Structured Variational Inference in Unstable Gaussian Process State Space Models

Silvan Melchior  
Department of Computer Science  
ETH Zürich  
silvan.melchior@alumni.ethz.ch

Felix Berkenkamp  
Department of Computer Science  
ETH Zürich  
befelix@inf.ethz.ch

Sebastian Curi  
Department of Computer Science  
ETH Zürich  
sebastian.curi@inf.ethz.ch

Andreas Krause  
Department of Computer Science  
ETH Zürich  
krausea@inf.ethz.ch

Abstract

Gaussian processes are expressive, non-parametric statistical models that are well-suited to learn nonlinear dynamical systems. However, large-scale inference in these state space models is a challenging problem. In this paper, we propose CBF-SSM, a scalable model that employs a structured variational approximation to maintain temporal correlations. In contrast to prior work, our approach applies to the important class of unstable systems, where state uncertainty grows unbounded over time. For these systems, our method contains a probabilistic, model-based backward pass that infers latent states during training. We demonstrate state-of-the-art performance in our experiments. Moreover, we show that CBF-SSM can be combined with physical models in the form of ordinary differential equations to learn a reliable model of a physical flying robotic vehicle.

1 Introduction

Learning state space models (SSMs) (A. Billings, 2013) of nonlinear dynamical systems is a key challenge for time series prediction tasks and model-based reinforcement learning (RL). While deterministic SSMs with parametric cells, such as long short-term memory (Hochreiter and Schmidhuber, 1997), can achieve great performance in expectation (Venugopalan et al., 2014), they do not quantify uncertainty in their predictions. However, many advanced methods require explicit model error estimates to, for example, improve exploration or provide safety guarantees in model-based reinforcement learning (Kamthe and Deisenroth, 2017; Berkenkamp et al., 2017).

Probabilistic inference in SSMs is one way to obtain reliable uncertainty estimates. Since closed-form inference in nonlinear SSMs with latent states is usually intractable, we use variational inference to obtain an approximation of the true posterior. Previous work by Doerr et al. (2018) proposed a variational distribution that keeps the temporal dependencies by combining forward sampling through the model with variational inference. While this approximation works well for many systems, we show that it can fail to learn on systems that are not mean square stable (MSS), see Fig. 1 (orange) for an example. Informally, a system is not MSS when the state uncertainty increases with time. These systems are ubiquitous in real-world control and robotics applications.

In this paper, we propose a new variational approximation that can reliably learn on systems that are not MSS, see Fig. 1 (blue). Like Doerr et al. (2018), we use a non-factorized variational approximation and place a Gaussian Process (GP, Rasmussen (2004)) prior on the transition function.
Figure 1: Open-loop predictions on the test-set for a noisy Dubin’s car model. Unlike the true posterior (3), PR-SSM’s variational approximation (4) does not condition the transition probabilities on measurements during training. As a result, when trained on long sequences, it explains the observations with high-variance observation noise in Fig. 1b and yields low-accuracy predictions in Fig. 1a. In contrast, CBF-SSM conditions on measurements in (6) during training (not at test-time) and can learn an accurate and reliable model of the system.

to obtain calibrated predictions. However, we capture additional structure of the true posterior in our variational approximation by conditioning the transition probabilities on observations during training. In partially observable unstable systems, we introduce a probabilistic system-based smoother that runs backward over the observations to summarize them in pseudo-observations. We then use these pseudo-observations in our tractable approximation of the posterior and call the resulting approximation CBF-SSM (conditional backward-forward). In summary, the main contributions of this paper are (i) a variational approximation that retains correlations and captures additional structure of the posterior, (ii) a probabilistic, system-based recognition model, and (iii) experiments that demonstrate state-of-the-art performance on challenging systems, including a combinations with physical models to learn from real-world data of a flying robotic vehicle.

Related work Archer et al. (2015) use variational inference in SSMs with a neural network model to approximate the posterior distribution. To obtain a tractable algorithm, they propose a posterior covariance based on that of a linear dynamical system. Krishnan et al. (2017) relax this assumption and add a smoother backward pass with a bi-directional recursive neural network to their variational approximation. While both approaches illustrate the power of variational inference, Guo et al. (2017) show that neural networks can yield unreliable uncertainty estimates on test data.

Instead, GPs quantify uncertainty based on a measure of the similarity to training examples and the corresponding GP-SSMs are commonly used to describe dynamical systems (Wang et al., 2006). Traditional approaches in GP-SSM system identification use Markov Chain Monte Carlo methods to obtain exact posterior samples (Frigola et al., 2013, 2014). However, these methods do not scale to larger and more complex problems. Mattos et al. (2015) introduce variational inference in such models with a mean-field approximate posterior distribution, which does not capture temporal correlations of the latent state. This can lead to sub-optimal solutions (Doerr et al., 2018).

To capture the temporal correlations in the approximate posterior, Eleftheriadis et al. (2017) propose a structured approximation for linear time-varying systems. Doerr et al. (2018) introduce PR-SSM, a method that uses this approximation and additionally incorporates nonlinear GP dynamics. To improve performance in systems with large process noise, Ialongo et al. (2018) add a smoothing scheme that introduces latent observations for each time step. In this work, we instead use a variational distribution that conditions on pseudo observations that are inferred with a probabilistic backward pass, without adding a new variational variable for each time step.

Krishnan et al. (2017) consider a similar, but deterministic, backward pass to infer hidden states. While Doerr et al. (2018) also use a backward pass, they only infer the initial state distribution based on a small number of future observations during training. In this work, we combine ideas from both approaches: we use a probabilistic backward pass for all observations during training, but run it only over a small number of future observations. Moreover, we modify this probabilistic backward pass to improve reliability in the open-loop predictions.
2 Problem Statement and Background

In this paper, we consider the problem of time series prediction: At test time, we are given a sequence of control actions \( u_{1:T} \) together with initial observations \( y_{1:t'} \) and we must predict future observations \( y_{t:T} \) in open-loop. For training, we have access to a training data set that consists of sequences of actions and corresponding observations. To make predictions, we must make assumptions about the process that generates these observations. In this section, we introduce GP-SSMs and review the variational inference method by Doerr et al. (2018) that our paper builds on.

**State space model** We model the process that generates observations with a SSM. Given a control input \( u \), the Markovian latent state \( x \in \mathbb{R}^{d_x} \) evolves over time based on a transition function \( f \). At every time step \( t \), we obtain measurements \( y_t \in \mathbb{R}^{d_y} \) of the state \( x_t \). The state transitions and observations are corrupted by zero-mean Gaussian noise \( \epsilon_t \) and \( \gamma_t \) with diagonal covariance matrices \( \Sigma_x \) and \( \Sigma_y \), respectively. The resulting SSM,

\[
x_{t+1} = f(x_t, u_t) + \epsilon_t, \quad y_t = Cx_t + \gamma_t,
\]

uses a linear observation model \( p(y_t | x_t) = \mathcal{N}(y_t | Cx_t, \Sigma_y) \) without loss of generality, since nonlinear observation functions can be absorbed into the transition function \( f \) (Frigola-Alcande, 2015). In particular, we set \( C = [I, 0] \) so that only the first \( d_y \) latent dimensions are observed. In the following, we omit the inputs \( u \) and consider an autonomous system \( f(x) \). However, our derivations remain valid with control inputs and we use them in our experiments.

