Deep Variational Bayes Filters: Unsupervised Learning of State Space Models from Raw Data

Maximilian Karl, Maximilian Soelch, Justin Bayer*, Patrick van der Smagt†
Chair of Robotics and Embedded Systems, Department of Informatics,
Technische Universität München, Germany
zip([karlma, m.soelch, bayer, smagt], [@in.tum.de, @tum.de, @sensed.io, @tum.de])

Abstract

We introduce Deep Variational Bayes Filters (DVBF), a new method for unsupervised learning of latent Markovian state space models. Leveraging recent advances in Stochastic Gradient Variational Bayes, DVBF can overcome intractable inference distributions by means of variational inference. Thus, it can handle highly nonlinear input data with temporal and spatial dependencies such as image sequences without domain knowledge. Our experiments show that enabling backpropagation through transitions enforces state space assumptions and significantly improves information content of the latent embedding. This also enables realistic long-term prediction.

1 Introduction

Estimating probabilistic models for sequential data is central to many domains, such as audio, natural language or physical plants [7, 19, 3, 4, 13]. The goal is to obtain a model \( p(x_{1:T}) \) that best reflects a data set of observed sequences \( x_{1:T} \). Recent advances in deep learning have paved the way to powerful models capable of representing high-dimensional sequences with temporal dependencies, e.g. [7, 19, 3, 1].

A typical model assumption in systems theory is that the observed sequence \( x_{1:T} \) is generated by a corresponding latent sequence \( z_{1:T} \). More specifically, state space models assume the latent sequence to be Markovian, i.e., \( z_t \) contains all information on the distribution of \( z_{t+1} \). Moreover, the emission distribution of \( x_t \) is assumed to be determined by the corresponding \( z_t \). In short, we assume a latent state \( z_t \) that holds all information available at time step \( t \). Efficient inference of such latent states is only partially solved with state-space models. Under strong assumptions on the system, one can derive optimal Bayesian filters, such as the classical Kalman filter [11] for linear Gaussian models (LGMs). Yet, for less restrictive models, posterior distributions \( p(z_{1:T} \mid x_{1:T}) \) are often intractable.

Leveraging a recently proposed estimator based on variational inference, stochastic gradient variational Bayes (SGVB, [12]), approximate inference of latent variables becomes tractable. Extensions to time series [1, 3, 1] resulted in considerable improvements of modeling quality in terms of compression, i.e., marginal likelihood of the data. Yet, in a wide range of applications, compression is of less importance than the recovery of interpretable, full-information latent states—a lacking feature in current approaches. This is crucial if the latent spaces are used in follow-up applications, of which model-based control is the most prominent and focus of this work.

The contribution of this work is, to our knowledge, the first model that (i) enforces the state-space model assumptions in latent space allowing for reliable and plausible long-term prediction of the observable system, (ii) inherits the merit of neural architectures to be trainable on raw data such as

*Justin Bayer is also affiliated with sensed.io UG (haftungsbeschränkt), München, Germany.
†Patrick van der Smagt is affiliated with fortiss, An-Institut der Technischen Universität München, Germany.
images, audio or other sensory inputs and (iii) scales to large data due to optimization of parameters based on stochastic gradient descent \cite{2}.

\section{Background and Related Work}

\subsection{Probabilistic Modeling and Filtering of Dynamical Systems}

We consider modeling a time-discrete, non-linear dynamical system with observations in some space $\mathcal{X} \subset \mathbb{R}^{n_x}$, depending on control inputs (or actions) from the space $\mathcal{U} \subset \mathbb{R}^{n_u}$. Elements of $\mathcal{X}$ can be high-dimensional sensory data such as raw images, or any other state observation. With $x_t \in \mathcal{X}$, let $x_{1:T} = (x_1, x_2, \ldots, x_T)$ be a sequence of length $T$ of observations. Similarly, with $u_t \in \mathcal{U}$, let $u_{1:T} = (u_1, u_2, \ldots, u_T)$ be a corresponding sequence of equal length $T$ of control inputs, which we consider as given. We are interested in deriving a probabilistic model

