Filtering Variational Objectives

Chris J. Maddison[*, 1, 2] **Dieterich Lawson**[*, 3] **George Tucker**[*, 3] **Nicolas Heess**[1] **Mohammad Norouzi**[3]
Andriy Mnih[1] **Arnaud Doucet**[2] **Yee Whye Teh**[1, 2]

Abstract

When used as a surrogate objective for maximum likelihood estimation in latent variable models, the evidence lower bound (ELBO) produces state-of-the-art results. Inspired by this, we consider the extension of the ELBO to a family of lower bounds defined by a particle filter's estimator of the marginal likelihood, the *filtering variational objectives* (FIVOs). FIVOs take the same arguments as the ELBO, but can exploit a model's sequential structure to produce tighter bounds. Experimentally, we show uniform improvements over models trained with ELBO on sequential data.

1. Introduction

There is a demand for rich generative models of high dimensional data with a mixture of known and unknown structure. For example, video data has known temporal structure, but possibly unknown spatial structure. Neural models are candidates, however, training them in the presence of latent variables is challenging. We introduce *filtering variational objectives* (FIVOs), a tractable family of objectives for maximum likelihood estimation (MLE) in latent variable models with sequential structure.

Denote the observations by x, an $\mathcal{X}$-valued random variable. We assume that x was generated via an unobserved $\mathcal{Z}$-value random variable z and joint density $p(x, z)$ in some family $\mathcal{P}$. The goal of MLE is to recover $p \in \mathcal{P}$ that maximizes the marginal log-likelihood, $\log \left(\int p(x, z)\, dz\right)$. The difficulty is that the likelihood function is defined via an intractable integral.

To circumvent marginalization, it is common to optimize a variational lower bound (e.g., the ELBO) on the marginal log-likelihood (Jordan et al., 1999; Beal, 2003). The ELBO is defined by a variational posterior distribution $q(z|x)$[1],

$$\mathcal{L}(x, p, q) = \mathop{\mathbb{E}}_{q(z|x)} \left[\log \frac{p(x, z)}{q(z|x)} \right] \\ = \log p(x) - \mathrm{KL}(q(z|x) \parallel p(z|x)) \leq \log p(x) \ ,$$

and lower bounds the log marginal log-likelihood for any choice of $q(z|x)$. The bound is tight when $q(z|x)$ matches the true posterior $p(z|x)$. Thus, the joint optimum of $\mathcal{L}(x, p, q)$ in p and q is the MLE. In practice, we parameterize p and q and jointly optimize the ELBO over both sets of parameters with stochastic gradient ascent (Hoffman et al., 2013; Kingma & Welling, 2014; Rezende et al., 2014).

In practice, the family of variational posteriors is restricted for tractability (e.g., a factored distribution). Because of this, optimizing the ELBO tends to force the model's posterior to satisfy the factorizing assumptions of the variational family. One strategy for addressing this is to decouple the tightness of the bound from the quality of the variational posterior. For example, (Burda et al., 2016) observed that the typical ELBO is obtained as the log of an importance weight with proposal given by the variational posterior, and that using N samples from the same proposal produces a tighter bound, known as IWAE. The filtering variational objectives (FIVOs) build on this idea by treating a particle filter's marginal log-likelihood estimator as an objective function. It is well-known that a particle filter's marginal likelihood estimator has variance that scales more favourably than simple importance sampling for models with sequential structure (Cérou et al., 2011; Bérard et al., 2014). For this reason, we expect that FIVO will generally be a much tighter bound on the marginal likelihood.

Other approaches to learning in neural latent variable models include (Bornschein & Bengio, 2015), who use importance sampling to approximate gradients under the posterior, and (Gu et al., 2015), who use sequential Monte Carlo to approximate gradients under the posterior. These are distinct from our contribution in the sense that inference for the sake of estimation is the ultimate goal. To our knowledge the idea of treating the output of inference as an objective in and of itself, while not completely novel, has not

[*] Equal contribution [1] DeepMind, London, UK [2] University of Oxford, Oxford, UK [3] Google Brain, Mountain View, USA. Correspondence to: Chris Maddison, Dieterich Lawson, George Tucker <{cmaddis, dieterichl, gjt}@google.com>.

[1] An underlying assumption here is that q puts mass on any event with positive mass under p.