**Gaussian process** We model the transition function \( f \) with a parametric Gaussian distribution. While our derivations hold for any model, we focus on GPs specifically. A GP is a distribution over functions \( f' \colon \mathbb{R}^{d_x} \to \mathbb{R} \) that is parametrized by a mean and covariance function \( m(\cdot) \) and \( k(\cdot, \cdot) \), respectively, which encode the expected value and similarities in the input space. Conditioned on observed function values \( f = (f'(x_1), \ldots, f'(x_N)) \) at inputs \( X = \{x_1, \ldots, x_N\} \), the mean and variance of the Gaussian posterior distribution for the function value \( f'(x) \) at input \( x \) is

\[
p(f'(x) \mid x, f, X) = \mathcal{N}(f'(x) \mid \mu_{GP}(x), \sigma_{GP}^2(x))
\]

where

\[
\mu_{GP}(x) = m(x) + k_x k_{XX}^{-1} (f - m_X), \quad \sigma_{GP}^2(x) = k(x, x) - k_x k_{XX}^{-1} k_x,
\]

with

\[
m_X = (m(x_1), \ldots, m(x_N)), \quad k_x = k(x, X), \quad k_{XX} = k(X, X), \quad k_{XX}^{-1} = k(X, X) - k_{X}^2. \tag{2}
\]

For multidimensional transition functions \( f \) with \( d_x \geq 1 \) we use independent GPs for each dimension to reduce computational complexity, although our method is not limited to this choice.

We use a sparse approximation scheme by Snelson and Ghahramani (2006) to avoid the cubic scaling in the number of observations in (2). This method conditions the GP on a finite number of inducing points \( \zeta \) at inputs \( \zeta \) and assumes independence of all other observations, i.e., \( p(f(x) \mid x, \zeta) \approx p(f(x) \mid x, \zeta, \zeta) \). While subsequent state transitions are correlated under a GP model, this assumption enforces independent, Markovian transitions. Together with the process noise, the state transition probability \( p(x_t \mid x_{t-1}, \zeta) \) is a sparse GP predictive distribution. As suggested by Snelson and Ghahramani (2006) and Doerr et al. (2018), we use the prior distribution

\[
p(\zeta) = \mathcal{N}(\zeta \mid m_c, K_{c,c}),
\]

which encourages the inducing points to take values consistent with the GP prior distribution.

**Variational approximation** We want to infer the posterior distribution over the latent states \( x_{1:T} \) and inducing points conditioned on the observations \( y_{1:T} \), which is given by (Ialongo et al., 2018)

\[
p(x_{1:T}, \zeta_f \mid y_{1:T}) = p(x_1 \mid y_{1:T}) p(\zeta_f \mid y_{1:T}) \prod_{t=2}^{T} p(x_t \mid x_{t-1}, \zeta_f, y_{1:T}). \tag{3}
\]

The first two terms encode the posterior probability of the initial state and the inducing points, while the last term quantifies the state transitions. Due to the Markov assumption, the state \( x_t \) is independent of past observations \( y_{1:t-1} \) given the previous latent state \( x_{t-1} \) and the inducing points \( \zeta_f \).

Computing the posterior distribution in (3) is intractable. We use variational inference instead, which approximates (3) with a tractable proposal distribution \( q \) by minimizing the KL-distance between the two distributions. This turns out to be equivalent to maximizing the so-called evidence lower bound (ELBO). The PR-SSM method by Doerr et al. (2018) uses the variational approximation

\[
q(x_{1:T} \mid y_{1:T}) = q(x_1 \mid y_{1:T}) q(\zeta_f) \prod_{t=2}^{T} p(x_t \mid x_{t-1}, \zeta_f). \tag{4}
\]
Figure 2: PR-SSM’s variational approximation only uses the first $y_{1:t'}$ measurements for its variational approximation. Instead, CBF-SSM approximates the conditioning on $y_{t:T}$ present in the true posterior (3) with a one-step conditioning on pseudo-observations $\tilde{y}$. This enables CBF-SSM to learn unstable systems (Fig. 1). During training we interleave multiple independent backward passes (Fig. 2b) to be closer to the test conditions with open-loop predictions in Fig. 2c.

The distribution $q(x_1 | y_{1:t'})$ over the first latent state $x_1$ is called the recognition model. It is implemented as a bi-directional recurrent neural network (Schuster and Paliwal, 1997) and acts as a smoother on the first $t'$ observations. For the transition dynamics, PR-SSM uses the exact GP transition probabilities $p(x_t | x_{t-1}, z_f)$. However, unlike the true posterior (3), they are not conditioned on the observations $y_{t:T}$, as this is intractable to compute. Lastly, for the GP’s inducing point distribution $q(z_f)$, they use a mean field assumption and fit a Gaussian with a diagonal covariance matrix. This assumption retains temporal correlations between states through the transition model.

To maximize the ELBO induced by (4), Doerr et al. (2018) use a doubly stochastic optimization scheme (Salimbeni and Deisenroth, 2017). They first sample a mini-batch of sub-trajectories from the dataset and then, given the distribution $q(x_1 | y_{1:t'})$, estimate the distribution over $x_{1:T}$ by sampling transitions from the GP transition model $f$, see Fig. 2a. They differentiate through this sampled representation using the re-parametrization trick (Kingma and Welling, 2013) and optimize the ELBO over all parameters jointly: GP hyperparameters, process- and observation-noise, recognition model parameters, inducing point positions $\zeta$, and the outputs’ mean and variance in $q(z_f)$.

3 Variational Inference in Unstable Systems

In this section, we show that PR-SSM can fail to learn on unstable systems. Next, we develop CBF-SSM, a structured variational approximation that retains the performance of PR-SSM on stable systems, but additionally allows us to train on unstable systems.

Unstable systems As an illustrative example, we consider a simple, noisy car model: The state $x \in \mathbb{R}^3$ describes the two-dimensional position and scalar rotation of the car. We observe only the positions, $y \in \mathbb{R}^2$ so we must estimate the rotation. The control input $u \in \mathbb{R}^2$ consists of the target speed and the curvature of the steering command. When we train PR-SSM on data from this system and use the learned model for open-loop predictions, we obtain the mean trajectory shown in Fig. 1a. PR-SSM predicts a mean close to the center of the dataset and explains the measurements through observations noise with large variance $\Sigma_y$, see Fig. 1b. While this prediction is reliable since the uncertainty estimate contains the true observations, it is not accurate enough to be of practical use.

