Supplementary Material for 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]

This supplementary material provides details about the derivations and configuration of the proposed PR-SSM in Sec. 1. Sec. 2 elaborates on the reference methods and the employed datasets in the model learning benchmark. Finally, additional experimental results from PR-SSM learning, the model learning benchmark, and the large scale experiment are summarized in Sec. 3.

1. Probabilistic Recurrent State-Space Model

1.1. Evidence Lower Bound (ELBO)

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

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

The variational distribution over the unknown model variables is defined as

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

Table 1. 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 benchmark section.

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	$\sigma^2_{\mathrm{X},i} = 0.002^2 \quad \forall i \in [1, D_{\mathrm{X}}]$ $\sigma^2_{\mathrm{Y},i} = 1.0^2 \quad \forall i \in [1, D_{\mathrm{Y}}]$
KERNEL HYPER-PARAMETERS	$\sigma_f^2 = 0.5^2$ $l_i^2 = 2 \quad \forall i \in [1, D_x]$

Together, the derivation of the ELBO is given below in (3) to (8).

In the ELBO, as derived in (8), the last term is a regularization on the initial state distribution. For the full gradient-based optimization in the main paper, an 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.

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. 1. This default configuration has been employed for all benchmark experiments presented in the main paper.

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 do-

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

Proceedings of the 35^{th} *International Conference on Machine Learning*, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

$$\log p(\boldsymbol{y}_{1:T}) \geq \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{y}_{1:T},\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\right] \tag{3}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\log \frac{\left[\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right]\left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t)\right]\left[\prod_{t=2}^{T}\prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) p(\boldsymbol{z}_d)\right] p(\boldsymbol{x}_1)}{\left[\prod_{t=2}^{T} p(\boldsymbol{x}_t \mid \boldsymbol{f}_t)\right]\left[\prod_{t=2}^{T}\prod_{d=1}^{D_x} p(f_{t,d} \mid \hat{\boldsymbol{x}}_{t-1}, \boldsymbol{z}_d) q(\boldsymbol{z}_d)\right] q(\boldsymbol{x}_1)}\right] \tag{4}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\log \frac{\left[\prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right]\left[\prod_{d=1}^{D_x} p(\boldsymbol{z}_d)\right] p(\boldsymbol{x}_1)}{\left[\prod_{d=1}^{D_x} q(\boldsymbol{z}_d)\right] q(\boldsymbol{x}_1)}\right] \tag{5}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\log \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\sum_{d=1}^{D_x} \log \frac{p(\boldsymbol{z}_d)}{q(\boldsymbol{z}_d)}\right] + \mathbb{E}_{q(\boldsymbol{x}_{1:T},\boldsymbol{f}_{2:T},\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x}_1)}{q(\boldsymbol{x}_1)}\right] \tag{6}$$

$$= \mathbb{E}_{q(\boldsymbol{x}_{1:T})}\left[\log \prod_{t=1}^{T} p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \mathbb{E}_{q(\boldsymbol{z})}\left[\sum_{d=1}^{D_x} \log \frac{p(\boldsymbol{z}_d)}{q(\boldsymbol{z}_d)}\right] + \mathbb{E}_{q(\boldsymbol{x}_1)}\left[\log \frac{p(\boldsymbol{x}_1)}{q(\boldsymbol{x}_1)}\right] \tag{7}$$

$$= \sum_{t=1}^{T} \mathbb{E}_{q(\boldsymbol{x}_t)}\left[\log p(\boldsymbol{y}_t \mid \boldsymbol{x}_t)\right] + \sum_{d=1}^{D_x} \mathrm{KL}(q(\boldsymbol{z}_d) \parallel p(\boldsymbol{z}_d)) + \mathrm{KL}(q(\boldsymbol{x}_1) \parallel p(\boldsymbol{x}_1)) \tag{8}$$

Table 2. Structural configuration of the PR-SSM as utilized in the benchmark experiments.

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

Table 3. 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

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

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., 2017). 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., 2017) will be pointed out in the following. These have been introduced to enable fair comparison between all benchmark methods.

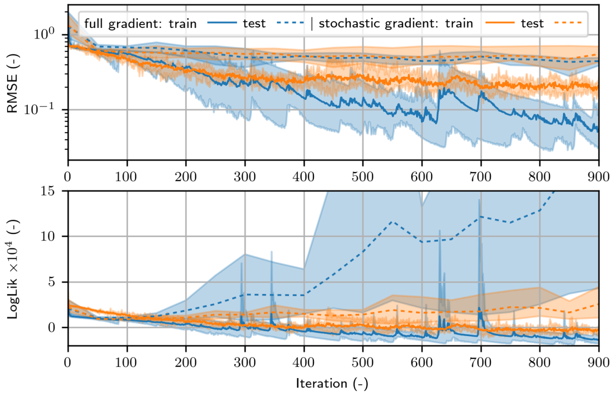

Figure 1. Comparison of the learning progress of the proposed method on the *Drive* dataset given the full ELBO gradient (blue) and the stochastic gradient, based on minibatches and the recognition model (orange). RMSE and log likelihood results over learning iterations are shown for the free simulation on training and test dataset. The full gradient optimization scheme overfitts (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.

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 determination. 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 timestep). 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., 2017): 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., 2017), 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).

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. 3.

