Variational Structured Stochastic Network

Hao Liu [*1] **Xinyi Yang** [*2] **Zenglin Xu** [2]

Abstract

High dimensional sequential data exhibits complex structure, a successful generative model for such data must involve highly dependent, structured variables. Thus it is desired or even necessary to model correlations and dependencies between the multiple input, output variables and latent variables in such scenario. To achieve this goal, we introduce Variational Structured Stochastic Network(VSSN), a new method for modeling high dimensional structured data. Leveraging recent advances in Stochastic Gradient Variational Bayes, VSSN can overcome intractable inference distributions via stochastic variational inference(Hoffman et al., 2013; Ranganath et al., 2014). To evaluate the proposed model, we apply it to speech recording data, music data, and several dynamic image sequence modeling tasks. Experimental results have demonstrated that our proposed method can outperform most state-of-the-art methods.

1. Introduction

Learning structured generative models for high dimensional sequential data is a critical yet challenging research topic in machine learning(Eyjolfsdottir et al., 2016; Johnson et al., 2016; Graves, 2013; Watter et al., 2015; Chung et al., 2015). This problem has been studied for many decades using State Space Models(SSMs) such as Hidden Markov Models (HMMs) and Kalman filters (Roweis & Ghahramani, 1999). Another popular method is Recurrent Neural Network(RNN), a recurrent type of neural network, is employed to handle both variable-length inputs and outputs. Some recent work also consider compose both of them for structured prediction(Johnson et al., 2016; Eyjolfsdottir et al., 2016; Fraccaro et al., 2016; Chung et al., 2015). However, for complicated input data, it is difficult for standard SSMs or RNNs to accurately model the underlying dependency structures (Chung et al., 2015; Gu et al., 2015; Gan et al., 2015; Sutskever et al., 2014). Therefore, it is essential to develop learning algorithm that with a good representation of the structure of input variables, output variables and latent states.

To address these issues, we propose an efficient and scalable model called Variational Structured Stochastic Network (VSSN), which enables encode long-term structured dependency in generative model by providing the corresponding inference mechanism with rich capacity.

2. Model

We start by specifying our generative and variational models and then introduce the proposed algorithm.

2.1. Generative model

Consider non-linear dynamical systems with observations $x_t \in X \subset R^{n_x}$, depending on control inputs $u_t \in U \subset R^{n_u}$. Elements of X can be high-dimensional sensory data, e.g., raw images. In particular they may exhibit complex non-Markovian transitions. Corresponding time-discrete sequences of length T are denoted as $x_{1:T} = (x_1, x_2, ..., x_T)$ and $u_{1:T} = (u_1, u_2, ..., u_T)$. We write VSSN as a generative model that temporally interlocks an SSM with a RNN, as illustrated in Figure 1 for a single sequence model. For a single sequence setting, we have

$$\mathcal{L}(\theta) = \log p_\theta(x_{1:T}|u_{1:T}, d_0, z_0), \tag{1}$$

where θ denotes the set of parameters, d_0 and z_0 are initial latent states. It is important to note that when there are N sequences in a dynamic system, we can write the likelihood of the i-th sequence, i.e., $\mathcal{L}_i(\theta)$, in the form of Equation 1 and formulate the whole likelihood as $\mathcal{L}(\theta) = \sum_{i=1}^{N} \mathcal{L}_i(\theta)$. For convenience of presentation, throughout the paper, we omit the index i when only one sequence is referred to, or when it is clear from the context. We set a prior $p(\beta)$ on $\beta_{1:T}$ and write the joint probability of the observation

[*]Equal contribution [1]SMILE Lab & Yingcai Honors College, University of Electronic Science and Technology of China [2]SMILE Lab & Big Data Research Center School of Computer Science and Engineering, University of Electronic Science and Technology of China. Correspondence to: Zenglin Xu <zlxu@gmail.com>.

Proceedings of the ICML 17 Workshop on Deep Structured Prediction, Sydney, Australia, PMLR 70, 2017.