$$p(x_{1:T} \mid u_{1:T}).$$

We assume a state-space model with an underlying latent dynamical system over the space $\mathcal{Z} \subset \mathbb{R}^{n_z}$. Let $z_{1:T} = (z_1, z_2, \ldots, z_T), z_t \in \mathcal{Z}$, be the corresponding latent sequence. Contrary to $\mathcal{Z}$, the output space $\mathcal{X}$ is observable and may exhibit complex non-Markovian transitions. Formally, we assume the following graphical model:

$$p(x_{1:T} \mid u_{1:T}) = \int p(x_{1:T} \mid z_{1:T}, u_{1:T}) p(z_{1:T} \mid u_{1:T}) \, dz_{1:T}$$

(1)

Obtaining a hypothesis for the system dynamics is done via the estimation of the so-called emission model $p(x_{1:T} \mid z_{1:T}, u_{1:T})$ and transition model $p(z_{1:T} \mid u_{1:T})$. The latter is imperative for achieving good long-term results: a bad transition model can lead to divergence of the latent state.

Accordingly, we put special emphasis on it through a Bayesian treatment. Assuming that the transitions are non-stationary (and hence differ for each time step), we impose a prior distribution on transition parameters $\beta_{1:T}$ and marginalize it out subsequently:

$$\mathbb{E}[1] = \int \int p(x_{1:T} \mid z_{1:T}, u_{1:T}) p(z_{1:T} \mid \beta_{1:T}, u_{1:T}) p(\beta_{1:T} \mid \beta_{1:T}, u_{1:T}) \, d\beta_{1:T} \, dz_{1:T}$$

(2)

To obtain state-space models, we impose assumptions on emission and state transition model:

$$p(x_{1:T} \mid z_{1:T}, u_{1:T}) = \prod_{t=1}^{T} p(x_t \mid z_t)$$

(3)

$$p(z_{1:T} \mid \beta_{1:T}, u_{1:T}) = \prod_{t=0}^{T-1} p(z_{t+1} \mid z_t, u_t, \beta_t)$$

(4)

Eq. (3) assumes that the current state $z_t$ contains all necessary information about $x_t$. Likewise, Eq. (4) tells us that $z_t$ contains all necessary information for transition to $z_{t+1}$ (given the current control input $u_t$ and transition parameters $\beta_t$). A typical instance of these assumptions are Linear Gaussian Models (LGMs), i.e., both state transition and emission model are affine transformations with Gaussian offset noise:

$$z_{t+1} = F_t z_t + B_t u_t + w_t \quad w_t \sim \mathcal{N}(0, Q_t)$$

(5)

$$x_t = H_t z_t + y_t \quad y_t \sim \mathcal{N}(0, R_t)$$

(6)

Here, $F_t$ is referred to as the state transition matrix, $B_t$ as the control-input matrix. In our notation, one would typically set $\beta_t = w_t$, though other variants like $\beta_t = (F_t, B_t, w_t)$ are possible.

Finding efficient inference distributions $p(z_{1:T} \mid x_{1:T})$ is in general an unsolved problem. Three examples are prediction, filtering, and smoothing: inference of $z_t$ from $x_{1:t-1}, x_{1:t}$, or $x_{1:T}$, respectively. In classical systems theory, we consider the parameters of state transition and emission model as given. Under the strong assumptions (5) and (6) of LGMs, we can derive the provably optimal, well-known Kalman filters. While extensions of Kalman filters to nonlinear dynamical systems exist

\footnote{The case without control inputs can be recovered by setting $\mathcal{U} = \emptyset$, i.e., not conditioning on control inputs.}
and are successfully applied in many areas, they suffer from two major drawbacks: Firstly, its assumptions are restrictive and are violated in practical applications, leading to suboptimal results. Secondly, the parameters, particularly state-transition matrix $F_t$ and control-input matrix $B_t$, have to be known before filtering can be done. There have been efforts to learn such system dynamics, cf. [5], based on the expectation maximization (EM) algorithm. However, these algorithms are not applicable in cases where the true posterior distribution is intractable. This is the case if, e.g., image sequences are used, since the posterior is then highly nonlinear. Our new approach will tackle both issues.

