Probabilistic Recurrent State-Space Models

Andreas Doerr [1 2] **Christian Daniel** [1] **Martin Schiegg** [1] **Duy Nguyen-Tuong** [1] **Stefan Schaal** [2 3] **Marc Toussaint** [4]
Sebastian Trimpe [2]

Abstract

State-space models (SSMs) are a highly expressive model class for learning patterns in time series data and for system identification. Deterministic versions of SSMs (e.g., LSTMs) proved extremely successful in modeling complex time-series data. Fully probabilistic SSMs, however, unfortunately often prove hard to train, even for smaller problems. To overcome this limitation, we propose a scalable initialization and training algorithm based on doubly stochastic variational inference and Gaussian processes. In the variational approximation we propose in contrast to related approaches to fully capture the latent state temporal correlations to allow for robust training.

1. Introduction

System identification, i.e. learning dynamics models from data (Ljung, 1998; 2010), is a key ingredient of model-predictive control (Camacho & Alba, 2013) and model-based reinforcement learning (RL) (Deisenroth & Rasmussen, 2011; Doerr et al., 2017b). State-Space Models (SSMs) are one popular class of representations (Billings, 2013), which describe a system with input u_t and output y_t in terms of a latent Markovian state x_t. Based on a transition model f and an observation model g as well as process and measurement noise ϵ_t and γ_t, a time-discrete SSM is given by

$$x_{t+1} = f(x_t, u_t) + \epsilon_t,$$
$$y_t = g(x_t, u_t) + \gamma_t. \qquad (1)$$

Typically, in real systems, the latent state x_t cannot be measured directly but has to be inferred from a series of noisy, less informative output observations. For *linear* SSMs, this inference problem and the model learning can be solved simultaneously by subspace identification (Van Overschee & De Moor, 2012). Solutions also exist for the *deterministic* but *non-linear* counterpart, e.g. recurrent neural networks (RNNs) (Hochreiter & Schmidhuber, 1997).

However, for tasks such as learning control, probabilistic models are required to enable safe learning and to avoid model bias (Deisenroth & Rasmussen, 2011). Robust training of *probabilistic, non-linear* SSMs is a challenging and only partially solved problem, especially for higher dimensional systems (Frigola et al., 2013; 2014; Eleftheriadis et al., 2017; Svensson & Schön, 2017). This paper proposes the Probabilistic Recurrent State-Space Model[1] (PR-SSM[2]), a framework which tackles the key challenges preventing robust training of probabilistic, non-linear SSMs. PR-SSM takes inspiration from RNN model learning. In particular, the latent state transition model is unrolled over time, therefore accounting for temporal correlations whilst simultaneously allowing learning by backpropagation through time and mitigating the problem of latent state initialization. Grounded in the theory of Gaussian Processes (GPs), the proposed method enables probabilistic model predictions, complex latent state distributions, and principled model complexity regularization. Furthermore, we propose an adapted form of a recognition model for scalability through batch learning and learning of slow or unstable system dynamics.

In summary, the key contributions of this paper are:

- Combining gradient-based and sample-based inference for efficient learning of complex posteriors.
- Tractable variational approximation, maintaining the latent state posterior and temporal correlations.
- Doubly stochastic inference scheme for scalability.
- Recognition model which allows robust training by initializing the latent state distribution.

Together, these contributions allow for robust training of the PR-SSM. The proposed framework is evaluated on a set of real-world system identification datasets and benchmarked against a range of state-of-the art methods.

---

[1]Bosch Center for Artificial Intelligence, Renningen, Germany [2]Autonomous Motion Dept., Max Planck Institute for Intelligent Systems, Tübingen, Germany [3]Computational Learning and Motor Control Lab, University of Southern California, USA [4]Machine Learning and Robotics Lab, University of Stuttgart, Germany. Correspondence to: Andreas Doerr <andreasdoerr@gmx.net>.

---

[1]Code available after publication at: `https://github.com/andreasdoerr/PR-SSM`.

[2]Pronounced *prism*

2. Related Work

Modeling the behavior of systems with only partially observable states has been an active field of research for many years and several schools of thought have emerged. Representations range from SSMs (Van Overschee & De Moor, 2012) over Predictive State Representations (PSRs) (Littman & Sutton, 2002; Singh et al., 2003; Rudary & Singh, 2004) to autoregressive models (Murray-Smith & Girard, 2001; Girard et al., 2003; Likar & Kocijan, 2007; Billings, 2013) as well as hybrid versions combining these approaches (Mattos et al., 2015; 2016; Doerr et al., 2017a).

Where SSMs aim to capture the true latent state of the observed system, PSRs aim to represent a state which allows the best predictions for future observations (Littman & Sutton, 2002). While the concept of PSRs is intuitively appealing, the process of training them successfully is often hard which has prevented wide-spread adoption so far.

Autoregressive (history-based) methods, on the other hand, avoid the complex inference of a latent state and instead directly learn a mapping from a history of h past inputs and observations to the next observation, i.e. $\boldsymbol{y}_{t+1} = f(\boldsymbol{y}_{t:t-h}, \boldsymbol{u}_{t:t-h})$. These models face the issue of feeding back observation noise into the dynamics model. Recent works address this problem by either actively accounting for input noise (McHutchon & Rasmussen, 2011) or reverting to an autoregressive formulation in a latent but noise free state (Mattos et al., 2016; Doerr et al., 2017a). Such models can be made deep and trained in a recurrent manner as presented in (Mattos et al., 2015). In theory, a horizon h identical to the true latent state dimensionality $|\boldsymbol{x}_t|$ is sufficient to model all relevant dependencies of the system under consideration (Ljung, 1998). In practice, however, autoregressive models typically need a much larger history horizon to cope with noisy observations and arbitrary sampling frequencies.

Thus, in this paper, we focus on SSMs based on a compact, Markovian state representations. Furthermore, SSMs allow the direct application of many existing control algorithms which rely on the explicit representation of the latent state. Within the field of latent state models, exact solutions are known for linear SSMs and can be obtained by the well known Kalman filter/smoother (Kalman, 1960) and subspace identification (Van Overschee & De Moor, 2012). In the case of non-linear latent state transition dynamics, both deterministic and probabilistic variants are active fields of research. Deterministic variants such as LSTMs have been shown to be powerful representations for tasks such as natural language processing (Venugopalan et al., 2014) or text understanding (Sutskever et al., 2011). However, for the purpose of system identification and control, experience has shown that probabilistic predictions are required to make model errors explicit (Deisenroth & Rasmussen, 2011).

Gaussian Process State-Space Models (GP-SSMs) are one popular class of probabilistic SSMs. The use of GPs allows a fully Bayesian treatment of the modeling problem resulting in an automatic complexity trade-off which regularizes the learning problem.

The inference problem, i.e. filtering and smoothing in GP-SSMs, has already been covered extensively: deterministic (e.g. linearization) as well as stochastic (e.g. particles) methods are presented by (Ko & Fox, 2009; Deisenroth et al., 2012). These methods, however, assume an established system model which is generally not available without prior knowledge.

Methods for deriving GP-SSM models from data based on maximum likelihood estimates are proposed by (Wang et al., 2008; Ko & Fox, 2009; Turner et al., 2010). In contrast, (Frigola et al., 2013) presents Bayesian learning of GP-SSMs using sampling methods for latent state inference and model learning. (Frigola et al., 2014) presents a variational approach for model parameter learning and sampling from the smoothing distribution, based on the variational sparse GP framework (Titsias, 2009). These methods rely on particle Markov chain Monte Carlo methods which makes them computationally expensive for higher dimensionality and larger datasets.

To prevent a linear growth of latent state parameters in the variational distribution, a recognition model is proposed in (Mattos et al., 2015; Eleftheriadis et al., 2017) following ideas from Dai et al. 2015. The recognition model (e.g., a RNN) acts as learned latent state smoother given a system's input/output data. However, in (Mattos et al., 2015), temporal correlations in the latent state distribution are dropped due to the mean field assumption, which is made for the latent states. (Eleftheriadis et al., 2017) proposes to directly incorporate linear dependencies between consecutive states instead of recovering temporal correlation in the model optimization step. However, they employ a time-varying linear transition model in the variational approximation, which has to be constrained by a powerful recognition model in order to enable learning.

3. Gaussian Process State-Space Model

This section presents the proposed PR-SSM (Sec. 3.3) after a short recap of GPs (Sec. 3.1) and the description of a specific sparse GP prior (Sec. 3.2).