The true dynamics used to generate the dataset are not asymptotically stable (not a contraction mapping). As a consequence, in open-loop predictions, the uncertainty introduced by the process noise $\epsilon_t$ in (1) compounds over time. This can be seen in Fig. 1b (blue shaded), where the variance over the state increases with the prediction horizon. More precisely, in systems that are not mean square stable (MSS) (Soong, 1973; Khasminskii, 2012), the predictive state covariance matrix $\lim_{t \to \infty} E [x_t x_t^T | x_0]$ is not bounded for open-loop predictions. This also implies that the uncertainty over the predicted observations $y$ grows unbounded over time. For example, a linear system with at least one eigenvalue of the transition matrix greater or equal than one combined with additive, non-zero process noise is not MSS. If the GP model accurately approximates the true transitions, it cannot be MSS either.
Non-MSS systems cause problems when optimizing the ELBO of PR-SSM. The true posterior in (3) conditions the transitions on the observations \( p(x_t | x_{t-1}, z_f, y_1:T) \) which counteracts the process noise and leads to a low-entropy distribution over \( x_{1:T} \). PR-SSM approximates this posterior distribution with an open-loop forward pass in (4), which iteratively samples from the transition probabilities \( p(x_t | x_{t-1}, z_f) \). For non-MMS systems, the state variance for this open-loop prediction increases as the prediction horizon gets longer. Eventually, the variance is so large, that the closest approximation to the posterior is no longer obtained by learning the transition model, but rather by assigning high observations noise to the system, together with a transition model that keeps the state close to the center of the output space, see Fig. 1b (orange shaded).

In summary, the main challenge in optimizing the ELBO of PR-SSM is that over long time horizons the open-loop predictions have necessarily high-variance on non-MSS systems and thus, do not agree with the observations. While in principle one could train on shorter sequences to avoid this problem, we show in Sec. 4 that, due to the measurement noise together with the partially observed nature of the system, the best results can only be obtained by training on long sequences.

**Conditioned variational distribution**

We want to retain the performance of PR-SSM on MSS systems, but also to be able to train on long sequences on unstable systems to obtain the best results. To this end, we have to account for the fact that the true posterior conditions the transition probabilities on the observations \( y_{1:T} \) in our variational approximation. Since \( p(x_t | x_{t-1}, z_f, y_1:T) \) cannot be computed in closed-form, we propose a two-step procedure to approximate this term. First, we summarize the information present in \( y_{1:T} \) with a distribution over a pseudo-observation of the full state, \( \tilde{y}_t \in \mathbb{R}^{d_f} \). As a second step, we condition the Gaussian transition probabilities on the pseudo-observation \( \tilde{y}_t \). The resulting forward pass is given by

\[
q(x_{1:T}, z_f | \tilde{y}_{1:T}) = q(z_f)q(x_1 | \tilde{y}_1) \prod_{t=2}^{T} p(x_t | x_{t-1}, z_f, \tilde{y}_t),
\]

and is similar to the true posterior (3), but uses the summarized information in \( \tilde{y}_1 \) instead of \( y_{1:T} \). This structured variational approximation uses the information in \( \tilde{y}_t \) to counteract the process noise and thereby leads to reliable inference. For now, we assume that computing \( p(x_t | x_{t-1}, x_f, \tilde{y}_t) \) is tractable and focus on the first step.

To summarize the information in \( y_{1:T} \), we use an additional model that runs backward over the observations analogously to the forward model, see Fig. 2a: given a current summary \( \tilde{y}_{t+1} \) and an observation \( y_t \), the backward model outputs a distribution over the previous summary \( \tilde{y}_t \). Krishnan et al. (2017) propose a similar, deterministic method. Instead, we use a probabilistic model to explicitly encode uncertainty about observations. For convenience, we use the same model as in the forward pass, which is a GP with a new set of inducing points \( z_0 \). The resulting structured variational distribution is

\[
q(x_{1:T}, \tilde{y}_{1:T}, z_f, z_0 | y_{1:T}) = q(x_{1:T}, z_f | \tilde{y}_{1:T}) q(z_0) q(\tilde{y}_T | y_T) \prod_{t=1}^{T-1} q(\tilde{y}_t | \tilde{y}_{t+1}, z_0, y_t),
\]

where the second part runs backward over the sequence of observations to infer the pseudo-observations \( \tilde{y} \), while the first part in (5) runs forward and is conditioned on the pseudo-observations. The resulting distribution is comparable to a forward-backward smoother and visualized in Fig. 2a.

We derive the corresponding ELBO in Appendix B.1, which yields

\[
\mathcal{L} = \lambda_1 \sum_{t=1}^{T} \mathbb{E}_q \left[ \log p(y_t | x_t) \right] - \lambda_1 \sum_{t=2}^{T} \mathbb{E}_q \left[ D_{KL}(q(x_t | x_{t-1}, z_f, \tilde{y}_t) \parallel p(x_t | x_{t-1}, z_f)) \right] \\
+ \lambda_2 \sum_{t=1}^{T-1} \mathbb{E}_q \left[ H(q(\tilde{y}_t | \tilde{y}_{t+1}, z_0, y_t)) \right] - D_{KL}(q(z_f) \parallel p(z_f)) - D_{KL}(q(z_0) \parallel p(z_0)).
\]

The first term maximizes the likelihood of the observations given the latent states as in PR-SSM. The second term penalizes the KL-divergence between the forward model conditioned on the pseudo-observations and the unconditioned distribution. It also ensures that the forward model does not blindly use information from the backward model and allows our training distribution to generalize to open-loop predictions where no observations are available. The third term maximizes the expected entropy of the backward model, which ensures that it does not become over-confident about its pseudo-observations. Finally, the last two terms regularize the inducing points as in PR-SSM and force them to stay close to their prior distributions. We have introduced additional tuning parameters \( \lambda_1 \) and \( \lambda_2 \) to account for the sampling batch-size, which we discuss later.

**Conditioned transition probabilities**

To optimize (7), we must compute the KL divergence between the transition probabilities with and without conditioning on \( \tilde{y}_t \). Since the latter term has a
Gaussian distribution, we approximate \( q(x_t | x_{t-1}, z_f, \tilde{y}_t) \) with a Gaussian distribution too in order to obtain a closed-form solution. While one could consider several approximation schemes, we focus on a method that is both probabilistic and allows for multi-modal distributions. In particular, we propose a joint forward-backward sampling scheme. To obtain a sample from our variational approximation \( q \) in (6), we first propagate a particle through the backward model in order to obtain sampled pseudo-observations. Then, in the forward-pass, we condition each particle on these sampled observations, assuming the same Gaussian observation noise as in (1). Since the measurement likelihood is Gaussian, this distribution can be computed analytically, as in the update step of the Kalman filter (Kalman, 1960; Archer et al., 2015). This update is conceptually similar to the resampling step in a particle filter (Gordon et al., 1993) but can be differentiated analytically (Jonschkowski et al., 2018). While the distribution over pseudo-observations can be multi-modal, we obtain a tractable approximation. When we employ this sampling scheme to optimize the ELBO (7) on the car example of Sec. 2, we obtain accurate and reliable predictions on the test set, as shown in Fig. 1.

### 3.1 Practical considerations

In this section, we discuss several practical modifications to our method that can improve performance.

