Deep Kalman Filters

Rahul G. Krishnan
Courant Institute of Mathematical Sciences
New York University

Uri Shalit
Courant Institute of Mathematical Sciences
New York University

David Sontag
Courant Institute of Mathematical Sciences
New York University

Abstract

Kalman Filters are some of the most influential models of time-varying phenomena. They admit an intuitive probabilistic interpretation, have a simple functional form, and enjoy widespread adoption in a variety of disciplines. Motivated by recent variational methods for learning deep generative models, we introduce a unified algorithm to efficiently learn a broad spectrum of Kalman filters. Of particular interest is the use of temporal generative models for counterfactual inference. We investigate the efficacy of such models for counterfactual inference, and to that end we introduce the "Healing MNIST" dataset where long-term structure, noise and actions are applied to sequences of digits.

1 Introduction

Electronic Health Records (EHRs) are collected nationwide and machine learning is increasingly used to discover patterns within them. A patient record may be viewed as a sequence of diagnoses, surgeries, laboratory values and drugs prescribed over time. These records yield the potential for machine learning to answer medical queries: What is the best course of treatment for a patient? Which of two drugs will save a patient? Can we find patients who are "similar" to each other? We introduce new techniques for learning generative temporal models from noisy high-dimensional data, and use the learned models within a causal framework, the first step towards addressing such questions. We learn a representation of the patient that (1) evolves over time and (2) is sensitive to the effect of the actions taken by doctors.

We show that recent techniques in variational inference [15, 10] can be adopted to learn a broad set of Kalman Filters [8] with a single algorithm. Using deep neural networks, we can enhance Kalman Filters with arbitrarily complex transition dynamics and emission distributions. We evaluate our model in two settings. First we introduce "Healing MNIST", a dataset of perturbed, noisy and rotated MNIST digits. We show our model captures both short- and long-range effects of actions performed on these digits. Second, we use EHR data from $8,000$ diabetic and pre-diabetic patients gathered over 4.5 years (similar to [19]). We present here the results on "Healing MNIST". The results on the EHR data will be presented in the full version of this paper [1].

Related Work We point the reader to [5] for a summary of some approaches to learn Kalman Filters. [4] learn a sequential model over multiple observations using an attention mechanism. [2] model sequences of length T using T variational autoencoders. They use a single Recurrent Neural Network (RNN) that share parameters. Earlier instances of learning Kalman Filters with Multi-Layer Perceptrons are considered by [14]. They approximate the posterior using non-linear dynamic factor analysis [18], which scales quadratically with the latent dimension. Closest to our work is that

[1] Full Version: `http://arxiv.org/abs/1511.05121`

of [21] who use temporal generative models for optimal control using a training algorithm based on maximizing the likelihood of consecutive pairs occurring within the sequence.

2 Background

Kalman Filters Assume we have a sequence of unobserved variables $z_1, \ldots, z_T \in \mathbb{R}^s$. For each unobserved variable z_t we have a corresponding *observation* $x_t \in \mathbb{R}^d$, and a corresponding *action* $u_t \in \mathbb{R}^c$, which is also observed. In the medical domain, the variables z_t might denote the true state of a patient, the observations x_t indicate known diagnoses and lab test results, and the actions u_t correspond to prescribed medications and medical procedures which aim to change the state of the patient. The classical Kalman Filter models the observed sequence $x_1, \ldots x_T$ as follows:

$$z_t = G_t z_{t-1} + B_t u_{t-1} + \epsilon_t \textit{ (action-transition)} \qquad x_t = F_t z_t + \eta_t \textit{ (observation)},$$

where $\epsilon_t \sim \mathcal{N}(0, \Sigma_t)$, $\eta_t \sim \mathcal{N}(0, \Gamma_t)$ are zero-mean i.i.d. normal random variables, with covariance matrices which may vary with t. In the next section, we show how to replace all the linear transformations with non-linear transformations parameterized by neural nets, and how to overcome the resulting intractability of posterior inference.

