A General Framework for Amortizing Variational Filtering

Joseph Marino [1] **Milan Cvitkovic** [1] **Yisong Yue** [1]

Abstract

We introduce the *variational filtering EM algorithm*, a simple, general-purpose method for performing filtering variational inference in deep dynamical latent variable models. We derive the algorithm from the variational objective in the filtering setting. By performing inference optimization with iterative amortized inference models, we obtain a computationally efficient implementation of the algorithm, which we call *amortized variational filtering*. We present experiments demonstrating that this general-purpose method improves inference performance on several recent dynamical latent variable models.

1. Introduction

Complex tasks with time-series inputs, like audio comprehension or robotic manipulation, must often be performed online, where the model can only consider past and present information at each step. Models for such tasks frequently operate by inferring the hidden state state of the world at each time-step, as in e.g. a Hidden Markov Model. This type of online inference procedure is known as *filtering*.

Learning filtering models purely through supervised labels or rewards can be impractical, requiring massive collections of labeled data or significant efforts at reward shaping. But deep generative models can learn and infer hidden structure and states directly from data. Latent variable models (Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014), in particular, offer a promising direction; they infer latent representations using expressive deep networks, commonly using variational methods to perform inference (Jordan et al., 1998). Recent works have extended deep latent variable models to the time-series setting, e.g. (Chung et al., 2015; Fraccaro et al., 2016). However, inference procedures for these dynamical models have been proposed on the basis of intuition rather than from a rigorous inference optimization perspective. These hand–designed methods are theoretically lacking, potentially limiting performance.

We introduce *variational filtering EM*, a general algorithm for performing filtering variational inference and learning that is rigorously derived from the filtering variational objective. As detailed below, the result of applying the standard variational objective to a sequence of observations is a series of inference optimization objectives: one at each time-step. If, in optimizing each of these objectives, a model initializes its estimate of the latent state from the previous time-step's prior distribution, a classic Bayesian filtering prediction-update loop naturally emerges. This contrasts with existing filtering approaches for deep dynamical models, which use inference models that do not explicitly account for prior predictions during inference, presumably because the inference optimizations at each step can be difficult. However, using *iterative* inference models (Marino et al., 2018), which overcome this difficulty, we develop a computationally efficient implementation of the variational filtering EM algorithm, which we refer to as amortized variational filtering (AVF).

The main contributions of this paper are the variational filtering EM algorithm and its amortized implementation, AVF. This is a general-purpose filtering algorithm, widely applicable to dynamical latent variable models (as we demonstrate in our experiments). Moreover, the variational filtering EM algorithm is derived from the filtering variational objective, providing a solid theoretical framework for filtering inference. By precisely specifying the inference optimization procedure, this method takes a simple form compared to previous hand-crafted methods. Using several deep dynamical latent variable models, we demonstrate that this filtering approach compares favorably against current methods across a variety of benchmark sequence data sets.

2. Background

Section 2.1 provides the form of a general dynamical latent variable model. Section 2.2 covers variational inference, an approximate inference technique. Many deep latent variable models are trained efficiently by amortizing inference optimization (Section 2.3). Applying this technique to dynamical models is non-trivial, leading many prior works to use handcrafted amortized inference methods (Section 2.4).

[*]Equal contribution [1]California Institute of Technology, Pasadena, CA, USA. Correspondence to: Joseph Marino <jmarino@caltech.edu>.

Presented at the ICML 2018 workshop on Theoretical Foundations and Applications of Deep Generative Models, Stockholm, Sweden.

2.1. General Dynamical Latent Variable Model

A sequence of T observations, $\mathbf{x}_{\leq T}$, can be modeled using a dynamical latent variable model, $p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})$, which models the joint distribution between $\mathbf{x}_{\leq T}$ and a sequence of latent variables, $\mathbf{z}_{\leq T}$, with parameters θ. It is typically assumed that $p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})$ can be factorized into conditional joint distributions at each step, $p_\theta(\mathbf{x}_t, \mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t})$, which are conditioned on preceding variables. This results in the following auto-regressive formulation:

$$p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p_\theta(\mathbf{x}_t, \mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}) \tag{1}$$
$$= \prod_{t=1}^{T} p_\theta(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leq t}) p_\theta(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t}).$$

$p_\theta(\mathbf{x}_t | \mathbf{x}_{<t}, \mathbf{z}_{\leq t})$ is the *observation* model, and $p_\theta(\mathbf{z}_t | \mathbf{x}_{<t}, \mathbf{z}_{<t})$ is the *dynamics* model, both of which can be arbitrary functions of their conditioning variables. However, while eq. 1 provides the general form of a dynamical latent variable model, further assumptions about the conditional dependencies of the random variables, e.g. Markov, or the functional forms of the observation and dynamics models, e.g. linear, are often necessary for tractability.

2.2. Variational Inference

Given a model and a set of observations, we typically want to *infer* the posterior, $p(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$, for each sequence and *learn* the model parameters, θ. Inference can be performed online or offline through Bayesian filtering or smoothing respectively (Särkkä, 2013), and learning can be performed through maximum likelihood estimation. Unfortunately, exact inference and learning are intractable for all but the simplest model classes. For non-linear functions, which are present in *deep* latent variable models, we must resort to approximate inference. Variational inference (Jordan et al., 1998) reformulates inference as optimization by introducing an approximate posterior, $q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$, then minimizing the KL-divergence to the true posterior, $p(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$. To avoid evaluating $p(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$, one can express the KL-divergence as

$$D_{KL}(q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T}) || p(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})) = \log p_\theta(\mathbf{x}_{\leq T}) + \mathcal{F}, \tag{2}$$

where $\mathcal{F}$ is the *variational free energy*, also referred to as the (negative) *evidence lower bound* or ELBO, defined as

$$\mathcal{F} \equiv -\mathbb{E}_{q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})} \left[\log \frac{p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})}{q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})} \right]. \tag{3}$$

In eq. 2, $\log p_\theta(\mathbf{x}_{\leq T})$ is independent of $q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$, so one can minimize the KL-divergence to the true posterior, thereby performing approximate inference, by minimizing $\mathcal{F}$ w.r.t. $q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$. Further, as KL-divergence is non-negative, eq. 2 implies that free energy upper bounds the negative log likelihood. Therefore, upon minimizing $\mathcal{F}$ w.r.t. $q(\mathbf{z}_{\leq T} | \mathbf{x}_{\leq T})$, one can use the gradient $\nabla_\theta \mathcal{F}$ to learn the model parameters. These two optimization procedures are respectively the expectation and maximization steps of the variational EM algorithm (Neal & Hinton, 1998), which alternate until convergence. To scale this algorithm, stochastic gradients can be used for both inference (Ranganath et al., 2014) and learning (Hoffman et al., 2013).