Algorithm 1 Simulating $\mathcal{L}^{\text{FIVO}}_{\text{N}}(x_{1:n}, p, q)$

```
FIVO($x_{1:n}, p, q, N$):
$\{w_0^i\}_{i=1}^N = \{1/N\}_{i=1}^N$
for $k \in \{1, \dots, n\}$ do
    for $i \in \{1, \dots, N\}$ do
        $z_k^i \sim q_k(z_k | x_{1:k}, z_{1:k-1}^i)$
        $z_{1:k}^i = \text{CONCAT}(z_{1:k-1}^i, z_k^i)$
    end for
    $\alpha_k(z_{1:k}^i) = \frac{p_k(x_k, z_k^i | x_{1:k-1}, z_{1:k-1}^i)}{q_k(z_k^i | x_{1:k}, z_{1:k-1}^i)}$
    $\hat{p}_k = \left(\sum_{i=1}^N w_{k-1}^i \alpha_k(z_{1:k}^i)\right)$
    $\hat{p}_N(x_{1:k}) = \hat{p}_N(x_{1:k-1})\hat{p}_k$
    $\{w_k^i\}_{i=1}^N = \{w_{k-1}^i \alpha_k(z_{1:k}^i)/\hat{p}_k\}_{i=1}^N$
    if resampling criteria satisfied by $\{w_k^i\}_{i=1}^N$ then
        $\{w_k^i, z_{1:k}^i\}_{i=1}^N = \text{RSAMP}(\{w_k^i, z_{1:k}^i\}_{i=1}^N)$
    end if
end for
return $\log \hat{p}_N(x_{1:n})$
RSAMP($\{w^i, z^i\}_{i=1}^N$):
for $i \in \{1, \dots, N\}$ do
    $a \sim \text{Categorical}(\{w^i\}_{i=1}^N)$
    $y^i = z^a$
end for
return $\{\frac{1}{N}, y^i\}_{i=1}^N$
```

been fully appreciated in the literature.

Also related to this work is the idea of increasing the expressiveness of the variational posterior family. For example, (Rezende & Mohamed, 2015; Kingma et al., 2016) augment the variational posterior with deterministic transformations with fixed Jacobians, (Salimans et al., 2015) extend the variational posterior to admit a Markov chain.

We note that the idea to optimize the log estimator of a particle filter was independently and concurrently considered in (Naesseth et al., 2017; Le et al., 2017). In (Naesseth et al., 2017) the bound we call FIVO is cast as a tractable lower bound on the ELBO defined by the particle filter's non-parameteric approximation to the posterior. (Le et al., 2017) additionally derive an expression for FIVO's bias as the KL between the filter's distribution and a certain target process.

2. Filtering Variational Objectives (FIVOs)

Let our observations be sequences of n $\mathcal{X}$-valued random variables denoted $x_{1:n}$, where $x_{i:j} \equiv (x_i, \dots, x_j)$ represents a sequence from x_i to x_j, inclusive. We also assume that the data generation process relies on a sequence of n unobserved $\mathcal{Z}$-valued latent variables denoted $z_{1:n}$. We focus on sequential latent variable models that factorize as a series of conditionals, $p(x_{1:n}, z_{1:n}) = p_1(x_1, z_1) \prod_{k=2}^n p_k(x_k, z_k | x_{1:k-1}, z_{1:k-1})$. An example is the hidden Markov model (HMM).

A particle filter is a sequential Monte Carlo algorithm which propagates a population of N weighted particles for n steps using a combination of importance sampling and resampling steps, see Algorithm 1.

The quantity $\hat{p}_N(x_{1:n})$ computed by Algorithm 1 is an unbiased, strongly consistent estimator of $p(x_{1:n})$ (Del Moral, 2004; 2013). By Jensen's inequality, $\mathbb{E}[\log \hat{p}_N(x_{1:n})] \leq \log p(x_{1:n})$. Thus, the basic idea behind FIVO is to treat $\mathbb{E}[\log \hat{p}_N(x_{1:n})]$ as an objective in an of itself.

Definition. Filtering Variational Objectives. Let $\log \hat{p}_N(x_{1:n})$ be the output of Algorithm 1 with inputs $(x_{1:n}, p, q, N)$, then $\mathcal{L}^{\text{FIVO}}_{\text{N}}(x_{1:n}, p, q) = \mathbb{E}[\log \hat{p}_N(x_{1:n})]$ is a filtering variational objective.