Stochastic backpropagation In order to overcome the intractability of posterior inference during learning, we make use of recently introduced variational autoencoders [15, 10] to optimize a variational lower bound on the model log-likehood. The key technical innovation is the introduction of a *recognition network* (denoted q_ϕ), a neural network parameterized by ϕ which approximates the intractable posterior in the standard variational formulation. The challenge in the resulting optimization problem is that the lower bound includes an expectation w.r.t. q_ϕ, which implicitly depends on the network parameters ϕ. This difficulty is overcome by using *stochastic backpropagation*: assuming that the latent state is normally distributed $q_\phi(z|x) \sim \mathcal{N}(\mu_\phi(x), \Sigma_\phi(x))$, a simple transformation allows one to take stochastic gradients of $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)]$ with respect to ϕ.

Counterfactual estimation Counterfactual estimation is the task of inferring the probability of a result given different circumstances than those empirically observed. For example, in the medical setting, one is often interested in questions such as "What would the patient's blood sugar level be had she taken a different medication?". Knowing the answers to such questions could lead to better and more efficient healthcare. We are interested in providing better answers to this type of questions, by leveraging the power of large-scale Electronic Health Records. [13] framed the problem of counterfactual estimation in the language of graphical models and *do*-calculus. If one knows the graphical model of the variables in question, then for some structures estimation of counterfactuals is possible by setting a variable of interest (e.g. medication prescribed) to a given value and performing inference on a derived sub-graph.

3 Model

Our goal is to fit a generative model to a sequence of observations and actions, motivated by the nature of patient health record data. Denote the sequence of observations $\vec{x} = (x_1, \ldots, x_T)$ and actions $\vec{u} = (u_1, \ldots, u_{T-1})$, with corresponding latent states $\vec{z} = (z_1, \ldots, z_T)$. As previously, we assume that $x_t \in \mathbb{R}^d$, $u_t \in \mathbb{R}^c$, and $z_t \in \mathbb{R}^s$. The generative model for the deep Kalman Filter is then given by:

$$z_1 \sim \mathcal{N}(\mu_0; \Sigma_0) \quad z_t \sim \mathcal{N}(G_\alpha(z_{t-1}, u_{t-1}, \Delta_t), S_\beta(z_{t-1}, u_{t-1}, \Delta_t)) \quad x_t \sim \Pi(F_\kappa(z_t)). \qquad (1)$$

Specifically, the functions $G_\alpha, S_\beta, F_\kappa$ are assumed to be parameterized by deep neural networks. We set $\mu_0 = 0$, $\Sigma_0 = I_d$, and therefore we have that $\theta = \{\alpha, \beta, \kappa\}$ are the parameters of the generative model. We use a diagonal covariance matrix $S_\beta(\cdot)$, and employ a log-parameterization, thus ensuring that the covariance matrix is positive-definite.

The key point here is that (1) subsumes a large family of linear and non-linear latent space models. By restricting the functional forms of $G_\alpha, S_\beta, F_\kappa$, we can train different kinds of Kalman Filters within the framework we propose. For example, by setting $G_\alpha(z_{t-1}, u_{t-1}) = G_t z_{t-1} + B_t u_{t-1}, S_\beta = \Sigma_t, F_\kappa = F_t z_t$ where G_t, B_t, Σ_t, F_t are matrices, we can learn classical Kalman Filters. Within the framework we propose any parametric differentiable function can be substituted in for one of $G_\alpha, S_\beta, F_\kappa$. Learning such a model can be done using backpropagation as will be detailed in the next section.

Learning using Stochastic Backpropagation We aim to fit the generative model (see Figure 1a) parameters θ which maximize the conditional likelihood of the data given the external actions, i.e. we desire $\max_\theta \log p_\theta(x_1 \ldots, x_T | u_1 \ldots u_{T-1})$. Using the variational principle, we maximize a lower bound on the log-likelihood (denoted $\mathcal{L}$) of the observations $\vec{x}$ conditioned on the actions. We derive an extension of [15, 10] to the the temporal setting where we use the factorization of the prior implied by (1) and an approximation to $q_\phi(\vec{z}|\vec{x}, \vec{u})$ that decomposes with time. We condition q_ϕ not just on the inputs $\vec{x}$ but also on the actions $\vec{u}$. We bound the conditional likelihood by (see supplementary for the full derivation):

$$\mathcal{L} = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{z_t} \left[\log p_\theta(x_t|z_t)\right] - \mathrm{KL}(q_\phi(z_1)||p_0(z_1)) - \sum_{t=2}^{T} \mathop{\mathbb{E}}_{z_{t-1}} \left[\mathrm{KL}(q_\phi(z_t|z_{t-1})||p_0(z_t|z_{t-1}, u_{t-1}))\right]. \tag{2}$$

Equation (2) is differentiable in the parameters of the model (θ, ϕ), and we can apply backpropagation for updating θ and the stochastic backpropagation trick for obtaining a Monte-Carlo estimate of the gradient of the expectation terms w.r.t. ϕ.