2.3. Amortized Variational Inference

Performing inference optimization using conventional stochastic gradient descent techniques can be computationally demanding, potentially requiring many inference iterations. To increase efficiency, a separate *inference model* can learn to map data examples to approximate posterior estimates (Dayan et al., 1995; Gregor et al., 2014; Kingma & Welling, 2014; Rezende et al., 2014), thereby *amortizing* inference across examples (Gershman & Goodman, 2014). Denoting the distribution parameters of q as $\boldsymbol{\lambda}^q$ (e.g. Gaussian mean and variance), standard inference models take the form

$$\boldsymbol{\lambda}^q \leftarrow f_\phi(\mathbf{x}), \tag{4}$$

where the inference model is denoted as f with parameters ϕ. These models, though efficient, have limitations. Notably, because the models only receive the data as input, they are unable to account for *empirical priors*. Empirical priors arise in the dynamics of dynamical models, forming priors across time steps, as well as in hierarchical models, forming priors across levels. Previous works have attempted to overcome their noninclusion of empirical priors through heuristics, using "top-down" inference in hierarchical models (Sønderby et al., 2016) and recurrent inference models in dynamical models, e.g. (Chung et al., 2015).

Iterative inference models (Marino et al., 2018) directly account for these priors, instead performing inference optimization by iteratively encoding approximate posterior estimates and gradients:

$$\boldsymbol{\lambda}^q \leftarrow f_\phi(\boldsymbol{\lambda}^q, \nabla_{\boldsymbol{\lambda}^q} \mathcal{F}). \tag{5}$$

The gradients, $\nabla_{\boldsymbol{\lambda}^q} \mathcal{F}$, can be estimated through black box methods (Ranganath et al., 2014) or the reparameterization trick (Kingma & Welling, 2014; Rezende et al., 2014) when applicable. Analogously to learning to learn (Andrychowicz et al., 2016), iterative inference models learn to perform inference optimization, thereby *learning to infer*. Eq. 5 provides a viable encoding form for an iterative inference model, but other forms, such as additionally encoding the data, $\mathbf{x}$, can potentially lead to faster inference convergence. Empirically, iterative inference models have also been shown to yield improved modeling performance over comparable standard models (Marino et al., 2018).

2.4. Related Work

Many *deterministic* deep dynamical latent variable models have been proposed for sequential data (Chung et al., 2014; Srivastava et al., 2015; Lotter et al., 2016; Finn et al., 2016). While these models often capture many aspects of the data, they cannot account for the uncertainty inherent in many domains, typically arising from partial observability of the environment. By averaging over multi-modal distributions, these models often produce samples in regions of low probability, e.g. blurry video frames. This inadequacy necessitates moving to probabilistic models, which can explicitly model uncertainty to accurately capture the distribution of possible futures.

Amortized variational inference (Kingma & Welling, 2014; Rezende et al., 2014) has enabled many recently proposed *probabilistic* deep dynamical latent variable models, with applications to video (Walker et al., 2016; Karl et al., 2016; Xue et al., 2016; Johnson et al., 2016; Gemici et al., 2017; Fraccaro et al., 2017; Babaeizadeh et al., 2018; Denton & Fergus, 2018; Li & Mandt, 2018), speech (Chung et al., 2015; Fraccaro et al., 2016; Goyal et al., 2017; Hsu et al., 2017; Li & Mandt, 2018), handwriting (Chung et al., 2015), music (Fraccaro et al., 2016), etc. While these models differ in their functional mappings, most fall within the general form of eq. 1. Crucially, simply encoding the observation at each step is insufficient to accurately perform approximate inference, as the prior can vary across steps. Thus, with each deep dynamical latent variable model, a hand–crafted amortized inference procedure has been proposed. For instance, many filtering inference methods re-use various components of the generative model (Chung et al., 2015; Fraccaro et al., 2016; Gemici et al., 2017; Denton & Fergus, 2018), while some methods introduce separate recurrent neural networks into the filtering procedure (Bayer & Osendorfer, 2014; Denton & Fergus, 2018) or encode the previous latent sample (Karl et al., 2016). Specifying a filtering method has been an engineering effort, as we have lacked a theoretical framework.

The variational filtering EM algorithm precisely specifies the inference optimization procedure implied by the filtering variational objective. The main insight from this analysis is that, having drawn approximate posterior samples at previous steps, inference becomes a *local* optimization, depending only on the current prior and observation. This suggests one unified approach that explicitly performs inference optimization at each step, allowing us to replace the current collection of custom filtering methods. When the approximate posterior at each step is initialized at the corresponding prior, this approach entails a prediction-update loop, with the update composed of a gradient (error) signal.

Perhaps the closest technique in the probabilistic modeling literature is the "residual" inference method from (Fraccaro et al., 2016), which updates the approximate posterior mean from the prior. Other similar ideas have been proposed on an empirical basis for deterministic models (Lotter et al., 2016; Henaff et al., 2017). PredNet (Lotter et al., 2016) is a deterministic model that encodes prediction errors to perform inference. This approach is inspired by *predictive coding* (Rao & Ballard, 1999; Friston, 2005), a theory from neuroscience postulating that feedforward pathways in sensory processing areas of the brain use prediction errors to update state estimates from prior predictions. In turn, this theory is motivated by classical Bayesian filtering (Särkkä, 2013), which updates the posterior from the prior using the likelihood of the prediction. For linear Gaussian models, this manifests as the Kalman filter (Kalman et al., 1960), which uses prediction errors to perform exact inference.

