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:

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

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:

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

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

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

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

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.