When $N = 1$, $\mathcal{L}^{\text{FIVO}}_{1}(x_{1:n}, p, q)$ reduces to the ELBO $\mathcal{L}(x_{1:n}, p, q)$. If we never resample, then FIVO reduces to IWAE. Crucially, the resampling step can dramatically decrease the relative variance of the estimator $\left(\text{var}\left(\frac{\hat{p}_N(x_{1:n})}{p(x_{1:n})}\right)\right)$ over simple importance sampling. In some scenarios, for example in an HMM, resampling reduces the scaling order of the relative variance from exponential in n to linear in n (Cérou et al., 2011; Doucet & Johansen, 2011). With some restrictions, we can show that the relative variance of the estimator is asymptotically related to the tightness of the objective $\mathcal{L}^{\text{FIVO}}_{\text{N}}(x_{1:n}, p, q)$.

Proposition. *Let $\hat{p}_N(x)$ be an unbiased positive estimator of $p(x)$. Let $g(N) = \mathbb{E}[(\hat{p}_N(x) - p(x))^6]$ be the 6th central moment. If the 1st inverse moment $\limsup \mathbb{E}[\hat{p}_N(x)^{-1}] < \infty$ is bounded, then*

$$\log p(x) - \mathcal{L}_{\text{N}}(x, p) = \frac{1}{2} \text{var}\left(\frac{\hat{p}_N(x)}{p(x)}\right) + \mathcal{O}(\sqrt{g(N)}).$$

Proof. See Appendix. □

Intuitively, resampling allows us to discard particles with low weight, and refocus the distribution of particles to regions of higher mass under the posterior. Resampling is a greedy process, so particles discarded at step k, could have attained a high mass at step n. Thus, we use the standard effective sample size (ESS) resampling criterion (Doucet & Johansen, 2011).

2.1. Optimization

FIVOs can be optimized with the same stochastic gradient framework used for the ELBO. We assume that p and q are parameterized in a differentiable way by θ and ϕ. If we do not make assumptions on the sampling process, then the score functions of each stochastic decision (sampling z_k^i and resampling indices a) scaled by the future learning signal provide unbiased estimators. If z_k^i are reparameterized, the gradient arising from sampling z_k^i flows through the lattice of states z_k^i created in the forward process (see

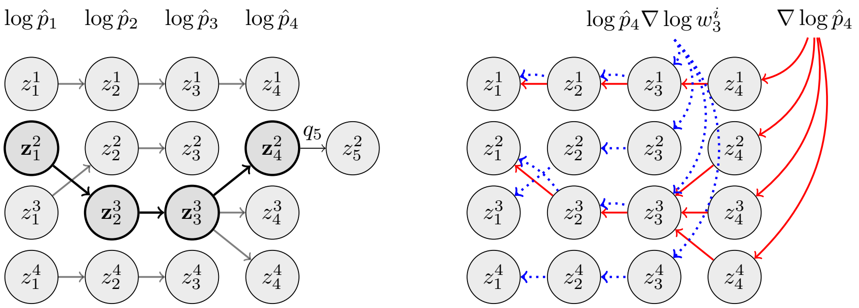
$\log \hat{p}_1$
$\log \hat{p}_2$
$\log \hat{p}_3$
$\log \hat{p}_4$
q_5
z_1^1 z_2^1 z_3^1 z_4^1
$\mathbf{z}_1^2$ z_2^2 z_3^2 $\mathbf{z}_4^2$ z_5^2
z_1^3 $\mathbf{z}_2^3$ $\mathbf{z}_3^3$ z_4^3
z_1^4 z_2^4 z_3^4 z_4^4
$\log \hat{p}_4 \nabla \log w_3^i$
$\nabla \log \hat{p}_4$

Figure 1: Visualizing FIVO. (left) The forward computation produces a lattice of latent states. (right) The backward gradient (in the reparameterized case) flows through that lattice, gradients from the objective at time 4 shown in solid red and resampling gradients at time 3 shown in dotted blue.

Figure 1). There are are also terms of the gradient corresponding to the adaptive resampling criteria decisions. In practice, we drop those terms as well as the resampling gradients $\nabla_{\theta,\phi} \log w_k^i$, which can add orders of magnitude to the variance of this gradient estimator. Thus, we found the best results were achieved by following just the reparameterization gradient $\nabla_{\theta,\phi} \log \hat{p}_N(x_{1:n})$. See the Appendix for the full gradient and further discussion.

