RECURRENT GAUSSIAN PROCESSES

César Lincoln C. Mattos[1], **Zhenwen Dai**[2], **Andreas Damianou**[3], **Jeremy Forth**[4], **Guilherme A. Barreto**[5] **& Neil D. Lawrence**[6]
[1,5]Federal University of Ceará, Fortaleza, Ceará, Brazil
[2,3,6]University of Sheffield, Sheffield, UK
[1]cesarlincoln@terra.com.br
[2,3]{z.dai,andreas.damianou}@sheffield.ac.uk
[4]jforth@iweng.org
[5]gbarreto@ufc.br
[6]N.Lawrence@dcs.sheffield.ac.uk

ABSTRACT

We define Recurrent Gaussian Processes (RGP) models, a general family of Bayesian nonparametric models with recurrent GP priors which are able to learn dynamical patterns from sequential data. Similar to Recurrent Neural Networks (RNNs), RGPs can have different formulations for their internal states, distinct inference methods and be extended with deep structures. In such context, we propose a novel deep RGP model whose autoregressive states are latent, thereby performing representation and dynamical learning simultaneously. To fully exploit the Bayesian nature of the RGP model we develop the Recurrent Variational Bayes (REVARB) framework, which enables efficient inference and strong regularization through coherent propagation of uncertainty across the RGP layers and states. We also introduce a RGP extension where variational parameters are greatly reduced by being reparametrized through RNN-based sequential recognition models. We apply our model to the tasks of nonlinear system identification and human motion modeling. The promising obtained results indicate that our RGP model maintains its highly flexibility while being able to avoid overfitting and being applicable even when larger datasets are not available.

1 INTRODUCTION

The task of learning patterns from sequences is an ongoing challenge for the machine learning community. Recurrent models are able to learn temporal patterns by creating internal memory representations of the data dynamics. A general recurrent model, comprised of external inputs $\boldsymbol{u}_i$, observed outputs $\boldsymbol{y}_i$ and hidden states $\boldsymbol{x}_i$, is given by

$$\boldsymbol{x}_i = f(\boldsymbol{x}_{i-1}, \boldsymbol{u}_{i-1}) + \boldsymbol{\epsilon}_i^x, \tag{1}$$

$$\boldsymbol{y}_i = g(\boldsymbol{x}_i) + \boldsymbol{\epsilon}_i^y, \tag{2}$$

where i is the instant of observation, $f(\cdot)$ and $g(\cdot)$ are unknown nonlinear functions respectively called *transition* and *observation* functions, $\boldsymbol{\epsilon}_i^x \sim \mathcal{N}(\boldsymbol{\epsilon}_i^x|\boldsymbol{0}, \sigma_x^2\boldsymbol{I})$ and $\boldsymbol{\epsilon}_i^y \sim \mathcal{N}(\boldsymbol{\epsilon}_i^y|\boldsymbol{0}, \sigma_y^2\boldsymbol{I})$ are respectively Gaussian transition and observation noises, and $\boldsymbol{I}$ is the identity matrix. The recurrent nature of the model is expressed by the state variables $\boldsymbol{x}_i$, which are dependent on their past values, allowing past patterns to have influence in future outputs.

In recurrent parametric models, such as Recurrent Neural Networks (RNN), both the transition and observation functions are modeled with weight matrices $\boldsymbol{W}$, $\boldsymbol{U}$, $\boldsymbol{V}$ and nonlinear element-wise activation functions $\phi_f(\cdot)$ and $\phi_g(\cdot)$:

$$\boldsymbol{x}_i = \phi_f(\boldsymbol{W}^\top \boldsymbol{x}_{i-1}, \boldsymbol{U}^\top \boldsymbol{u}_{i-1}), \tag{3}$$

$$\boldsymbol{y}_i = \phi_g(\boldsymbol{V}^\top \boldsymbol{x}_i). \tag{4}$$

As argued by Pascanu et al. (2013), the basic recurrent structure in Eq. 3 can be made *deep*, for example by adding multiple hidden layers comprised of multiple transition functions, where the output of each layer is used as the input of the next one.

The difficulties related to learning dynamical structures from data (Bengio et al., 1994) have motivated the proposal of several RNN architectures in the literature, such as time-delay neural networks (Lang et al., 1990), hierarchical RNNs (El Hihi & Bengio, 1996), nonlinear autoregressive with exogenous inputs (NARX) neural networks (Lin et al., 1996), long short-term memory networks (Hochreiter & Schmidhuber, 1997), deep RNNs (Pascanu et al., 2013) and the RNN encoder-decoder (Cho et al., 2014). The usefulness of RNNs has been demonstrated in interesting applications, such as music generation (Boulanger-lewandowski et al., 2012), handwriting synthesis, (Graves, 2013) speech recognition (Graves et al., 2013), and machine translation (Cho et al., 2014).

However, one well known limitation of parametric models, such as RNNs, is that they usually require large training datasets to avoid overfitting and generalization degradation. In contrast Bayesian nonparametric methods, such as Gaussian Processes (GP) models, often perform well with smaller datasets. In particular GP-based models are able to propagate uncertainty through their different structural components, something which ensures that when data is not present in a particular region of input space the predictions do not become over confident.

The general recurrent Eqs. 1 and 2 have been widely studied in the control and dynamical system identification community as either non-linear auto-regressive models with exogenous inputs (NARX) models or state-space models (SSM). Here we are particularly interested in the Bayesian approach to those models (Peterka, 1981). In this context, several GP-NARX models have been proposed in the literature (Murray-Smith et al., 1999; Solak et al., 2003; Kocijan et al., 2005). However, these models do not propagate the past states' uncertainty through the transition function during the training or prediction phase. Girard et al. (2003); Damianou & Lawrence (2015) rectify this problem. Nevertheless, in all of the above standard NARX approaches the autoregressive structure is performed directly with the observed outputs, which are noisy.

A more general alternative to standard NARX models is the use of SSMs. Such structures have been explored recently by the GP community. Frigola et al. (2014) proposed a variational GP-SSM where both the transition and observation functions can have GP priors. Although they present results exclusively for the case where only the transition is modeled by a GP, while the observation has a parametric form. Conversely, Moreover, the inference required an additional smoothing step with, for example, a sequential Monte Carlo algorithm. Svensson et al. (2015) also consider a GP-SSM, but with a reduced-rank structure, and perform inference following a fully Bayesian approach, using a particle MCMC algorithm.