3.1. Gaussian Process

A GP (Williams & Rasmussen, 2005) is a distribution over functions $f : \mathbb{R}^D \to \mathbb{R}$ that is fully defined by a mean function $m(\cdot)$ and covariance function $k(\cdot, \cdot)$. For each finite set of points $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ from the function's domain, the corresponding function evaluations

$\boldsymbol{f} = [f(\boldsymbol{x}_1), \dots, f(\boldsymbol{x}_N)]$ are jointly Gaussian as given by

$$p(\boldsymbol{f} \mid \boldsymbol{X}) = \mathcal{N}(\boldsymbol{f} \mid \boldsymbol{m}_X, \boldsymbol{K}_{X,X}), \qquad (2)$$

with mean vector $\boldsymbol{m}_X$ having elements $m_i = m(\boldsymbol{x}_i)$ and covariance matrix $\boldsymbol{K}_{X,X}$ with elements $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ Given observed function values $\boldsymbol{f}$ at input locations $\boldsymbol{X}$, the GP's predictive distribution at a new input location $\boldsymbol{x}^*$ is obtained as the conditional distribution

$$p(f^* \mid \boldsymbol{x}^*, \boldsymbol{f}, \boldsymbol{X}) = \mathcal{N}(f^* \mid \mu, \sigma^2), \qquad (3)$$

with mean and variance

$$\mu = m_{x^*} + \boldsymbol{k}_{x^*,X} \boldsymbol{K}_{X,X}^{-1}(\boldsymbol{f} - \boldsymbol{m}_X), \qquad (4)$$

$$\sigma^2 = k_{x^*,x^*} - \boldsymbol{k}_{x^*,X} \boldsymbol{K}_{X,X}^{-1} \boldsymbol{k}_{X,x^*}, \qquad (5)$$

where $k_{A,B}$ denotes the scalar or vector of covariances for each pair of elements in $\boldsymbol{A}$ and $\boldsymbol{B}$. In this work, the squared exponential kernel with Automatic Relevance Determination (ARD) (Williams & Rasmussen, 2005) with hyperparameters θ_{GP} is employed. Due to the proposed sampling-based inference scheme (cf. Sec. 4), any other differentiable kernel might be incorporated instead.

3.2. GP Sparsification

Commonly, the GP prediction in Eq. (3) is obtained by conditioning on all training data $\boldsymbol{X}, \boldsymbol{y}$. To alleviate the computational cost of $\mathcal{O}(N^3)$, which is undesirable for large datasets, several sparse approximations of the full GP posterior have been presented (Snelson & Ghahramani, 2006). By introducing P inducing GP targets $\boldsymbol{z} = [z_1, \dots, z_P]$ at pseudo input points $\boldsymbol{\zeta} = [\boldsymbol{\zeta}_1, \dots, \boldsymbol{\zeta}_P]$, which are jointly Gaussian with the latent function f, the true GP predictive distribution is approximated conditioned only on this set of inducing points.

$$p(f^* \mid \boldsymbol{x}^*, \boldsymbol{f}, \boldsymbol{X}) \approx p(f^* \mid \boldsymbol{x}^*, \boldsymbol{z}, \boldsymbol{\zeta}), \qquad (6)$$

$$p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z} \mid \boldsymbol{m}_\zeta, \boldsymbol{K}_{\zeta,\zeta}). \qquad (7)$$

The predicted function values consequently become mutually independent given the inducing points.

3.3. PR-SSM Model Definition

The PR-SSM is build upon a GP prior on the transition function $f(\cdot)$ and a parametric observation model $g(\cdot)$. This is a common model structure without loss of generality compared to the general system in Eq. (1), since any observation model can be absorbed into an sufficiently large latent state (Frigola-Alcade, 2015). Eliminating the non-parametric observation model however mitigates the problem of *'severe non-identifiability'* between transition model $f(\cdot)$ and observation model $g(\cdot)$ (Frigola et al., 2014). Independent GP priors are employed for each latent state dimension d given individual inducing points $\boldsymbol{\zeta}_d$ and $\boldsymbol{z}_d$, .

In the following derivations, the system's latent state, input and output at time t are denoted by $\boldsymbol{x}_t \in \mathbb{R}^{D_x}$, $\boldsymbol{u}_t \in \mathbb{R}^{D_u}$, and $\boldsymbol{y}_t \in \mathbb{R}^{D_y}$, respectively. The shorthand $\hat{\boldsymbol{x}}_t = (\boldsymbol{x}_t, \boldsymbol{u}_t)$ denotes the transition model's input at time t. The output of the transition model is denoted by $\boldsymbol{f}_{t+1} = f(\hat{\boldsymbol{x}}_t)$. A time series of observations from time a to time b (including) is abbreviated by $\boldsymbol{y}_{a:b}$ (analogously for the other model variables).

The joint distribution of all PR-SSM variables is given by

$$p(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) = \left[\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) \right] p(\boldsymbol{x}_1) p(\boldsymbol{z}) \quad (8)$$
$$\left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t) p(\boldsymbol{f}_t \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}) \right],$$

where $p(\boldsymbol{f}_t \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}) = \prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d)$ and $\boldsymbol{z} \equiv [\boldsymbol{z}_1, \dots \boldsymbol{z}_{D_x}]$. A graphical model of the resulting PR-SSM is shown in Fig. 1.

The individual contributions to (8) are given by the observation model and the transition model which are now described in detail. The observation model is governed by

$$p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) = \mathcal{N}(\boldsymbol{y}_t \mid g(\boldsymbol{x}_t), \text{diag}(\sigma_{y,1}^2, \dots, \sigma_{y,D_y}^2)), \quad (9)$$

with observation function

$$g(\boldsymbol{x}_t) = \boldsymbol{C} \boldsymbol{x}_t. \qquad (10)$$

In particular, the matrix $\boldsymbol{C}$ is chosen to select the D_y first entries of $\boldsymbol{x}_t$ by defining $\boldsymbol{C} := [\boldsymbol{I}, \boldsymbol{0}] \in \mathbb{R}^{D_y \times D_x}$ with $\boldsymbol{I}$ being the identity matrix. This model is particularly suitable for observation spaces which are low-dimensional compared to the latent state dimensionality, i.e. $D_y < D_x$, as it is often the case for physical systems with a restricted number of sensors. For high-dimensional observation spaces (e.g. images), a more involved observation model (e.g. a neural network) may be seamlessly incorporated into the presented framework for any differentiable mapping $g(\cdot)$. Process noise is modeled as

$$p(\boldsymbol{x}_t \mid \boldsymbol{f}_t) = \mathcal{N}(\boldsymbol{x}_t \mid \boldsymbol{f}_t, \text{diag}(\sigma_{x,1}^2, \dots, \sigma_{x,D_x}^2)). \quad (11)$$

The transition dynamics is described independently for each latent state dimension d by $p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) p(\boldsymbol{z}_d)$. This probability is given by the sparse GP prior (7) and predictive distribution (6), where $x^* = \hat{\boldsymbol{x}}_t$ and $f^* = f_{t,d}$. The initial system state distribution $p(\boldsymbol{x}_1)$ is unknown and has to be estimated. We will discuss this process in detail in Sec. 5.2.

The latent space GP dynamics model, as visualized in Fig. 1, could be made deep by introducing additional layers in between latent state and observations. An example for such a deep latent state can be found in (Mattos et al., 2015). In their method, however, the latent state is one-dimensional,
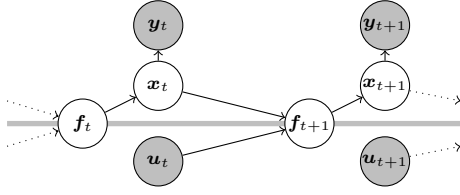
Figure 1. Graphical model of the PR-SSM. Gray nodes are observed variables in contrast to latent variables in white nodes. Thick lines indicate variables, which are jointly Gaussian under a GP prior.

therefore not Markovian and needs to capture additional information by employing history of previous states and inputs. Our method, in contrast, employs a multi-dimensional, Markovian latent state.

4. PR-SSM Inference

Computing the log likelihood or a posterior based on (8) is generally intractable due to the nonlinear GP dynamics model in the latent state. However, the log marginal likelihood $\log p(\boldsymbol{y}_{1:T})$ (evidence) can be bounded below by the Evidence Lower BOund (ELBO). This ELBO is derived via Jensen's inequality by introducing a computationally simpler, variational distribution $q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})$ of the true model's posterior distribution $p(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z} \mid \boldsymbol{y}_{1:T})$ (cf. eq. (8)). In contrast to previous work (Frigola et al., 2014; Mattos et al., 2015; Eleftheriadis et al., 2017), the proposed approximation explicitly incorporates the true temporal correlations in the latent state whilst being scalable to large datasets. The inference scheme is inspired by doubly stochastic variational inference for deep GPs as presented in (Salimbeni & Deisenroth, 2017).