3. Experiments

We sought to understand: (a) how optimizing the ELBO, IWAE, and FIVO bounds compare in terms of final model log-likelihoods, (b) whether there are differences in how the trained models use the stochastic state, and (c) how IWAE and FIVO scale with the number of particles. To explore these questions, we trained variational recurrent neural networks (VRNN) (Chung et al., 2015) with the ELBO, IWAE, and FIVO bounds on two benchmark sequential modeling tasks: modeling natural speech waveforms and modeling polyphonic music. When comparing to ELBO we increased the batch size by the number of particles given to IWAE or FIVO. To evaluate each model p, we computed $\mathcal{L}(x_{1:n}, p, q), \mathcal{L}_{64}^{\mathrm{IWAE}}(x_{1:n}, p, q), \mathcal{L}_{64}^{\mathrm{FIVO}}(x_{1:n}, p, q)$ and report the maximum, because all are stochastic lower bounds on the log likelihood. For training set performance see the Appendix.

3.1. Polyphonic Music

We evaluated VRNNs trained with the ELBO, IWAE, and FIVO bounds on 4 polyphonic music datasets: the Nottingham folk tunes, the JSB chorales, the MuseData library of classical piano and orchestral music, and the piano-midi.de MIDI archive (Boulanger-Lewandowski et al., 2012). We report bounds on average log likelihood per timestep.

Models trained on the FIVO bound significantly outperformed models trained with either the ELBO or the IWAE bounds on all four datasets (Table 1). In some cases, the improvements exceeded 1.0 nat *per timestep*, and in all cases, optimizing FIVO with $N = 4$ outperformed optimizing either IWAE or ELBO for $N = \{4, 8, 16\}$. A known pathology when training stochastic latent state models with the ELBO bound is that the stochastic states are unused, and as a result, the inference network collapses to the model (Bowman et al., 2015). To investigate this, we plot the KL divergence from $q(z_{1:n}|x_{1:n})$ to $p(z_{1:n})$ averaged over the dataset (Appnedix Figure 3). Indeed, the KL of models trained with ELBO collapsed during training, whereas the KL of models trained with FIVO remained high, even while achieving a higher log likelihood bound. In Figure 2, we also investigated how the log likelihood bound of models trained with IWAE and FIVO scaled with the number of particles, N. FIVO continued to benefit as N increased through $\{4, 8, 16\}$ while IWAE suffered diminishing returns.

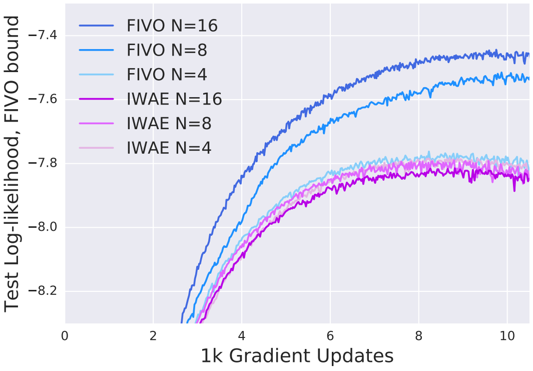

Figure 2: Learning curves comparing models trained with $\mathcal{L}_{\mathrm{N}}^{\mathrm{FIVO}}$ and $\mathcal{L}_{\mathrm{N}}^{\mathrm{IWAE}}$ for different N on the Piano-midi.de dataset.

N	Bound	Nottingham	JSB	MuseData	piano-midi
	ELBO	-3.23	-8.61	-7.12	-7.79
4	IWAE	-3.21	-8.59	-7.17	-7.81
	FIVO	**-2.86**	**-6.95**	**-6.55**	**-7.72**
	ELBO	-3.60	-8.60	-7.11	-7.83
8	IWAE	-3.30	-7.53	-7.10	-7.81
	FIVO	**-2.62**	**-6.69**	**-6.36**	**-7.49**
	ELBO	-3.54	-8.60	-7.17	-7.83
16	IWAE	-2.95	-7.55	-7.08	-7.81
	FIVO	**-2.58**	**-6.60**	**-6.09**	**-7.19**

		TIMIT	
N	Bound	64 units	512 units
	ELBO	35,908	36,981
4	IWAE	35,984	34,067
	FIVO	**40,211**	**41,834**
	ELBO	35,612	37,902
8	IWAE	36,835	38,074
	FIVO	**40,912**	**41,666**

Table 1: Test set lower bound on log likelihood comparison of models trained with ELBO, IWAE, and FIVO objectives and varying numbers of particles. Each set of rows delimited by a bar matches the computation between methods.