All the aforementioned dynamic GP models contain recurrent structures. Each model makes a particular choice for the definition of the states $\boldsymbol{x}_i$ and the algorithm used to perform inference. Because all these GP models incorporate recurrent structures we refer to this general class of models as the *Recurrent GP (RGP)* family of methods. These are models such as in Eqs (1) and (2) which incorporate GP priors for the transition and/or observation functions. Inspired by developments in the RNN community we propose a novel RGP model which introduces *latent autoregression* and is embedded in a new variational inference procedure named *Recurrent Variational Bayes (REVARB)*. Our formulation aims to tackle some issues of past RGP structures. Our algorithm allows the RGP class of models to easily be extended to have deep structures, similar to deep RNNs. Furthermore, we develop an extension which combines the RGP and RNN technologies by reparameterizing the means of REVARB's variational distributions through a new RNN-based recognition model. This idea results in simpler optimization and faster inference in larger datasets.

Recently, Sohl-Dickstein & Kingma (2015) have detailed interesting similarities between the log-likelihood training of RNNs and the variational Bayes training objective in the context of generative models. In the present work we also follow a variational approach with the proposed REVARB framework, but with respect to RGP models.

The rest of the paper is structured as follows. In Section 2 we briefly summarize the standard GP for regression. In Section 3 we define the structure of our proposed RGP model. In Section 4 we describe the REVARB inference method. In Section 5 we present some experiments with REVARB in some challenging applications. We conclude the paper with hints to further work in Section 6.

2 STANDARD GP MODEL FOR REGRESSION

In the GP framework, a multiple input single output nonlinear function $f(\cdot)$ applied to a collection of N examples of D-dimensional inputs $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ is given a multivariate Gaussian prior:

$$\boldsymbol{f} = f(\boldsymbol{X}) \sim \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, \boldsymbol{K}), \tag{5}$$

where a zero mean vector was considered, $\boldsymbol{f} \in \mathbb{R}^N$ and $\boldsymbol{K} \in \mathbb{R}^{N \times N}, K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, is the covariance matrix, obtained with a covariance (or *kernel*) function $k(\cdot, \cdot)$, which must generate a semidefinite positive matrix $\boldsymbol{K}$, for example the exponentiated quadratic kernel:

$$k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\left[-\frac{1}{2}\sum_{d=1}^{D} w_d^2 (x_{id} - x_{jd})^2\right], \tag{6}$$

where the vector $\boldsymbol{\theta} = [\sigma_f^2, w_1^2, \ldots, w_D^2]^\top$ is comprised of the hyperparameters which characterize the covariance of the model.

If we consider a Gaussian likelihood $p(\boldsymbol{y}|\boldsymbol{f}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \sigma_y^2 \boldsymbol{I})$ relating the observations $\boldsymbol{y}$ and the unknown values $\boldsymbol{f}$, inference for a new output f_*, given a new input $\boldsymbol{x}_*$, is calculated by:

$$p(f_*|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{x}_*) = \mathcal{N}\left(f_*|\boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_y^2 \boldsymbol{I})^{-1}\boldsymbol{y}, k_{**} - \boldsymbol{k}_{*N}(\boldsymbol{K} + \sigma_y^2 \boldsymbol{I})^{-1}\boldsymbol{k}_{N*}\right), \tag{7}$$

where $\boldsymbol{k}_{*N} = [k(\boldsymbol{x}_*, \boldsymbol{x}_1), \cdots, k(\boldsymbol{x}_*, \boldsymbol{x}_N)]$, $\boldsymbol{k}_{N*} = \boldsymbol{k}_{*N}^\top$ and $k_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$. The predictive distribution of y_* is similar to the one in Eq. (7), but the variance is added by σ_y^2.

The vector of hyperparameters $\boldsymbol{\theta}$ can be extended to include the noise variance σ_y^2 and be determined with the maximization of the marginal log-likelihood $\log p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta})$ of the observed data, the so-called *evidence* of the model. The optimization is guided by the gradients of the evidence with respect to each component of the vector $\boldsymbol{\theta}$. It is worth mentioning that such optimization can be seen as the model selection step of obtaining a plausible GP model from the training data.

3 OUR RECURRENT GP MODEL

We follow an alternative SSM approach where the states have an autoregressive structure. Differently from standard NARX models, the autoregression in our model is performed with *latent* (non-observed) variables. Thus, given L lag steps and introducing the notation $\bar{\boldsymbol{x}}_i = [x_i, \cdots, x_{i-L+1}]^\top$ we have

$$x_i = f(\bar{\boldsymbol{x}}_{i-1}, \bar{\boldsymbol{u}}_{i-1}) + \epsilon_i^x, \tag{8}$$

$$\boldsymbol{y}_i = g(\bar{\boldsymbol{x}}_i) + \boldsymbol{\epsilon}_i^y, \tag{9}$$

where $\bar{\boldsymbol{u}}_{i-1} = [u_{i-1}, \cdots, u_{i-L_u}]^\top$ and L_u is the number of past inputs used. Even if the output of the transition function in Eq. 8 is chosen to be 1-dimensional, it should be noticed that the actual hidden state $\bar{\boldsymbol{x}}_i \in \mathbb{R}^L$ is multidimensional for $L > 1$.

