf{b}_t, \mathbf{S}_t^*)$, to make the expectation over $q(f)$ tractable. We choose $\mathbf{A}_t$ and $\mathbf{b}_t$ to be free variational parameters in all cases. The dependence between X and f is controlled through $\mathbf{f}_t$ and $\mathbf{S}_t^*$. The parameters $\mathbf{A}_t$ and $\mathbf{b}_t$ are crucial to allow information from the observations to make its way into the filtering distribution. Down-weighting of information coming from the previous state is done by $\mathbf{A}_t$, while $\mathbf{b}_t$ summarises the influence from the observations.

We now discuss various choices for $\mathbf{f}_t$ and $\mathbf{S}_t^*$, and the influence they have on the flexibility of the approximate posterior, which we summarise in table 1. The densities of all these variational distributions can still be evaluated, and exact samples can be drawn for unbiased evaluation of the bound.

4.3 A natural choice

Another way to justify our choice of posterior is that, for Gaussian emissions, the exact posterior "filtering" factors: $p(\mathbf{x}_{t+1} | f, \mathbf{x}_t, \mathbf{y}_{1:t+1})$ are Gaussians of precisely the form we propose. In fact, we can compute the corresponding $\mathbf{A}_t$,

	$q(X \mid f)$	$\mathbf{f}_t$	$\mathbf{S}_t^*$	sampling
1) Factorised - linear	$q(X)$	$\mathbf{x}_t$	$\mathbf{S}_t$	$\mathcal{O}(T)$
2) Factorised - non-linear	$q(X)$	$K_{\mathbf{x}_t,Z}K_{Z,Z}^{-1}\boldsymbol{\mu}_\mathbf{u}$	$\mathbf{S}_t + \mathbf{A}_t\mathbf{C}_f(\mathbf{x}_t)\mathbf{A}_t^\top$	$\mathcal{O}(T)$
3) Non-Factorised - non-linear	$q(X \mid f)$	$f(\mathbf{x}_t)$	$\mathbf{S}_t$	$\mathcal{O}(T^3)$
4) U-Factorised - non-linear	$q(X \mid \mathbf{u})$	$K_{\mathbf{x}_t,Z}K_{Z,Z}^{-1}\mathbf{u}$	$\mathbf{S}_t + \mathbf{A}_t\mathbf{C}_{f\mid\mathbf{u}}(\mathbf{x}_t)\mathbf{A}_t^\top$	$\mathcal{O}(T)$

Table 1: Variations of approximate posteriors. $\mathbf{A}_t, \mathbf{b}_t, \mathbf{S}_t$ are free parameters in all cases. $\mathbf{C}_f(\mathbf{x}_t)$ is the sparse GP's marginal posterior variance $\mathbf{C}_f(\mathbf{x}_t) \triangleq K_{\mathbf{x}_t,\mathbf{x}_t} + K_{\mathbf{x}_t,Z}K_{Z,Z}^{-1}(\boldsymbol{\Sigma}_\mathbf{u} - K_{Z,Z})K_{Z,Z}^{-1}K_{Z,\mathbf{x}_t}$ whereas $\mathbf{C}_{f\mid\mathbf{u}}(\mathbf{x}_t)$ is the conditional variance of $f(\mathbf{x}_t) \mid \mathbf{u}$: $\mathbf{C}_{f\mid\mathbf{u}}(\mathbf{x}_t) \triangleq K_{\mathbf{x}_t,\mathbf{x}_t} - K_{\mathbf{x}_t,Z}K_{Z,Z}^{-1}K_{Z,\mathbf{x}_t}$.