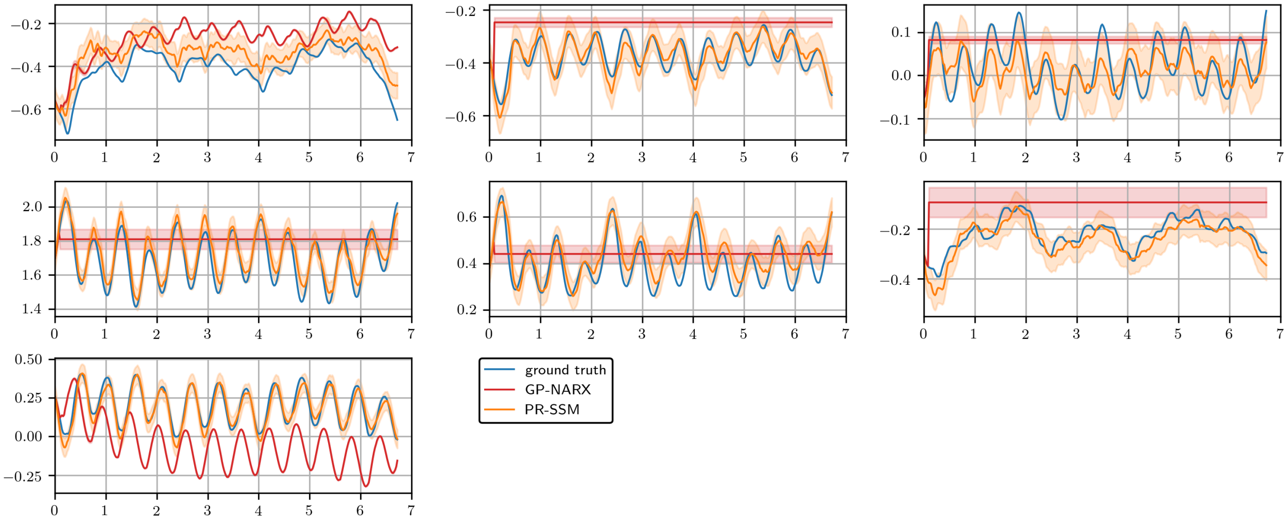

Figure 2. Detailed results from the Sarcos large scale task: Predictions from the GP-NARX model (red) and the PR-SSM (green) for all seven joint positions as obtained for the first test experiment. The ground truth, measured joint positions are shown in blue. PR-SSM is clearly able to capture the robot arm dynamics, whereas the GP-NARX model only succesfully captures a rough model of the robot arm dynamics for two out of seven joints.

3. Additional Results

3.1. Optimization Schemes Comparison

In Fig. 1, 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. 1 depicts mean (lines) and minimum to maximum intervals (shaded areas) of incurred loss, based on five independent model trainings.

3.2. Detailed Benchmark Results

In Tab. 4, detailed results are provided for the benchmark experiments. The reference learning methods in the presented benchmark are highly deceptive to changes in the data pre-processing and the long-term prediction method. 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-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. 5. The results from the unnormalized datasets and moment matching are in line with the results published in (Doerr et al., 2017).

3.3. Large Scale Experiment Details

The *Sarcos* task is based on a publicly available dataset comprising joint positions, velocities, acceleration and torques of a seven degrees-of-freedom SARCOS anthropomorphic robot arm. This dataset has been previously used in (Vijayakumar & Schaal, 2000; Williams & Rasmussen, 2005) in the task of learning the system's inverse dynamics, therefore mapping joint position, velocities, and accelerations to the required joint torques. This task can be framed as a standard regression problem, which is solved in a supervised fashion. In contrast, in this paper, we consider the task of learning the forward dynamics, i.e. predicting the joint positions given a sequence of joint torques. The system output is therefore given by the seven joint positions ($D_y = 7$). Joint velocities and acceleration, as latent states, are not available for learning but have to be inferred. The system input is given by the seven joint torques ($D_u = 7$).

The original training dataset (44.484 datapoints) recorded at 100 Hz has been downsampled to 50 Hz. It is split into 66

Table 4. 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)

independent experiments as indicated by the discontinuities in the original time-series data. Six out of 66 experiments have been utilized for testing whereas the other 60 experiments remain for training. None of the reference methods from the model learning benchmark is out-of-the-box applicable to this large scale dataset. To obtain a baseline, the sparse GP-NARX model has been trained on a subset of training experiments (400 inducing points, approx. 2000 training data points). The PR-SSM can be directly trained on the full training dataset utilizing its stochastic, mini-batched optimization scheme. PR-SSM is setup similar to the configuration described in the benchmark experiment but based on a 14 dimensional latent state ($D_x = 14$). Long-term prediction results on one of the test experiments are visualized in Fig. 2. PR-SSM robustly predicts the robot arm motions for all joints and clearly improves over the GP-NARX baseline. In contrast, the GP-NARX baseline can not predict the dynamics on 5 out of 7 joints.

References

Doerr, A., Daniel, C., Nguyen-Tuong, D., Marco, A., Schaal, S., Toussaint, M., and Trimpe, S. Optimizing long-term predictions for model-based policy search. In *Conference on Robot Learning (CORL)*, pp. 227–238, 2017.

Girard, A., Rasmussen, C. E., 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.

Kocijan, J., Girard, A., Banko, B., and Murray-Smith, R. Dynamic systems identification with Gaussian processes. *Mathematical and Computer Modelling of Dynamical Systems*, 11(4):411–424, 2005.

Mattos, C. L. C., Dai, Z., Damianou, A., Forth, J., Barreto, G. A., and Lawrence, N. D. Recurrent Gaussian processes. *arXiv preprint arXiv:1511.06644*, 2015.

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

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

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

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

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

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

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

Titsias, M. 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.

Vijayakumar, S. and Schaal, S. LWPR: An O(n) algorithm for incremental real time learning in high dimensional space. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, 2000.

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

Williams, C. K. and Rasmussen, C. E. *Gaussian processes for machine learning*. MIT Press, 2005.