If we have H transition functions, each one comprising a hidden layer, it naturally gives rise to the deep structure

$$x_i^{(h)} = f^{(h)}\left(\hat{\boldsymbol{x}}_i^{(h)}\right) + \epsilon_i^{(h)}, \qquad \boldsymbol{f}^{(h)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(h)}\right), \qquad 1 \le h \le H \tag{10}$$

$$\boldsymbol{y}_i = f^{(H+1)}\left(\hat{\boldsymbol{x}}_i^{(H+1)}\right) + \epsilon_i^{(H+1)}, \qquad \boldsymbol{f}^{(H+1)} \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{K}_f^{(H+1)}\right) \tag{11}$$

where we put GP priors with zero mean and covariance matrix $\boldsymbol{K}_f^{(h)}$ on the unknown functions $f(\cdot)^{(h)}$, the noise in each layer is defined as $\epsilon_i^{(h)} \sim \mathcal{N}(0, \sigma_h^2)$ and the upper index differentiates variables and functions from distinct layers. We also introduce the notation

$$\hat{\boldsymbol{x}}_i^{(h)} = \begin{cases} \left[\bar{\boldsymbol{x}}_{i-1}^{(1)}, \bar{\boldsymbol{u}}_{i-1}\right]^\top = \left[\left[x_{i-1}^{(1)}, \cdots, x_{i-L}^{(1)}\right], [u_{i-1}, \cdots, u_{i-L_u}]\right]^\top, & \text{if } h = 1, \\ \left[\bar{\boldsymbol{x}}_{i-1}^{(h)}, \bar{\boldsymbol{x}}_i^{(h-1)}\right]^\top = \left[\left[x_{i-1}^{(h)}, \cdots, x_{i-L}^{(h)}\right], \left[x_i^{(h-1)}, \cdots, x_{i-L+1}^{(h-1)}\right]\right]^\top, & \text{if } 1 < h \le H, \\ \bar{\boldsymbol{x}}_i^{(H)} = \left[x_i^{(H)}, \cdots, x_{i-L+1}^{(H)}\right]^\top, & \text{if } h = H+1. \end{cases} \tag{12}$$

The graphical model for the RGP is presented in Fig. 1, where we kept the general states $\bar{\boldsymbol{x}}^{(h)}$ to make the recurrent connections more clear. It should be noted that the standard GP-NARX and GP-SSM are also RGPs, but with different states structure.

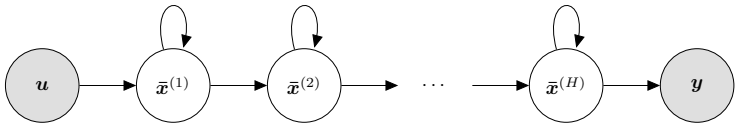


Figure 1: RGP graphical model with H hidden layers.

Our RGP model, as defined by Eqs. 10 and 11, can be seen as a special case of the Deep GP framework (Damianou & Lawrence, 2013; Damianou, 2015) where the priors of the latent variables in each hidden layer follow the autoregressive structure of Eq. 12.

We emphasize that our model preserves the non-observed states of standard SSMs but avoids the ambiguities of generic multidimensional states by imposing a latent autoregressive structure. In the next section, we explain how this novel RGP model can be trained using the REVARB framework.

4 Recurrent Variational Bayes (REVARB)

Inference is intractable in our RGP model because we are not able to get analytical forms for the posterior of $\boldsymbol{f}^{(h)}$ or the marginal likelihood of $\boldsymbol{y}$. In order to tackle such intractabilities, we apply a novel variational approximation scheme named REVARB.

REVARB is based on the variational sparse framework proposed by Titsias (2009), thus, we start by including to each layer h a number of M inducing points $\boldsymbol{z}^{(h)} \in \mathbb{R}^M$ evaluated in M pseudo-inputs $\boldsymbol{\zeta}^{(h)} \in \mathbb{R}^D$ such as that $\boldsymbol{z}^{(h)}$ are extra samples of the GP that models $f^{(h)}(\cdot)$ and $p\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{0}, \boldsymbol{K}_z^{(h)}\right)$, where $\boldsymbol{K}_z^{(h)}$ is the covariance matrix obtained from $\boldsymbol{\zeta}^{(h)}$. Considering a model with H hidden layers and 1-dimensional outputs, the joint distribution of all the variables is given by:

$$
\begin{aligned}
&p\left(\boldsymbol{y}, \boldsymbol{f}^{(H+1)}, \boldsymbol{z}^{(H+1)}, \left\{\boldsymbol{x}^{(h)}, \boldsymbol{f}^{(h)}, \boldsymbol{z}^{(h)}\right\}\middle|_{h=1}^{H}\right) = \\
&\left(\prod_{i=L+1}^{N} p\left(y_i \middle| f_i^{(H+1)}\right) p\left(f_i^{(H+1)} \middle| \boldsymbol{z}^{(H+1)}, \hat{\boldsymbol{x}}_i^{(H)}\right) \prod_{h=1}^{H} p\left(x_i^{(h)} \middle| f_i^{(h)}\right) p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)\right) \\
&\left(\prod_{h=1}^{H+1} p\left(\boldsymbol{z}^{(h)}\right)\right)\left(\prod_{i=1}^{L}\prod_{h=1}^{H} p\left(x_i^{(h)}\right)\right).
\end{aligned}
\tag{13}
$$

By applying Jensen's inequality, similar to the standard variational approach, we can obtain a lower bound to the log-marginal likelihood $\log p(\boldsymbol{y})$ (Bishop, 2006):

$$
\log p(\boldsymbol{y}) \geq \int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} Q \log\left[\frac{p\left(\boldsymbol{y}, \boldsymbol{f}^{(H+1)}, \boldsymbol{z}^{(H+1)}, \left\{\boldsymbol{x}^{(h)}, \boldsymbol{f}^{(h)}, \boldsymbol{z}^{(h)}\right\}\middle|_{h=1}^{H}\right)}{Q}\right], \tag{14}
$$

where Q is the variational distribution. We choose the following factorized expression:

$$
Q = \left(\prod_{h=1}^{H} q\left(\boldsymbol{x}^{(h)}\right)\right)\left(\prod_{h=1}^{H+1} q\left(\boldsymbol{z}^{(h)}\right)\right)\left(\prod_{i=L+1}^{N}\prod_{h=1}^{H+1} p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right)\right). \tag{15}
$$

Considering a mean-field approximation, each term is given by

$$q\left(\boldsymbol{x}^{(h)}\right) = \prod_{i=1}^{N} \mathcal{N}\left(x_i^{(h)} \middle| \mu_i^{(h)}, \lambda_i^{(h)}\right), \tag{16}$$

$$q\left(\boldsymbol{z}^{(h)}\right) = \mathcal{N}\left(\boldsymbol{z}^{(h)} \middle| \boldsymbol{m}^{(h)}, \boldsymbol{\Sigma}^{(h)}\right), \tag{17}$$

$$p\left(f_i^{(h)} \middle| \boldsymbol{z}^{(h)}, \hat{\boldsymbol{x}}_i^{(h)}\right) = \mathcal{N}\left(f_i^{(h)} \middle| \left[\boldsymbol{a}_f^{(h)}\right]_i, \left[\boldsymbol{\Sigma}_f^{(h)}\right]_{ii}\right), \tag{18}$$

$$\text{where } \boldsymbol{a}_f^{(h)} = \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \boldsymbol{z}^{(h)} \quad \text{and} \quad \boldsymbol{\Sigma}_f^{(h)} = \boldsymbol{K}_f^{(h)} - \boldsymbol{K}_{fz}^{(h)} \left(\boldsymbol{K}_z^{(h)}\right)^{-1} \left(\boldsymbol{K}_{fz}^{(h)}\right)^{\top}.$$