4.1. Variational Sparse GP

PR-SSM employs a variational sparse GP (Titsias, 2009) based on a variational distribution $q(\boldsymbol{z})$ on the GP's inducing outputs as previously used in (Frigola et al., 2014; Eleftheriadis et al., 2017). For the standard regression problem, the inducing output distribution can be optimally eliminated and turns out to be a Gaussian distribution. Eliminating the inducing outputs, however, results in dependencies between inducing outputs and data which, in turn, leads to a complexity of $\mathcal{O}(NP^2)$, where N is the number of data points and P the number of inducing points. Unfortunately, this complexity is still prohibitive for large datasets. Therefore, we revert to an explicit representation of the variational distribution over inducing outputs as previously detailed in (Hensman et al., 2013). This explicit representation enables scalability by utilizing stochastic gradient-based optimization since individual GP predictions become independent given the explicit inducing points. Following a mean-field

variational approximation the inducing output distribution is given as $q(\boldsymbol{z}) = \prod_{d=1}^{D_x} \mathcal{N}(\boldsymbol{z}_d \mid \mu_d, \Sigma_d)$ for each latent state dimension d with diagonal variance Σ_d. Marginalizing out the inducing outputs, the GP predictive distribution is obtained as Gaussian with mean and variance given by

$$\mu_d = m_{\hat{\boldsymbol{x}}_t} + \boldsymbol{\alpha}(\hat{\boldsymbol{x}}_t)^T (\mu_d - m_{\boldsymbol{\zeta}_d}) , \qquad (12)$$

$$\sigma_d^2 = k_{\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_t} - \boldsymbol{\alpha}(\hat{\boldsymbol{x}}_t)^T (k_{\boldsymbol{\zeta}_d, \boldsymbol{\zeta}_d} - \Sigma_d) \boldsymbol{\alpha}(\hat{\boldsymbol{x}}_t) , \qquad (13)$$

with $\boldsymbol{\alpha}(\hat{\boldsymbol{x}}_t) := k_{\hat{\boldsymbol{x}}_t, \boldsymbol{\zeta}_d} K_{\boldsymbol{\zeta}_d, \boldsymbol{\zeta}_d}^{-1}$.

4.2. Variational Approximation

In previous work (Mattos et al., 2015), a factorized variational distribution is considered based on a mean-field approximation for the latent states $\boldsymbol{x}_{1:T}$. Their variational distribution is given by $q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) = \left[\prod_{t=1}^{T} q(\boldsymbol{x}_t) \right] \left[\prod_{d=1}^{D_x} q(\boldsymbol{z}_d) \left[\prod_{t=2}^{T} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) \right] \right]$. This choice, however, leads to several caveats: (i) The number of model parameters grows linearly with the length of the time series since each latent state is parametrized by its individual distribution $q(\boldsymbol{x}_t)$ for every time step. (ii) Initializing the latent state is non-trivial, since the observation mapping is unknown and generally not bijective. (iii) The model design does not represent correlations between time steps. Instead, these correlations are only introduced by enforcing pairwise couplings during the optimization process. The first two problems have been addressed in (Mattos et al., 2015; Eleftheriadis et al., 2017) by introducing a *recognition* model, e.g. a Bi-RNN[3], which acts as a smoother which can be learned through backpropagation and which allows to obtain the latent states given the input/output sequence.

The issue of representing correlations between time steps, however, is currently an open problem which we aim to address with our proposed model structure. Properly representing these correlations is a crucial step in making the optimization problem tractable in order to learn GP-SSMs for complex systems.

In contrast to previous work, the proposed variational distribution in our model does not factorize over the latent state but takes into account the true transition model, based on the sparse GP approximation from (8). In previous work, stronger approximations have been required to achieve an analytically tractable ELBO. This work, however, deals with the more complex distribution by combining sampling and gradient-based methods. The variational distribution is

---

[3]A bi-directional RNN operates on a sequence from left to right and vice versa to obtain predictions based on past and future inputs.

given by

$$q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) = \left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t)\right] q(\boldsymbol{x}_1) \cdot \qquad (14)$$
$$\left[\prod_{t=2}^{T}\prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d)q(\boldsymbol{z}_d)\right],$$

with

$$q(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_1 \mid \boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1}), \quad q(\boldsymbol{z}_d) = \mathcal{N}(\boldsymbol{z}_d \mid \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d).$$

In work by (Frigola et al., 2014), the variational distribution over inducing outputs has been optimally eliminated. This leads to a smoothing problem in a second system requiring computationally expensive, e.g. sample-based, smoothing methods. Instead, we approximate the distribution by a Gaussian, which is the optimal solution in case of sparse GP regression (cf. (Titsias, 2009)).

The PR-SSM model parameters include the variational parameters for the initial state and inducing points as well as deterministic parameters for noise models and GP kernel hyper-parameters: $\theta_{\text{PR-SSM}} = (\boldsymbol{\mu}_{x_1}, \boldsymbol{\Sigma}_{x_1}, \boldsymbol{\mu}_{1:D_x}, \boldsymbol{\Sigma}_{1:D_x}, \boldsymbol{\zeta}_{1:D_x}, \sigma_{\text{x},1:D_x}^2, \sigma_{\text{y},1:D_y}^2, \theta_{\text{GP},1:D_x})$. Note that in the PR-SSM, the number of parameters grows only with the number of latent dimensions, but not with the length of the time series.

4.3. Variational Evidence Lower Bound

The ELBO is given by

$$\log p(\boldsymbol{y}_{1:T}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \frac{p(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\right]$$
$$=: \mathcal{L}_{PR-SSM}. \qquad (15)$$

Maximizing the ELBO is equivalent to minimizing $\text{KL}(q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) \parallel p(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z} \mid \boldsymbol{y}_{1:T}))$, therefore this is a way to optimize the approximated model parameter distribution with respect to the intractable, true model parameter posterior.

Based on (8) and (14) and using standard variational calculus, the ELBO (15) can be transformed into

$$\mathcal{L}_{PR-SSM} = \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t)}[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)]$$
$$- \sum_{d=1}^{D_x} \text{KL}(q(\boldsymbol{z}_d) \parallel p(\boldsymbol{z}_d; \boldsymbol{\zeta}_d)). \qquad (16)$$

The first part is the expected log-likelihood of the observed system outputs $\boldsymbol{y}$ based on the observation model and the variational latent state distribution $q(\boldsymbol{x})$. This term captures the capability of the learned latent state model to explain the

observed system behavior. The second term is a regularizer on the inducing output distribution that penalizes deviations from the GP prior. Due to this term, PR-SSM automatically trades off data fit against model complexity. A detailed derivation of the ELBO can be found in the supplementary material.

4.4. Stochastic ELBO Evaluation

Training the proposed PR-SSM requires maximizing the ELBO in (16) with respect to the model parameters $\theta_{\text{PR-SSM}}$. Whilst the second term, as KL between two Gaussian distributions, can be easily computed, the first term requires evaluation of an expectation with respect to the latent state distribution $q(\boldsymbol{x})$. Since the true non-linear, latent dynamics is maintained in the variational approximation (14), analytic evaluation of $q(\boldsymbol{x})$ is still intractable. To make this process tractable, the Markovian structure of the unobserved states and the sparse GP approximation can be exploited to enable a differentiable, sampling-based estimation of the expectation term. Specifically, the marginal latent state distribution $q(\boldsymbol{x}_t)$ at time t is *conditionally* independent of past time steps, given the previous state distribution $q(\boldsymbol{x}_{t-1})$ and the explicit representation of GP inducing points. Samples $\tilde{\boldsymbol{x}}_t \sim q(\boldsymbol{x}_t)$ can therefore be obtained by recursively drawing from the sparse GP posterior in (13) for $t = 1, \ldots, T$. Drawing samples from a Gaussian distribution can be made differentiable with respect to its parameters using the 'reparametrisation trick' (Kingma & Welling, 2013) by first drawing samples $\epsilon \sim \mathcal{N}(0, 1)$ and then computing

$$\tilde{x}_{t+1,d} = \mu_d(\hat{\boldsymbol{x}}_t) + \epsilon\sqrt{\sigma_d^2(\hat{\boldsymbol{x}}_t, \hat{\boldsymbol{x}}_t) + \sigma_{x,d}^2}, \qquad (17)$$

with $\hat{\boldsymbol{x}}_t = (\tilde{\boldsymbol{x}}_t, \boldsymbol{u}_t)$ and $\tilde{\boldsymbol{x}}_1 \sim q(\boldsymbol{x}_1)$. Note that the gradient is propagated back trough time due to this reparamatrization and unrolling of the latent state. Using (17), an unbiased estimator of the first expectation in the ELBO in (16) is given by

$$\mathbb{E}_{q(\boldsymbol{x}_t)}[\log(\boldsymbol{y}_t \mid \boldsymbol{x}_t)] \approx \frac{1}{N}\sum_{i=1}^{N}\log p(\boldsymbol{y}_t \mid \tilde{\boldsymbol{x}}_t^{(i)}). \qquad (18)$$

Based on the stochastic ELBO evaluation, analytic gradients of (16) can be derived to facilitate stochastic gradient-descent-based model optimization.