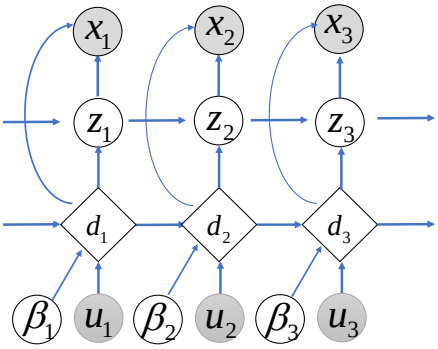

Figure 1. Graphical model of generative network: a recurrent model with state space model. x_t denotes an observation, d_t denotes a hidden state, z_t denotes a label, and the control variable d_t denotes the duration of x_t.

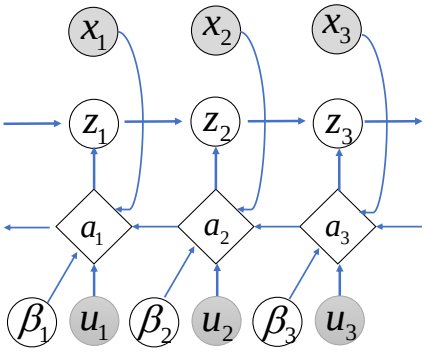

Figure 2. Graphical model of inference network. Inference by backward recurrent function through each time step. x_t denotes an observation, a_t denotes a backward sequence variable.

sequence and its latent states as follows:

$$p_\theta(x_{1:T}, z_{1:T}, d_{1:T}, \beta_{1:T}|u_{1:T}, z_0, d_0) = p_{\theta_x}(x_{1:T}|z_{1:T}, d_{1:T}) \cdot p_{\theta_z}(z_{1:T}|d_{1:T}, z_0) \cdot p_{\theta_d}(d_{1:T}|u_{1:T}, \beta_{1:T}, d_0) \cdot p(\beta_{1:T}) = \prod_{t=1}^{T} p_{\theta_x}(x_t|z_t, d_t) p_{\theta_z}(z_t|z_{t-1}, d_t) p_{\theta_d}(d_t|d_{t-1}, \beta_t, u_t) p(\beta_t),$$

where θ_x, θ_z, and θ_d denote the parameters related to the corresponding conditional distributions. And we have $\theta = \{\theta_x, \theta_z, \theta_d\}$.

The log likelihood term $\mathcal{L}(\theta)$ in Equation 1 could be calculated by averaging out the latent states $z_{1:T}$ and $d_{1:T}$ from Equation 2. Following Figure 1, the states $d_{1:T}$ are determined from d_0, $\beta_{1:T}$ and $u_{1:T}$ through the recursion $d_t = f_{\theta_d}(d_{t-1}, u_t, \beta_t)$. In our implementation f_{θ_d} is a GRU network with parameters θ_d.

We assume $p_{\theta_z}(z_i|z_{t-1}, d_t)$ is subject to a Gaussian distribution with a diagonal covariance structure, namely $p_{\theta_z}(z_t|z_{t-1}, d_t) = N(z_t; \mu_t, v_t)$. In particularly, its mean and log-variance are parameterized by neural networks depending on z_{t-1} and d_t, as follows,

$$\mu_t = f_1(z_{t-1}, d_t), \log v_t = f_2(z_{t-1}, d_t). \tag{2}$$

where $f_i(\cdot)$ for $i = 1, 2$ denotes a neural network, respectively. We split $\beta_t = (w_t, v_t)$, where w_t is the a sample-specific process noise which can be inferred from incoming data, while v_t are universal transition parameters, which are sample-independent (and are only inferred from data during training). Thus it leads to a decomposition as follows: $q_{\phi_2}(\beta_{1:T}|x_{1:T}, u_{1:T}) = q_{\phi_2}(w_{1:T}|x_{1:T}) q_{\phi_2}(u_{1:T})$.

2.2. Inference Network

Instead of maximizing $L(x)$ with respect to θ, we maximize a variational evidence lower bound(ELBO) over $L(x)$, i.e., $F(\theta, \phi) \leq L(x)$, with respect to both θ and the variational parameters ϕ.