### 2.2 Stochastic Gradient Variational Bayes (SGVB) for Time Series Distributions

Replacing the bottleneck layer of a deterministic auto-encoder with stochastic units $z$, the variational auto-encoder (VAE, [12, 16]) learns complex marginal data distributions on $x$ in an unsupervised fashion from simpler distributions via the graphical model

$$p(x) = \int p(x, z) \, dz = \int p(x \mid z)p(z) \, dz.$$  

In VAEs, $p(x \mid z) \equiv p_\theta(x \mid z)$ is typically parametrized by a neural network with parameters $\theta$. Within this framework, models are trained by maximizing a lower bound to the marginal data log-likelihood via stochastic gradients:

$$\ln p(x) \geq E_{q_\phi(z \mid x)}[\ln p_\theta(x \mid z)] - \text{KL}(q_\phi(z \mid x) \mid \mid p(z)) =: \mathcal{L}_{SGVB}(x, \phi, \theta) \tag{7}$$

This is provably equivalent to minimizing the KL-divergence between the approximate posterior or recognition model $q_\phi(z \mid x)$ and the true, but usually intractable posterior distribution $p(z \mid x)$. $q_\phi$ is parametrized by a neural network with parameters $\phi$.

The principle of VAEs has been transferred to time series [11, 3, 9]. These models employ nonlinear state transitions in latent space, but violate Eq. (4). Observations are directly included in the transition process. Consequently, the state space $Z$ does not reflect all information available. The recognition model $q_\phi(z \mid x)$ becomes a powerful compression algorithm, prohibiting plausible long-term generative prediction. Such phenomena with generative models have been explained in [18].

Others have been specifically interested in applying variational inference for controlled dynamical systems. In [19], a VAE is used to learn the mapping to and from latent space. The state transition is modeled by a neural network that outputs a local transition matrix. The regularization of this network is clearly motivated by Eq. (7). However, it fails to be a mathematically correct lower bound to the marginal data likelihood. Moreover, their recognition model needs all observations that contain information w.r.t. the current state, i.e., a temporal i.i.d. assumption on data. This requires significant domain knowledge for data pre-processing.

In [14], the state-space assumptions (3) and (4) are only softly encoded in the KL-divergence term of their loss function (a variant of Eq. (7)). No gradient information about reconstruction error (the expectation term in (7)) is ever back-propagated through the transition. Hence, firstly, a latent representation that enables good reconstruction is learned. This does not require information about time derivatives, i.e., information that can only be extracted from multiple observations. Secondly, the transition is learned to match this latent space. Indeed, experiments, cf. Section 3, show that their model fails to extract information such as velocity (and in general time derivatives). The latent space $Z$ violates Eq. (4).

A key contribution of this paper is the reversal this effect, i.e., forcing the latent space to fit the transition, thus achieving the state-space model assumptions and full information in the latent states.

### 3 Deep Variational Bayes Filters

#### 3.1 Reparametrizing the Transition

The central problem for learning latent states system dynamics is efficient inference of a latent space that obeys state-space model assumptions. If the latter are fulfilled, the latent space must contain all information. Previous approaches emphasized good reconstruction, so that the space only contains information necessary for reconstruction of one time step. To overcome this, we establish gradient paths through transitions over time so that the transition becomes the driving factor for shaping the
latent space, rather than adjusting the transition to the recognition model’s latent space. The key is to prevent the recognition model \( q_\phi(z_{1:T} \mid x_{1:T}) \) from directly drawing the latent state \( z_t \).