$\mathbf{b}_t$, and $\mathbf{S}_t^*$ in closed form:

$$\mathbf{S}_t^* = (\mathbf{Q}^{-1} + \mathbf{C}^\mathsf{T}\mathbf{R}^{-1}\mathbf{C})^{-1}, \qquad (18)$$

$$\mathbf{A}_t = \mathbf{S}_t^*\mathbf{Q}^{-1}, \qquad (19)$$

$$\mathbf{b}_t = \mathbf{S}_t^*\mathbf{C}^\mathsf{T}\mathbf{R}^{-1}(\mathbf{y}_{t+1} - \mathbf{d}). \qquad (20)$$

Notice that here $\mathbf{A}_t$ and $\mathbf{S}_t^*$ are constant through time, reducing the time and memory footprint of the algorithm. While the exact posterior "smoothing" factors $p(\mathbf{x}_{t+1}|f,\mathbf{x}_t,\mathbf{y}_{1:T})$ are not generally Gaussian or closed-form (hence leaving $\mathbf{A}_t, \mathbf{b}_t, \mathbf{S}_t$ free), in settings where the observation noise is small, filtering and smoothing will give very similar results, making the filtering factors very close to optimal. Likewise, if future observations hold little information about the present due to process noise corruption, the filtering factors will provide good approximations to the conditional state posterior.

4.4 Factorised approximations
We can recover the existing Gaussian factorised approximation of McHutchon (2014) by taking $\mathbf{f}_t = \mathbf{x}_t$ (see 1 in table 1). The linear relationship between consecutive states implies a linearisation of the transition function. We know that the true $p(\mathbf{x}_{t+1}|f(\mathbf{x}_t),\mathbf{y}_{t+1:T})$ will contain influence from $\mathbf{x}_t$ only through the *non-linear* transition function.

We can break the linearisation constraint by correlating $\mathbf{x}_{t+1}$ with $\mathbf{x}_t$ pushed through the approximate GP mean (2 in table 1). This results in a non-Gaussian $q(X)$ which is still independent from $q(f)$. Similarly to how $\mathbf{A}_t$ weights influence pfrom the previous state and observations, we also use $\mathbf{A}_t$ to weigh the predictive covariance from f with $\mathbf{S}_t$, which is learned directly through the influence the observed data has on the bound.

The main reason we introduce this modified factorised method, is so we can directly isolate the effect of dependence between X and f, compared to simply non-Gaussianity in X.

Both approximate posteriors can be sampled in $\mathcal{O}(T)$ due to their Markov structure. The factorised - linear posterior is particularly convenient as the marginal variance of any $\mathbf{x}_t$ is directly parameterised. This means that sequences can be subsampled to allow for minibatching within sequences as well as between sequences. This property is lost in the factorised - nonlinear approximation, as samples from $\mathbf{x}_t$

can only be generated by sequential sampling through the non-linear relationships between neighbouring time points.

4.5 Dependence on f
The factorised, non-linear approximation from the previous section is attempting to directly incorporate the learned transition function into the approximation over X. Instead of summarising the information using the mean of f, we want to break the factorisation between X and f by using the actual f in the approximation. This is summarised in table 1, line 2. The *actual* function value $f(\mathbf{x}_t)$ of a function in the posterior is weighed by $\mathbf{A}_t$ against $\mathbf{b}_t$.

If there is no process noise, choosing $\mathbf{A}_t = \mathbf{I}$, $\mathbf{b}_t = \mathbf{0}$, and $\mathbf{S}_t = \mathbf{0}$ would be the exact form for $q(\mathbf{x}_{t+1}|f,\mathbf{x}_t)$, since knowledge of the previous state and the transition function gives perfect knowedge about the next state. In this situation, the only uncertainty can come from sampling different transition functions from $q(f)$.

Sampling from this approximate posterior poses a challenge. From eq. (16), we note that every time a $f(\mathbf{x}_t)$ is sampled, it has to be consistent with a single function throughout the entire time series. This means that $f(\mathbf{x}_t)$ has to be sampled conditioned on the function values sampled for earlier t, resulting in an $\mathcal{O}(T^3)$ cost. This high cost prohibits learning from long time series. This can be alleviated somewhat by breaking correlations within a time series in the approximate posterior. We do this by cutting the approximate posterior into subsequences with lengths $\tau_1, \ldots, \tau_n$. The state immediately after a subsequence, e.g. $\mathbf{x}_{\tau_1+1}$ is given a non-conditional posterior:

$$q(X \mid f) = \left[q(\mathbf{x}_1) \prod_{t=1}^{\tau_1-1} q(\mathbf{x}_{t+1} \mid f, \mathbf{x}_t) \right] \times$$
$$\left[q(\mathbf{x}_{\tau_1+1}) \prod_{t=\tau_1+1}^{\tau_1+\tau_2-1} q(\mathbf{x}_{t+1} \mid f, \mathbf{x}_t) \right] \ldots, \quad (21)$$