In the above, $\mu_i^{(h)}$, $\lambda_i^{(h)}$, $\boldsymbol{m}^{(h)}$ and $\boldsymbol{\Sigma}^{(h)}$ are variational parameters, $\boldsymbol{K}_f^{(h)}$ is the standard kernel matrix obtained from $\hat{\boldsymbol{x}}^{(h)}$, $\boldsymbol{K}_z^{(h)}$ is the sparse kernel matrix calculated from the pseudo-inputs $\boldsymbol{\zeta}^{(h)}$ and $\boldsymbol{K}_{fz}^{(h)} = k(\hat{\boldsymbol{x}}^{(h)}, \boldsymbol{\zeta}^{(h)}) \in \mathbb{R}^{N \times M}$.

The variational distribution in Eq. 16 indicates that the latent variables $\boldsymbol{x}^{(h)}$ are related to $2N$ variational parameters. In standard variational GP-SSM, such as in Frigola et al. (2014) we would have a total of $2ND$ parameters, for D-dimensional states, even for a diagonal covariance matrix in the posterior. Such reduction of parameters in the mean-field approximation was enabled by the latent autoregressive structure of our model.

Replacing the variational distribution in the Eq. 14 and working the expressions we are able to optimally eliminate the variational parameters $\boldsymbol{m}^{(h)}$ and $\boldsymbol{\Sigma}^{(h)}$, obtaining the final form of the lower bound, presented in the included appendix. We have to compute some statistics that come up in the full bound:

$$\begin{aligned} \Psi_0^{(h)} &= \operatorname{Tr}\left(\left\langle \boldsymbol{K}_f^{(h)} \right\rangle_{q(\cdot)^{(h)}}\right) \\ \boldsymbol{\Psi}_1^{(h)} &= \left\langle \boldsymbol{K}_{fz}^{(h)} \right\rangle_{q(\cdot)^{(h)}} \\ \boldsymbol{\Psi}_2^{(h)} &= \left\langle \left(\boldsymbol{K}_{fz}^{(h)}\right)^{\top} \boldsymbol{K}_{fz}^{(h)} \right\rangle_{q(\cdot)^{(h)}} \end{aligned} \quad \Rightarrow q(\cdot)^{(h)} = \begin{cases} q\left(\boldsymbol{x}^{(1)}\right), & \text{if } h = 1, \\ q\left(\boldsymbol{x}^{(h)}\right) q\left(\boldsymbol{x}^{(h-1)}\right), & \text{if } 1 < h \leq H, \\ q\left(\boldsymbol{x}^{(H)}\right), & \text{if } h = H+1, \end{cases} \tag{19}$$

where $\langle \cdot \rangle_{q(\boldsymbol{x}^{(h)})}$ means expectation with respect to the distribution $q\left(\boldsymbol{x}^{(h)}\right)$, which itself depends only on the variational parameters $\mu_i^{(h)}$ and $\lambda_i^{(h)}$. All the expectations are tractable for our choice of the exponentiated quadratic covariance function and follow the same expressions presented by **?**. The bound can be optimized with the help of analytical gradients with respect to the kernel and variational hyperparameters.

The REVARB framework allows for a natural way to approximately propagate the uncertainty during both training and prediction. For testing, given a new sequence of external inputs, we can calculate the moments of the predictive distribution of each layer by recursively applying the results introduced in Girard et al. (2003), with predictive equations presented in the included appendix.

4.1 Sequential RNN-based recognition model

From Eq. (16) it is obvious that the number of variational parameters in REVARB grows linearly with the number of output samples. This renders optimization challenging in large N scenarios. To alleviate this problem we propose to constrain the variational means $\left\{\mu_i^{(h)}\right\}, \forall h, i$ using RNNs. More specifically, we have:

$$\mu_i^{(h)} = g^{(h)}\left(\hat{\boldsymbol{x}}_{i-1}^{(h)}\right), \text{where } g(\boldsymbol{x}) = \boldsymbol{V}_{L_N}^{\top} \boldsymbol{\phi}_{L_N}(\boldsymbol{W}_{L_N-1} \boldsymbol{\phi}_{L_N-1}(\cdots \boldsymbol{W}_2 \boldsymbol{\phi}_1(\boldsymbol{U}_1 \boldsymbol{x}))), \tag{20}$$

$\boldsymbol{W}, \boldsymbol{U}$ and $\boldsymbol{V}$ are parameter matrices, $\boldsymbol{\phi}(\cdot)$ denotes the hyperbolic tangent activation function and L_N denotes the depth of the neural network. We refer to this RNN-based constraint as the *sequential recognition model*. Such model directly captures the transition between the latent representation across time. This provides a constraint over the variational posterior distribution of the RGP that emphasizes free simulation. The recognition model's influence is combined with that of the analytic

lower bound in the same objective optimization function. In this way, we no longer need to optimize the variational means but, instead, only the set of RNN weights, whose number does not increase linearly with N. Importantly, this framework also allows us to kick-off optimization by random initialization of the RNN weights, as opposed to more elaborate initialization schemes. The recognition model idea relates to the work of (Kingma & Welling, 2013; Rezende et al., 2014). In our case, however, the recognition model is sequential to agree with the latent structure and its purpose is distinct, because it acts as a constraint in an already analytic variational lower bound. Furthermore, our sequential recognition model acts upon a nonparametric Bayesian model.

5 EXPERIMENTS

In this section we evaluate the performance of our RGP model in the tasks of nonlinear system identification and human motion modeling.