4.5. Model Predictions

After model optimization based on the ELBO (16), model predictions for a new input sequence $\boldsymbol{u}_{1:T}$ and initial latent state $\boldsymbol{x}_1$ can be obtained based on the approximate, variational posterior distribution in (14). In contrast to (Mattos et al., 2015), no approximations such as moment matching are required for model predictions. Instead, the complex latent state distribution is approximated based on samples as in (17). The predicted observation distribution can then be

computed from the latent distribution according to the observation model in (9). Instead of a fixed, uninformative initial latent state, a learned recognition model can be utilized to find a more informative model initialization (cf. 5.2).

5. Extensions for Large Datasets

In the following, two extensions to PR-SSM are presented to allow for model learning on large real-world datasets.

5.1. Stochastic Gradient Optimization

Evaluating the ELBO in (16) can be prohibitive for long data sequences. Because of the independence assumption in Sec. 3.2, the transition dynamics is independent of the full input/output data sequence and an estimate of the ELBO gradient can be computed by evaluating the ELBO for mini-batches of sub-trajectories of length T_{sub}.

5.2. Recognition Model

In the variational distribution (14), the initial latent state distribution $q(\boldsymbol{x}_1)$ is a model parameter, which is optimized during training to fit the observed input/output data. In contrast, during prediction, the initial latent system state is unknown. Deep recurrent models are usually initialized with random or zero latent state. This is possible for stable systems, since transient effects caused by the initial state are decaying (Vidyasagar, 2002). Thus, after an initial phase, the predictions of the recurrent model are solely caused by reactions to the given system input. However, in case of slow or unstable dynamics, model learning is derailed if the initial state is chosen inappropriately. For model learning based on mini-batches of subtrajectories (cf. Sec. 5.1), the initial state transient and therefore the choice of initial state becomes even more relevant.

In recent work on SSMs (Mattos et al., 2015; Eleftheriadis et al., 2017), a recognition model is introduced to parametrize the smoothing distribution $p(\boldsymbol{x}_{1:T} \mid \boldsymbol{y}_{1:T}, \boldsymbol{u}_{1:T})$. In this work, a similar idea is employed but only to model the initial latent state

$$q(\boldsymbol{x}_1) = \mathcal{N}(\boldsymbol{x}_1 \mid \boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1) \approx q(\boldsymbol{x}_1 \mid \boldsymbol{y}_{1:L}, \boldsymbol{u}_{1:L}), \quad (19)$$
$$\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1 = h(\boldsymbol{y}_{1:L}, \boldsymbol{u}_{1:L}; \theta_{\text{recog}}). \quad (20)$$

The initial latent state distribution is approximated by a Gaussian, where mean and variance are modeled by a recognition model h. The recognition model acts as a smoother, operating on the first L elements of the system input/output data to infer the first latent state. Instead of directly optimizing $q(\boldsymbol{x}_1)$ during training, errors are propagated back into the recognition model h, which is parametrized by θ_{recog}. This learned recognition model can then be employed in the prediction phase to infer the initial latent state (cf. Sec.4.5).

6. Comparison to Related Models

Model learning and learning SSMs are embedded in a large body of literature from several communities, such as control theory, deep learning or probabilistic inference, as depicted in Sec. 2. In the following, we highlight connections to related methods in the GP-SSM literature and point out close relations to a deterministic counterpart in deep recurrent models.

6.1. Probabilistic State-Space Models

Approaches to probabilistic GP-SSMs mainly differ in their approximations to the model's joint distribution (e.g. when solving for the smoothing distribution or for the observations likelihood). One class of approaches aims to solve for the true distribution which requires sample-based methods, e.g. Particle Markov Chain Monte Carlo (PMCMC), as in (Frigola et al., 2013). Similarly, (Frigola et al., 2014) employs a variational approximation but optimally eliminates the latent state distribution which again requires sample-based smoothing. These methods are close to exact but computationally inefficient and intractable for higher latent state dimensions or larger datasets.

A second class of approaches is based on variational inference and mean field approximations (Mattos et al., 2015; Föll et al., 2017). These methods, however, operate on latent autoregressive models which can be initialized by the observed output time series, such that the learned latent representation acts as a smoothed version of the observations. In Markovian latent spaces, no such prior information is available and therefore initialization is non-trivial. Model optimization based on mean field approximations empirically leads to highly suboptimal local solutions.

Bridging the gap between both classes, recent methods strive to recover (temporal) latent state structure. In (Eleftheriadis et al., 2017), a linear, time-varying latent state structure is enforced as a tractable compromise between the true non-linear dependencies and no dependencies as in mean field variational inference. However, to facilitate learning, a complex recognition model over the linear time-varying dynamics is required. In contrast, the proposed PR-SSM can efficiently incorporate the true dynamics by combining sampling- and gradient-based learning.

6.2. Deep Recurrent Models

PR-SSM learning can be interpreted as a probabilistic version of the learning procedure in deep recurrent models. Both procedures share the explicit unrolling of transition and observation model over a finite horizon. Errors between the predicted and the observed system output are propagated back over time. Therefore, the transition dynamics is to be learned but the latent state (distribution) is implicitly given.
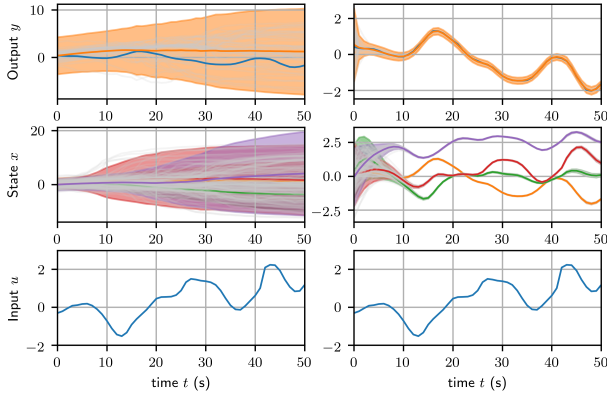
Figure 2. Predictions of the initial, untrained PR-SSM (left) and the final, trained model (right) based on the full gradient ELBO optimization. The system input/output data (blue) is visualized together with the model prediction (orange) for the first 50 time steps of the *furnace* dataset. Samples of the latent space distribution and output distribution are shown in gray. For latent state and observation distribution, the shaded areas visualize mean +/- σ. The initial model exhibits a random walk behavior in the latent space. In the trained model, the decay of the initial state uncertainty can be observed in the first time steps. In this experiment, no recognition model has been used (cf. Sec. 5).

This way, the challenging initialization and optimization of latent state variables is prevented. In contrast to deep recurrent models, the PR-SSM loss and model regularization is not a heuristic but automatically obtained from the model in (8). Furthermore, PR-SSMs obtain predictive distributions and the proposed initial state recognition model facilitates learning on shorter sub-trajectories and unstable systems, which is not possible in deep recurrent models.

7. Experimental Evaluation

In the following we evaluate the performance of the proposed method and present comparisons to existing work. We start by investigating the effects of the proposed recognition model and by giving some intuition on the algorithms inner working. Subsequently, we present a comparison to existing model learning algorithms from the fields of both auto-regressive as well as SSM based methods. Details about the PR-SSM default configuration as used in the following experiments can be found in the supplementary material.

7.1. PR-SSM Learning

For small datasets (i.e. short training trajectory lengths), the model can be trained based on the full gradient of the ELBO in (16). A comparison of the model predictions before and after training with the full ELBO gradient is visualized in Fig. 2.
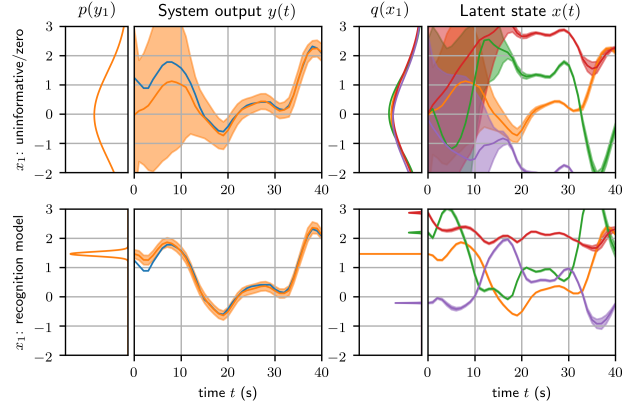
Figure 3. Comparison of the fully trained PR-SSM predictions with (lower row) and without (upper row) initial state recognition model. The initial transient based on the uncertainty from an uninformative initial state distribution $q(\boldsymbol{x}_0) = \mathcal{N}(\boldsymbol{x}_0 \mid \boldsymbol{0}, \boldsymbol{I})$ decays, as shown in upper plots. Below the outcome is shown when $q(\boldsymbol{x}_0)$ is initialized by the smoothing distribution $p(\boldsymbol{x}_0 \mid \boldsymbol{y}_{1:n}, \boldsymbol{u}_{1:n})$, given the first n steps of system input/output. Using the recognition model yields a significantly improved latent state initialization and therefore decreases the initial state uncertainty and the initial transient behavior.