$$F(\theta, \phi) = \int\int\int q_\phi(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, S) \cdot \log \frac{p_\theta(x_{1:T}, d_{1:T}, z_{1:T}|S)}{q_\phi(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, S)} \mathrm{d}d_{1:T}\mathrm{d}z_{1:T}\mathrm{d}\beta_{1:T},$$

where $S = \{u_{1:T}, d_0, z_0\}$ denote the set of fixed variables. Maximizing $F(\theta, \phi)$ with parameters θ and ϕ is done by stochastic gradient ascent, and in doing so, both the posterior and its approximation q_ϕ change simultaneously. In general, intractable expectations in the objective function can typically be approximated by the reparameterization trick (Kingma & Welling, 2013; Gu et al., 2015; Chung et al., 2016) or control variates (Paisley et al., 2012) to obtain low-variance estimators of its gradients. In order to obtain efficient solution for $F(\theta, \phi)$, we adopt the reparameterization trick. We add initial structure to q_ϕ by noticing that the prior $p_\theta(z_{1:T}|d_{1:T}, z_0)$ in the generative model is a delta function over the computed $z_{1:T}$, and so is the posterior $p_\theta(z_{1:T}|x_{1:T}, d_{1:T}, z_0)$. Consequently, we let the inference network use exactly the same deterministic state setting $z_{1:T}$ as that of the generative model, and we decompose it as:

$$q_\phi(d_{1:T}, z_{1:T}, \beta_{1:T}|x_{1:T}, u_{1:T}, d_0, z_0) = q(z_{1:T}|x_{1:T}, z_0, d_{1:T}) \cdot q(d_{1:T}|u_{1:T}, x_{1:T}, \beta_{1:T}, d_0) \cdot q(\beta_{1:T}|x_{1:T}, u_{1:T}). \tag{3}$$

Note that $q(d_{1:T}|u_{1:T}, x_{1:T}, \beta_{1:T}, d_0)$ is exactly equals to $p_{\theta_z}(d_{1:T}|u_{1:T}, \beta_{1:T}, z_0)$ based on the generative graphical model Figure 1. Then we substitute Eq. 3 into $F(\theta, \phi)$ and

by some math manipulation:

$$F(\theta,\phi) \geq \int\{\int q_{\phi_2}(\beta|x_{1:T},u_{1:T})\log p_{\theta_z}(x_{1:T}|z_{1:T},d_{1:T})p(\beta_{1:T}) \frac{q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0)q_{\phi_2}(\beta_{1:T}|x_{1:T},u_{1:T})}{q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0)q_{\phi_2}(\beta_{1:T}|x_{1:T},u_{1:T})}\mathrm{d}\beta\}\mathrm{d}z$$

Then we factorize $p_{\theta_z}(x_{1:T}|z_{1:T},d_{1:T})$ to two terms $\sqrt{p_{\theta_z}(x_{1:T}|z_{1:T},d_{1:T})}$ under log and the bound in Eq.(4) can be further optimized as follow:

$$\text{Eq.(4)} \geq \frac{1}{2}E_{q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0)}\{E_{q_{\phi_2}(\beta_{1:T}|x_{1:T},u_{1:T})} [\log p_{\theta_x}(x_{1:T}|z_{1:T},d_{1:T})] - KL(q_{\phi_2}(\beta_{1:T}|x_{1:T},u_{1:T})||p(\beta_{1:T}))\} + \frac{1}{2}E_{q_{\phi_2}(\beta_{1:T}|x_{1:T},u_{1:T})}\{q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0) \int\log\frac{p_{\theta_x}(x_{1:T}|z_{1:T},d_{1:T})}{q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0)}\mathrm{d}z\} := \mathrm{L}(\theta,\phi)$$

Here KL denotes the Kullback-Leibler divergence between two distributions. The lower bound is denoted as $\mathrm{L}(\theta,\phi)$. We denote the first term in $F(\theta,\phi)$ as A while the second term denoted as B for ease of simplification.

The true posterior distribution of the stochastic states $h_{1:T}$, given both the data and the deterministic state $z_{1:T}$, factorizes as:

$$p_\theta(z_{1:T}|d_{1:T},u_{1:T},x_{1:T},z_0) = \prod_t p_\theta(z_t|z_{t-1},d_{t:T},x_{t:T}).$$

This shows that, knowing z_{t-1}, the posterior distribution of z_t does not depend on the past outputs, but only on the present and future ones; this was also noted in (Krishnan et al., 2015; 2016). Instead of factorizing q_ϕ as a mean-field approximation across time steps, we keep the structured form of the posterior factors, including z_ts dependence on z_{t-1}, in the variational approximation:

$$\begin{aligned} q_{\phi_1}(z_{1:T}|d_{1:T},x_{1:T},z_0) &= \prod_t q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T}) \\ &= \prod_t q_{\phi_1}(z_t|z_{t-1},a_t), \end{aligned} \tag{4}$$

where $a_t = g_{\phi_a}(a_{t+1},[d_t,x_t])$ and $[d_t,x_t]$ is the concatenation of the vectors z_t and x_t. We mimic each posterior factor's nonlinear long-term dependence on $d_{t:T}$ and $x_{t:T}$ through a backwards-recurrent function g_{ϕ_a}. The inference network is therefore parameterized by $\phi = \{\phi_1,\phi_2,\phi_a\}$. The inference procedure is illustrated in Figure 2.