Similar to the reparameterization trick from [12] for making the Monte Carlo estimate differential w.r.t. the parameters, we make the transition differentiable w.r.t. the last state and its parameters:

\[
z_{t+1} = f(z_t, u_t, \beta_t)
\]  

Given the stochastic parameters \( \beta_t \), the state transition is deterministic (which in turn means that by marginalizing \( \beta_t \), we still have a stochastic transition). The immediate and crucial consequence is that errors in reconstruction of \( x_t \) from \( z_t \) are backpropagated directly through time.

This reparameterization has a couple of other important implications: The recognition model no longer infers latent states \( z_t \), but transition parameters \( \beta_t \). In particular, the gradient \( \partial z_{t+1} / \partial z_t \) is well-defined from (8)—gradient information can be backpropagated through the transition.

This is different to the method used in [14], where the transition is optimized by minimizing a KL divergence. No gradient from the generative model is backpropagated through the transitions.

Much like in Eq. (5), the stochastic parameters includes a corrective offset term \( w_t \), which emphasizes the notion of the recognition model as a filter. In theory, the learning algorithm could still learn the transition as \( z_{t+1} = w_t \). However, the introduction of \( \beta_t \) also enables us to regularize the transition with meaningful priors, which not only prevents overfitting the recognition model, but also enforces meaningful manifolds in the latent space via transition priors. Ignoring the potential of the transition over time yields large penalties from these priors. Thus, the problems outlined in Section 2 are overcome by construction.

To install such transition priors, we split \( \beta_t = (w_t, v_t) \). The interpretation of \( w_t \) is a sample-specific process noise which can be inferred from incoming data, like in Eq. (5). On the other hand, \( v_t \) are universal transition parameters, which are sample-independent (and are only inferred from data during training). This corresponds to the idea of weight uncertainty in [8]. This interpretation implies the factorization of the recognition model:

\[
q_\phi(\beta_{1:T} \mid x_{1:T}) = q_\phi(w_{1:T} \mid x_{1:T}) q_\phi(v_{1:T})
\]  

When using the fully trained model for generative sampling, i.e., sampling without input, the universal state transition parameters can still be drawn from \( q_\phi(v_{1:T}) \), whereas \( w_{1:T} \) is drawn from the prior in the absence of input data.

Fig. (1) shows the underlying graphical model and the inference procedure. Fig. (2a) shows a generic view on our new computational architecture. An example of a locally linear transition parametrization will be given in Section 5.3.
Based on a sampled $\beta$, our experiments show that an annealed version of (10) is beneficial to the overall performance:

$$p(x_{1:T} | u_{1:T}) = \int \int \int \int \int p(\beta_{1:T}) \prod_{t=1}^{T} p(\theta_t | x_t) \prod_{t=0}^{T-1} p(z_{t+1} | z_t, u_t, \beta_t) d\beta_{1:T} d\theta_{1:T}$$

Due to the deterministic transition given $\beta_{t+1}$, the last term is a product of Dirac distributions and the overall distribution simplifies greatly:

$$p(x_{1:T} | u_{1:T}) = \int \int \int \int \int p(\beta_{1:T}) \prod_{t=1}^{T} p(\theta_t | x_t) \prod_{t=0}^{T-1} \delta(z_{t+1} = f(z_{t-1}, u_{t}, \beta_{t+1-1})) d\beta_{1:T}$$

The last formulation is for notational brevity: The term $p_\theta(x_{1:T} | \theta_{1:T})$ is not independent of $\beta_{1:T}$ and $u_{1:T}$. We now derive the objective function, a lower bound to the data likelihood:

$$\ln p(x_{1:T} | u_{1:T}) = \ln \int \int \int \int \int p(\beta_{1:T}) p_\theta(x_{1:T} | z_{1:T}) q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T}) \frac{p(\beta_{1:T})}{q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T})} d\beta_{1:T}$$

$$\geq \int q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T}) \ln \left( p_\theta(x_{1:T} | z_{1:T}) \frac{p(\beta_{1:T})}{q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T})} \right) d\beta_{1:T}$$

$$= \mathbb{E}_{q_\phi} [\ln p_\theta(x_{1:T} | z_{1:T}) - \ln q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T}) + \ln p(\beta_{1:T})]$$

$$= \mathbb{E}_{q_\theta} [\ln p_\theta(x_{1:T} | z_{1:T}) - \mathcal{L}(q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T}) \| p(\beta_{1:T}))$$