Finally, several recent works have used particle filtering, in conjunction with amortized inference, to provide a tighter lower bound on the log likelihood for sequential data (Maddison et al., 2017; Naesseth et al., 2017; Le et al., 2017). The techniques developed here can also be applied to this tighter bound.

3. Variational Filtering

3.1. Variational Filtering Expectation Maximization

In the filtering setting, the approximate posterior at each step depends only on past and present variables, enabling *online* approximate inference. Making the mean-field assumption across steps, then $q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})$ takes the factorized form

$$q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T}) = \prod_{t=1}^{T} q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}). \tag{6}$$

Note that the conditioning variables in q denote an *indirect* dependence that arises through free energy minimization and does not necessarily constitute a direct functional mapping. Under a filtering approximate posterior, the free energy (eq. 3) can be expressed as

$$\mathcal{F} = \sum_{t=1}^{T} \mathbb{E}_{\prod_{\tau=1}^{t-1} q(\mathbf{z}_\tau|\mathbf{x}_{\leq \tau}, \mathbf{z}_{<\tau})} [\mathcal{F}_t] = \sum_{t=1}^{T} \tilde{\mathcal{F}}_t, \tag{7}$$

(see Appendix A for the derivation) where $\mathcal{F}_t$ is the *step free energy*, defined as

$$\mathcal{F}_t \equiv -\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})} \left[\log \frac{p_\theta(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})}{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})} \right], \tag{8}$$

and we have also defined $\tilde{\mathcal{F}}_t$ as the t^{th} term in the summation. To gain further intuition, note that with a single step, the filtering free energy reduces to the first step free energy. As in the static setting, this term can be re-expressed as a reconstruction term and a "regularizing" KL-divergence

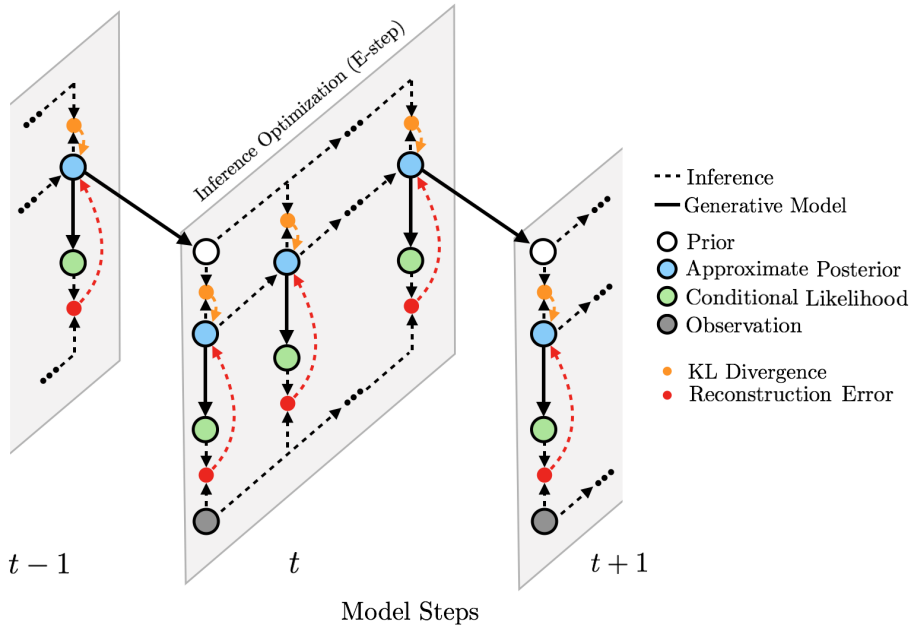


Figure 1. **Variational filtering EM.** Diagram shows filtering inference within a Markov dynamical latent variable model, as detailed in Algorithm 1. The central gray region depicts inference optimization of the approximate posterior, $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$, at step t, which can be initialized at or near the corresponding prior, $p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$. Sampling from the approximate posterior generates the conditional likelihood, $p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})$, which is evaluated at the observation, $\mathbf{x}_t$, to calculate the reconstruction error. This term, combined with the KL divergence between the approximate posterior and prior, comprise the step free energy, $\mathcal{F}_t$ (see eq. 9). Inference optimization (E-step) involves finding the approximate posterior that minimizes the step free energy. Learning (M-step), which is not shown in the figure, corresponds to updating the model parameters, θ, to minimize the total filtering free energy, $\mathcal{F}$. Note that the figure outlines a gradient-based inference procedure using red and orange lines. Many previously proposed filtering techniques instead rely on direct amortized mappings from observations and hidden states, however they are still attempting to optimize the same inference objective.

term:

$$\begin{aligned}\mathcal{F}_t = &- \mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\left[\log p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t})\right] \\ &+ D_{KL}(q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})||p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})).\end{aligned} \tag{9}$$

The filtering free energy in eq. 7 is the sum of these step free energy terms, each of which is evaluated according to expectations over past latent sequences. To perform filtering variational inference, we must find the set of T terms in $q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})$ that minimize the filtering free energy summation.

We now describe the variational filtering EM algorithm, given in Algorithm 1 and depicted in Figure 1, which optimizes eq. 7. This algorithm sequentially optimizes each of the approximate posterior terms to perform filtering inference. Consider the approximate posterior at step t, $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$. This term appears in $\mathcal{F}$, either directly or in expectations, in terms t through T of the summation:

$$\mathcal{F} = \underbrace{\tilde{\mathcal{F}}_1 + \cdots + \tilde{\mathcal{F}}_{t-1}}_{\text{steps } q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) \text{ depends on}} + \overbrace{\tilde{\mathcal{F}}_t + \tilde{\mathcal{F}}_{t+1} + \cdots + \tilde{\mathcal{F}}_T}^{\text{terms } q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) \text{ appears in}}. \tag{10}$$

However, the filtering setting requires that the approximate posterior at each step can only depend on past and present variables, i.e. steps 1 through t. Therefore, of the T terms in $\mathcal{F}$, the *only* term through which we can optimize $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ is the t^{th} term:

$$q^*(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) = \underset{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}{\arg\min}\ \mathcal{F} = \underset{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}{\arg\min}\ \tilde{\mathcal{F}}_t. \tag{11}$$

Optimizing $\tilde{\mathcal{F}}_t$ requires evaluating expectations over previous approximate posteriors. Again, because approximate posterior estimates cannot be influenced by future variables, these past expectations remain *fixed* through the future. Thus, filtering variational inference (the variational E-step) can be performed by sequentially minimizing each $\mathcal{F}_t$ w.r.t. $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$, holding the expectations over past variables fixed. Importantly, once the past expectations have been evaluated or estimated, inference optimization is entirely defined by the step free energy at the current step.