Experimentally, three major shortcomings of the full gradient-based optimization schemes are observed: (i) Computing the full gradient for long trajectories is expensive and prone to the well-known problems of exploding and vanishing gradients (Pascanu et al., 2013). (ii) An uninformative initial state is prohibitive for unstable systems or systems with only slowly decaying initial state transients. (iii) Momentum-based optimizers (e.g. Adam) exhibit fragile optimization performance and are prone to overfitting on the training data.

The proposed method addresses these problems by employing the stochastic ELBO gradient based on minibatches of sub-trajectories (cf. Sec. 5.1) and the initial state recognition model (cf. Sec. 5.2). Fig. 3 visualizes the initial state distribution $q(\boldsymbol{x}_1)$ and the corresponding predictive output distribution $p(\boldsymbol{y}_1)$ for the fully trained model based on the full gradient (top row) as well as for the model based on the stochastic gradient and recognition model (bottom row). The transient dynamics and the associated model uncertainty is clearly visible for the first 15 time steps until the initial transient decays and approaches the true system behavior. In contrast, the learned recognition model almost perfectly initializes the latent state , leading to much smaller deviations in the predicted observations and far less predictive uncertainty. Notice how the recognition model is most certain about the distribution of the first latent state dimension (orange) which is directly coupled to the observation through the parametric observation model (cf. (9)). The uncertainty for the remaining, non-observed latent states, in contrast, is

Table 1. Comparison of model learning methods on five real-world benchmark examples. The RMSE result (mean (std) over 5 independently learned models) is given for the free simulation on the test dataset. For each dataset, the best result (solid underline) and second best result (dashed underline) is indicated. The proposed PR-SSM consistently outperforms the reference (SS-GP-SSM) in the class of Markovian state space models and robustly achieves performance comparable to the one of state-of-the-art latent, autoregressive models.

TASK	ONE-STEP-AHEAD, AUTOREGRESSIVE		MULTI-STEP-AHEAD, LATENT SPACE AUTOREGRESSIVE			MARKOVIAN STATE-SPACE MODELS	
	GP-NARX	NIGP	REVARB 1	REVARB 2	MSGP	SS-GP-SSM	**PR-SSM**
ACTUATOR	0.627 (0.005)	0.599 (0)	0.438 (0.049)	0.613 (0.190)	0.771 (0.098)	0.696 (0.034)	0.502 (0.031)
BALLBEAM	0.284 (0.222)	0.087 (0)	0.139 (0.007)	0.209 (0.012)	0.124 (0.034)	411.550 (273.043)	0.073 (0.007)
DRIVES	0.701 (0.015)	0.373 (0)	0.828 (0.025)	0.868 (0.113)	0.451 (0.021)	0.718 (0.009)	0.492 (0.038)
FURNACE	1.201 (0.000)	1.205 (0)	1.195 (0.002)	1.188 (0.001)	1.277 (0.127)	1.318 (0.027)	1.249 (0.029)
DRYER	0.310 (0.044)	0.268 (0)	0.851 (0.011)	0.355 (0.027)	0.146 (0.004)	0.152 (0.006)	0.140 (0.018)

slightly higher.

Comparing the full ELBO gradient-based model learning and the stochastic version with the recognition model, the stochastic model learning is far more robust and counteracts the overfitting tendencies in the full gradient-based model learning. A comparison of the model learning progress for both methods is depicted in the supplementary material. Due to the improved optimization robustness and the applicability to larger datasets, the stochastic, recognition-model-based optimization scheme is employed for the model learning benchmark presented in the next section.

7.2. Model Learning Benchmark

The performance of the proposed PR-SSM is assessed compared to state-of-the-art model learning methods on several real-world datasets as previously utilized by (Mattos et al., 2015). The main focus is thereby on long-term prediction quality given a sequence of system inputs. The suite of reference methods is composed of: One-step ahead autoregressive GP models: GP-FITC (Snelson & Ghahramani, 2006) and NIGP (McHutchon & Rasmussen, 2011); Multi-step-ahead autoregressive and recurrent GP models in latent space: REVARB based on 1 respectively 2 hidden layers (Mattos et al., 2015) and MSGP (Doerr et al., 2017a). GP-SSMs, based on a full Markovian state: SS-GP-SSM (Svensson & Schön, 2017) and the proposed PR-SSM. Currently, no published and runnable code exists for the model learning frameworks presented in (Turner et al., 2010; Frigola et al., 2013; 2014; Eleftheriadis et al., 2017). Reproducing these methods is not straightforward and outside the scope of this work. To enable a fair comparison, all methods is given access to a predefined amount of input/output data for initialization. Details about the benchmark methods, their configuration as well as the benchmark datasets can be found in the supplementary material.

7.3. Benchmark Results

Tab. 1 summarizes benchmark results in terms of RMSE between predicted mean and observed system output for all combinations of benchmark methods and datasets. A detailed visualization of the resulting model predictions on the *Drives* dataset is shown in Fig. 4. To obtain long-term predictive distributions from one-step-ahead models (GP-NARX, NIGP) two methods are used: Propagating only the mean of the current prediction (no uncertainty propagation); Approximating the true posterior by a Gaussian using exact moment matching (Girard et al., 2003).

In this benchmark, PR-SSM consistently outperforms the SS-GP-SSM learning method. Similarly, the performance is improved in comparison to the one-step-ahead optimized autoregressive models (GP-NARX and NIGP) on four out of five problems. In the ensemble of long-term optimize models based on autoregressive structure (REVARB, MSGP), no method is clearly superior. However, the performance of PR-SSM is in all cases close to the one of the best performing method out of this ensemble. Although PR-SSM is not outperforming the ensemble of state-of-the-art methods, it demonstrates robust model learning performance and does not exhibit severe failure on some datasets as some of the contestants do.

8. Conclusion

In this work, we presented PR-SSM, a novel model structure and inference scheme for learning probabilistic state-space models. PR-SSM can efficiently learn complex nonlinear models based on a Markovian state and GP priors on the latent transition dynamics. Therein, it differs from the branch of model learning techniques centered around history-based state representations (e.g. GP-NARX, RE-VARB). The proposed method uses a doubly stochastic inference technique, which is previously known from deep GPs, and which combines sampling-based elements with variational inference techniques. Based on this technique, a
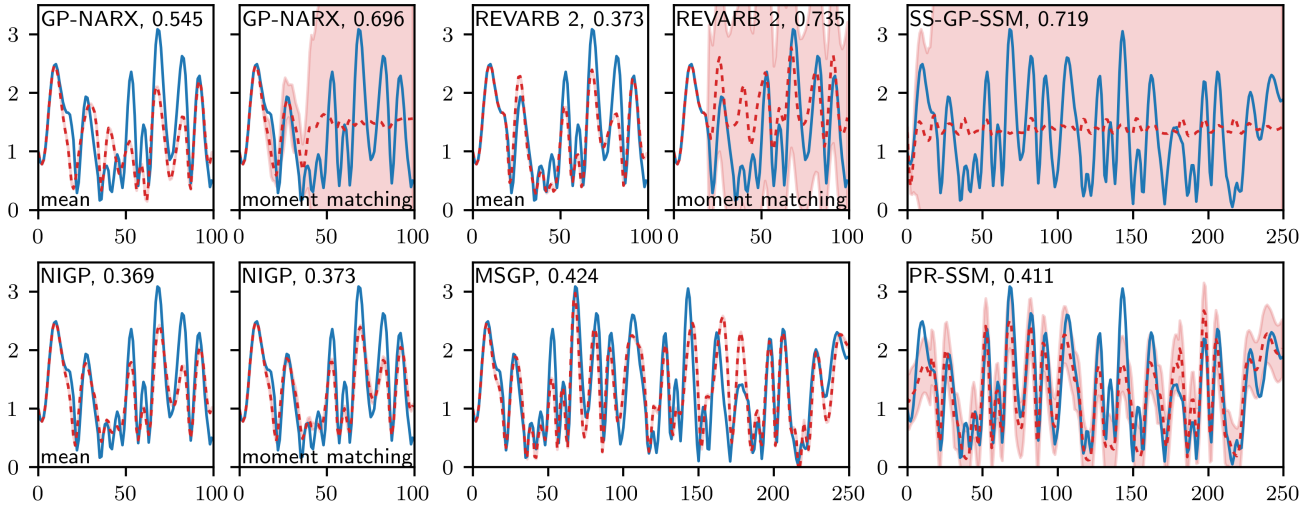
Figure 4. Free simulation results for the benchmark methods on the *Drives* test dataset. The true, observed system output (blue) is compared to the individual model's predictive output distribution (red, showing $+/-2\sigma$). Results are presented for the one-step-ahead models GP-NARX and NIGP in the left column. REVARB and MSGP (shown in the middle column) are both based on multi-step optimized autoregressive GP models in latent space. In the right column, the SS-GP-SSMs, as a model based on a Markovian latent state, is compared to the proposed PR-SSM.

novel model optimization criterion is derived for this probabilistic model, which is closely related to the optimization scheme in the very successful, but deterministic, RNNs or LSTMs. By maintaining the true latent state distribution and thereby enabling long-term gradients, efficient inference of truly unobserved parts of the latent spaces becomes feasible, which proved to be challenging for mean field variational approaches. Secondly, this work introduces a specific recognition model for the first latent state, thus enabling learning models for unstable or slow dynamics. The utilized explicit variational sparse GP in combination with the recognition model are furthermore key ingredients to the presented stochastic and mini-batched version of our algorithm which facilitates robust and scalable model learning. Robustness and high-performance in model learning is demonstrated on a set of real-world benchmark problems in comparison to state-of-the-art methods. The proposed method clearly improves over the baseline methods and a reference method from related GP-SSM work. At the same time, our method consistently performance on a par with an ensemble of state-of-the-art methods based on different model structures and inference techniques.