3.2. Speech

Next, we evaluated the bounds on a speech waveform dataset, TIMIT, a standard benchmark for sequential models that contains 6300 waveform utterances with an average duration of 3.1 seconds spoken by 630 different speakers. We report the average log likelihood bound per sequence. Again, models optimized with the FIVO bound significantly outperformed models optimized with IWAE or ELBO, see Table 1.

References

Beal, Matthew J. *Variational algorithms for approximate Bayesian inference*. 2003.

Bérard, Jean, Del Moral, Pierre, and Doucet, Arnaud. A lognormal central limit theorem for particle approximations of normalizing constants. *Electron. J. Probab.*, 19(94):1–28, 2014.

Bornschein, Jörg and Bengio, Yoshua. Reweighted wake-sleep. *ICLR*, 2015.

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

Bowman, Samuel R, Vilnis, Luke, Vinyals, Oriol, Dai, Andrew M, Jozefowicz, Rafal, and Bengio, Samy. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

Burda, Yuri, Grosse, Roger, and Salakhutdinov, Ruslan. Importance weighted autoencoders. *ICLR*, 2016.

Cérou, Frédéric, Del Moral, Pierre, and Guyader, Arnaud. A nonasymptotic theorem for unnormalized Feynman–Kac particle models. *Ann. Inst. H. Poincaré B*, 47(3):629–649, 2011.

Chung, Junyoung, Kastner, Kyle, Dinh, Laurent, Goel, Kratarth, Courville, Aaron C, and Bengio, Yoshua. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

Del Moral, Pierre. *Feynman-Kac formulae: genealogical and interacting particle systems with applications*. Springer Verlag, 2004.

Del Moral, Pierre. *Mean field simulation for Monte Carlo integration*. CRC Press, 2013.

Doucet, Arnaud and Johansen, Adam M. A tutorial on particle filtering and smoothing: Fifteen years later. In Crisan, D. and Rozovsky, B. (eds.), *The Oxford Handbook of Nonlinear Filtering*, pp. 656–704. Oxford University Press, 2011.

Fraccaro, Marco, Sønderby, Søren Kaae, Paquet, Ulrich, and Winther, Ole. Sequential neural models with stochastic layers. In *NIPS*, 2016.

Glorot, Xavier and Bengio, Yoshua. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Gu, Shixiang, Ghahramani, Zoubin, and Turner, Richard E. Neural adaptive sequential Monte Carlo. In *NIPS*, 2015.

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

Jordan, Michael I, Ghahramani, Zoubin, Jaakkola, Tommi S, and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. *ICLR*, 2015.

Kingma, Diederik P and Welling, Max. Auto-encoding variational Bayes. *ICLR*, 2014.

Kingma, Diederik P, Salimans, Tim, Jozefowicz, Rafal, Chen, Xi, Sutskever, Ilya, and Welling, Max. Improved variational inference with inverse autoregressive flow. In *NIPS*, 2016.

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

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

Rezende, Danilo Jimenez and Mohamed, Shakir. Variational inference with normalizing flows. *ICML*, 2015.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *ICML*, 2014.

Salimans, Tim, Kingma, Diederik, and Welling, Max. Markov chain Monte Carlo and variational inference: Bridging the gap. In *ICML*, 2015.

4. Appendix of Filtering Variational Objectives

4.1. Proof of Proposition

Let $\hat{p}_N(x)$ be an unbiased positive estimator of $p(x)$. Let $g(N) = \mathbb{E}[(\hat{p}_N(x) - p(x))^6]$ be the 6th central moment. If the 1st inverse moment $\limsup \mathbb{E}[\hat{p}_N(x)^{-1}] < \infty$ is bounded, then define the relative error

$$\Delta = \frac{\hat{p}_N(x) - p(x)}{p(x)} \tag{1}$$

Then the bias $\log p(x) - \mathcal{L}_N(x, p) = -\mathbb{E}[\log(1 + \Delta)]$. Now, Taylor expand $\log(1 + \Delta)$ about 0,

$$\log(1 + \Delta) = \Delta - \frac{1}{2}\Delta^2 + \int_0^\Delta \left(\frac{1}{1 + x} - 1 + x \right) dx \tag{2}$$

$$= \Delta - \frac{1}{2}\Delta^2 + \int_0^\Delta \left(\frac{x^2}{1 + x} \right) dx \tag{3}$$

and in expectation

$$-\mathbb{E}[\log(1 + \Delta)] = \frac{1}{2}\Delta^2 - \mathbb{E}\left[\int_0^\Delta \left(\frac{x^2}{1 + x} \right) dx \right] \tag{4}$$