Our experiments show that an annealed version of (10) is beneficial to the overall performance:

$$\mathcal{L}_{DVBF}(x_{1:T}, \theta, \phi | u_{1:T}) = \mathbb{E}_{q_\theta} [\ln p_\theta(x_{1:T} | z_{1:T}) - \ln q_\phi(\beta_{1:T} | x_{1:T}, u_{1:T}) + \ln p(\beta_{1:T})]$$

Here, $c_i = \max(1, 0.01 + i/T_A)$ is an inverse temperature that increases linearly in the number of gradient updates $i$ until reaching 1 after $T_A$ annealing iterations. Similar annealing schedules have been applied in, e.g., [6] [15] [17]. Additionally, the transition prior $p(\beta_{1:T})$ was estimated during optimization, i.e., through an empirical Bayes approach. In all experiments, we used a diagonal Gaussian parameterized by its means and variances.
3.3 Example: Locally Linear Transitions

We have derived a learning algorithm for time series with particular focus on transitions in latent space. Inspired by [19], this section will show how to learn locally linear state transitions in latent space. To parametrize the transition, we have to specify Eq. (8). In this case we set

$$\mathbf{z}_{t+1} = \mathbf{A}_t \mathbf{z}_t + \mathbf{B}_t \mathbf{u}_t + \mathbf{C}_t \mathbf{w}_t,$$

where \( \mathbf{w}_t \) is a stochastic sample from the the recognition model and \( \mathbf{A}_t, \mathbf{B}_t, \) and \( \mathbf{C}_t \) are matrices of matching dimensions. They are stochastic functions of \( \mathbf{z}_t \) and \( \mathbf{u}_t \) (thus local linearity). We set \( \mathbf{v}_t = \{ \mathbf{A}_t^{(i)}, \mathbf{B}_t^{(i)}, \mathbf{C}_t^{(i)} \mid i = 1, \ldots, M \} \), i.e., by drawing from \( q_{\phi}(\mathbf{v}_t) \), which is independent of observations, we draw a set of \( 3M \) basis matrices, and finally yield \( \mathbf{A}_t, \mathbf{B}_t, \) and \( \mathbf{C}_t \) as state-dependent and control-dependent linear combinations:

$$\alpha_t = f_\psi(\mathbf{z}_t, \mathbf{u}_t) \in \mathbb{R}^M$$

$$\mathbf{A}_t = \sum_{i=1}^{M} \alpha_t^{(i)} \mathbf{A}_t^{(i)} \quad \mathbf{B}_t = \sum_{i=1}^{M} \alpha_t^{(i)} \mathbf{B}_t^{(i)} \quad \mathbf{C}_t = \sum_{i=1}^{M} \alpha_t^{(i)} \mathbf{C}_t^{(i)}$$

The computation is depicted in Fig. 2(b). The function \( f_\psi \) can be, e.g., a (deterministic) neural network with weights \( \psi \). As a subset of the generative parameters \( \theta, \psi \) is part of the trainable parameters of our model. The weight vector \( \alpha_t \) is shared between the three matrices. There is a correspondence to Eq. (5): \( \mathbf{A}_t, \mathbf{F}_t, \mathbf{B}_t, \) and \( \mathbf{C}_t \), as well as \( \mathbf{C}_t \mathbf{C}_t^T \) and \( \mathbf{Q}_t \) are related.

4 Experiments and Results

In this section we validate that DVBF with locally linear transitions (DVBF-LL) (Section 3.3) outperforms Deep Kalman Filters (DKF, [14]) in recovering latent spaces with full information. We do not include E2C [19] since it necessitates specific data pre-processing and does not provide a correct lower bound. The experimental setup is described in the Supplementary Material.