At test time, we measure performance on open-loop predictions, where we have access to the first \( t' \) observations and want to predict the state evolution of the system, see Fig. 2c. To reflect this situation during training, we run replicated recognition models over \( t' \) steps of the sub-trajectory to infer the pseudo-observations, see Fig. 2b. This optimizes the backward inference model for the short-horizon inference tasks during open-loop predictions and can be parallelized along the sub-trajectory.

Similarly, although the conditioned transition probabilities in our variational distribution (6) are necessary to learn unstable systems, this optimizes the forward GP model under the assumption that its predictions are corrected by pseudo-observations at every step. As a result, the forward model focuses on learning one-step predictions. Although this might be desired for interpolation, i.e., inferring the states \( x_{1:T} \) given the observations \( y_{1:T} \), during open-loop predictions, the model should trade-off one-step and long-term accuracy.

We propose to increase the observation noise of the pseudo-observations, so that we retain the ability to train on unstable systems, but allow for better open-loop predictions. Given \( x_{t-1} \), the distribution over the next state is given by the combination of the forward model prediction and the process noise \( \epsilon_t \), so that \( p(x_t | x_{t-1}, z_f) = N(x_t | \mu_{\text{GP}}, \Sigma_{\text{GP}} + \Sigma_x) \). Instead of conditioning this transition on \( \tilde{y}_t \) at every step, we approximate the case when pseudo-observations are available only every \( k \) time steps. That is, we approximate the measurement model \( p(\tilde{y}_t | x_t) = N(\tilde{y}_t | x_t, \Sigma_y) \) with \( q(\tilde{y}_t | x_t, x_{t-1}, z_f) = N(\tilde{y}_t | x_t, \Sigma_y + (k-1)(\Sigma_{\text{GP}} + \Sigma_x)) \). This approximation assumes that the variance of the GP is roughly equal over the next \( k \) time steps and is equivalent to introducing a bound of \( 1/k \) on the Kalman filter gain, see Appendix B.6 for a derivation. Consequently, this reduces the effect of the pseudo-observations on the posterior state distribution.

---

**Figure 3:** Effect of the weakened conditioning factor \( k \) on a simple, synthetic dataset from a nonlinear spring. For small values of \( k \) in Fig. 3a, each model performs the best \( k \)-step ahead predictions for the \( k \) it was trained with. For larger values in Fig. 3b eventually the conditioning is too weak to be useful for training. The lines show the mean ± std. deviations over 5 independently learned models.
We illustrate this trade-off on a simple, synthetic dataset from a mechanical, nonlinear spring in Fig. 3. For small values of $k$ in Fig. 3a, the models trained with a $k$-step conditioning achieve the lowest RMSE after a $k$-step open-loop prediction. For large values of $k$, the conditioning becomes too weak to be useful for training and the performance begins to decrease for $k \geq 100$ in Fig. 3b. More examples can be found in Appendix C.1.

Tuning factors In our ELBO (7), we introduced two tuning parameters $\lambda_1$ and $\lambda_2$, which allow to trade-off prediction likelihood and regularization. In standard stochastic variational inference (Hoffman et al., 2013), these are used to reweigh the prior (regularization) to account for the batch size relative to the full dataset. In practice, this leads to poor results both in our method, CBF-SSM, and PR-SSM by Doerr et al. (2018). The reason for this is that the sampled sub-trajectories are not independent of each other, but also that, e.g., the mean field assumption on the inducing points is violated by datasets generated by certain linear systems. In practice, we use a validation set to tune $\lambda_1$ and $\lambda_2$ to obtain high-accuracy yet reliable predictions. See Appendix D.1 for details.

4 Experiments

In this section, we compare CBF-SSM and PR-SSM on several partially-observable datasets. While (7) conditions the GP model on samples from $q(z_f)$ to obtain a correct posterior (Ialongo et al., 2018), in our implementation we instead follow Doerr et al. (2018) and marginalize the inducing point distribution $q(z_f)$ out. While this correlates transitions and leads to incorrect samples, in practice it improves performance. The code is available at https://github.com/silvanmelchior/CBF-SSM.

Synthetic unstable system We evaluate the performance on the toy car dataset introduced in Sec. 3. Fig. 1 shows a qualitative comparison of PR-SSM and CBF-SSM when trained on sequence lengths of 300 and the resulting test error for different sequence lengths is shown in Fig. 4a. CBF-SSM achieves lower test error when training on longer sequences, while PR-SSM fails to learn the system accurately on long sequences. Beyond the mean error, we show in Appendix C.2 that both CBF-SSM and PR-SSM quantify the model uncertainty reliably. This suggests that CBF-SSM achieves better mean predictions without overfitting. While the more complicated backward pass renders our method 3.7 times slower than PR-SSM during training, the prediction cost at test-time is equal between both models, see Appendix C.3 for details.

MSS systems We compare CBF-SSM against PR-SSM on the datasets used by Doerr et al. (2018), where PR-SSM outperforms other methods. Fig. 5a shows that both methods reach comparable accuracy. To evaluate the reliability of the uncertainty quantification, we evaluate the proportion of the test-observations contained in the scaled confidence intervals, relative to what would be expected from the Gaussian cumulative distribution function (CDF). Fig. 5b shows that CBF-SSM achieves reliable predictions on the ACTUATOR dataset. The remaining plots can be found in Appendix C.2.

Incorporating physical knowledge (VoliroX) To demonstrate that CBF-SSM can be applied to real-world, complex, and unstable systems, we use it to learn the dynamics of a flying robotic vehicle. VoliroX (Bodie et al., 2018) is a drone consisting of twelve rotors mounted on six tiltable arms, which cause airflow interference that is difficult to model. The dataset includes measured position and orientation $p \in \mathbb{R}^6$, while linear and angular velocities $v \in \mathbb{R}^6$ are unobserved and must be estimated. Control inputs are the arm tilt angles $\alpha \in \mathbb{R}^6$ and motor signals $\eta \in \mathbb{R}^6$. Bodie et al. (2018) model the
### Table 1: Test RMSE [mean (std)] over five runs.

| Dataset      | PR-SSM     | CBF-SSM    |
|--------------|------------|------------|
| ACTUATOR     | 0.502 (0.031) | 0.522 (0.020) |
| BALLBEAM     | 0.073 (0.007) | 0.070 (0.010) |
| DRIVES       | 0.492 (0.038) | 0.456 (0.040) |
| DRYER        | 0.140 (0.018) | 0.178 (0.050) |
| FURNACE      | 1.249 (0.029) | 1.204 (0.024) |
| SARCOs       | 0.049 (-)    | 0.045 (0.004) |

(a) Test RMSE [mean (std)] over five runs.

Figure 5: Comparison between PR-SSM and CBF-SSM on stable datasets. Both models reach comparable accuracy in terms of RMSE, see Fig. 5a. Fig. 5b shows that the proportion of datapoints contained in the scaled confidence intervals matches the Gaussian CDF, indicating reliable uncertainty.