As both the generative model and inference network factorize over time steps, the ELBO separates as a sum over the time steps:

$$B = \sum_t \mathrm{E}_{q^\star_{\phi_1}(z_{t-1})}[\mathrm{E}_{q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T})}[\log p_\theta(x_t|z_t,d_t)] - \mathrm{KL}(q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T})||p_\theta(z_t|z_{t-1},d_t))].$$

Here $q^\star_{\phi_1}(z_{t-1})$ denotes the marginal distribution of h_{t-1} in the variational approximation to the posterior $q_{\phi_1}(z_{1:t-1}|d_{1:T},x_{1:T},z_0)$. Substituting the above equation into $F(\theta,\phi)$ we have:

$$\begin{aligned} L(\theta,\phi) =& \frac{1}{2}\sum_t \mathrm{E}_{q^\star_{\phi_1}(z_{t-1})}\{\mathrm{E}_{q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T})} [\sum_t \mathrm{E}_{q_{\phi_2}(\beta_t|x_t,u_t)}[\log p_\theta(x_t|z_t,d_t) \\ & - KL(q_{\phi_2}(\beta_t|x_t,u_t)||p(\beta_t))]]\} \\ & + \frac{1}{2}\sum_t \mathrm{E}_{q_{\phi_2}(\beta_t|x_t,u_t)}\{\sum_t \mathrm{E}_{q^\star_{\phi_1}(z_{t-1})} [\mathrm{E}_{q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T})}[\log p_\theta(x_t|z_t,d_t)] \\ & - \mathrm{KL}(q_{\phi_1}(z_t|z_{t-1},d_{t:T},x_{t:T})||p_\theta(z_t|z_{t-1},d_t))]\} \end{aligned}$$

3. Experiment

3.1. Synthetic Experiment

To validate that VSSN is able to model high dimensional data with complex dependency, we simulated a dynamic torque-controlled pendulum governed by the differential equation to test VSSN on non-Markovian observations of a dynamical system: $ml^2\frac{d^2\phi(t)}{dt^2} = -\mu\frac{d\phi(t)}{dt} + mgl\sin\phi(t) + u(t)$. For fair comparison with (Karl et al., 2016), we set $m = l = 1$, $\mu = 0.5$, $g = 9.81$, via numerical integration, and then converted the ground-truth angle into an image observation. The one-dimensional control corresponds to angle acceleration. Angle and angular velocity fully describe the system. The OLS regression results are shown in Table 2.2, VSSN is clearly better than DVBF-LL and DKF in predicting $\sin\phi$, $\cos\phi$ and $\frac{d\phi}{dt}$. VSSN achieves a higher goodness-of-fit than other methods.

3.2. Speech modelling

We also evaluate VSSN on the modelling of speech data, i.e., Blizzard and TIMIT datasets. Here Blizzard is a dataset of 300 hours of English speech by a single female speaker and TIMIT is a dataset of 6300 English sentences read by 630 speakers.This is a challenging task since they have shown to be difficult to model without a good representation of the uncertainty in the latent states (Chung et al., 2015; Gu et al., 2015; Gan et al., 2015; Sutskever et al., 2014).We report the results in Table 3 . It can be seen that VSSN performs slightly better than SRNN(smooth+Res_q) and other methods on TIMIT, while outperforms current state-of-the-art methods on Blizzard by a large margin.