This reduces the cost of sampling to $\mathcal{O}\left(\frac{T}{\tau}\tau^3\right)$, and allows for minibatching over subsequences. However, adding these factorisations reintroduces the properties of the orginal factorising bounds. In the extreme case of $\tau = 1$, we actually obtain an approximate posterior which does not only factorise between X and f, but also between all $\mathbf{x}_t$ as in Mattos et al. (2016).

4.6 U-Factorisation

The previous sections introduce an approximate posterior exhibiting dependence between X and f. This will generally come at a cubic cost. The reason for this is the same as that for the cost of sampling from the prior (section 3.4): our transition function has as many degrees of freedom as time-steps. Parametric models avoid this cubic cost since they can sample a transition function with fixed degrees of freedom before sampling the entire time series. Luckily, our sparse variational GP posterior separates the finite degrees of freedom that we can learn about, from the rest of the potentially infinite degrees of freedom of our prior.

In order to gain an approximate posterior with linear time sampling scheme, we choose $q(X|f) = q(X|\mathbf{u})$ ((4) in table 1). Crucially, we can sample from this posterior in linear time, as only a single $\mathbf{u}$ needs to be sampled from $q(\mathbf{u})$ for an entire sample trajectory. We are effectively being parametric about our sample trajectories while minimising a KL distance from the full non-parametric model. By the properties of sparse variational GPs (Matthews et al., 2016), by adding more inducing points we can reduce this distance.

Intuitively, by finely tiling with inducing points the parts of the state-space we are likely to traverse with our sample paths we can fully specify the behaviour of our transition function in those regions. In this case, conditioning on $\mathbf{u}$ is sufficient to "tie" down the transitions arbitrarily tightly leaving very little to be gained by additionally conditioning on sampled f values. The elegance of this approach lies in the fact that we can assess how many inducing points are required by finding the saturation point where the converged ELBO values stop improving as more points are added.

4.7 Comparison to PR-SSM

Doerr et al. (2018) were, to the best of our knowledge, the first to consider a non-factorised variational posterior for the GPSSM. Their work, however, has two significant shortcomings. Firstly, $q(\mathbf{x}_{t+1}|f, \mathbf{x}_t)$ is taken to be the same as the prior transition. This leaves no free parameters, except for those in $q(\mathbf{x}_1)$ and $q(\mathbf{u})$, to incorporate information from the observations into our posterior: no filtering or smoothing is possible. This gives a good approximation only when the process noise is low and/or the observed sequence is short, as low noise levels can compound in a long sequence. Even in the absence of process noise, using the observations to update our beliefs about the states can help optimisation greatly by effectively setting up easier "targets" for the GP to go through.

Secondly, Doerr et al. (2018) employ a sampling scheme which gives incorrect marginal samples of $q(\mathbf{x}_t)$, even assuming, as they did, that:

$$p(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_{T-1})|\mathbf{u}) = \prod_{t=1}^{T-1} p(f(\mathbf{x}_t)|\mathbf{u}) \qquad (22)$$

Despite this conditional factorisation assumption, once $\mathbf{u}$ is integrated out, as in PR-SSM, the $f(\cdot)$'s become correlated. Thus, in order to correctly sample $f(\mathbf{x}_t)$, we need to condition on all previous samples $f(\mathbf{x}_{1:t-1})$, introducing a cubic time cost. This is ignored in PR-SSM and samples from the posterior over f are effectively drawn *independently* for every time-step t. The resulting samples from $q(\mathbf{x}_t)$ are biased. A mismatch is thus introduced between the functional form of the approximate posterior (which has correlations between function values), and the samples we use to compute its expectations. Hence, PR-SSM's objective does not correspond to a valid variational lower bound.