### Figure 6: CBF-SSM learns improved forces on top of an existing physical model. Visualized are the test-set predictions. The reference is calculated from the measured positions and never seen by the model. The lines in Fig. 6b are the predicted mean ±1.96 the predicted std. deviation.

The physical model does not model airflow interference, which leads to significant prediction errors in Fig. 6a. In contrast, CBF-SSM provides accurate predictions with reliable uncertainty information in Fig. 6b. We compare these predictions to PR-SSM for different training sequence lengths in Fig. 4b. Since the drone is unstable and has large process noise, PR-SSM can only train on short sequences. In contrast, CBF-SSM can reliably train on all sequence lengths. Moreover, on the best sequence length for PR-SSM, CBF-SSM achieves a MSE that is 47.2% lower than that of PR-SSM. At the same time, the variance statistics in Appendix C.2 indicate reliable uncertainty estimates.

**5 Conclusions**

We introduced CBF-SSM, a novel variational method for system identification in GP-SSMs. While we match the performance of existing methods on stable systems, our variational distribution is closer to the true posterior and can also reliably learn on unstable systems. This is achieved conditioning the forward pass on pseudo-observations inferred by a probabilistic backward pass. In our experiments, we demonstrate state-of-the-art accuracy of CBF-SSM and prove its applicability to more challenging and unstable real-world problems, including the ability to incorporate prior physical knowledge.
Acknowledgments

This project has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme grant agreement No 815943. It was also supported by a fellowship from the Open Philanthropy Project. We would like to thank Karen Bodie and Maximilian Brunner for the Voliro robot data and valuable discussions.

References

Stephen A. Billings. Nonlinear system identification: Narmax methods in the time, frequency, and spatio-temporal domains. Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, 08 2013.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathan Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski. Black box variational inference for state space models. arXiv preprint arXiv:1511.07367, 2015.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In Advances in Neural Information Processing Systems, pages 908–918, 2017.

Karen Bodie, Zachary Taylor, Mina Kamel, and Roland Siegwart. Towards efficient full pose omnidirectionality with overactuated mavs. CoRR, abs/1810.06258, 2018.

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

Andreas Doerr, Christian Daniel, Martin Schiegg, Nguyen-Tuong Duy, Stefan Schaal, Marc Toussaint, and Trimpe Sebastian. Probabilistic recurrent state-space models. In Proceedings of the 35th International Conference on Machine Learning, pages 1280–1289, 2018.

Stefanos Eleftheriadis, Tom Nicholson, Marc Deisenroth, and James Hensman. Identification of gaussian process state space models. In Advances in Neural Information Processing Systems, pages 5309–5319, 2017.

Roger Frigola, Fredrik Lindsten, Thomas B Schön, and Carl Edward Rasmussen. Bayesian inference and learning in gaussian process state-space models with particle mcmc. In Advances in Neural Information Processing Systems, pages 3156–3164, 2013.

Roger Frigola, Yutian Chen, and Carl Edward Rasmussen. Variational gaussian process state-space models. In Advances in Neural Information Processing Systems, pages 3680–3688, 2014.

Roger Frigola-Alcàde. Bayesian time series learning with gaussian processes. University of Cambridge, 2015.

Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In IEE Proceedings F-radars and signal processing, volume 140, pages 107–113. IET, 1993.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1321–1330. JMLR. org, 2017.
Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation, 9*(8): 1735–1780, 1997.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research, 14*(1):1303–1347, 2013.

Alessandro Davide Ialongo, Mark van der Wilk, James Hensman, and Carl Edward Rasmussen. Non-factorised variational inference in dynamical systems. *arXiv preprint arXiv:1812.06067*, 2018.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors. *arXiv preprint arXiv:1805.11122*, 2018.

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

Sanket Kamthe and Marc Peter Deisenroth. Data-efficient reinforcement learning with probabilistic model predictive control. *arXiv preprint arXiv:1706.06491*, 2017.

Rafail Khasminskii. *Stochastic Stability of Differential Equations*. Stochastic Modelling and Applied Probability. Springer-Verlag, Berlin Heidelberg, 2 edition, 2012.

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

Rahul G Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer, 2004.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing, 45*(11):2673–2681, Nov 1997. ISSN 1053-587X. doi: 10.1109/78.650093.

Edward Snelson and Zoubin Ghahramani. Sparse gaussian processes using pseudo-inputs. In *Advances in neural information processing systems*, pages 1257–1264, 2006.

Tsu T Soong. *Random differential equations in science and engineering*. Elsevier, 1973.

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

Jack Wang, Aaron Hertzmann, and David J Fleet. Gaussian process dynamical models. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 1441–1448. MIT Press, 2006.
Appendix A describes CBF-SSM-HALF, a reduced version of CBF-SSM for datasets without unstable hidden dimensions. Appendix B presents mathematical derivations and additional results neglected in the main paper. Appendix C shows additional experiments to gain further insights into the proposed method. Appendix D contains further details about the data collection processes and parameter tuning scheme.

A CBF-SSM Without a Backward Model

Datasets without unstable latent dimensions (i.e. no latent dimension which diverges over time or no latent dimensions at all) allow for a simpler and more efficient version of CBF-SSM without a backward model. Instead of using the backward model to infer $\tilde{y}_t$, the forward pass can condition on $y_t$ directly, which conditions the observed dimensions only. We call this variant CBF-SSM-HALF.

Fig. 7 compares PR-SSM and CBF-SSM-HALF on a variant of the car toy example in Sec. 3 with full observations (i.e. position and orientation). We can see that with full observations, CBF-SSM-HALF is sufficient to learn accurate predictions.

![Figure 7: A comparison of PR-SSM and CBF-SSM-HALF, a variant of CBF-SSM without a backward model and $\tilde{y}_t = y_t$. As a system we use the car example from Sec. 3, but with fully observed state (position and orientation). In this case, CBF-SSM-HALF is able to learn accurate predictions.](image)

However, on datasets with unstable hidden dimensions CBF-SSM-HALF can only make very short predictions before diverging, see Fig. 8 for a demonstration on the original example, where the orientation is not observed.

![Figure 8: A comparison of CBF-SSM-HALF and CBF-SSM. Visualized are the open-loop mean predictions on the test-set of the car example from Sec. 3.](image)
B Mathematical Derivations

In this Sec. we present various derivations neglected in the main paper for brevity.