For simple models, such as linear Gaussian models, these expectations may be computed exactly. However, in general, the expectations must be estimated through Monte Carlo samples from q, with inference optimization carried out using stochastic gradients (Ranganath et al., 2014). As in the

Algorithm 1 Variational Filtering Expectation Maximization

1: **Input:** observation sequence $\mathbf{x}_{1:T}$, model $p_\theta(\mathbf{x}_{1:T}, \mathbf{z}_{1:T})$
2: $\nabla_\theta \mathcal{F} = 0$
3: **for** $t = 1$ **to** T **do**
4: initialize $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ ▷ from $p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$
5: $\tilde{\mathcal{F}}_t := \mathbb{E}_{q(\mathbf{z}_{<t}|\mathbf{x}_{<t}, \mathbf{z}_{<t-1})}[\mathcal{F}_t]$
6: $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t}) = \arg\min_q \tilde{\mathcal{F}}_t$ ▷ inference (E-step)
7: $\nabla_\theta \mathcal{F} = \nabla_\theta \mathcal{F} + \nabla_\theta \tilde{\mathcal{F}}_t$
8: **end for**
9: $\theta = \theta - \alpha \nabla_\theta \mathcal{F}$ ▷ learning (M-step)

static setting, we can initialize $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ at (or near) the prior, $p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$. This yields a simple interpretation: starting at the prior, we run the model forward, thereby generating a *prediction*, in order to evaluate the free energy at the current step. Then, using the approximate posterior gradient, we perform an inference *update* to the estimate. This agrees with classical Bayesian filtering, where the posterior is updated from the prior prediction according to the likelihood of observations. Unlike the classical setting, here, reconstruction and update steps can be repeated until inference convergence.

After inferring an optimal approximate posterior, learning (the variational M-step) can be performed by minimizing the total filtering free energy w.r.t. the model parameters, θ. As eq. 7 is a summation and differentiation is a linear operation, $\nabla_\theta \mathcal{F}$ is the sum of contributions from each of these terms:

$$\nabla_\theta \mathcal{F} = \sum_{t=1}^{T} \nabla_\theta \left[\mathbb{E}_{\prod_{\tau=1}^{t-1} q(\mathbf{z}_\tau|\mathbf{x}_{\leq \tau}, \mathbf{z}_{<\tau})} [\mathcal{F}_t] \right]. \tag{12}$$

Parameter gradients can be estimated online by accumulating the result from each term in the filtering free energy. The parameters are then updated at the end of the sequence. For large data sets, stochastic estimates of parameter gradients can be obtained from a mini-batch of data examples (Hoffman et al., 2013).

3.2. Amortized Variational Filtering

Performing approximate inference optimization (Algorithm 1, Line 6) with traditional techniques can be computationally costly, requiring many iterations of gradient updates and hand-tuning of optimizer hyper-parameters. In online settings, particularly with large models and data sets, this may be impractical. An alternative approach is to employ an amortized inference model, which can learn to minimize $\mathcal{F}_t$ w.r.t. $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ more efficiently at each step. Note that $\mathcal{F}_t$ (eq. 8) contains $p_\theta(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t}) = p_\theta(\mathbf{x}_t|\mathbf{x}_{<t}, \mathbf{z}_{\leq t}) p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$. The prior, $p_\theta(\mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})$, varies across steps, constituting the latent dynamics. Standard inference models (eq. 4), which only encode $\mathbf{x}_t$, do not have access to the prior and therefore cannot properly optimize $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$. Many inference models in the sequential setting attempt to account for this information by including hidden states or encoding previous latent samples, e.g. (Chung et al., 2015; Fraccaro et al., 2016; Denton & Fergus, 2018). However, given the complexities of many generative models, it can be difficult to determine how to properly route the necessary prior information into the inference model to perform inference optimization. As a result, each dynamical latent variable model has been proposed with an accompanying custom inference model set-up.

We propose a simple and general alternative method for amortizing filtering inference that is agnostic to the particular form of the generative model. Iterative inference models (Marino et al., 2018) naturally account for the changing prior through the approximate posterior gradients. These models are thus a natural candidate for performing inference at each step. Similar to eq. 5, when $q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})$ is a parametric distribution with parameters $\boldsymbol{\lambda}_t^q$, the inference update takes the form:

$$\boldsymbol{\lambda}_t^q \leftarrow f_\phi(\boldsymbol{\lambda}_t^q, \nabla_{\boldsymbol{\lambda}_t^q} \tilde{\mathcal{F}}_t). \tag{13}$$

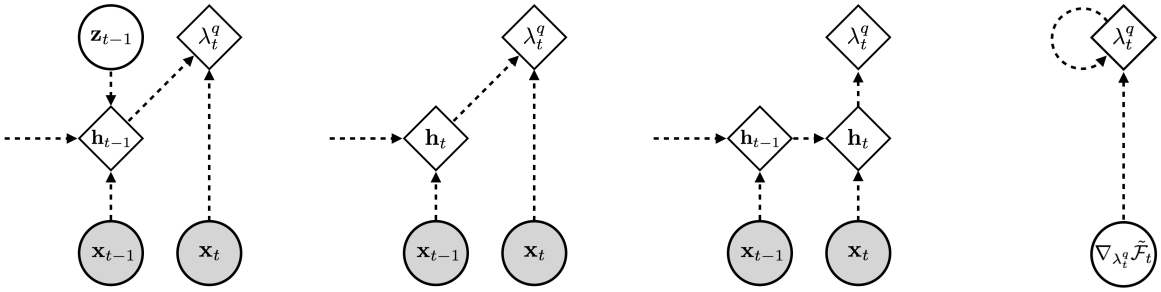
We refer to this set-up as *amortized variational filtering* (AVF). As in eq. 5, we note that eq. 13 offers just one particular encoding form for an iterative inference model. For instance, $\mathbf{x}_t$ could be additionally encoded at each step. Marino et al. also note that in latent Gaussian models, precision-weighted errors provide an alternative inference optimization signal (Marino et al., 2018). There are two main benefits to using iterative inference models in the filtering setting:

- These models contain all of the terms necessary to perform inference optimization, providing a simple model form that does not require any additional states or inputs.
- The approximate posterior is updated from the prior, so model capacity is utilized for inference *corrections* rather than re-estimating the approximate posterior at each step.