5.1 NONLINEAR SYSTEM IDENTIFICATION

We use one artificial benchmark, presented by Narendra & Li (1996), and two real datasets. The first real dataset, named *Actuator* and described by Sjöberg et al. (1995) [1], consists of a hydraulic actuator that controls a robot arm, where the input is the size of the actuator's valve opening and the output is its oil pressure. The second dataset, named *Drives* and introduced by Wigren (2010), is comprised by a system with two electric motors that drive a pulley using a flexible belt. The input is the sum of voltages applied to the motors and the output is the speed of the belt.

In the case of the artificial dataset we choose $L = L_u = 5$ and generate 300 samples for training and 300 samples for testing, using the same inputs described by Narendra & Li (1996). For the real datasets we use $L = L_u = 10$ and apply the first half of the data for training and the second one for testing. The evaluation is done by calculating the root mean squared error (RMSE) of the free simulation on the test data. We emphasize that the predictions are made based only on the test inputs and past predictions.

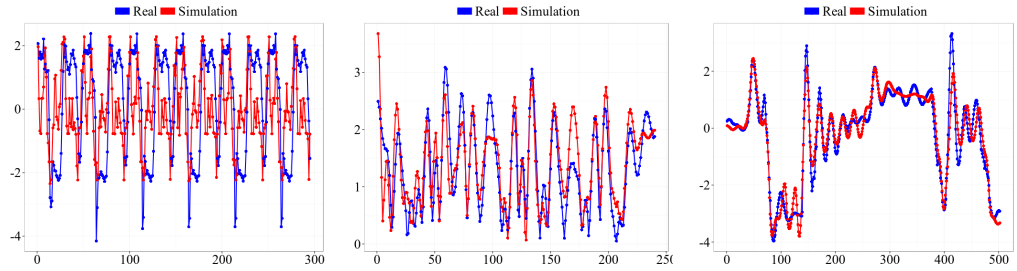
We compare our RGP model with 2 hidden layers, REVARB inference and 100 inducing inputs with two models commonly applied to system identification tasks: standard GP-NARX and MLP-NARX. We use the MLP implementation from the MATLAB Neural Network Toolbox with 1 hidden layer. We also include experiments with the LSTM network, although the task itself probably does not require long term dependences. The original LSTM architecture by Hochreiter & Schmidhuber (1997) was chosen, with a network depth of 1 to 3 layers and the number of cells at each layer selected to be up to 2048. LSTM memory length was unlimited, and sequence length was chosen initially to be a multiple of the longest duration memory present in the data generative process, and reduced gradually. During experiments with varying LSTM network configurations, it became clear that it was possible in most cases to obtain convergence on the training sets, using a carefully chosen network model size and hyperparameters. Training was organized around batches, and achieved using a learning rate selected to fall slightly below loop instability, and it was incrementally reduced when instability re-appeared. A batch in this context is the concatenation of fixed length sub-sequences of the temporal data set. Neither gradient limits nor momentum were used.

The results are summarized in Tab. 1 and the obtained simulations are illustrated in Fig. 2. The REVARB model was superior in all cases, with large improvements over GP-NARX. Although worse than REVARB, the MLP-NARX model presented good results, specially for the *Actuator* dataset. The higher RMSE values obtained by the LSTM model is possibly related to the difficulties we have encountered when trying to optimize its architecture for this given task.

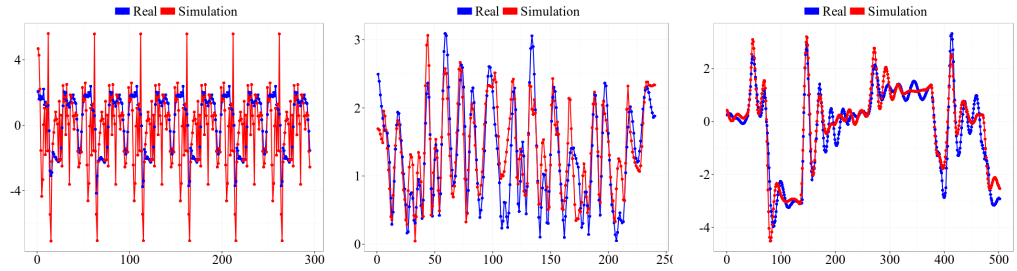
Table 1: Summary of RMSE values for the free simulation results on system identification test data.

	MLP-NARX	LSTM	GP-NARX	REVARB
Artificial	1.6334	2.2438	1.9245	**0.4513**
Drive	0.4403	0.4329	0.4128	**0.2491**
Actuator	0.4621	0.5170	1.5488	**0.3680**

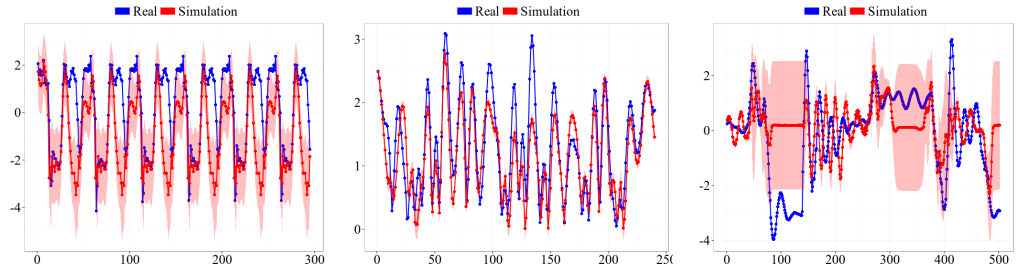
[1] Available in the DaISy repository at `http://www.iau.dtu.dk/nnbook/systems.html`.



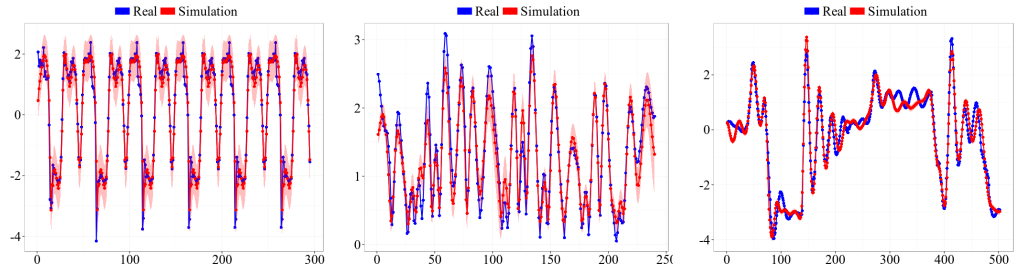
(a) MLP-NARX - Artificial dataset. (b) MLP-NARX - *Drives* dataset. (c) MLP-NARX - *Actuator* dataset.