4.1 Dynamic Pendulum

In order to test our algorithm on truly non-Markovian observations of a dynamical system, we simulated a dynamic torque-controlled pendulum governed by the differential equation

$$ml^2 \ddot{\varphi}(t) = -m\mu \dot{\varphi}(t) + mgl \sin \varphi(t) + u(t),$$

where \( m = l = 1, \mu = 0.5, g = 9.81 \), via numerical integration, and then converted the ground-truth angle \( \varphi \) into an image observation in \( \mathcal{X} \). The one-dimensional control corresponds to angle acceleration (which is proportional to joint torque). Angle and angular velocity fully describe the system.

Fig. 3 shows the latent spaces for identical input data learned by DVBF-LL and DKF, respectively, colored with the ground truth in the top row. It should be noted that latent samples are shown, not means of posterior distributions. The state-space model was allowed to use three latent dimensions. As we can see in Fig. (3a), DVBF-LL learned a two-dimensional manifold embedding, i.e., it encoded the angle in polar coordinates (thus circumventing the discontinuity of angles modulo \( 2\pi \)). The bottom row shows regressions underlining the performance: There exists a high correlation between latent states and ground-truth angle and angular velocity for DVBF-LL. On the contrary, Fig. (3b) verifies our prediction that DKF is equally capable of learning the angle, but extracts little to no information on angular velocity.

The OLS regression results shown in Table (1) validate this observation\footnote{Linear regression is a natural choice: After transforming the ground truth to polar coordinates, an affine transformation should be a good fit for predicting ground truth from latent states. We also tried nonlinear regression with vanilla neural networks. While not being shown here, the results underlined the same conclusion.}. Predicting \( \sin(\varphi) \) and \( \cos(\varphi) \), i.e., polar coordinates of the ground-truth angle \( \varphi \), works almost equally well for DVBF-LL and DKF, with DVBF-LL slightly outperforming DKF. For predicting the ground truth velocity \( \dot{\varphi} \), DVBF-LL shows remarkable performance. DKF, instead, contains hardly any information, resulting in a very low goodness-of-fit score of \( R^2 = 0.035 \).
Figure 3: (a) Our DVBF-LL model trained on pendulum image sequences. The upper plots show the latent space with coloring according to the ground truth with angles on the left and angular velocities on the right. The lower plots show regression results for predicting ground truth from the latent representation. The latent space plots show clearly that all information for representing the full state of a pendulum is encoded in each latent state. (b) DKF from [14] trained on the same pendulum dataset. The latent space plot shows that DKF fails to learn velocities of the pendulum. It is therefore not able to capture all information for representing the full pendulum state.

| Dependent ground truth | DVBF-LL Log-Likelihood | $R^2$ | DKF Log-Likelihood | $R^2$ |
|------------------------|------------------------|-------|-------------------|-------|
| $\sin(\phi)$          | 3990.8                 | 0.961 | 1737.6            | 0.929 |
| $\cos(\phi)$          | 7231.1                 | 0.982 | 6614.2            | 0.979 |
| $\dot{\phi}$          | -11139                 | 0.916 | -20289            | 0.035 |

Table 1: Results for pendulum OLS regressions of all latent states on respective dependent variable.

Fig. (4) shows that the strong relation between ground truth and latent state is beneficial for generative sampling. All plots show 100 time steps of a pendulum starting from the exact same latent state and not being actuated. The top row plots show a purely generative walk in the latent space on the left, and a walk in latent space that is corrected by filtering observations on the right. We can see that both follow a similar trajectory to an attractor. The generative model is more prone to noise when approaching the attractor.

The bottom plot shows the first 45 steps of the corresponding observations (top row), reconstructions (middle row), and generative samples (without correcting from observations). Interestingly, DVBF works very well even though the sequence is much longer than all training sequences (indicated by the red line).