4 Experimental Section

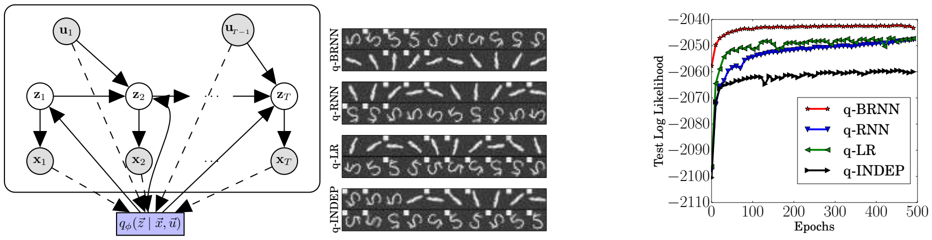
We implement and train models in Torch [3] using ADAM [9]. In the experiments that follow, we fix the generative model as follows: G_α is a two-layer Multi-layer perceptron (MLP), S_β is a constant, learned diagonal matrix, F_κ is a four-layer MLP. Our code is implemented to parameterize $\log S_\beta$ during learning. For the sequential variational model q_ϕ we use a two-layer Long-Short Term Memory Recurrent Neural Net (LSTM-RNN)[22].

Introducing Healing MNIST Healthcare data exhibits diverse structural properties. Surgeries and drugs vary in their effect as a function of age, gender and ethnicity. Lab measurements are noisy, and diagnoses may be tentative, redundant or delayed. In health claims data, the situation is further complicated by arcane, institutional specific practices that determine how decisions by doctors are repurposed into codes used for reimbursements.

To mimic learning under such harsh conditions, we consider a synthetic dataset derived from the MNIST Handwritten Digits [11]. We create a dataset where rotations are performed to the digits. The rotations are encoded as the actions ($\vec{u}$) and the rotated images as the observations ($\vec{x}$). This realizes a sequence of rotated images. To each such generated training sequence, exactly one sequence of three consecutive squares is superimposed on the top-left corner of the images in a random starting location. Finally, we consider learning under 20% bit-flip noise. We consider two experiments: "Small Healing MNIST"(40000 sequences of length 5 of a single example of 1 and 5), "Large Healing MNIST" (140000 sequences of length 5 with 200 different 1's and 5's). The large dataset represents the temporal evolution of two distinct subpopulations of patients (of size 100 each). The squares within the sequences are intended to be analogous to seasonal flu or other ailments that a patient could exhibit which are independent of the actions and last several timesteps.

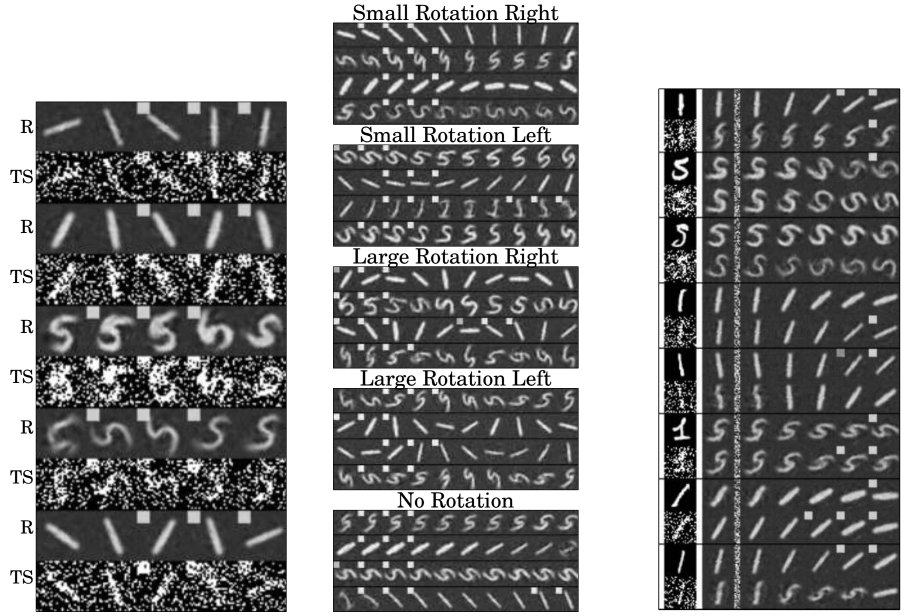
Figure 2a shows examples of training sequences (marked **TS**) from "Large Healing MNIST" provided to the model, and their corresponding reconstructions (marked **R**) representing mean probabilities output by the model.