(d) LSTM - Artificial dataset. (e) LSTM - *Drives* dataset. (f) LSTM - *Actuator* dataset.



(g) GP-NARX - Artificial dataset. (h) GP-NARX - *Drives* dataset. (i) GP-NARX - *Actuator* dataset.



(j) REVARB - Artificial dataset. (k) REVARB - *Drives* dataset. (l) REVARB - *Actuator* dataset.

Figure 2: Free simulation on system identification test data.

5.2 Human Motion Modeling

The motion capture data from the CMU database[2] was used to model walking and running motions. Training was performed with the trajectories 1 to 4 (walking) and 17 to 20 (running) from subject 35. The test set is comprised by the trajectories 5 to 8 (walking) and 21 to 24 (running) from the

[2]Available at `http://mocap.cs.cmu.edu/`.

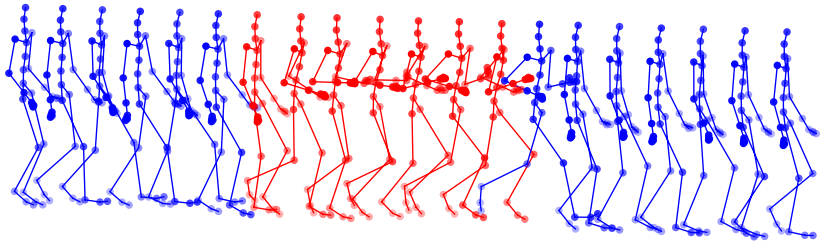


Figure 3: The generated motion with a step function signal, starting with walking (blue), switching to running (red) and switching back to walking (blue).

same subject. The original dataset contains 59 outputs, but 2 are constant, so we remove those and use the remaining 57.

In order to perform free simulation in the test set, we include a control input given by the y coordinate of the left toes. Following the previous system identification experiments, predictions are made based only on such control input and previous predictions. We normalize the inputs and outputs with zero mean and unitary standard deviation.

We evaluate a 2 hidden layer REVARB with 200 inducing inputs, the standard GP-NARX model and a 1 hidden layer MLP with 1000 hidden units. The orders are fixed at $L = L_u = 20$. Note that the data related to both walking and running is used in the same training step. The latent autoregressive structure of REVARB allow us to train a single model for all outputs. In the case of GP-NARX, we had to train separate models for each output, since training a single model with $57 \times 20 + 20 = 1160$ dimensional regressor vector was not feasible.

The mean of the test RMSE values are summarized in Tab. 2. The REVARB model obtained better results than both the other models. We emphasize that REVARB has an additional advantage over GP-NARX because its latent autoregressive structure allows the training of a single mode for all the outputs.

Table 2: Summary of RMSE values for the free simulation results on human motion test data.

MLP-NARX	GP-NARX	REVARB
1.2141	0.8987	**0.8600**

5.3 Avatar control

We demonstrate the capability of RGP by applying it to synthesize human motions with simple control signals such as the velocity. Such system ideally can be used to generate realistic human motion according to human instruction in virtual environment such as video games. We use the 5 walking and 5 running sequences from CMU motion database and take the average velocity as the control signal. We train a 1 hidden layer REVARB model with the RNN sequential recognition model (two hidden layer 500-200 units). After training, we use the model to synthesize motions with unseen control signals. Figure 3 shows the frames of the generated motion with a step function signal (the training sequences do not contain any switch of motions). The video of this and some more motions are available at `https://youtu.be/FuF-uZ83VMw`, `https://youtu.be/FR-oeGxV6yY`, `https://youtu.be/AT0HMtoPgjc`.

6 Discussion and Further Work

We defined the broad family of Recurrent Gaussian Processes models, which, similarly to RNNs, are able to learn, possibly deep, temporal representations from data. We also proposed a novel RGP model with a latent autoregressive structure where the intractabilities brought by the recurrent GP priors are tackled with a variational approximation approach, resulting in the REVARB framework.

Furthermore, we extended REVARB with a sequential RNN-based recognition model that simplifies the optimization.

We applied REVARB to the tasks of nonlinear system identification and human motion modeling. The good results obtained by our model indicate that the latent autoregressive structure and our variational approach were able to better capture the dynamical behavior of the data.

In the work Turner & Sahani (2008), the authors present some concerns with respect to the use of mean-field approximations within a time-series context, suggesting that such approximation has a hard time propagating uncertainty through time. However, we observed in practice that our proposed REVARB framework is able to better account for uncertainty in the latent space with its autoregressive deep structure. This may be because the next layer is able to 'compensate' the mean-field assumption of the previous layer, accounting for additional (temporal) correlations. Since each latent variable x_i and, thus, its associated variational parameters, is present in two layers (see Eq. 12), this effect is enabled for all latent variables of the model. A similar observation is made for regular deep GPs by Damianou (2015).

The flexibility of GP modeling along with expressive recurrent structures is a theme for further theoretical investigations and practical applications. For instance, we intend to verify if some of the recommendations for deep modeling described by Duvenaud et al. (2014) would be helpful for our RGP model. Finally, we hope that our paper opens up new directions in the study of the parallels between RGPs and RNNs. To this end, we intend to explore the REVARB approach within longer term memory tasks and extend it with non-Gaussian likelihood distributions.

Acknowledgements. The authors thank the financial support of CAPES, FUNCAP, NUTEC, CNPq (Maresia, grant 309451/2015-9, and Amalia, grant 407185/2013-5), RADIANT (EU FP7-HEALTH Project Ref 305626) and WYSIWYD (EU FP7-ICT Project Ref 612139).

REFERENCES

Bengio, Yoshua, Simard, Patrice, and Frasconi, Paolo. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166, 1994.

Bishop, Christopher M. *Pattern recognition and machine learning*. Springer, 2006.

Boulanger-lewandowski, Nicolas, Bengio, Yoshua, and Vincent, Pascal. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1159–1166, 2012.