3.3. Bouncing Balls

The bouncing balls dataset is a common test set for models that generate high dimensional sequences. It consists

	DVBF-LL		DKF		VSSN	
	ll	R^2	ll	R^2	ll	R^2
$\sin\phi$	3990.8	0.961	1737.6	0.929	4304.4	0.973
$\cos\phi$	7231.1	0.982	6614.2	0.979	8021.3	0.991
$\frac{d\phi}{dt}$	-11139	0.916	-20289	0.035	-9731	0.930

Table 1. The results measured on the log-likelihood(denoted as ll) and the goodness-of-fit (R^2) given by three methods on the prediction of all latent states on respective dependent variables in pendulum dynamics. For both measures, the higher the better.

Models	Error
DTSBN	2.79 ± 0.39
SRTRBM	3.31 ± 0.33
PGN(MSE)	$\mathbf{0.65 \pm 0.11}$
TSBN	3.07 ± 0.40
VSSN	0.74 ± 0.25

Table 2. Average prediction error for the bouncing balls dataset.

MODELS	Blizzard	TIMIT
VRNN-GMM	≥ 9107	≥ 28982
VRNN-GAUSS	≥ 9223	≥ 28805
VRNN-I-GAUSS	≥ 9223	≥ 28805
SRNN(smooth+Res_q)	≥ 11991	≥ 60550
SRNN(smooth)	≥ 10991	≥ 59269
SRNN(filt)	≥ 10846	50524
RNN-GMM	7413	26643
RNN-GAUSS	3539	-1900
VSSN	$\mathbf{\geq 14141}$	$\mathbf{\geq 65334}$

Table 3. Comparision between different models on Blizzard and TIMIT speech datasets. Among them, results of SRNN cited from (Fraccaro et al., 2016), VRNN from (Chung et al., 2015).

MODELS	Nottingham	JSB chorales	MuseData	Piano-middle
SRNN	≥ -2.94	≥ -4.74	≥ -6.28	≥ -8.20
TSBN	≥ -3.67	≥ -7.48	≥ -6.83	≥ -7.94
NASMC	≈ -2.71	$\approx \mathbf{-3.98}$	≈ -6.88	≈ -7.62
STORN	≈ -2.85	≈ -6.93	≈ -6.17	≈ -7.15
RNN-NADE	$\approx \mathbf{-2.31}$	≈ -5.19	$\approx \mathbf{-5.60}$	$\approx \mathbf{-7.05}$
RNN	≈ -4.45	≈ -8.72	≈ -8.11	≈ -8.34
DMM	≈ -2.770	≈ -6.388	≈ -6.831	≈ -7.835
HMSBN	≥ -7.98	≥ -5.13	≥ -9.79	≥ -8.90
VSSN	≥ -2.344	≥ -4.021	≥ -5.725	≥ -7.130

Table 4. Evaluation against Baselines on Polyphonic Music Generation dataset. Compared results are cited from their original papers: HMSBN, TSBN(Gan et al., 2015), NASMC (Gu et al., 2015), STORN (Bayer & Osendorfer, 2014) , RNN-NADE & RNN (Boulanger-Lewandowski et al., 2012), SRNN (Fraccaro et al., 2016), DMM (Krishnan et al., 2016).

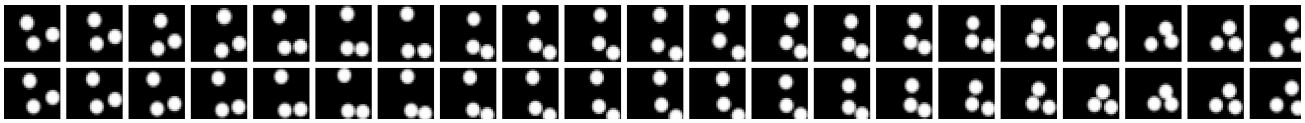

Figure 3. Visualization of the three balls dynamic modelling experiment. The first eleventh columns are 17 consecutive timesteps of training and the later 6 columns are 6 consecutive testing timesteps. At each timestep, the above one is the groundtruth and the bottom one is the corresponding prediction generated by VSSN.

of simulations of three balls bouncing in a box. We followed standard procedure to create 4000 training videos and 200 testing videos (Sutskever et al., 2009; Gan et al., 2015) and used an additional 200 videos for validation. It features three ball rolling within a bounding box in a plane. If the ball hits the wall and another ball, it bounces off, so that the true dynamics are highly dependent on the current position and velocity of the ball. Each video is of length 100 and of resolution 30×30. As can be seen, the model is able both to represent the ball almost accurately and to make long-term predictions while modelling uncertainty. VSSN outperforms the Deep Temporal Sigmoid Belief Network (Gan et al., 2015), the recurrent temporal RBM (RTRBM) and the structured RTRBM (SRTRBM)(Mittelman et al., 2014). VSSN also compete favorably with the Predictive Generative Networks(PGN) (Lotter et al., 2015) although PGN is a rather complex deep neural network.Results shown in Table 2.2. An example of prediction sequence is shown in Figure 3.