Table 2 shows values of the lower bound to the marginal data likelihood (for DVBF-LL, this corresponds to Eq. (11)). We see that DVBF-LL outperforms DKF in terms of compression, but only with a slight margin, which does not reflect the better generative sampling as [18] argues.

| Lower Bound = Reconstruction Error − KL divergence |
|------------|------------|------------|
| DVBF-LL    | 798.56     | 802.06     | 3.50       |
| DKF        | 784.70     | 788.58     | 3.88       |
4.2 Bouncing Ball

The bouncing ball experiment features a ball rolling within a bounding box in a plane. The system has a two-dimensional control input, added to the directed velocity of the ball. If the ball hits the wall, it bounces off, so that the true dynamics are highly dependent on the current position and velocity of the ball. The system’s state is four-dimensional, two dimensions each for position and velocity.

Consequently, we use a DVBF-LL with four latent dimensions. Fig. 5 shows that DVBF again captures the entire system dynamics in the latent space.
5 Conclusion

We have proposed Deep Variational Bayes Filters (DVBF), a new method to learn state space models from raw non-Markovian sequence data. DVBFs make use of stochastic gradient variational Bayes to overcome intractable inference and thus naturally scale to large data sets. In a series of vision-based experiments we demonstrated that latent states can be recovered which identify the underlying physical quantities. The generative model showed stable long-term predictions far beyond the sequence length used during training.

Acknowledgements

This work has been supported in part by the TACMAN project, EC Grant agreement no. 610967, within the FP7 framework programme.

We would like to thank Jost Tobias Springenberg, Adam Kosiorek, and Moritz Münst for valuable input.

References

[1] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. arXiv preprint arXiv:1411.7610, 2014.

[2] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT’2010, pages 177–186. Springer, 2010.

[3] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. CoRR, abs/1506.02216, 2015.

[4] Marc Deisenroth and Carl E Rasmussen. Pilco: A model-based and data-efficient approach to policy search. In Proceedings of the 28th International Conference on machine learning (ICML-11), pages 465–472, 2011.

[5] Zoubin Ghahramani and Geoffrey E Hinton. Parameter estimation for linear dynamical systems. Technical report, Technical Report CRG-TR-96-2, University of Toronto, Dept. of Computer Science, 1996.

[6] Zoubin Ghahramani and Geoffrey E Hinton. Variational learning for switching state-space models. Neural computation, 12(4):831–864, 2000.

[7] Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

[8] Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In Proceedings of the sixth annual conference on Computational learning theory, pages 5–13. ACM, 1993.

[9] Matthew J Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Structured VAEs: Composing probabilistic graphical models and variational autoencoders. arXiv preprint arXiv:1603.06277, 2016.

[10] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In AeroSense’97, pages 182–193. International Society for Optics and Photonics, 1997.

[11] Rudolph E Kalman and Richard S Bucy. New results in linear filtering and prediction theory. Journal of basic engineering, 83(1):95–108, 1961.

[12] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. arXiv preprint arXiv:1312.1172, 2013.

[13] Jonathan Ko and Dieter Fox. Learning gp-bayesfilters via gaussian process latent variable models. Autonomous Robots, 30(1):3–23, 2011.

[14] Rahul G Krishnan, Uri Shalit, and David Sontag. Deep Kalman filters. arXiv preprint arXiv:1511.05121, 2015.

[15] Stephan Mandt, James McInerney, Farhan Abrol, Rajesh Ranganath, and David Blei. Multicanonical stochastic variational inference. arXiv preprint arXiv:1411.1810, 2014.
[16] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In Tony Jebara and Eric P. Xing, editors, Proceedings of the 31st International Conference on Machine Learning (ICML-14), pages 1278–1286. JMLR Workshop and Conference Proceedings, 2014.

[17] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. arXiv preprint arXiv:1505.05770, 2015.

[18] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844, 2015.