In practice, these advantages permit the use of relatively simple iterative inference models that can perform filtering inference efficiently and accurately. We demonstrate this in the following section.

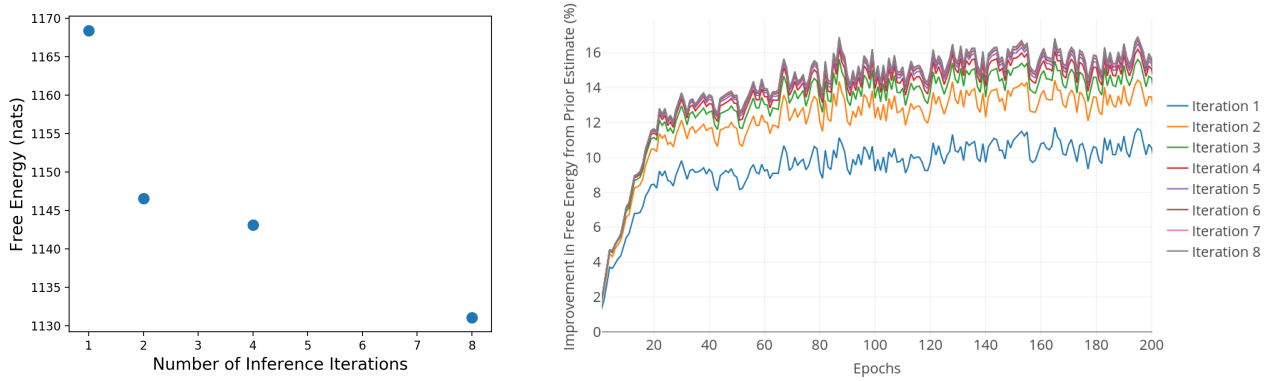
4. Experiments

We empirically evaluate amortized variational filtering using multiple deep dynamical latent Gaussian model ar-



(a) VRNN (Inference) (b) SRNN (Inference) (c) SVG (Inference) (d) AVF

Figure 2. **Inference computation graphs** for originally proposed filtering methods for (a) VRNN, (b) SRNN, and (c) SVG. Each diagram shows the procedure for inferring the approximate posterior parameters, $\boldsymbol{\lambda}_t^q$, at step t. Previously proposed methods (a – c) rely on hand-crafted architectures of observations, hidden states, and latent variables. AVF (d) is a general filtering method that only requires the current approximate posterior estimate and gradient (see eq. 13) to explicitly optimize $\tilde{\mathcal{F}}_t$, the current expected step free energy.



(a) (b)

Figure 3. **Improvement with inference iterations.** (a) Average free energy per step using VRNN on TIMIT (validation set) for AVF with varying numbers of inference iterations during training. Additional iterations lead to improved performance. (b) Relative improvement (decrease) in free energy at each inference iteration for the same model during training. The initial inference iteration results in a large improvement over the prior estimate. Successive inference iterations provide further, smaller improvements.

chitectures on a variety of sequence data sets. Specifically, we use AVF to train VRNN (Chung et al., 2015), SRNN (Fraccaro et al., 2016), and SVG (Denton & Fergus, 2018) on speech (Garofolo et al., 1993), music (Boulanger-Lewandowski et al., 2012), and video (Schuldt et al., 2004) data. In each setting, we compare AVF against the originally proposed filtering method for the model. Diagrams of the filtering methods are shown in Figure 2. Implementations of the models are based on code provided by the respective authors of VRNN[1], SRNN[2], and SVG[3]. Accompanying code can be found on GitHub at https://github.com/joelouismarino/amortized-variational-filtering.

[1] https://github.com/jych/nips2015_vrnn
[2] https://github.com/marcofraccaro/srnn
[3] https://github.com/edenton/svg

4.1. Experiment Set-Up

Iterative inference models are implemented as specified in eq. 13, encoding the approximate posterior parameters and their gradients at each inference iteration at each step. Following (Marino et al., 2018), we normalize the inputs to the inference model using layer normalization (Ba et al., 2016). The generative models that we evaluate contain non-spatial latent variables, thus, we use fully-connected layers to parameterize the inference models. Importantly, minimal effort went into engineering the inference model architectures: across all models and data sets, we utilize the *same* inference model architecture for AVF. Further details are found in Appendix B.

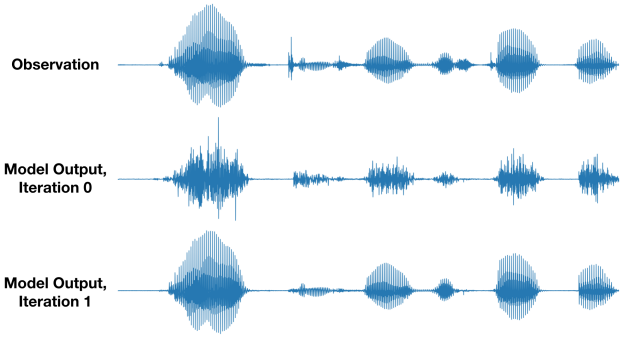


Figure 4. Test data (top), output predictions (middle), and reconstructions (bottom) for TIMIT using SRNN with AVF. Sequences run from left to right. The predictions made by the model already contain the general structure of the data. AVF explicitly updates the approximate posterior from the prior prediction, focusing on inference *corrections* rather than re-estimation.

4.1.1. Speech Modeling

Models For speech modeling, we use VRNN and SRNN, attempting to keep the model architectures consistent with the original implementations. The most notable difference in our implementation occurs in SRNN, where we use an LSTM rather than a GRU as the recurrent module. As in (Fraccaro et al., 2016), we anneal the KL divergence initially during training. In both models, we use a Gaussian output density. Unlike (Chung et al., 2015; Fraccaro et al., 2016; Goyal et al., 2017), which evaluate log *densities*, we evaluate and report log *probabilities* by integrating the output density over the data discretization window, as in modeling image pixels. This permits comparison across different output distributions.