Comparing Recognition Models Using "Small Healing MNIST" we evaluated the impact of different variational models on learning, by examining test log-likelihood and by visualizing the samples generated by the models. We experiment with four choices of variational models of increasing complexity: **q-INDEP** where $q(z_t|x_t)$ is parameterized by an MLP, **q-LR** where $q(z_t|x_{t-1}, x_t, x_{t+1})$ is parameterized by an MLP, **q-RNN** where $q(z_t|x_1, \ldots, x_t)$ is parameterized by an RNN, and **q-BRNN** where $q(z_t|x_1, \ldots, x_T)$ is parameterized by a bi-directional RNN. Figures 1b and 1c depict test log likelihood and samples from the models trained using different recognition networks. Unsurprisingly, the Bidirectional LSTM RNN, a model capable of summarizing the past and future while approximating the posterior in a manner similar to the Forward-Backward algorithm, outperforms the others in log-likelihood and samples.



(a) (b) (c)

Figure 1: (a) Graphical Model of the Deep Kalman Filter. “Small Healing MNIST”: (b) Mean probabilities sampled under different variational models with a constant, large rotation applied to the right. (c) Test log-likelihood under different recognition models.



(a) (b) (c)

Figure 2: “Large Healing MNIST”. (a) Pairs of Training Sequences (TS) and Mean Probabilities of Reconstructions (R) shown above. (b) Mean probabilities sampled from the model under different, constant rotations. (c) Counterfactual Reasoning. We reconstruct variants of the digits $5, 1$ *not* present in the training set, with (bottom) and without (top) bit-flip noise. We infer a sequence of 2 timesteps and display the reconstructions from the posterior. We then keep the latent state and perform forward sampling and reconstruction from the generative model under a constant right rotation.

Results on “Large Healing MNIST” Figure 2a (left) depicts pairs of training sequences, and the mean probabilities obtained after reconstruction, as learning progresses. The reconstructions show that the model learns different styles of the digits (corresponding to variations within individual patients). Figure 2b has samples under varying degrees of rotation, corresponding for example to the intensity of a treatment. The model shows that it is capable of learning variations within the digit, as well as realizing the effect of the action and its intensity.

Figure 2c shows what happens when we ask the model to reconstruct on data which from a previously unseen test set. The input image is on the left (with a clean and noisy version of the digit displayed) and the following sample represents the reconstruction by the variational model from the input images. Following this, we forward sample from the model using the inferred latent representation under a constant action. This idea has parallels within the medical setting where one asks about the course of action for a new patient. On this unseen patient, the model would infer a latent state similar to one that exists in the training set. To simulate the medical question: The consequent samples mimic a response to the question, what would happen if the doctor prescribed the drug “rotate right mildly” to the new digit at hand.

References

[1] Léon Bottou, Jonas Peters, Joaquin Quinonero-Candela, Denis X Charles, D Max Chickering, Elon Portugaly, Dipankar Ray, Patrice Simard, and Ed Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research*, 14(1):3207–3260, 2013.

[2] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *arXiv preprint arXiv:1506.02216*, 2015.

[3] Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[4] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015.

[5] Simon Haykin. *Kalman filtering and neural networks*, volume 47. John Wiley & Sons, 2004.

[6] M Höfler. Causal inference based on counterfactuals. *BMC medical research methodology*, 5(1):28, 2005.

[7] Andrew H Jazwinski. *Stochastic processes and filtering theory*. Courier Corporation, 2007.

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

[9] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[11] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2010.