B.1 ELBO derivation CBF-SSM

Since \(z_b\) and \(z_f\) learn similar dynamics, we would prefer to use the same regularization scheme on them. However, the lack of \(z_b\) in the original model results in an ELBO without this property. To overcome this issue, we define \(p(z_b)\) equivalent to \(p(z_f)\) and add the cross-entropy

\[
H(q(z_b), p(z_b)) = E_{q(z_b)} [-\log p(z_b)]
\]

(8)

This results in the ELBO

\[
\mathcal{L}_{CBF-SSM}
\]

\[
= E_q \left[ \log \frac{p(y_{1:T}, x_{1:T}, z_f)}{q(x_{1:T}, \tilde{y}_{1:T}, z_f, z_b | y_{1:T})} \right] - E_q [ -\log p(z_b)] \tag{9}
\]

\[
= E_q \left[ \log \frac{p(z_f)p(x_1)}{q(z_f)q(x_1 | \tilde{y}_1)} \prod_{t=2}^{T} p(x_t | x_{t-1}, z_f) \prod_{t=2}^{T} p(x_t | x_{t-1}, z) q(z_b)q(\tilde{y}_T | y_T) \prod_{t=1}^{T-1} \log p(\tilde{y}_t | \tilde{y}_{t+1}, z_b, y_t) \right] \tag{10}
\]

\[
= E_q \left[ \log \prod_{t=1}^{T} p(y_t | x_t) \right] + E_q \left[ \log \prod_{t=2}^{T} \frac{p(x_t | x_{t-1}, z)}{q(x_t | x_{t-1}, z_f, \tilde{y}_t)} \right] \tag{12}
\]

\[
+ E_q \left[ \log \frac{p(x_1)}{q(x_1 | \tilde{y}_1)} \right] + E_q \left[ \log \frac{p(z_f)}{q(z_f)} \right] + E_q \left[ \log \frac{p(z_b)}{q(z_b)} \right] \tag{13}
\]

\[- E_q \left[ \log q(\tilde{y}_T | y_T) \right] - E_q \left[ \log \prod_{t=1}^{T-1} q(\tilde{y}_t | \tilde{y}_{t+1}, z_b, y_t) \right] \tag{14}
\]

\[
= \sum_{t=1}^{T} E_q [\log p(y_t | x_t)] - \sum_{t=2}^{T} E_q [D_{KL}(q(x_t | x_{t-1}, z_f, \tilde{y}_t) \| p(x_t | x_{t-1}, z))] \tag{15}
\]

\[- D_{KL}(q(x_1 | \tilde{y}_1) \| p(x_1)) - D_{KL}(q(z_f) \| p(z_f)) - D_{KL}(q(z_b) \| p(z_b)) \tag{16}
\]

\[+ H(q(\tilde{y}_T | y_T)) + \sum_{t=1}^{T-1} E_q [H(q(\tilde{y}_t | \tilde{y}_{t+1}, z_b, y_t))] \tag{17}
\]

As in PR-SSM, we chose a fixed, uninformative initial distribution for \(x_1\), such that the third term disappears from the loss. In particular, we set \(q(x_1) = q(\tilde{y}_1)\) based on the particle distribution, so that we have \(D_{KL}(q(x_1 | \tilde{y}_1) \| p(x_1)) \propto \sum_y \log p(y_1)\), which is constant as the variance of \(p(x_1)\) goes to infinity. Additionally, the sixth term disappears if we initialize \(\tilde{y}_T\) based on the measurement \(y_T\). Moreover, this term looses importance for longer trajectories.
B.2 ELBO derivation CBF-SSM-HALF

The ELBO of CBF-SSM-HALF can be reformulated to

\[
\mathcal{L}_{\text{CBF-SSM-HALF}} = \mathbb{E}_{q(x_{1:T}, z | y_{1:T})} \left[ \log p(y_{1:T}, x_{1:T}, z) \right] - \mathbb{E}_{q(z)} \left[ \log \frac{p(z)}{q(z)} \right] + \mathbb{E}_{q(x_1)} \left[ \log \frac{p(x_1)}{q(x_1)} \right]
\]

\[
= \mathbb{E}_{q(x_{1:T}, z | y_{1:T})} \left[ \log \prod_{t=2}^T p(y_t | x_{t-1}, z) \right] + \mathbb{E}_{q(x_1)} \left[ \log \prod_{t=2}^T p(y_t | x_{t-1}, z) \right]
\]

\[
+ \mathbb{E}_{q(z)} \left[ \log \frac{p(z)}{q(z)} \right] + \mathbb{E}_{q(x_1)} \left[ \log \frac{p(x_1)}{q(x_1)} \right]
\]

\[
= \sum_{t=2}^T \mathbb{E}_{q(x_{t-1} | y_{1:t-1})} \left[ \log p(y_t | x_{t-1}, z) \right] + \mathbb{E}_{q(x_1)} \left[ \log p(y_1 | x_1) \right]
\]

\[
- D_{\text{KL}}(q(z) \parallel p(z)) - D_{\text{KL}}(q(x_1) \parallel p(x_1)),
\]

where we applied Bayes rule in (22). Again, an uninformative initial distribution for \(x_1\) is chosen and fixed.
B.3 Approaching PR-SSM with k-factor

In the case of full observation, we can show that CBF-SSM approaches PR-SSM for large $k$. We have $\tilde{y} = y$ and thus can reformulate the ELBO to

$$L_{\text{CBF-SSM:HALF}} = E_{q(x_{1:T}, z | y_{1:T})} \left[ \log \frac{p(y_{1:T}, x_{1:T}, z)}{q(x_{1:T}, z | y_{1:T})} \right]$$

(28)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=1}^{T} p(y_{t} | x_{t}) \right) \left( \prod_{t=2}^{T} p(x_{t} | x_{t-1}, z) \right) \right]$$

(29)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=1}^{T} p(y_{t} | x_{t}) \right) \left( \prod_{t=2}^{T} p(x_{t} | x_{t-1}, z) \right) \right]$$

(30)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=1}^{T} p(y_{t} | x_{t}) \right) \left( \prod_{t=2}^{T} p(x_{t} | x_{t-1}, z) \right) \right]$$

(31)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=1}^{T} p(y_{t} | x_{t}) \right) \left( \prod_{t=2}^{T} p(x_{t} | x_{t-1}, z) \right) \right]$$

(32)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=1}^{T} p(y_{t} | x_{t}) \right) \left( \prod_{t=2}^{T} p(x_{t} | x_{t-1}, z) \right) \right]$$

(33)

$$= E_{q(x_{1:T} | y_{1:T})} \left[ \log \frac{p(z)p(x_{1})}{q(z)q(x_{1})} \left( \prod_{t=2}^{T} p(y_{t} | x_{t-1}, z) \right) \right] + E_{q(x_{1})} \left[ \log p(y_{1} | x_{1}) \right]$$

(34)

$$+ E_{q(x)} \left[ \log \frac{p(z)}{q(z)} \right] + E_{q(x_{1})} \left[ \log \frac{p(x_{1})}{q(x_{1})} \right]$$

(35)

$$= \sum_{t=2}^{T} E_{q(x_{t-1} | y_{t-1})} \left[ \log \frac{p(y_{t} | x_{t-1}, z)}{q(y_{t} | x_{t-1}, z, x_{t})} \right] + E_{q(x_{1})} \left[ \log p(y_{1} | x_{1}) \right]$$

(36)

$$- D_{KL}(q(z) \| p(z)) - D_{KL}(q(x_{1}) \| p(x_{1})).$$

(37)