[19] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In Advances in Neural Information Processing Systems, pages 2728–2736, 2015.
A Supplementary to Lower Bound

A.1 Annealed KL-Divergence

We used the analytical solution of the annealed KL-divergence in (10) for optimization:

$$
\mathbb{E}_{q_\phi}[-\ln q_\phi(w_{1:T} \mid x_{1:T}, u_{1:T}) + c_i \ln p(w_{1:T})] = 
\frac{c_i}{2} \ln(2\pi\sigma_p^2) - \frac{1}{2} \ln(2\pi\sigma_q^2) + c_i \frac{\sigma_q^2 + (\mu_q - \mu_p)^2}{2\sigma_p^2} - \frac{1}{2}
$$

B Supplementary to Implementation

B.1 Experimental setup

In all our experiments, we use sequences of 15 raw images of the respective system with $16 \times 16$ pixels each, i.e., observation space $\mathcal{X} \subset \mathbb{R}^{256}$, as well as control inputs of varying dimension and interpretation depending on the experiment. We used training, validation and test sets with 500 sequences each. Control input sequences were drawn randomly (“motor babbling”).

B.2 Implementation details for DVBF in Pendulum Experiment

- Input: 15 timesteps of $16^2$ observation dimensions and 1 action dimension
- Latent Space: 3 dimensions
- Observation Network $p(x_t \mid z_t) = \mathcal{N}(x_t; \mu(z_t), \sigma)$: 128 ReLU + $16^2$ identity output
- Recognition Model: 128 ReLU + 6 identity output
  $$
  q(w_t \mid z_t, x_{t+1}, u_t) = \mathcal{N}(w_t; \mu, \sigma),
  \quad (\mu, \sqrt{\sigma}) = f(z_t, x_{t+1}, u_t)
  $$
- Transition Network $\alpha_t(z_t)$: 16 softmax output
- Initial Network $w_1 \sim \mathcal{N}(x_{1:T})$: Fast Dropout BiRNN with: 128 ReLU + 3 identity output
- Initial Transition $z_1(w_1)$: 128 ReLU + 3 identity output
- Optimizer: adadelta, 0.1 step rate
- Inverse temperature: $c_0 = 0.01$, updated every 10th gradient update, $T_A = 4000$ epochs
- Batch-size: 20

B.3 Implementation details for DVBF in Bouncing Ball Experiment

- Input: 15 timesteps of $16^2$ observation dimensions and 2 action dimension
- Latent Space: 4 dimensions
- Observation Network $p(x_t \mid z_t) = \mathcal{N}(x_t; \mu(z_t), \sigma)$: 128 ReLU + $16^2$ identity output
- Recognition Model: 128 ReLU + 6 identity output
  $$
  q(w_t \mid z_t, x_{t+1}, u_t) = \mathcal{N}(w_t; \mu, \sigma),
  \quad (\mu, \sqrt{\sigma}) = f(z_t, x_{t+1}, u_t)
  $$
- Transition Network $\alpha_t(z_t)$: 16 softmax output
- Initial Network $w_1 \sim \mathcal{N}(x_{1:T})$: Fast Dropout BiRNN with: 128 ReLU + 3 identity output
- Initial Transition $z_1(w_1)$: 128 ReLU + 3 identity output
- Optimizer: adadelta, 0.1 step rate
- Inverse temperature: $c_0 = 0.01$, updated every 10th gradient update, $T_A = 4000$ epochs
- Batch-size: 20
B.4 Implementation details for DKF in Pendulum Experiment

- Input: 15 timesteps of $16^2$ observation dimensions and 1 action dimension
- Latent Space: 3 dimensions
- Observation Network $p(x_t | z_t) = \mathcal{N}(x_t; \mu(z_t), \sigma(z_t))$: 128 Sigmoid + 128 Sigmoid + 2 $16^2$ identity output
- Recognition Model: Fast Dropout BiRNN 128 Sigmoid + 128 Sigmoid + 3 identity output
- Transition Network $p(z_t | z_{t-1}, u_{t-1})$: 128 Sigmoid + 128 Sigmoid + 6 output
- Optimizer: adam, 0.001 step rate
- Inverse temperature: $c_0 = 0.01$, updated every 10th gradient update, $T_A = 2000$ iterations
- Batch-size: 200