[12] Stephen L Morgan and Christopher Winship. *Counterfactuals and causal inference*. Cambridge University Press, 2014.

[13] Judea Pearl. *Causality*. Cambridge university press, 2009.

[14] Tapani Raiko and Matti Tornio. Variational bayesian learning of nonlinear hidden state-space models for model predictive control. *Neurocomputing*, 72(16):3704–3712, 2009.

[15] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

[16] Paul R Rosenbaum. *Observational studies*. Springer, 2002.

[17] Sam Roweis and Zoubin Ghahramani. An EM algorithm for identification of nonlinear dynamical systems. 2000.

[18] Harri Valpola and Juha Karhunen. An unsupervised ensemble learning method for nonlinear dynamic state-space models. *Neural computation*, 14(11):2647–2692, 2002.

[19] Finale Doshi Velez. Partially-observable markov decision processes as dynamical causal models. 2013.

[20] Eric Wan, Ronell Van Der Merwe, et al. The unscented kalman filter for nonlinear estimation. In *Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000*, pages 153–158. IEEE, 2000.

[21] Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *arXiv preprint arXiv:1506.07365*, 2015.

[22] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *arXiv preprint arXiv:1410.4615*, 2014.

A Related Work

Modelling temporal data is a widely studied problem in machine learning. Models such as the Hidden Markov Models (HMM), Dynamic Bayesian Networks (DBN), and Recurrent Neural Networks (RNN) have been proposed.Here, we consider a widely used probabilistic model: Kalman Filters [8]. In classical Kalman Filters, the latent state evolution as well as the emission distribution and external effects are modelled as linear functions perturbed by Gaussian noise. For real world applications the use of linear transition and emission distribution limits the capacity to model complex phenomena, and modifications to the functional form of Kalman Filters have been proposed. For example, the Extended Kalman Filter [7] and the Unscented Kalman Filter [20] are two different methods to learn temporal models with non-linear transition and emission distributions (see also [17] and [5]).

The literature on sequential modeling and Kalman Filters is vast and here we review some of the relevant work on the topic with particular emphasis on recent work in machine learning.

[2] model sequences of length T using T variational autoencoders. They use a single Recurrent Neural Network (RNN) that (1) shares parameters in the inference and generative network and (2) models the prior and approximation to the posterior at time $t \in [1, \dots T]$ as a deterministic function of the hidden state of the RNN. There are a few key differences between their work and ours. (1) they do not model the effect of external actions on the data, (2) their choice of model ties together inference and sampling from the model whereas we consider decoupled generative and recognition networks, and (3) The time varying "memory" of their resulting generative model is both deterministic and stochastic whereas ours is entirely stochastic. i.e our model retains the Markov Property and other conditional independence statements held by Kalman Filters.

This latter property means that [2]'s method cannot be readily adopted for counterfactual inference, since there is no clean way of letting interventions persist in the model.

Early instances of learning Kalman Filters with Multi-Layer Perceptrons was considered by [14]. They approximate the posterior using non-linear dynamic factor analysis [18], which scales quadratically with the latent dimension. Closest to our work is that of [21] who use temporal generative models for optimal control. While [21] aim to learn a locally linear latent dimension within which to perform optimal control, our goal is different: we wish to model the data in order to perform counterfactual inference. Their training algorithm relies on approximating the bound on the likelihood by training on consecutive pairs of images.

In broad strokes, our work extends that of [21] to training with arbitrarily long sequences. The factorization of the prior and posterior, also made use of in [2], enables us to retain a tractable bound on the log likelihood. By varying the functional form of $G_\alpha, S_\beta, F_\kappa$, we can learn different variants of Kalman Filters using the same algorithm.

In general, control applications deal with domains where the effect of action is instantaneous, unlike in the medical setting. In addition, most control scenarios involve a setting such as controlling a robot arm where the control signal has a major effect on the observation; we contrast this with the medical setting where medication can often have a weak impact on the patient's state, compared with endogenous and environmental factors.

There is a vast literature about estimating expected counterfactual effects over a population - see [12, 6, 16] for overviews. Another line of work exists in the computational advertising literature, when one is often interested in more specific counterfactuals such as "how would the page-views change if I had used a different advertisement". [1]model a complex machine learning and ad-placement system, for which much of the causal structure is known. They are able to derive estimates and confidence intervals for counterfactual questions pertaining to the system.