Cho, Kyunghyun, Van Merriënboer, Bart, Gulcehre, Caglar, Bahdanau, Dzmitry, Bougares, Fethi, Schwenk, Holger, and Bengio, Yoshua. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Damianou, Andreas. Deep Gaussian processes and variational propagation of uncertainty. *PhD Thesis, University of Sheffield*, 2015.

Damianou, Andreas and Lawrence, Neil. Deep Gaussian processes. In *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*, pp. 207–215, 2013.

Damianou, Andreas and Lawrence, Neil. Semi-described and semi-supervised learning with Gaussian processes. In *31st Conference on Uncertainty in Artificial Intelligence (UAI)*, 2015.

Duvenaud, David, Rippel, Oren, Adams, Ryan P, and Ghahramani, Zoubin. Avoiding pathologies in very deep networks. *arXiv preprint arXiv:1402.5836*, 2014.

El Hihi, Salah and Bengio, Yoshua. Hierarchical recurrent neural networks for long-term dependencies. In *Advances in Neural Information Processing Systems*, pp. 493–499, 1996.

Frigola, Roger, Chen, Yutian, and Rasmussen, Carl. Variational Gaussian process state-space models. In *Advances in Neural Information Processing Systems 27 (NIPS)*, pp. 3680–3688, 2014.

Girard, A., Rasmussen, CE., Quiñonero-Candela, J., and Murray-Smith, R. Multiple-step ahead prediction for non linear dynamic systems: A gaussian process treatment with propagation of the uncertainty. In *Advances in Neural Information Processing Systems 15*, pp. 529–536. MIT Press, Cambridge, MA, USA, 2003.

Graves, Alan, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 6645–6649. IEEE, 2013.

Graves, Alex. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Hochreiter, Sepp and Schmidhuber, Jürgen. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

King, Nathaniel J and Lawrence, Neil D. Fast variational inference for gaussian process models through kl-correction. In *Machine Learning: ECML 2006*, pp. 270–281. Springer, 2006.

Kingma, Diederik P and Welling, Max. Auto-Encoding Variational Bayes. In *ICLR*, 2013.

Kocijan, Juš, Girard, Agathe, Banko, Blaž, and Murray-Smith, Roderick. Dynamic systems identification with Gaussian processes. *Math Comp Model Dyn*, 11(4):411–424, 2005.

Lang, Kevin J, Waibel, Alex H, and Hinton, Geoffrey E. A time-delay neural network architecture for isolated word recognition. *Neural networks*, 3(1):23–43, 1990.

Lin, Tsungnam, Horne, Bil G, Tiňo, Peter, and Giles, C Lee. Learning long-term dependencies in narx recurrent neural networks. *Neural Networks, IEEE Transactions on*, 7(6):1329–1338, 1996.

Murray-Smith, Roderick, Johansen, Tor A, and Shorten, Robert. On transient dynamics, off-equilibrium behaviour and identification in blended multiple model structures. In *European Control Conference (ECC'99), Karlsruhe, BA-14*. Springer, 1999.

Narendra, Kumpati S and Li, Sai-Ming. Neural networks in control systems. *Mathematical Perspectives on Neural Networks*, pp. 347–394, 1996.

Pascanu, Razvan, Gulcehre, Caglar, Cho, Kyunghyun, and Bengio, Yoshua. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

Peterka, V. Bayesian approach to system identification. *Trends and Prog in Syst ident*, 1:239–304, 1981.

Rezende, D J, Mohamed, S, and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Sjöberg, Jonas, Zhang, Qinghua, Ljung, Lennart, Benveniste, Albert, Delyon, Bernard, Glorennec, Pierre-Yves, Hjalmarsson, Håkan, and Juditsky, Anatoli. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

Sohl-Dickstein, Jascha and Kingma, Diederik P. Technical note on equivalence between recurrent neural network time series models and variational bayesian models. *arXiv preprint arXiv:1504.08025*, 2015.

Solak, Ercan, Murray-Smith, Roderick, Leithead, William E, Leith, Douglas J, and Rasmussen, Carl Edward. Derivative observations in Gaussian process models of dynamic systems. *Advances in Neural Information Processing Systems*, 16, 2003.

Svensson, Andreas, Solin, Arno, Särkkä, Simo, and Schön, Thomas B. Computationally efficient bayesian learning of gaussian process state space models. *arXiv preprint arXiv:1506.02267*, 2015.

Titsias, Michalis K. Variational learning of inducing variables in sparse gaussian processes. In *International Conference on Artificial Intelligence and Statistics*, pp. 567–574, 2009.

Turner, Richard E and Sahani, Maneesh. Two problems with variational expectation maximisation for time-series models. In *Proceedings of the Workshop on Inference and Estimation in Probabilistic Time-Series Models*, pp. 107–115, 2008.

Wigren, Torbjörn. Input-output data sets for development and benchmarking in nonlinear identification. *Technical Reports from the department of Information Technology*, 20:2010–020, 2010. Dataset available in `http://www.it.uu.se/research/publications/reports/2010-020/NonlinearData.zip` as *DATAPRBS.mat*, with input u1 and output z1.

APPENDIX

REVARB LOWER BOUND

Replacing the definition of the joint distribution (Eq. 13) and the factorized variational distribution Q (Eq. 15) in the Jensen's inequality of Eq. 14, we are able to cancel the terms $p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)$ inside the $\log$:

$$\begin{aligned}
\log p(\boldsymbol{y}) \geq & \sum_{i=L+1}^{N}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}} q\left(\boldsymbol{x}^{(H)}\right) q\left(\boldsymbol{z}^{(H+1)}\right) p\left(f_i^{(H+1)}\middle|\boldsymbol{z}^{(H+1)},\hat{\boldsymbol{x}}_i^{(H)}\right)\log p\left(y_i\middle|f_i^{(H+1)}\right)\\
& + \sum_{i=L+1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{f},\boldsymbol{x},\boldsymbol{z}}\left(\prod_{h'=1}^{H} q\left(\boldsymbol{x}^{(h')}\right)\right) q\left(\boldsymbol{z}^{(h)}\right) p\left(f_i^{(h)}\middle|\boldsymbol{z}^{(h)},\hat{\boldsymbol{x}}_i^{(h)}\right)\log p\left(x_i^{(h)}\middle|f_i^{(h)}\right)\\
& - \sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log q\left(\boldsymbol{z}^{(h)}\right) + \sum_{h=1}^{H+1}\int_{\boldsymbol{z}} q\left(\boldsymbol{z}^{(h)}\right)\log p\left(\boldsymbol{z}^{(h)}\right)\\
& - \sum_{i=L+1}^{N}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\sum_{h=1}^{H}\int_{\boldsymbol{x}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right),
\end{aligned} \tag{21}$$

where the integrals are tractable, since all the distributions are Gaussians. The expectations with respect to $\boldsymbol{x}_i^{(h)}$ give rise to the statistics $\Psi_0^{(h)}$, $\boldsymbol{\Psi}_1^{(h)}$ and $\boldsymbol{\Psi}_2^{(h)}$, defined in Eq. 19.