Data We train and evaluate on TIMIT (Garofolo et al., 1993), which consists of audio recordings of 6,300 sentences spoken by 630 individuals. As performed by (Chung et al., 2015), we sample the audio waveforms at 16 kHz, split the training and validation sets into half second clips, and group each sequence into bins of 200 consecutive samples. Thus, each training and validation sequence consists of 40 model steps. Test evaluation is performed on the full duration of each test sequence, averaging roughly 3 seconds.

4.1.2. Music Modeling

Model We model polyphonic music using SRNN. The generative model architecture is the same as in the speech modeling experiments, with changes in the number of layers and units to match (Fraccaro et al., 2016). To model the binary music notes, we use a Bernoulli output distribution. Again, we anneal the KL divergence initially during training.

Data We use four data sets of polyphonic (MIDI) music (Boulanger-Lewandowski et al., 2012): Piano-midi.de, MuseData, JSB Chorales, and Nottingham. Each data set contains between 100 and 1,000 songs, with each song between 100 to 4,000 steps. For training and validation, we break the sequences into clips of length 25, and we test on the entire test sequences.

4.1.3. Video Modeling

Model Our implementation of SVG differs from the original model in that we evaluate conditional log-likelihood under a Gaussian output density rather than mean squared output error. All other architecture details are identical to the original model. However, (Denton & Fergus, 2018) down-weight the KL-divergence by a factor of 1e-6 at all steps. We instead remove this factor to use the free energy during training and evaluation. As to be expected, this results in the model using the latent variables to a lesser extent. We also train and evaluate SVG using filtering inference at all steps, rather than predicting multiple steps into the future.

Data We train and evaluate SVG on KTH Actions (Schuldt et al., 2004), which contains 760 train / 768 val / 863 test videos of people performing various actions, each of which is between roughly 50 - 150 frames. Frames are re-sized to 64×64. For training and validation, we split the data into clips of 20 frames.

4.2. Results

4.2.1. Additional Inference Iterations

The variational filtering EM algorithm involves approximate posterior optimization (Algorithm 1, Line 6). AVF handles this optimization through a model that learns to perform iterative updates (eq. 13). Thus, additional inference iterations can potentially lead to further inference improvement. We explore this aspect on TIMIT using VRNN. In Figure 3a, we plot the average free energy on validation sequences for varying numbers of inference iterations during training. Figure 3b shows relative inference improvement over the prior estimate for a single model throughout training. Additional inference iterations improve performance, with each iteration providing a slight improvement. Note that this aspect is distinct from many baseline filtering methods for deep dynamical latent variable models, which do not permit multiple inference iterations per step.

Figure 4 illustrates example reconstructions over inference iterations, using SRNN on TIMIT. At the initial inference iteration, the approximate posterior is initialized from the prior, resulting in an output prediction at iteration 0. The approximate posterior is updated at the following inference iteration, improving the output reconstruction by adjusting the parameters of the Gaussian conditional likelihood.

Table 1. Average free energy (in *nats per step*) on the TIMIT speech data set for SRNN and VRNN with the respective proposed filtering procedures and with AVF.

	TIMIT
VRNN	
baseline	1,082
AVF	**1,071**
SRNN	
baseline	1,026
AVF	**1,024**

Table 2. Average free energy (in *nats per step*) on the KTH Actions video data set for SVG with the proposed filtering procedure and with AVF.

	KTH Actions
SVG	
baseline	15,097
AVF	**11, 714**

Table 3. Average free energy (in *nats per step*) on polyphonic music data sets for SRNN with and without AVF. Results from (Fraccaro et al., 2016) are provided for comparison, however, our implementation of SRNN differs in several aspects (see Appendix B).

	Piano-midi.de	MuseData	JSB Chorales	Nottingham
SRNN				
baseline (Fraccaro et al., 2016)	8.20	6.28	4.74	2.94
baseline	8.19	6.27	6.92	3.19
AVF	**8.12**	**5.99**	**6.77**	**3.13**

4.2.2. Quantitative Comparison

Tables 1, 2, and 3 present quantitative comparisons of average filtering free energy between AVF and baseline filtering methods for TIMIT, KTH Actions, and the polyphonic music data sets respectively. On TIMIT, training with AVF performs comparably or slightly better than using the baseline methods for both VRNN and SRNN. Similar improvements are also observed on each of the polyphonic music data sets. AVF significantly improves the performance of SVG on KTH Actions. We attribute this, in part, to the absence of the KL down-weighting factor in our training objective. The baseline filtering procedure seems to struggle to a greater degree than AVF.

5. Conclusion

We introduced the variational filtering EM algorithm for filtering in dynamical latent variable models. We noted that filtering inference can be expressed as a sequence of optimization objectives, linked across steps through previous latent samples. Using iterative inference models to perform inference optimization, we arrived at an efficient implementation of the algorithm: amortized variational filtering. This general filtering algorithm scales to large models and data sets. Numerous methods have been proposed for filtering in deep dynamical latent variable models, with each method hand-crafted for each model. The variational filtering EM algorithm provides a framework for analyzing and constructing these methods. Amortized variational filtering is a simple, theoretically-motivated, and general filtering method that we have shown performs on-par with or better than multiple existing state-of-the-art methods.

References

Andrychowicz, Marcin, Denil, Misha, Gomez, Sergio, Hoffman, Matthew W, Pfau, David, Schaul, Tom, and de Freitas, Nando. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pp. 3981–3989, 2016.

Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Babaeizadeh, Mohammad, Finn, Chelsea, Erhan, Dumitru, Campbell, Roy H, and Levine, Sergey. Stochastic variational video prediction. In *Proceedings of the International Conference on Learning Representations*, 2018.

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 *Proceedings of the International Conference on Machine Learning*, 2012.

Chung, Junyoung, Gulcehre, Caglar, Cho, KyungHyun, and Bengio, Yoshua. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

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.

Dayan, Peter, Hinton, Geoffrey E, Neal, Radford M, and Zemel, Richard S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.