Our aim is to show

$$\left| \mathbb{E}\left[\int_0^\Delta \frac{x^2}{1 + x}\, dx \right] \right| \in \mathcal{O}(g(N)^{1/2}) \tag{5}$$

In particular, by Cauchy-Schwarz

$$\left| \mathbb{E}\left[\int_0^\Delta \left(\frac{x^2}{1 + x} \right) dx \right] \right| \tag{6}$$

$$\leq \mathbb{E}\left[\left| \int_0^\Delta \frac{1}{(1 + x)^2}\, dx \right|^{1/2} \left| \int_0^\Delta x^4\, dx \right|^{1/2} \right] \tag{7}$$

$$= \mathbb{E}\left[\left| \frac{\Delta}{1 + \Delta} \right|^{1/2} \left| \frac{\Delta^5}{5} \right|^{1/2} \right] \tag{8}$$

$$= \mathbb{E}\left[\left| \frac{1}{1 + \Delta} \right|^{1/2} \left| \frac{\Delta^6}{5} \right|^{1/2} \right] \tag{9}$$

and again by Cauchy-Schwarz

$$\leq \left(\mathbb{E}\left[\left| \frac{1}{1 + \Delta} \right| \right] \right)^{1/2} \left(\mathbb{E}\left[\frac{\Delta^6}{5} \right] \right)^{1/2} \tag{10}$$

and we're done.

4.2. Gradients of $\mathcal{L}_{\mathrm{N}}^{\mathrm{FIVO}}(x_{1:n}, p, q)$

We formulate unbiased gradients of $\mathcal{L}_{\mathrm{N}}^{\mathrm{FIVO}}(x_{1:n}, p, q)$ by considering Algorithm 1 as a method for simulating a FIVO. We consider the cases when the sampling of z_k^i is and is not reparameterized. We also consider the case where we make adaptive resampling decisions.

First, we assume that the decision to resample is not adaptive (i.e., depends on some way on the random variables already produced until that point in Algorithm 1), and are fixed ahead of time. When the sampling z_k^i is not reparameterized there are three terms to the gradient: (1) the gradients of $\log \hat{p}_N(x_{1:n})$ with respect to the parameters conditional on the latent states, (2) gradients of the densities q_k with respect to their parameters, and (3) gradients of the resampling probabilities with respect to the parameters. All together, the following is an unbiased gradient of FIVO,

$$\nabla_{\theta,\phi} \log \hat{p}_N(x_{1:n}) + \sum\nolimits_{k=1}^{n} \sum\nolimits_{i=1}^{N} \Big(\log \frac{\hat{p}_N(x_{1:n})}{\hat{p}_N(x_{1:k-1})} \nabla_\phi \log q_{k,\phi}(z_k^i | x_{1:k}, z_{1:k-1}^i) + \mathbb{I}(\text{resampling at step } k) \log \frac{\hat{p}_N(x_{1:n})}{\hat{p}_N(x_{1:k})} \nabla_{\theta,\phi} \log w_k^i \Big) \tag{11}$$

where $\mathbb{I}(A)$ is an indicator function. If z_k^i is reparameterized, then the first and third terms suffice for an unbiased gradient,

$$\nabla_{\theta,\phi} \log \hat{p}_N(x_{1:n}) + \sum\nolimits_{k=1}^{n} \Big(\mathbb{I}(\text{resampling at step } k) \sum\nolimits_{i=1}^{N} \log \frac{\hat{p}_N(x_{1:n})}{\hat{p}_N(x_{1:k})} \nabla_{\theta,\phi} \log w_k^i \Big) \tag{12}$$

In this work we only considered reparameterized q_ks, and we dropped the terms of the gradient that arise from resampling.

Second, when the decision to resample is adaptive, the domain of the random variables involved in simulating $\log \hat{p}_N(x_{1:n})$ can be partitioned into 2^n regions, over each of which the density is differentiable. Between those regions, the density experiences a jump discontinuity. Thus, there are additional terms to the gradient of $\mathcal{L}_{\mathrm{N}}^{\mathrm{FIVO}}(x_{1:n}, p, q)$ that correspond to the change in the regions of continuity as the parameters change. These terms can be written as surface integrals over the boundaries of the regions. We drop these terms in practice.