We solve this by explicitly sampling $\mathbf{u}$ and conditioning on it. Conditioning on the same $\mathbf{u}$ enforces consistency of the dynamics along a sample trajectory. However, we also wish to avoid the factorisation assumption of eq. (22) as that does not generally correspond to a valid GP prior. Taking $q(X \,|\, f) = q(X \,|\, \mathbf{u})$ we obtain a valid variational bound where we only need to condition on $\mathbf{u}$ to sample X, giving us an $\mathcal{O}(T)$ cost while maintaining a valid GP prior.

5 Experiments

In the experiments we set out to test how the proposed posterior (U-Factorised - non-linear) compares with the Factorised - linear, Factorised - non-linear, and PR-SSM (Doerr et al., 2018) approximations. Each model serves as a "control" for a specific experimental question:

1. Factorised - linear: what is gained by making our approximation non-linear? I.e. if our posterior $q(X)$ were not jointly Gaussian, but only its Markovian conditional factors.

2. Factorised - non-linear: what is gained by introducing dependence between our states and our transition function? I.e. if we do not marginalise both $f(\mathbf{x}_t)$ and $\mathbf{u}$ from our posterior over X.

3. PR-SSM: what is gained, or lost, by forcing our model to find transition functions that explain the data well, without accounting for process noise?

To answer these questions, we test model calibration and predictive performance in a number of environments: the "kink" function, a set of five system identification benchmark datasets (also used in (Doerr et al., 2018)), and a cart and pole system.

5.1 "Kink" Function

We generate data according to the transition function displayed in red in figure 1:

$$f(x) = 0.8 + (x + 0.2)(1 - \frac{5}{1 + \exp{(-2x)}})$$

This function generates cyclic trajectories through the state-space, as it combines two linear segments with slope greater than 1 (left) and less than -1 (right). It also contains a non-linearity at the joining of these segments, giving it a variety of dynamics and serving as an interesting testing environment. We train our models on the data described in figure 1, fixing their emissions $(\mathbf{C}, \mathbf{d}, \mathbf{R})$ to the true, generative ones so as to enable direct latent space comparison[3].
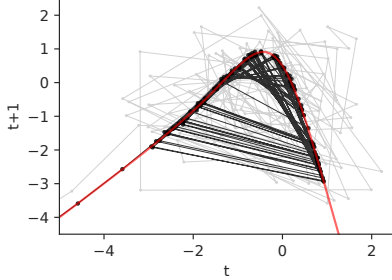
Figure 1: "Kink" data. In red is the true system transition function, in black are the latent states and in grey the observations. Each point's (x, y) coordinates are a pair $(\mathbf{x}_t, \mathbf{x}_{t+1})$ or $(\mathbf{y}_t, \mathbf{y}_{t+1})$ for some index t. Such pairs are connected through time, highlighting trajectories through the state-space. A sequence of 120 latent states were sampled beginning from a standard normal $\mathbf{x}_1$. Their trajectory was perturbed by Gaussian noise with s.d. 0.05. Observations were obtained by adding Gaussian noise with s.d. $\sqrt{0.8}$ to the latents.

In order to make inference challenging, we add a small amount of process noise (s.d. 0.05) and a significant amount of observation noise (s.d. $\sqrt{0.8}$). As we can see from figure 2, in this regime PR-SSM is unable to learn, given that its posterior encourages it to find trajectories with no process noise. Indeed, regardless of which data we are training on, during optimisation, PR-SSM always drives the process noise to zero. This leads it to converge to poor solutions when process noise exists and/or to drive up the observation noise (across all experiments, PR-SSM learned the highest values for the observation noise). On the other hand, the GPSSM models can handle process noise and recover inflated estimates of it in order to ascribe some transitions to noise and justify smoother dynamics. The Factorised - linear model's estimates of the process noise are consistently the highest (across many settings of noise parameters and sequence lengths), followed by the Factorised - non-linear model. As can be seen in figure 3, the U-Factorised model finds the most well-calibrated posterior. The factorised approaches (the linear one is shown, though they gave a very similar fit) result in a posterior that is not only less accurate, but also more confidently wrong. In general, the posteriors found by the factorised approaches favour higher process