Denton, Emily and Fergus, Rob. Stochastic video generation with a learned prior. In *Proceedings of the International Conference on Machine Learning*, 2018.

Finn, Chelsea, Goodfellow, Ian, and Levine, Sergey. Unsupervised learning for physical interaction through video prediction. In *Advances in Neural Information Processing Systems*, pp. 64–72, 2016.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pp. 2199–2207, 2016.

Fraccaro, Marco, Kamronn, Simon, Paquet, Ulrich, and Winther, Ole. A disentangled recognition and nonlinear dynamics model for unsupervised learning. In *Advances in Neural Information Processing Systems*, pp. 3603–3612, 2017.

Friston, Karl. A theory of cortical responses. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1456):815–836, 2005.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L. Darpa timit acoustic phonetic continuous speech corpus, 1993.

Gemici, Mevlana, Hung, Chia-Chun, Santoro, Adam, Wayne, Greg, Mohamed, Shakir, Rezende, Danilo J, Amos, David, and Lillicrap, Timothy. Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*, 2017.

Gershman, Samuel and Goodman, Noah. Amortized inference in probabilistic reasoning. In *Proceedings of the Cognitive Science Society*, volume 36, 2014.

Goyal, Anirudh, Sordoni, Alessandro, Côté, Marc-Alexandre, Ke, Nan, and Bengio, Yoshua. Z-forcing: Training stochastic recurrent networks. In *Advances in Neural Information Processing Systems*, pp. 6716–6726, 2017.

Gregor, Karol, Danihelka, Ivo, Mnih, Andriy, Blundell, Charles, and Wierstra, Daan. Deep autoregressive networks. In *Proceedings of the International Conference on Machine Learning*, 2014.

Henaff, Mikael, Zhao, Junbo, and LeCun, Yann. Prediction under uncertainty with error-encoding networks. *arXiv preprint arXiv:1711.04994*, 2017.

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

Hsu, Wei-Ning, Zhang, Yu, and Glass, James. Unsupervised learning of disentangled and interpretable representations from sequential data. In *Advances in Neural Information Processing Systems*, pp. 1876–1887, 2017.

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.

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An introduction to variational methods for graphical models. *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, 89:105–162, 1998.

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

Karl, Maximilian, Soelch, Maximilian, Bayer, Justin, and van der Smagt, Patrick. Deep variational bayes filters: Unsupervised learning of state space models from raw data. In *Proceedings of the International Conference on Learning Representations*, 2016.

Kingma, Diederik P and Welling, Max. Stochastic gradient vb and the variational auto-encoder. In *Proceedings of the International Conference on Learning Representations*, 2014.

Le, Tuan Anh, Igl, Maximilian, Jin, Tom, Rainforth, Tom, and Wood, Frank. Auto-encoding sequential monte carlo. *arXiv preprint arXiv:1705.10306*, 2017.

Li, Yingzhen and Mandt, Stephan. A deep generative model for disentangled representations of sequential data. In *Proceedings of the International Conference on Machine Learning*, 2018.

Lotter, William, Kreiman, Gabriel, and Cox, David. Deep predictive coding networks for video prediction and unsupervised learning. *arXiv preprint arXiv:1605.08104*, 2016.

Maddison, Chris J, Lawson, John, Tucker, George, Heess, Nicolas, Norouzi, Mohammad, Mnih, Andriy, Doucet, Arnaud, and Teh, Yee. Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pp. 6576–6586, 2017.

Marino, Joseph, Yue, Yisong, and Mandt, Stephan. Iterative amortized inference. In *Proceedings of the International Conference on Machine Learning*, 2018.

Naesseth, Christian A, Linderman, Scott W, Ranganath, Rajesh, and Blei, David M. Variational sequential monte carlo. *arXiv preprint arXiv:1705.11140*, 2017.

Neal, Radford M and Hinton, Geoffrey E. A view of the em algorithm that justifies incremental, sparse, and other variants. In *Learning in graphical models*, pp. 355–368. Springer, 1998.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David. Black box variational inference. In *Artificial Intelligence and Statistics*, pp. 814–822, 2014.

Rao, Rajesh PN and Ballard, Dana H. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1), 1999.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the International Conference on Machine Learning*, pp. 1278–1286, 2014.

Särkkä, Simo. *Bayesian filtering and smoothing*, volume 3. Cambridge University Press, 2013.

Schuldt, Christian, Laptev, Ivan, and Caputo, Barbara. Recognizing human actions: a local svm approach. In *Proceedings of the International Conference on Pattern Recognition*, volume 3, pp. 32–36. IEEE, 2004.

Sønderby, Casper Kaae, Raiko, Tapani, Maaløe, Lars, Sønderby, Søren Kaae, and Winther, Ole. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, pp. 3738–3746, 2016.

Srivastava, Nitish, Mansimov, Elman, and Salakhudinov, Ruslan. Unsupervised learning of video representations using lstms. In *International Conference on Machine Learning*, pp. 843–852, 2015.

Walker, Jacob, Doersch, Carl, Gupta, Abhinav, and Hebert, Martial. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, pp. 835–851. Springer, 2016.

Xue, Tianfan, Wu, Jiajun, Bouman, Katherine, and Freeman, Bill. Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 91–99, 2016.

A. Filtering Variational Free Energy

A.1. Derivation

This derivation largely follows that of (Gemici et al., 2017) and is valid for factorized filtering approximate posteriors. From eq. 3, we have the definition of variational free-energy:

$$\mathcal{F} \equiv -\mathbb{E}_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\log \frac{p_\theta(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T})}{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\right]. \tag{1}$$

Plugging in the forms of the joint distribution (eq. 1) and approximate posterior (eq. 6), we can write the term within the expectation as a sum:

$$\mathcal{F} = -\mathbb{E}_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\log\left(\prod_{t=1}^{T} \frac{p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})}{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\right)\right] \tag{2}$$

$$\mathcal{F} = -\mathbb{E}_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T} \log \frac{p(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})}{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\right] \tag{3}$$

$$\mathcal{F} = -\mathbb{E}_{q(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T} C_t\right] \tag{4}$$

where the term C_t is defined to simplify notation. We then expand the expectation:

$$\mathcal{F} = -\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)} \ldots \mathbb{E}_{q(\mathbf{z}_T|\mathbf{x}_{\leq T}, \mathbf{z}_{<T})}\left[\sum_{t=1}^{T} C_t\right] \tag{5}$$