Acknowledgements

References

Billings, Stephen A. *Nonlinear system identification: NARMAX methods in the time, frequency, and spatio-temporal domains*. John Wiley & Sons, 2013.

Camacho, Eduardo F and Alba, Carlos Bordons. *Model predictive control*. Springer Science & Business Media, 2013.

Deisenroth, Marc P and Rasmussen, Carl Edward. PILCO: A model-based and data-efficient approach to policy search. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 465–472, 2011.

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

Doerr, Andreas, Daniel, Christian, Nguyen-Tuong, Duy, Marco, Alonso, Schaal, Stefan, Marc, Toussaint, and Trimpe, Sebastian. Optimizing long-term predictions for model-based policy search. In *Conference on Robot Learning (CORL)*, pp. 227–238, 2017a.

Doerr, Andreas, Nguyen-Tuong, Duy, Marco, Alonso, Schaal, Stefan, and Trimpe, Sebastian. Model-based policy search for automatic tuning of multivariate PID controllers. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2017b.

Eleftheriadis, Stefanos, Nicholson, Tom, Deisenroth, Marc, and Hensman, James. Identification of Gaussian process state space models. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 5315–5325, 2017.

Föll, Roman, Haasdonk, Bernard, Hanselmann, Markus, and Ulmer, Holger. Deep recurrent Gaussian process with variational sparse spectrum approximation. *arXiv preprint arXiv:1711.00799*, 2017.

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

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

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

Girard, Agathe, Rasmussen, Carl Edward, Quinonero-Candela, J, Murray-Smith, R, Winther, O, and Larsen, J. Multiple-step ahead prediction for non linear dynamic systemsa Gaussian process treatment with propagation of the uncertainty. In *Advances in Neural Information Processing Systems (NIPS)*, volume 15, pp. 529–536, 2003.

Hensman, James, Fusi, Nicolo, and Lawrence, Neil D. Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*, 2013.

Hochreiter, Sepp and Schmidhuber, Jürgen. LSTM can solve hard long time lag problems. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 473–479, 1997.

Kalman, R.E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1): 35–45, 1960.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Kocijan, Juš, Girard, Agathe, Banko, Blaž, and Murray-Smith, Roderick. Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

Likar, Bojan and Kocijan, Juš. Predictive control of a gas–liquid separation plant based on a Gaussian process

model. *Computers & chemical engineering*, 31(3):142–152, 2007.

Littman, Michael L and Sutton, Richard S. Predictive representations of state. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1555–1561, 2002.

Ljung, Lennart. System identification. In *Signal analysis and prediction*, pp. 163–173. Springer, 1998.

Ljung, Lennart. Perspectives on system identification. *Annual Reviews in Control*, 34(1):1–12, 2010.

Mattos, César Lincoln C, Dai, Zhenwen, Damianou, Andreas, Forth, Jeremy, Barreto, Guilherme A, and Lawrence, Neil D. Recurrent Gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

Mattos, César Lincoln C, Damianou, Andreas, Barreto, Guilherme A, and Lawrence, Neil D. Latent autoregressive Gaussian processes models for robust system identification. *IFAC-PapersOnLine*, 49(7):1121–1126, 2016.

McHutchon, Andrew and Rasmussen, Carl Edwards. Gaussian process training with input noise. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 1341–1349, 2011.

Moor, De. DaISy: Database for the identification of systems. http://homes.esat.kuleuven.be/~smc/daisy/, 2017. [Online; accessed 30-Jan-2017].

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

Narendra, Kumpati S and Parthasarathy, Kannan. Identification and control of dynamical systems using neural networks. *IEEE Transactions on neural networks*, 1(1): 4–27, 1990.

Nrgaard, Magnus. Hydraulic actuator dataset. http://www.iau.dtu.dk/nnbook/systems.html, 2000. [Online; accessed 30-Jan-2017].

Pascanu, Razvan, Mikolov, Tomas, and Bengio, Yoshua. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pp. 1310–1318, 2013.

Rudary, Matthew R and Singh, Satinder P. A nonlinear predictive state representation. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 855–862, 2004.

Salimbeni, Hugh and Deisenroth, Marc. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems (NIPS)*, pp. 4591–4602, 2017.

Singh, Satinder P, Littman, Michael L, Jong, Nicholas K, Pardoe, David, and Stone, Peter. Learning predictive state representations. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 712–719, 2003.

Snelson, Edward and Ghahramani, Zoubin. Sparse Gaussian processes using pseudo-inputs. In *Advances in Neural Information Processing Systems (NIPS)*, volume 18, pp. 1257, 2006.

Sutskever, Ilya, Martens, James, and Hinton, Geoffrey E. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pp. 1017–1024, 2011.

Svensson, Andreas and Schön, Thomas B. A flexible state–space model for learning nonlinear dynamical systems. *Automatica*, 80:189–199, 2017.

Titsias, Michalis K. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 5, pp. 567–574, 2009.

Turner, Ryan, Deisenroth, Marc, and Rasmussen, Carl. State-space inference and learning with Gaussian processes. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 868–875, 2010.

Van Overschee, Peter and De Moor, BL. *Subspace identification for linear systems: TheoryImplementationApplications*. Springer Science & Business Media, 2012.

Venugopalan, Subhashini, Xu, Huijuan, Donahue, Jeff, Rohrbach, Marcus, Mooney, Raymond, and Saenko, Kate. Translating videos to natural language using deep recurrent neural networks. *arXiv preprint arXiv:1412.4729*, 2014.

Vidyasagar, Mathukumalli. *Nonlinear systems analysis*. SIAM, 2002.

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

Wigren, Torbjörn. Input-output data sets for development and benchmarking in nonlinear identification. Technical report, Department of Information Technology, Uppsala University, 2010.

Williams, Christopher KI and Rasmussen, Carl Edward. *Gaussian processes for machine learning*. MIT Press, 2005.

A. Supplementary Material

This supplementary material provides details about the derivations and configuration of the proposed PR-SSM in Sec. A.1. Sec. A.2 elaborates on the reference methods and the employed datasets in the model learning benchmark. Finally, additional results from the PR-SSM learning and the model learning benchmark are presented in Sec. A.3.1.

A.1. Probabilistic Recurrent State-Space Model

A.1.1. EVIDENCE LOWER BOUND (ELBO)

Summarizing the model assumptions from the main paper, the model's joint distribution is given by

$$
p(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) = \left[\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t) \right]
$$
$$
\left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t) \right]
$$
$$
\left[\prod_{t=2}^{T} \prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) p(\boldsymbol{z}_d) \right]
$$
$$
p(\boldsymbol{x}_1) . \tag{21}
$$

The variational distribution over the unknown model variables is defined as

$$
q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z}) = \left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t) \right]
$$
$$
\left[\prod_{t=2}^{T} \prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) q(\boldsymbol{z}_d) \right]
$$
$$
q(\boldsymbol{x}_1) . \tag{22}
$$

Together, the derivation of the evidence lower bound is given below in Eq. (23) to (28).

In the ELBO, as derived in (28), the last term is a regularization on the initial state distribution. For the full gradient-based optimization in the main paper, a uninformative initial distribution is chosen and fixed, such that the third term is dropped. In the stochastic optimization scheme, this term acts as a regularization preventing the recognition model to become overconfident in its predictions.

A.1.2. MODEL CONFIGURATION

The PR-SSM exhibits a large number of model (hyper-) parameters $\theta_{\text{PR-SSM}}$ which need to be initialized. However, empirically, most of these model parameters can be initialized to a default setting as given in Tab. 2. This default configuration has been employed for all experiments presented in the main paper.

Table 2. Default configuration for the initialization of the PR-SSM (hyper-) parameters $\theta_{\text{PR-SSM}}$. This configuration has been employed for all experiments in the main paper.