[3]This is to side-step any scale invariances built into our model.

noise to achieve smoother, over-confident dynamics. This is consistent with the behaviour discussed in (Turner & Sahani, 2011): mean-field variational inference favours more concentrated posteriors. By matching our true posterior's structure more closely and modelling the dependence between X and f, it seems we can overcome this issue.
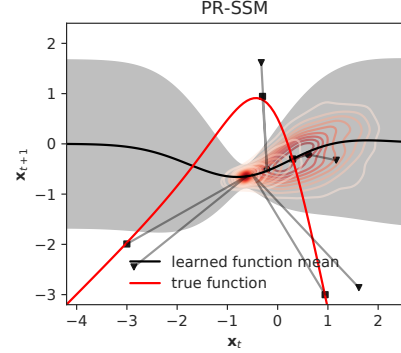

Figure 2: PR-SSM fails to learn in the presence of process noise, for moderately long time-series ($T = 120$). Plot elements as in figure 3.

5.2 System identification benchmarks

We also train our models on five benchmark datasets taken from (De Moor et al., 1997). Test set results are reported in table 2. As in (Doerr et al., 2018), we train on the first half of the sequence, we normalise the data, and we use a 4-dimensional latent state (plus 1 non-stochastic dimension for the control inputs). No mini-batching was used, the bound is evaluated using 100 samples from the posterior, and we use 100 variational inducing points. The test results are for predictions 30 steps into the future from the end of the training sequence.

We initialise all models at the solution found by the Factorised - non-linear method to assess differences in optima of the variational objectives. This turned out to be necessary for PR-SSM as learning on such long sequences, without mini-batching, proved very challenging without any filtering from the observations. As can be seen from table 2 PR-SSM can often find good solutions despite its constraints. This is particularly true for low noise regimes, as in the "Drive" dataset. The GPSSM models are more robust in general though, with the U-Factorised model in the lead among GPSSMs for NLPP in all but the last dataset.

5.3 Cart and pole

Finally, we consider a cart and pole system (or inverted pendulum). We add Gaussian noise to the states in order to obtain our observations. Standard deviations are: 0.03; 0.1047; 0.4; 1.3963; for the respective dimensions of our system: cart position, pendulum angle, cart velocity and angular velocity of the pendulum. The noise s.d. levels correspond to 3cm; 6 degrees; 3cm/0.075sec; 6 degrees/0.075sec. Results
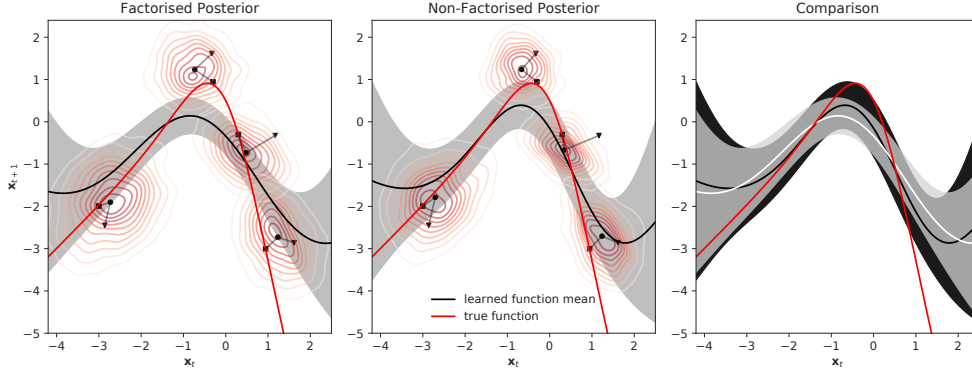
Figure 3: Posterior miscalibration due to factorisation. **Left**: fit for the factorised $q(X)q(f)$; **Middle**: fit for the U-Factorised $q(X \mid \mathbf{u})q(f)$; **Right**: superimposed posteriors (grey is factorised, black is U-Factorised). **All subplots**: the shaded regions indicate a 3σ confidence interval and the contour plots (left and middle) show the pairwise posteriors $q(\mathbf{x}_t, \mathbf{x}_{t+1})$ over some test latent states. The (x, y) coordinates of the squares, triangles and circles correspond to, respectively: the true latent states $(\mathbf{x}_t, \mathbf{x}_{t+1})$, the observed states $(\mathbf{y}_t, \mathbf{y}_{t+1})$ and the means of the marginal posteriors $(\mathbb{E}_{q(\mathbf{x}_t)}[\mathbf{x}_t], \mathbb{E}_{q(\mathbf{x}_{t+1})}[\mathbf{x}_{t+1}])$.