All but the first term appear in the ELBO of PR-SSM. To analyze the first term, we neglect the indices and introduce $\sigma_{B}^2, \sigma_{C}^2$ and $\sigma_{w}^2$ as

$$p(y | x)q(y | x, z) = \frac{N(y | x, \sigma_{B}^2)N(y | f, \sigma_{B}^2 + k\sigma_{A}^2)}{N(y | x, \sigma_{B}^2 + (k-1)\sigma_{A}^2)} \quad (38)$$

$$= \frac{N(y | x, \sigma_{B}^2)N(y | f, \sigma_{A}^2)}{N(y | x, \sigma_{w}^2)} \quad (39)$$

$$= \frac{\sqrt{2\pi}\sigma_{w}^2}{\sqrt{2\pi}\sigma_{B}^2} \exp \left[ -\frac{(y - x)^2}{2\sigma_{w}^2} - \frac{(y - f)^2}{2\sigma_{w}^2} + \frac{(y - x)^2}{2\sigma_{w}^2} \right] \quad (40)$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{w}^2} \exp \left[ -\frac{(y - x)^2}{2\sigma_{w}^2} - \frac{(y - f)^2}{2\sigma_{w}^2} + \frac{(y - x)^2}{2\sigma_{w}^2} \right], \quad (41)$$

where $f$ is the GP predictive mean value introduced in (Doerr et al., 2018). By completing the square we can rewrite term (A) to
If we analyze the mean, we get thus our likelihood term describes the data by a unnormalized Gaussian distribution.

If we analyze the variance, we get

\[ \mu_L = \frac{\sigma_u^2 \sigma_w^2 \mathbf{x} + \sigma_u^2 \sigma_w^2 \mathbf{f} - \sigma_u^2 \sigma_w^2 \mathbf{x}}{\sigma_u^2 \sigma_w^2 + \sigma_u^2 \sigma_w^2 - \sigma_u^2 \sigma_w^2} \]

with

\[ \alpha = \frac{\sigma_u^2 (\sigma_w^2 - \sigma_x^2)}{\sigma_u^2 (\sigma_w^2 - \sigma_x^2) + \sigma_u^2 \sigma_w^2} = \frac{(k-1)(\sigma_B^2 + k \sigma_A^2)\sigma^2_A}{(k-1)(2\sigma_B^2 + k \sigma_A^2)(\sigma_A^2 + \sigma_B^2)}, \]

so the mean is a linear combination of \( \mathbf{x} \) and \( \mathbf{f} \). For \( k = 1 \), we have \( \alpha = 0 \), for \( k \to \infty \), we have \( \alpha \to 1 \).

If we analyze the variance, we get

\[ \sigma_L = \frac{\sigma_u^2 \sigma_w^2}{\sigma_u^2 \sigma_w^2 + \sigma_u^2 \sigma_w^2 - \sigma_u^2 \sigma_w^2} \]

and see that for \( k = 1 \) we have \( \sigma_u^2 = \sigma_w^2 \) and thus \( \sigma_L^2 = \sigma_u^2 \) and for \( k \to \infty \) we have \( \sigma_u^2 \to \sigma_w^2 \) and thus \( \sigma_L^2 \to \sigma_u^2 \).
We can rewrite $\lambda$ to

$$
\lambda = \exp \left[ -\frac{1}{2a} \left( \frac{\sigma_w^2}{\lambda_1} \frac{\sigma_u^2}{\lambda_2} \right) (x - f)^2 - \lambda_4 \frac{(\sigma_w^2)^2}{\lambda_4} \right]
$$

(55)

and see that for $k = 1$ we have $\lambda_2 = 0$ and $\lambda_4 = 1$, for $k \to \infty$ we have $\lambda_1 \to 0$ and $\lambda_4 \to 1$. $\lambda_3$ penalizes if the conditioned $x$ is too far from $f$.

We see that for $k = 1$, we fully condition and get back the original CBF-SSM before introducing $k$. For $k \to \infty$, we no longer condition and get back PR-SSM. In between, we have a mixture of the two methods, where $\lambda$ can be ignored for sufficient small/large values of $k$.

This result can be generalized to the partially observed case. We later show that the Kalman gain is bounded by $\frac{1}{k}$, which prevents strong conditioning on $\bar{y}$. Thus, the recognition model becomes negligible with large $k$ and the same reformulation as in the fully observed case can be applied.

### B.4 CBF-SSM training distribution

The main paper only presents a graphical illustration of the variational distribution of CBF-SSM for training. The mathematical formulation is based on two backward passes $\bar{y}_{1:T}^A$ and $\bar{y}_{1:T}^B$. Let’s define the sets of re-initialization indices as

$$
\mathcal{R}_A = \{2ir \mid i \in \mathbb{N}, 2ir < T\} \cup \{T\},
$$

(56)

$$
\mathcal{R}_B = \{(2i - 1)r \mid i \in \mathbb{N}, (2i - 1)r < T\} \cup \{T\},
$$

(57)

where $r$ is the recognition length. Furthermore, let’s define the conditioning mapping as

$$
c(t) = \begin{cases} A & \text{if } \left\lfloor \frac{t - 1}{r} \right\rfloor \equiv 2 \ 0 \\ B & \text{otherwise.} \end{cases}
$$

(58)

The variational distribution then becomes

$$
q(x_{1:T}, \bar{y}_{1:T}; z_f, z_b \mid y_{1:T}) =
$$

$$
q(z_f)q(x_1 \mid \bar{y}_1^{(1)}) \prod_{t=2}^{T} q(x_t \mid x_{t-1}, z_f, \bar{y}_t^{(t)}) q(z_b)
$$

(59)

$$
\prod_{l \in \mathcal{R}_A} q(\bar{y}_l^A \mid y_t) \prod_{l \in \mathcal{R}_B} q(\bar{y}_l^B \mid y_t)
$$

(60)

$$
\prod_{l=1}^{T} q(\bar{y}_l^A \mid \bar{y}_{t+1}^A, z_b, y_t) \prod_{t=1}^{T} q(\bar{y}_t^B \mid \bar{y}_{t+1}^B, z_b, y_t). 
$$

(61)

(62)

### B.5 CBF-SSM Kalman update step

The distribution $p(x_t \mid x_{t-1}, z, y_t) \sim \mathcal{N}(\mu, \Sigma)$ in CBF-SSM-HALF can be calculated analytically using a 1-step Kalman filter. It is Gaussian with mean $\mu$ and covariance matrix $\Sigma$ as:

$$
\Sigma_F = \Sigma_x + \Sigma_{GP}
$$

(63)

$$
y = y_t - C \mu_{GP}
$$

(64)

$$
S = \Sigma_y + C \Sigma_F C^T
$$

(65)

$$
K = \Sigma_F C S^{-1}
$$

(66)

$$
\mu = \mu_{GP} + K y
$$

(67)

$$
\Sigma = (I - KC) \Sigma_F (I - KC)^T + K \Sigma_y K^T
$$

(68)

where $\Sigma_x$, $\Sigma_y$ and $C$ are the covariance matrices for the process and observation noise and the parametric observation model, $\mu_{GP}$ and $\Sigma_{GP}$ are the mean and covariance of the GP predictive distribution. The result naturally generalizes to CBF-SSM.
B.6 k-factor bound

Our introduced $k$-factor causes an upper bound on the Kalman gain. For each dimension (with its corresponding $\sigma_A$ and $\sigma_y$) we have

$$\frac{\sigma_A^2}{\sigma_A^2 + \sigma_y^2} \Rightarrow \frac{\sigma_A^2}{\sigma_A^2 + \sigma_y^2 + (k-1)\sigma_A^2} = \frac{\sigma_A^2}{k\sigma_A^2 + \sigma_y^2} \leq \frac{1}{k}.$$  

(69)
Further Experiments

In this Sec. we show additional experiments neglected in the main paper.