PARAMETER	INITIALIZATION
INDUCING INPUTS	$\boldsymbol{\zeta}_d \sim \mathcal{U}(-2, 2) \in \mathbb{R}^{P \times (D_x + D_u)}$
INDUCING OUTPUTS	$q(\boldsymbol{z}_d) = \mathcal{N}(\boldsymbol{z}_d \mid \boldsymbol{\mu}_d, \boldsymbol{\Sigma}_d) \in \mathbb{R}^P$ $\mu_{d,i} \sim \mathcal{N}(\mu_{d,i} \mid 0, 0.05^2)$ $\boldsymbol{\Sigma}_d = 0.01^2 \cdot \boldsymbol{I}$
PROCESS NOISE SENSOR NOISE	$\sigma_{\text{X},i}^2 = 0.002^2 \quad \forall i \in [1, D_\text{X}]$ $\sigma_{\text{Y},i}^2 = 1.0^2 \quad \forall i \in [1, D_\text{Y}]$
KERNEL HYPER-PARAMETERS	$\sigma_f^2 = 0.5^2$ $l_i^2 = 2 \quad \forall i \in [1, D_x]$

Table 3. Structural configuration of the PR-SSM as utilized in the experimental section.

PARAMETER	INITIALIZATION
INDUCING POINTS STATE SAMPLES	$P = 20$ $N = 50$
SUBTRAJECTORIES	$N_{\text{BATCH}} = 10$ $T_{sub} = 100$
LATENT SPACE	$D_x = 4$

The PR-SSM's latent state dynamics model and noise models are configured to initially exhibit a random walk behavior. This behavior is clearly visible for the prediction based on the untrained model in Fig. 2 of the main paper. The GP prior is approximating the identity function based on an identity mean function and almost zero inducing outputs (up to a small Gaussian noise term to avoid singularities). The inducing inputs are spread uniformly over the function's domain. The noise processes are initializes such as to achieve high correlations between latent states over time (i.e. small process noise magnitude). At the same time, a larger observation noise is required to obtain a inflation of predictive uncertainty over time. This inflation of predictive uncertainty is again clearly visible in Fig. 2 of the main paper. Both noise terms are chosen in a way to obtain numerically stable gradients for both the sample based log likelihood and the backpropagation through time in the ELBO evaluation.

The number of samples used in the ELBO approximation, number of inducing points in the GP approximation and batch size are in contrast a trade-off between model accuracy and computational speed. The proposed default configuration empirically showed good performance whilst being computationally tractable.

Two tuning parameters remain which are problem specific and have to be chosen for each dataset individually. Depending on the true system's timescales/sampling frequency and

$$\log p(\boldsymbol{y}_{1:T}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \frac{p(\boldsymbol{y}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\right] \tag{23}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \frac{\left[\prod_{t=1}^T p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right]\left[\prod_{t=2}^T p(\boldsymbol{x}_t \mid \boldsymbol{f}_t)\right]\left[\prod_{t=2}^T \prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d)p(\boldsymbol{z}_d)\right] p(\boldsymbol{x}_1)}{\left[\prod_{t=2}^T p(\boldsymbol{x}_t \mid \boldsymbol{f}_t)\right]\left[\prod_{t=2}^T \prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d)q(\boldsymbol{z}_d)\right] q(\boldsymbol{x}_1)}\right] \tag{24}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \frac{\left[\prod_{t=1}^T p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right]\left[\prod_{d=1}^{D_x} p(\boldsymbol{z}_d)\right] p(\boldsymbol{x}_1)}{\left[\prod_{d=1}^{D_x} q(\boldsymbol{z}_d)\right] q(\boldsymbol{x}_1)}\right] \tag{25}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \prod_{t=1}^T p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\sum_{d=1}^{D_x} \log \frac{p(\boldsymbol{z}_d)}{q(\boldsymbol{z}_d)}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T}, \boldsymbol{f}_{2:T}, \boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x}_1)}{q(\boldsymbol{x}_1)}\right] \tag{26}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T})}\left[\log \prod_{t=1}^T p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\sum_{d=1}^{D_x} \log \frac{p(\boldsymbol{z}_d)}{q(\boldsymbol{z}_d)}\right] + \mathbb{E}_{q(\boldsymbol{x}_1)}\left[\log \frac{p(\boldsymbol{x}_1)}{q(\boldsymbol{x}_1)}\right] \tag{27}$$

$$= \sum_{t=1}^T \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \sum_{d=1}^{D_x} \mathrm{KL}(q(\boldsymbol{z}_d) \parallel p(\boldsymbol{z}_d)) + \mathrm{KL}(q(\boldsymbol{x}_1) \parallel p(\boldsymbol{x}_1)) \tag{28}$$

system order, the length of subtrajectories T_{sub} for mini-batching and the latent state dimensionality D_x have to be specified manually. For the benchmark datasets we choose $T_{sub} = 100$ and $D_x = 4$.

A.2. Model Learning Benchmark Details

In the main paper, the proposed PR-SSM's long-term predictive performance is compared to several state-of-the-art methods. The benchmark is set up similar to the evaluation presented in (Doerr et al., 2017a). Details about the individual benchmark methods, their configuration and the employed datasets can be found in the following sections. Minor adjustments with respect to the set up in (Doerr et al., 2017a) will be pointed out in the following. These have been introduced to enable fair comparison between all benchmark methods.

A.2.1. BENCHMARK METHODS

The proposed PR-SSM is evaluated in comparison to methods from three classes: one-step ahead autoregressive models (GP-NARX, NIGP), multi-step ahead autoregressive models in latent space (REVARB, MSGP) and Markov state-space models (SS-GP-SSM). To enable a fair comparison, all methods have access to a predefined amount of input/output data for initialization.

(i) GP-NARX (Kocijan et al., 2005): The system dynamics is modeled as $y_{t+1} = f(y_t, \dots, y_{t-L_y}, u_t, \dots, u_{t-L_u})$ with a GP prior on f. The GP has a zero mean function and a squared exponential kernel with automatic relevance deter-

Table 4. Summary of the real-world, non-linear system identification benchmark tasks. All datasets are generated by recording input/output data of actual physical plants. For each dataset, the lengths of training and test set are given together with the number of past input and outputs used for the NARX dynamics models.

	N_{train}	N_{test}	L_u, L_y
ACTUATOR (NRGAARD, 2000)	512	512	10
BALLBEAM (MOOR, 2017)	500	500	10
DRIVES (WIGREN, 2010)	250	250	10
FURNACE (MOOR, 2017)	148	148	3
DRYER (MOOR, 2017)	500	500	2

mination. The kernel hyper-parameters, signal variance and lengthscales, are optimized based on the standard maximum likelihood objective. A sparse approximation (Snelson & Ghahramani, 2006), based on 100 inducing inputs is employed. Moment matching (Girard et al., 2003) is employed to obtain a long-term predictive distribution.

(ii) NIGP (McHutchon & Rasmussen, 2011): Noise Input GPs (NIGP) account for uncertainty in the input by treating input points as deterministic and inflating the corresponding output uncertainty, leading to state dependent noise, i.e. heteroscedastic GPs. The experimental results are based on the publicly available Matlab code. Since no sparse version is available, training is performed on the full training dataset. Training on the full dataset is however not possible for larger datasets and provides an advantage to NIGP. Experiments

based on a random data subset of size 100 lead to decreased performance in the order of the GP-NARX results or worse.

(iii) REVARB (Mattos et al., 2015): Recurrent Variational Bayes (REVARB) is a recent proposition to optimize the lower bound to the log-marginal likelihood $\log p(\boldsymbol{y})$ using variational techniques. This framework is based on the variational sparse GP framework (Titsias, 2009), but allows for computation of (time-)recurrent GP structures and deep GP structures (stacking multiple GP-layers in each time-step). For our benchmark, we run REVARB using one (REVARB1) respectively two (REVARB2) hidden layers, where each layer is provided with 100 inducing inputs. We closely follow the original setup as presented by (Mattos et al., 2015), performing 50 initial optimization steps based on fixed variances and up to 10000 steps based on variable variances. Unlike for the other benchmark methods, the autoregressive history of REVARB implicitly becomes longer when introducing additional hidden layers.

(iv) MSGP (Doerr et al., 2017a): MSGP is a GP-NARX model operating in a latent, noise free state, which is trained by optimizing its long-term predictions. The experimental results are obtained according to the configuration described in (Doerr et al., 2017a), again using 100 inducing points and moment matching.

(v) SS-GP-SSM (Svensson & Schön, 2017): The Sparse-Spectrum GP-SSM is employing a sparse spectrum GP approximation to model the system's transition dynamics in a Markovian, latent space. The available Matlab implementation is restricted to a 2D latent space. In the experimental results, a default configuration is employed as given by: $K = 2000, N = 40, n\_basis\_u = n\_basis\_x = 7$. The variables are defined as given in the code published for (Svensson & Schön, 2017).