There are T terms within the sum, but each C_t only depends on the expectations up to time t because we only condition on past and present variables. This allows us to write:

$$\begin{aligned}\mathcal{F} = &-\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}\left[C_1\right] \\ &-\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}\mathbb{E}_{q(\mathbf{z}_2|\mathbf{x}_{\leq 2}, \mathbf{z}_1)}\left[C_2\right] \\ &-\ldots \\ &-\mathbb{E}_{q(\mathbf{z}_1|\mathbf{x}_1)}\mathbb{E}_{q(\mathbf{z}_2|\mathbf{x}_{\leq 2}, \mathbf{z}_1)} \ldots \mathbb{E}_{q(\mathbf{z}_T|, \mathbf{x}_{\leq T}, \mathbf{z}_{<T})}\left[C_T\right]\end{aligned} \tag{6}$$

$$\mathcal{F} = -\sum_{t=1}^{T} \mathbb{E}_{q(\mathbf{z}_{\leq t}|\mathbf{x}_{\leq t})}\left[C_t\right] \tag{7}$$

$$\mathcal{F} = -\sum_{t=1}^{T} \mathbb{E}_{\prod_{\tau=1}^{t} q(\mathbf{z}_\tau|\mathbf{x}_{\leq \tau}, \mathbf{z}_{<\tau})}\left[C_t\right] \tag{8}$$

$$\mathcal{F} = -\sum_{t=1}^{T} \mathbb{E}_{\prod_{\tau=1}^{t-1} q(\mathbf{z}_\tau|\mathbf{x}_{\leq \tau}, \mathbf{z}_{<\tau})}\left[\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\left[C_t\right]\right] \tag{9}$$

As in Section 3, we define $\mathcal{F}_t$ as

$$\mathcal{F}_t \equiv -\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\left[C_t\right] \tag{10}$$

$$\mathcal{F}_t = -\mathbb{E}_{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\left[\log \frac{p_\theta(\mathbf{x}_t, \mathbf{z}_t|\mathbf{x}_{<t}, \mathbf{z}_{<t})}{q(\mathbf{z}_t|\mathbf{x}_{\leq t}, \mathbf{z}_{<t})}\right]. \tag{11}$$

This allows us to write eq. 9 as

$$\mathcal{F} = \sum_{t=1}^{T} \mathbb{E}_{\prod_{\tau=1}^{t-1} q(\mathbf{z}_\tau|\mathbf{x}_{\leq \tau}, \mathbf{z}_{<\tau})}\left[\mathcal{F}_t\right], \tag{12}$$

which agrees with eq. 7.

B. Implementation Details

For all iterative inference models, we follow (Marino et al., 2018), using two layer fully-connected networks with 1,024 units per layer, highway gating connections (Srivastava et al., 2015), and ELU non-linearities (Clevert et al., 2015). Unless otherwise noted, these models receive the current estimate of the approximate posterior and approximate posterior gradient (4 terms in total), normalizing each term separately using layer normalization (Ba et al., 2016). We use the same output gating update employed in (Marino et al., 2018). We also found that applying layer normalization to the approximate posterior mean estimates resulted in improved training stability.

B.1. Speech Modeling

The VRNN architecture is implemented as in (Chung et al., 2015), matching the number of layers and units in each component of the model, as well as the non-linearities. We train on TIMIT with sequences of length 40, using a batch size of 64. For the baseline method, we use a learning rate of 0.001, as specified in (Chung et al., 2015). For AVF, we use a learning rate of 0.0001. We anneal the learning rates by a factor of 0.999 after each epoch. For quantitative (test) results, we used 2 inference iterations for AVF.

We implement SRNN following (Fraccaro et al., 2016), with the exception of an LSTM in place of the GRU. All other architecture details, are kept consistent, including the use of clipped (± 3) leaky ReLU non-linearities. The sequence length and batch size are the same as above. We use a learning rate of 0.001 for the baseline method, following (Fraccaro et al., 2016). We use a learning rate of 0.0001 for AVF. We use the same learning rate annealing strategy as above. Following (Fraccaro et al., 2016), we anneal the KL-divergence of the baseline linearly over the first 20 epochs. We increase this duration to 50 epochs for AVF. The iterative inference model additionally encodes the data observation at each step, which we found necessary to overcome the local minima from the KL-divergence. We use a single inference iteration for AVF.

B.2. Music Modeling

Our SRNN implementation is the same as in the speech modeling setting, with the appropriate changes in the number of units and layers to match (Fraccaro et al., 2016). We use a sequence length of 25 and a batch size of 16. All models are trained with a learning rate of 0.0001, with a decay factor of 0.999 per epoch. We anneal the KL-divergence linearly over the first 50 epochs. Models trained with AVF using a single inference iteration, except with JSB Chorales, where we use 5 inference iterations.

B.3. Video Modeling

The SVG model architecture is implemented identically to (Denton & Fergus, 2018), with the addition of a variance term to the observation model to account for uncertainty in the output. We train on sequences of length 20 using a batch size of 20. For both methods, we use a learning rate of 0.0001, with decay of 0.999 after each epoch. We use a single inference iteration for AVF.

References

Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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.

Clevert, Djork-Arné, Unterthiner, Thomas, and Hochreiter, Sepp. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Denton, Emily and Fergus, Rob. Stochastic video generation with a learned prior. *arXiv preprint arXiv:1802.07687*, 2018.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, pp. 2199–2207, 2016.

Gemici, Mevlana, Hung, Chia-Chun, Santoro, Adam, Wayne, Greg, Mohamed, Shakir, Rezende, Danilo J, Amos, David, and Lillicrap, Timothy. Generative temporal models with memory. *arXiv preprint arXiv:1702.04649*, 2017.

Marino, Joseph, Yue, Yisong, and Mandt, Stephan. Iterative amortized inference. In *International Conference on Machine Learning*, 2018.

Srivastava, Rupesh Kumar, Greff, Klaus, and Schmidhuber, Jürgen. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.