Following similar argument of King & Lawrence (2006), we are able to optimally eliminate the variational parameters associated with the inducing points, $\boldsymbol{m}^{(h)}$ and $\boldsymbol{\Sigma}^{(h)}$ and get to the final form of the REVARB lower bound:

$$\begin{aligned}
\log p(\boldsymbol{y}) \geq & -\frac{N-L}{2}\sum_{h=1}^{H+1}\log 2\pi\sigma_h^2 - \frac{1}{2\sigma_{H+1}^2}\left(\boldsymbol{y}^\top\boldsymbol{y} + \Psi_0^{(H+1)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(H+1)}\right)^{-1}\boldsymbol{\Psi}_2^{(H+1)}\right)\right)\\
& + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(H+1)}\right| - \log\frac{1}{2}\left|\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right|\\
& + \frac{1}{2(\sigma_{H+1}^2)^2}\boldsymbol{y}^\top\boldsymbol{\Psi}_1^{(H+1)}\left(\boldsymbol{K}_z^{(H+1)} + \frac{1}{\sigma_{H+1}^2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top\boldsymbol{y}\\
& + \sum_{h=1}^{H}\left\{-\frac{1}{2\sigma_h^2}\left(\sum_{i=L+1}^{N}\lambda_i^{(h)} + \left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\mu}^{(h)} + \Psi_0^{(h)} - \operatorname{Tr}\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1}\boldsymbol{\Psi}_2^{(h)}\right)\right)\right.\\
& + \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)}\right| - \frac{1}{2}\log\left|\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right|\\
& + \frac{1}{2(\sigma_h^2)^2}\left(\boldsymbol{\mu}^{(h)}\right)^\top\boldsymbol{\Psi}_1^{(h)}\left(\boldsymbol{K}_z^{(h)} + \frac{1}{\sigma_h^2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top\boldsymbol{\mu}^{(h)}\\
& \left. - \sum_{i=L+1}^{N}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log q\left(x_i^{(h)}\right) + \sum_{i=1}^{L}\int_{x_i^{(h)}} q\left(x_i^{(h)}\right)\log p\left(x_i^{(h)}\right)\right\}.
\end{aligned} \tag{22}$$

Note that the parameters of the Gaussian priors $p\left(x_i^{(h)}\right) = \mathcal{N}\left(x_i^{(h)}\middle|\mu_{0i}^{(h)},\lambda_{0i}^{(h)}\right)$ of the initial latent variables $x_i^{(h)}|_{i=1}^{L}$ can be optimized along with the variational parameters and kernel hyperparameters.

REVARB PREDICTIVE EQUATIONS

Predictions in the REVARB framework are done iteratively, with approximate uncertainty propagation between each layer:

$$\mu_*^{(h)} = \mathbb{E}\left\{p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\} = \left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top, \tag{23}$$

$$\begin{aligned}\lambda_*^{(h)} = \mathbb{V}\left\{p\left(f_*^{(h)}\middle|\hat{\boldsymbol{x}}_*^{(h)}\right)\right\} &= \left(\boldsymbol{B}^{(h)}\right)^\top \left(\boldsymbol{\Psi}_{2*}^{(h)} - \left(\boldsymbol{\Psi}_{1*}^{(h)}\right)^\top \boldsymbol{\Psi}_{1*}^{(h)}\right)\boldsymbol{B}^{(h)} + \Psi_{0*}^{(h)} \\ &- \mathrm{Tr}\left(\left(\left(\boldsymbol{K}_z^{(h)}\right)^{-1} - \left(\boldsymbol{K}_z^{(h)} + \sigma_h^{-2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\right)\boldsymbol{\Psi}_{2*}^{(h)}\right),\end{aligned} \tag{24}$$

where $\hat{\boldsymbol{x}}_*^{(h)}$ is defined similar to the Eq. 12, $\boldsymbol{B}^{(h)} = \sigma_h^{-2}\left(\boldsymbol{K}_z^{(h)} + \sigma_h^{-2}\boldsymbol{\Psi}_2^{(h)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(h)}\right)^\top \boldsymbol{\mu}^{(h)}$, for $1 \leq h \leq H$, and $\boldsymbol{B}^{(H+1)} = \sigma_{H+1}^{-2}\left(\boldsymbol{K}_z^{(H+1)} + \sigma_{H+1}^{-2}\boldsymbol{\Psi}_2^{(H+1)}\right)^{-1}\left(\boldsymbol{\Psi}_1^{(H+1)}\right)^\top \boldsymbol{y}$. The terms $\Psi_{0*}^{(h)}$, $\boldsymbol{\Psi}_{1*}^{(h)}$ and $\boldsymbol{\Psi}_{2*}^{(h)}$ are computed as in the Eq. 19, but instead of the distributions $q\left(x_i^{(h)}\right)$ we use the new Gaussian approximation $q\left(x_*^{(h)}\right) = \mathcal{N}\left(x_*^{(h)}\middle|\mu_*^{(h)}, \lambda_*^{(h)}\right)$ and replace $\boldsymbol{K}_f^{(h)}$ and $\boldsymbol{K}_{fz}^{(h)}$ respectively by $\boldsymbol{K}_*^{(h)} = k\left(\hat{\boldsymbol{x}}_*^{(h)}, \hat{\boldsymbol{x}}_*^{(h)}\right)$ and $\boldsymbol{K}_{*z}^{(h)} = k\left(\hat{\boldsymbol{x}}_*^{(h)}, \boldsymbol{\zeta}^{(h)}\right)$.