C.1 k-factor examples

The main paper presented quantitative results on a synthetic dataset illustrating the trade-off between greedy and long-term predictions. Fig. 9 shows an example on the same synthetic dataset. A real-world example can be found in Fig. 10.

![Figure 9: Qualitative influence of $k$ on different prediction-lengths. The areas are the predicted 95% confidence interval.](image)

C.2 Prediction variance statistics

Fig. 11 shows variance statistics for all stable datasets on which CBF-SSM was tested on. They indicate overfitting tendencies for the FURNACE dataset, consisting of 148 training points only. All other statistics, including the one for DRIVES with 250 training points only, indicate reliable predictions. For completeness, we also show these statistics for the unstable datasets generated by the car example from Sec. 3 (see Fig. 12a) and VOLIROX (see Fig. 12b).

To get correct variance estimates on VOLIROX, a prior on the GP hyperparameters was necessary, due to non-identifiability of the noise (i.e. a position prediction error could be the result of many different sources, not necessarily the forces). We chose a beta distribution which prevents the length scales from getting too large and the noise from getting too small. The parameters were tuned on the validation set.

If we predict unstable dimensions as in the case of the car example from Sec. 3, the variance progression over time is actually much more interesting to study. Fig. 13 shows that the variance increases over time. However, there are short periods of contraction, which do not directly align with the system model. A more careful inspection showed that these happened at the border of the system,
Figure 10: Influence of $k$ on SARCOS dataset in the long run. The SARCOS dataset reacts very sensitive to $k$ because the given data does not allow to really learn the dynamics of the system, hence the optimizer needs to set focus on very accurate 1-step ahead predictions or a coarse approximation of the whole system. The areas are the predicted 95% confidence interval.

where the robot turns around. CBF-SSM learned that particles should not leave this territory and pushes particle further away back more quickly. A demonstration of this can be found in Fig. 14. Visualized is the mean path of particles during a contraction period. It shows that the GP learns a contraction at the border to regain certainty over the system by exploiting the reoccurring curve patterns in the data.

C.3 Computational performance comparison

Table 1 compares the training performance between PR-SSM, CBF-SSM-HALF and CBF-SSM. As all other experiments in this paper, the numbers were obtained on a machine with two Intel® Xeon® CPUs, utilizing a NVidia® GeForce® GTX 1080 Ti. As already mentioned in the main paper, the performance for prediction is the same for all three methods.
C.4 Recognition model variance

We investigate the variance of our fully probabilistic backward model on the car example from Sec. 3 by plotting the variance development over time. Fig. 15a shows that the longer the model runs backward over the sequence, the more confident it gets about the hidden dimensions. Contrary, if we
| Method         | Time [s] | Time [relative] |
|----------------|----------|-----------------|
| PR-SSM         | 24.3     | 1               |
| CBF-SSM-HALF   | 34.9     | 1.4             |
| CBF-SSM        | 89.8     | 3.7             |

Table 1: Training performance comparison. Measured is the duration of one complete training swipe through the dataset from the car example from Sec. 3 (i.e. epoch).

predict on unseen data (we scale the input by a factor of 100), Fig. 15b shows how the uncertainty increases, because each time step adds more uncertainty to the hidden dimensions.

![Figure 15](image)

Figure 15: Variance of hidden dimensions in BW-model for different time steps, both for seen and unseen data.

As a quick experiment, we replaced our backward model by a RNN. More concrete, we use a GRU-cell (Cho et al., 2014) which predicts both mean and variance of the hidden dimensions while running backward over in- and output. For seen data, the RNN learns a similar pattern as the probabilistic BW model, as can be seen in Fig. 16a. Surprisingly, the RNN also manages to recognize unseen data and adjusts it’s variance estimates, see Fig. 16b.

![Figure 16](image)

Figure 16: Variance of hidden dimensions in RNN for different time steps, both for seen and unseen data.

Note that CBF-SSM does not necessarily need a GP in the temporal dynamics, neither in the forward nor the backward pass. These first results hint that it might be possible to use more complex and powerful cells, while preserving reasonable uncertainty estimates both on seen and unseen test data.

21
C.5 VoliroX train- and validation-set

The main paper only shows the test set results on VoliroX. Fig. 17 shows them for the training and validation set as well.

![Graphs showing forces for VoliroX train, validate, and test sets.]

Figure 17: CBF-SSM learns improved forces on top of an existing physical model. The reference is calculated from the measured positions and never seen by the model. The lines are the predicted mean ±1.96 the predicted std. deviation.
D Training Details

In this Sec. we give further details to the training process and data-sets utilized.

D.1 Curriculum learning and parameter tuning

The CBF-SSM objective is highly non-convex and it’s sampling approximation inherently noisy. This caused problems during optimization in some experiments. To overcome them, we train CBF-SSM using a very simple two-step curriculum learning setting. We first train without the entropy regularizer and then retrain the model using the full ELBO.

This two-step scheme goes hand in hand with the tuning of the ELBO-parameters. In the first (train) phase, $\lambda_1$ is set such that the accuracy on the validation set is as high as possible, while $\lambda_2$ is set to 0. In the second (retrain) phase, $\lambda_1$ can be kept fix and $\lambda_2$ should be set such that the variance on the train- and validation-set is not under-estimated.

The remaining hyper-parameters were either set as in Doerr et al. (2018) or manually tuned using the same validation-set. The accompanying code contains the final values for all experiments in the main paper.

D.2 Data collection processes

All datasets used in this paper are already or will be made publicly available. For all simulation, we will furthermore publish the generating code which contains all the physical constants we used.

The data collection processes, sample sizes and train/test splits for the six datasets PR-SSM was already benchmarked on can be found in Doerr et al. (2018). For the Dubin’s car example from Sec. 3, we used a simulation of the system described in the main paper and sampled 25’000 train- and 5’000 test-points. For SPRINGNONLINEAR we simulated a one-dimensional damped spring where the input $u$ was mapped to a force $f$ through the non-linearity $f = \tanh(2u)$ and sampled 5’000 train- and 5’000 test-points. For VOLIROX we used two datasets called FLIP (2572 datapoints) and TILT (2300 datapoints) which were collected by recording a prototype of the drone. The exact collection process will be published together with the datasets. We split FLIP in half for training and validation, whereas TILT was used for testing.

We normalized all data before using them in our model and denormalized them for predictions afterwards, except for the VOLIROX experiments where no data pre-processing was used.