B Lower Bound on Likelihood

Figure 1a depicts both the graphical model and the variational approximation to the posterior. We derive the lower bound on the likelihood of the data.

$$\log p_\theta(\vec{x}|\vec{u}) \geq$$

(Jensen's Inequality)

$$\int_{\vec{z}} q_\phi(\vec{z}) \log \frac{p_0(\vec{z}|\vec{u}) p_\theta(\vec{x}|\vec{z},\vec{u})}{q_\phi(\vec{z})} \tilde{\mathrm{z}} =$$

$$\mathop{\mathbb{E}}_{q_\phi(\vec{z})} [\log p_\theta(\vec{x}|\vec{z},\vec{u})] - \mathrm{KL}(q_\phi(\vec{z}) || p_0(\vec{z}|\vec{u})) \geq$$

(Using $x_t \perp\!\!\!\perp x_{\neg t}|\vec{z}$)

$$\sum_{t=1}^{T} \mathop{\mathbb{E}}_{q_\phi(z_t)} [\log p_\theta(x_t|z_t, u_{t-1})] - \mathrm{KL}(q_\phi(\vec{z}) || p_0(\vec{z}|\vec{u})).$$

We can show that the KL divergence between the approximation to the posterior and the prior simplifies as:

$$\begin{aligned}
&\mathrm{KL}(q(z_1,\ldots,z_T)||p(z_1,\ldots,z_T))\\
&\quad=\int_{z_1}\ldots\int_{z_T} q(z_1)\ldots q(z_T)\log\frac{p(z_1,z_2,\ldots,z_T)}{q(z_1)\ldots q(z_T)}\\
&\quad\textit{(Factorization of the variational distribution)}\\
&\quad=\int_{z_1}\ldots\int_{z_T} q(z_1)\ldots q(z_T)\\
&\quad\log\frac{p(z_1)p(z_2|z_1,u_1)\ldots p(z_T|z_{T-1},u_{T-1})}{q(z_1)\ldots q(z_T)}\\
&\quad\textit{(Factorization of the prior)}\\
&\quad=\int_{z_1}\ldots\int_{z_T} q(z_1)\ldots q(z_T)\log\frac{p(z_1)}{q(z_1)}\\
&\quad+\sum_{t=2}^{T}\int_{z_1}\ldots\int_{z_T} q(z_1)\ldots q(z_T)\log\frac{p(z_t|z_{t-1})}{q(z_t)}\\
&\quad=\int_{z_1} q(z_1)\log\frac{p(z_1)}{q(z_1)}+\sum_{t=2}^{T}\int_{z_{t-1}}\int_{z_t} q(z_t)\log\frac{p(z_t|z_{t-1})}{q(z_t)}\\
&\quad\textit{(Each expectation over z_t is constant for $t\notin\{t,t-1\}$)}\\
&\quad=\mathrm{KL}(q(z_1)||p(z_1))\\
&\quad+\sum_{t=2}^{T-1}\mathop{\mathbb{E}}_{q(z_{t-1})}\left[\mathrm{KL}(q(z_t)||p(z_t|z_{t-1},u_{t-1}))\right]
\end{aligned} \tag{3}$$

For evaluating the marginal likelihood on the test set, we can use the following Monte-Carlo estimate:

$$p(\vec{x}) \approxeq \frac{1}{S}\sum_{s\ =1}^{S}\frac{p(\vec{x}|\vec{z}^{(s)})p(\vec{z}^{(s)})}{q(\vec{z}^{(s)}|\vec{x})} \quad \vec{z}^{(s)}\sim q(\vec{z}|\vec{x}) \tag{4}$$

This may be derived in a manner akin to the one depicted in Appendix E [15] or Appendix D [10].

The log likelihood on the test set is computed using:

$$\log p(\vec{x}) \approxeq \log\frac{1}{S}\sum_{s\ =1}^{S}\exp\log\left[\frac{p(\vec{x}|\vec{z}^{(s)})p(\vec{z}^{(s)})}{q(\vec{z}^{(s)}|\vec{x})}\right] \tag{5}$$

(5) may be computed in a numerically stable manner using the log-sum-exp trick.