MODEL	ACTUATOR	BALLBEAM	DRIVE	DRYER	GAS FURNACE
FACTORISED - LINEAR	0.156; -0.385	0.082; -0.508	0.445; 0.755	0.098; -0.713	0.170; -0.272
FACTORISED - NON-LINEAR	0.149; -0.568	0.076; -1.340	0.245; 0.360	0.048; -1.292	0.169; 3.424
U-FACTORISED - NON-LINEAR	0.141; -0.656	0.072; -1.402	0.280; 0.353	0.052; -1.312	0.169; 2.489
PR-SSM (INITALISED)	0.333; 3.362	0.993; 19.615	0.142; -0.649	0.051; -1.262	0.161; 0.274

Table 2: Test set performance (RMSE; NLPP) for the system identification datasets. Lower is better.

are shown in table 3. Because there is no process noise and the training sequences are short (20 time-steps), we see that PR-SSM performs the best in terms of predictive performance. Of course PR-SSM can be viewed as a special case of our approximation, albeit with a mismatched sampling scheme. Running the experiment with our approximation but fixing the variational parameters to: $\mathbf{A}_t = \mathbf{I}$, $\mathbf{b}_t = 0$, and $\mathbf{S}_t = \mathbf{Q}$, we learn the same solution as PR-SSM. In the GPSSM models, the U-Factorised model performs the best, and, as can be seen in figure 4, it also recovers the lowest noise levels, using the dynamics to explain more of the structure in the data.

Figure 4: Lower values of the learned process noise standard deviation.

MODEL	RMSE	NLPP
FACTORISED - LINEAR	3.36	2.92
FACTORISED - NON-LINEAR	3.06	2.48
U-FACTORISED - NON-LINEAR	2.80	2.27
PR-SSM	2.12	0.93

Table 3: Test set performance for the Cart and Pole dataset. Results averaged over 30 different test sequences where we filter on 5 test states and predict the next 15 steps ahead. Lower is better.

6 Conclusion

The GPSSM holds much promise for system identification applications, its main challenge being inference. Varia-
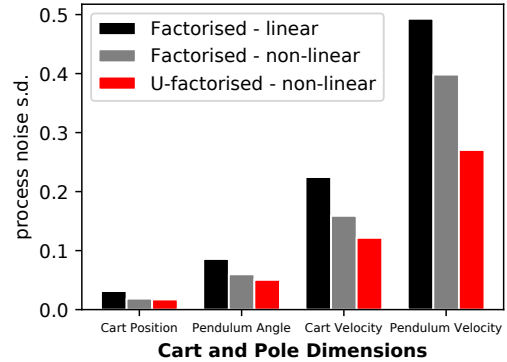
tional inference maintains many of the benefits of using non-parametric models, but presents particular challenges for time-series models due to their sensitivity to factorising assumptions. The most straightforward non-factorising approximations regain the poor computational scaling that variational methods were introduced to avoid. By exploiting structure of the approximate GP posterior, we were able to construct a non-factorised posterior with the desired computational scaling. This often leads to better predictive performance, calibration, and an improved estimation of model parameters.

References

Bauer, M., van der Wilk, M., and Rasmussen, C. E. Understanding probabilistic sparse Gaussian process approximations. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, 2016.

Billings, S. *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Wiley, 2013. ISBN 9781119943594.

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

Bui, T., Turner, R. E., and Rasmussen, C. E. Bayesian Gaussian process state-space models via Power-EP. In *ICML 2016 Workshop on Data efficient Machine Learning*, 2016.