A.2.2. Benchmark Datasets

The benchmarks datasets are composed of popular system identification datasets from related work (Narendra & Parthasarathy, 1990; Kocijan et al., 2005; Mattos et al., 2016). They incorporate measured input output data from technical systems like hydraulic actuators, furnaces, hair dryers or electrical motors. For all of these problems, both inputs and outputs are one-dimensional $D_u = D_y = 1$. However, the system's true state is higher dimensional such that an autoregressive history or an explicit latent state representation is required to capture the relevant dynamics. The number of historic inputs and outputs for the autoregressive methods is fixed a-priori for each dataset as previously used in other publications. For model training, datasets are normalized to zero mean and variance one based on the available training data. References to the individual datasets, training and test trajectory length and the utilized history for the GP-NARX models are summarized in Tab. 4.
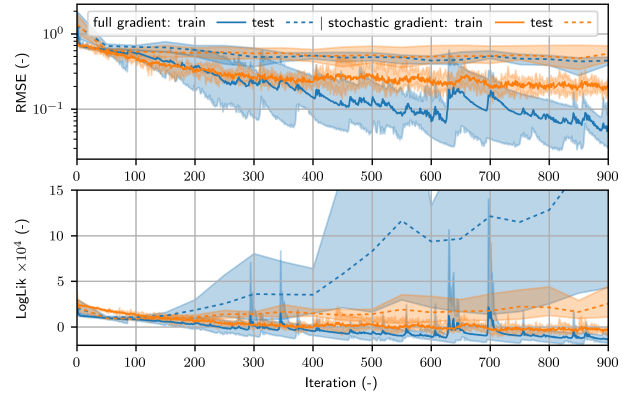

Figure 5. Comparison of the learning progress of the proposed method on the *Drive* dataset given the full ELBO gradient (blue) and the stochastic gradient, based on minibatches and the recognition model (orange). RMSE and log likelihood results over learning iterations are shown for the free simulation on training and test dataset. The full gradient optimization scheme overfits (in particular visible in the log likelihood) and exposes a difficult optimization objective (cf. spikes in model loss). Stochastically optimizing the model-based on the proposed minibatched ELBO estimates and employing the recognition model significantly reduces overfitting and leads to more robust learning.

A.3. Additional Results

A.3.1. Optimization Schemes Comparison

In Fig. 5, the RMSE and the negative log likelihood, which is obtained for the model's long-term prediction, is depicted over learning iterations for the training- (solid line) and test- (dotted line) set from the *Drives* dataset. The full gradient optimization (blue) obtains smaller training loss in comparison to the stochastic optimization scheme for both RMSE and negative log likelihood. The resulting test performance however indicates similar performance in terms of RMSE whilst showing clear overfitting of the full-gradient-based model in terms of log likelihood. Additionally, optimizing, based on the full gradient, is much more delicate and less robust as indicated by the spikes in loss and the higher variance of incurred optimization progress. Fig. 5 depicts mean (lines) and minimum to maximum intervals (shaded areas) of incurred loss, based on five independent model trainings.

A.3.2. Detailed Benchmark Results

In the following, detailed results are provided for the benchmark experiments. The reference learning methods in the presented benchmark are highly deceptive to changes in the model configuration and the data pre-processing. Therefore, results are detailed for GP-NARX, NIGP, REVARB 1, and REVARB 2 for all combinations of normalized/unnormalized training data and mean or moment matching predictions. The results for methods MSGP, SS-GP-

Table 5. Comparison of model learning methods on five real-world benchmark examples. The RMSE result (mean (std) over 5 independently learned models) is given for the free simulation on the test dataset. For each dataset, the best result (solid underline) and second best result (dashed underline) is indicated. The proposed PR-SSM consistently outperforms the reference (SS-GP-SSM) in the class of Markovian state space models and robustly achieves performance comparable to the one of state-of-the-art latent, autoregressive models.

	ONE-STEP-AHEAD AUTOREGRESSIVE		MULTI-STEP-AHEAD AUTOREGRESSIVE IN LATENT SPACE			MARKOVIAN STATE-SPACE MODELS	
	DATA UNNORMALIZED + MEAN PREDICTION			DEFAULT CONFIGURATION			
TASK	GP-NARX	NIGP	REVARB 1	REVARB 2	MSGP	SS-GP-SSM	PR-SSM
ACTUATOR	0.645 (0.018)	0.752 (0)	0.496 (0.057)	0.565 (0.047)	0.771 (0.098)	0.696 (0.034)	0.502 (0.031)
BALLBEAM	0.169 (0.005)	0.165 (0)	0.138 (0.001)	0.073 (0.000)	0.124 (0.034)	411.550 (273.043)	0.073 (0.007)
DRIVES	0.579 (0.004)	0.378 (0)	0.718 (0.081)	0.282 (0.031)	0.451 (0.021)	0.718 (0.009)	0.492 (0.038)
FURNACE	1.199 (0.001)	1.195 (0)	1.210 (0.034)	1.945 (0.016)	1.277 (0.127)	1.318 (0.027)	1.249 (0.029)
DRYER	0.278 (0.003)	0.281 (0)	0.149 (0.017)	0.128 (0.001)	0.146 (0.004)	0.152 (0.006)	0.140 (0.018)
	DATA UNNORMALIZED + MOMENT MATCHING			DEFAULT CONFIGURATION			
TASK	GP-NARX	NIGP	REVARB 1	REVARB 2	MSGP	SS-GP-SSM	PR-SSM
ACTUATOR	0.633 (0.018)	0.601 (0)	0.430 (0.026)	0.618 (0.047)	0.771 (0.098)	0.696 (0.034)	0.502 (0.031)
BALLBEAM	0.077 (0.000)	0.078 (0)	0.131 (0.005)	0.073 (0.000)	0.124 (0.034)	411.550 (273.043)	0.073 (0.007)
DRIVES	0.688 (0.003)	0.398 (0)	0.801 (0.032)	0.733 (0.087)	0.451 (0.021)	0.718 (0.009)	0.492 (0.038)
FURNACE	1.198 (0.002)	1.195 (0)	1.192 (0.002)	1.947 (0.032)	1.277 (0.127)	1.318 (0.027)	1.249 (0.029)
DRYER	0.284 (0.003)	0.280 (0)	0.878 (0.016)	0.123 (0.002)	0.146 (0.004)	0.152 (0.006)	0.140 (0.018)
	DATA NORMALIZATION + MEAN PREDICTION			DEFAULT CONFIGURATION			
TASK	GP-NARX	NIGP	REVARB 1	REVARB 2	MSGP	SS-GP-SSM	PR-SSM
ACTUATOR	0.665 (0.014)	0.791 (0)	0.506 (0.092)	0.559 (0.069)	0.771 (0.098)	0.696 (0.034)	0.502 (0.031)
BALLBEAM	0.357 (0.199)	0.154 (0)	0.141 (0.004)	0.206 (0.008)	0.124 (0.034)	411.550 (273.043)	0.073 (0.007)
DRIVES	0.564 (0.029)	0.369 (0)	0.605 (0.027)	0.376 (0.026)	0.451 (0.021)	0.718 (0.009)	0.492 (0.038)
FURNACE	1.201 (0.001)	1.205 (0)	1.196 (0.002)	1.189 (0.001)	1.277 (0.127)	1.318 (0.027)	1.249 (0.029)
DRYER	0.282 (0.001)	0.269 (0)	0.123 (0.001)	0.113 (0)	0.146 (0.004)	0.152 (0.006)	0.140 (0.018)
	DATA NORMALIZATION + MOMENT MATCHING			DEFAULT CONFIGURATION			
TASK	GP-NARX	NIGP	REVARB 1	REVARB 2	MSGP	SS-GP-SSM	PR-SSM
ACTUATOR	0.627 (0.005)	0.599 (0)	0.438 (0.049)	0.613 (0.190)	0.771 (0.098)	0.696 (0.034)	0.502 (0.031)
BALLBEAM	0.284 (0.222)	0.087 (0)	0.139 (0.007)	0.209 (0.012)	0.124 (0.034)	411.550 (273.043)	0.073 (0.007)
DRIVES	0.701 (0.015)	0.373 (0)	0.828 (0.025)	0.868 (0.113)	0.451 (0.021)	0.718 (0.009)	0.492 (0.038)
FURNACE	1.201 (0.000)	1.205 (0)	1.195 (0.002)	1.188 (0.001)	1.277 (0.127)	1.318 (0.027)	1.249 (0.029)
DRYER	0.310 (0.044)	0.268 (0)	0.851 (0.011)	0.355 (0.027)	0.146 (0.004)	0.152 (0.006)	0.140 (0.018)

SSM and PR-SSM are always computed for the normalized datasets using the method specific propagation of uncertainty schemes.

Obtaining uncertainty estimates is one key requirement for employing the long-term predictions, e.g. in model-based control. Therefore, only the predictive results based on the approximate propagation of uncertainty through moment matching is considered in the main paper, although better results in RMSE are sometimes obtained from employing only the mean predictions. A comparison of the predictive results based on mean and moment matching predictions on the *Drives* dataset is shown in Fig. 4. The results from the unnormalized datasets and moment matching are in line with the results published in (Doerr et al., 2017a).