3.4. Polyphonic Music

Additionally, we test VSSN for modelling sequences of polyphonic music, using the four data sets of MIDI songs introduced. Each data set contains more than 7 hours of polyphonic music of varying complexity: folk tunes (Nottingham data set), the four-part chorales by J. S. Bach (JSB chorales), orchestral music (MuseData) and classical piano music (Piano-midi.de). Table 4 compares the average log-likelihood on the test sets obtained with the models introduced in (Bayer & Osendorfer, 2014; Gan et al., 2015; Gu et al., 2015). It can be seen that VSSN performs slightly better or by a large margin than other methods.

4. Conclusion

To learn a generative model for high dimensional sequential data, Variaitonal Structured Stochastic Network is introduced here, which is a novel model for learning a good representation of the structure of input, output and latent variables.

References

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

Boulanger-Lewandowski, Nicolas, Bengio, Yoshua, and Vincent, Pascal. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. In *ICML*, 2012.

Chung, Junyoung, Kastner, Kyle, Dinh, Laurent, Goel, Kratarth, Courville, Aaron C, and Bengio, Yoshua. A recurrent latent variable model for sequential data. In *Advances in neural information processing systems*, pp. 2980–2988, 2015.

Chung, Junyoung, Ahn, Sungjin, and Bengio, Yoshua. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

Eyjolfsdottir, Eyrun, Branson, Kristin, Yue, Yisong, and Perona, Pietro. Learning recurrent representations for hierarchical behavior modeling. *CoRR*, abs/1611.00094, 2016.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

Gan, Zhe, Li, Chunyuan, Henao, Ricardo, Carlson, David E, and Carin, Lawrence. Deep temporal sigmoid belief networks for sequence modeling. In *Advances in Neural Information Processing Systems*, pp. 2467–2475, 2015.

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

Gu, Shixiang, Ghahramani, Zoubin, and Turner, Richard E. Neural adaptive sequential monte carlo. In *Advances in Neural Information Processing Systems*, pp. 2629–2637, 2015.

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

Johnson, Matthew, Duvenaud, David K, Wiltschko, Alex, Adams, Ryan P, and Datta, Sandeep R. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, pp. 2946–2954, 2016.

Karl, Maximilian, Soelch, Maximilian, Bayer, Justin, and van der Smagt, Patrick. Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*, 2016.

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

Krishnan, Rahul G, Shalit, Uri, and Sontag, David. Deep kalman filters. *arXiv preprint arXiv:1511.05121*, 2015.

Krishnan, Rahul G, Shalit, Uri, and Sontag, David. Structured inference networks for nonlinear state space models. *arXiv preprint arXiv:1609.09869*, 2016.

Lotter, William, Kreiman, Gabriel, and Cox, David. Unsupervised learning of visual structure using predictive generative networks. *arXiv preprint arXiv:1511.06380*, 2015.

Mittelman, Roni, Kuipers, Benjamin, Savarese, Silvio, and Lee, Honglak. Structured recurrent temporal restricted boltzmann machines. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1647–1655, 2014.

Paisley, John, Blei, David, and Jordan, Michael. Variational bayesian inference with stochastic search. *arXiv preprint arXiv:1206.6430*, 2012.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David M. Black box variational inference. In *AISTATS*, 2014.

Roweis, Sam and Ghahramani, Zoubin. A unifying review of linear gaussian models. *Neural computation*, 11(2): 305–345, 1999.

Sutskever, Ilya, Hinton, Geoffrey E, and Taylor, Graham W. The recurrent temporal restricted boltzmann machine. In *Advances in Neural Information Processing Systems*, pp. 1601–1608, 2009.

Sutskever, Ilya, Vinyals, Oriol, and Le, Quoc V. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pp. 3104–3112, 2014.

Watter, Manuel, Springenberg, Jost, Boedecker, Joschka, and Riedmiller, Martin. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems*, pp. 2746–2754, 2015.