Damianou, A. and Lawrence, N. Deep Gaussian processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 207–215, 2013.

Damianou, A. C., Titsias, M. K., and Lawrence, N. D. Variational inference for latent variables and uncertain inputs in Gaussian processes. *The Journal of Machine Learning Research*, 17(1):1425–1486, 2016.

De Moor, B., De Gersem, P., De Schutter, B., and Favoreel, W. DAISY: A database for identification of systems. *JOURNAL A*, 38:4–5, 1997. URL http://homes.esat.kuleuven.be/~smc/daisy/.

Deisenroth, M. and Rasmussen, C. E. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Doerr, A., Daniel, C., Schiegg, M., Nguyen-Tuong, D., Schaal, S., Toussaint, M., and Trimpe, S. Probabilistic recurrent state-space models. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Eleftheriadis, S., Nicholson, T., Deisenroth, M., and Hensman, J. Identification of Gaussian process state-space models. In *Advances in Neural Information Processing Systems 30*, 2017.

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

Frigola, R., Lindsten, F., Schön, T. B., and Rasmussen, C. E. Bayesian inference and learning in Gaussian process state-space models with particle MCMC. In *Advances in Neural Information Processing Systems 26*, 2013.

Frigola, R., Chen, Y., and Rasmussen, C. E. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27*, 2014.

Ghahramani, Z. and Roweis, S. T. Learning nonlinear dynamical systems using an EM algorithm. In *Advances in Neural Information Processing Systems 12*, 1999.

Hensman, J., Fusi, N., and Lawrence, N. D. Gaussian processes for Big Data. *Uncertainty in Artificial Intelligence*, 2013.

Hochreiter, S. and Schmidhuber, J. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems 9*, 1996.

Ialongo, A. D., van der Wilk, M., and Rasmussen, C. E. Closed-form inference and prediction in Gaussian process state-space models. *NIPS 2017 Time-Series Workshop*, 2017.

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

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state-space models. *AAAI*, 2017.

Matthews, A., Hensman, J., Turner, R., and Ghahramani, Z. On sparse variational methods and the Kullback-Leibler divergence between stochastic processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Matthews, A. G. d. G. *Scalable Gaussian process inference using variational methods*. PhD thesis, University of Cambridge, 2016.

Mattos, C. L. C., Dai, Z., Damianou, A., Forth, J., Barreto, G. A., and Lawrence, N. D. Recurrent Gaussian processes. In *International Conference on Learning Representations (ICLR)*, 2016.

McHutchon, A. *Nonlinear modelling and control using Gaussian processes*. PhD thesis, University of Cambridge, 2014.

Rasmussen, C. E. and Ghahramani, Z. Occam's razor. In *Advances in Neural Information Processing Systems 13*, 2000.

Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

Roweis, S. and Ghahramani, Z. A unifying review of linear Gaussian models. *Neural computation*, 11(2):305–345, 1999.

Salimbeni, H. and Deisenroth, M. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems 30*, pp. 4588–4599, 2017.

Snelson, E. and Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems 18*, 2005.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, 2014.

Titsias, M. Variational learning of inducing variables in sparse Gaussian processes. *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

Titsias, M. and Lawrence, N. D. Bayesian Gaussian process latent variable model. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 844–851, 2010.

Turner, R., Deisenroth, M., and Rasmussen, C. State-space inference and learning with Gaussian processes. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.

Turner, R. E. and Sahani, M. Two problems with variational expectation maximisation for time-series models. *Bayesian Time series models*, 1(3.1):3–1, 2011.

von Neumann, J., Morgenstern, O., Kuhn, H. W., and Rubinstein, A. *Theory of Games and Economic Behavior*. Princeton University Press, 1944. ISBN 9780691130613.

Wang, J., Hertzmann, A., and Fleet, D. J. Gaussian process dynamical models. In *Advances in Neural Information Processing Systems 18*, 2005.

Wang, J. M., Fleet, D. J., and Hertzmann, A. Gaussian process dynamical models for human motion. *IEEE transactions on pattern analysis and machine intelligence*, 30(2): 283–298, 2008.