4.3. Implementation details

4.3.1. VRNN MODEL

The VRNN is a sequential latent variable model that combines a deterministic recurrent neural network (RNN) with stochastic latent states z_k at each step. The observation distribution x_k is conditioned directly on z_k and indirectly on $z_{1:k-1}$ via the RNN's state $h_k(z_{k-1}, x_{k-1}, h_{k-1})$. For a length n sequence the model's posterior factorizes into the

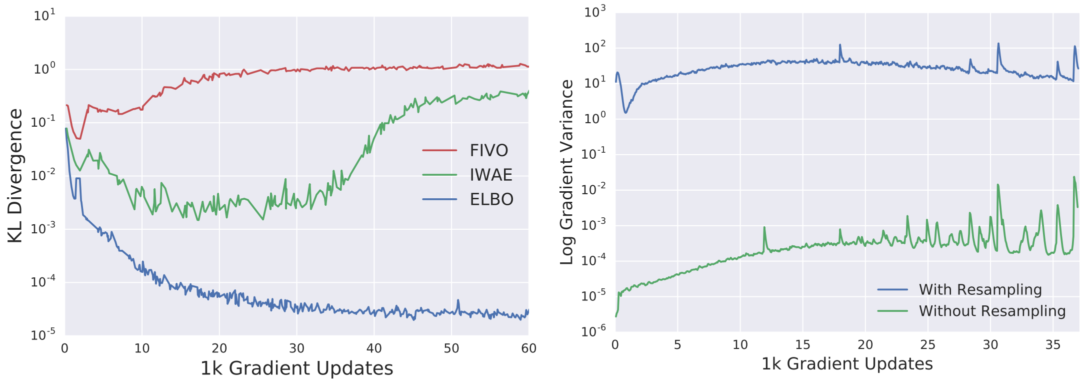

Figure 3: (Left) The KL divergence between $p(z_{1:n})$ and $q(z_{1:n}|x_{1:n})$ on the JSB chorales dataset with $N = 16$. (Right) Variance of FIVO gradients with and without resampling terms along the trajectory generated by a training run trained without resampling terms. The variance of the gradients with resampling terms is several orders of magnitude larger than the gradients without resampling terms, making it difficult to train with the resampling terms. These curves are generated from training on the JSB chorales.

conditionals

$$p_1(z_1)q_1(x_1|z_1) \tag{13}$$

$$\prod_{k=2}^{n} \Big(p_k(z_k|h_k(z_{k-1}, x_{k-1}, h_{k-1})) \tag{14}$$

$$g_k(x_k|z_k, h_k(z_{k-1}, x_{k-1}, h_{k-1}))\Big). \tag{15}$$

Similarly the variational posterior factorizes as

$$q_1(z_1|x_1) \prod_{k=2}^{n} q_k(z_k|h_k(z_{k-1}, x_{k-1}, h_{k-1}), x_k). \tag{16}$$

The latent variables at each step are factorized Gaussians, and the observation distributions depend on the dataset (Bernoulli for binary data and Gaussian for continuous data). The RNN is a single-layer LSTM and the conditionals are parameterized by fully connected neural networks with one hidden layer of the same size as the LSTM hidden layer. We used the residual parameterization (Fraccaro et al., 2016) for the variational posterior.

We initialized weights using the Xavier initialization (Glorot & Bengio, 2010) and used the Adam optimizer (Kingma & Ba, 2015) with a batch size of 4. We performed a grid search over learning rates $\{3 \times 10^{-4}, 1 \times 10^{-4}, 3 \times 10^{-5}, 1 \times 10^{-5}\}$ and picked the run and early stopping step by the validation performance. During training, we did not truncate sequences and performed full backpropagation through time.

To reduce the variance from the gradient terms arising from the resampling events, we used a linear baseline in the number of remaining timesteps. Still, we found that the unbiased FIVO gradients had high variance. This variance is almost entirely due to the gradients corresponding to resampling events, accounting for 6 orders of magnitude (Appendix Figure 3). Thus, we report results using only the first term of Eq. (11) to compute gradient estimates.

4.3.2. Polyphonic Music

We evaluated VRNNs trained with the ELBO, IWAE, and FIVO bounds on 4 polyphonic music datasets: the Nottingham folk tunes, the JSB chorales, the MuseData library of classical piano and orchestral music, and the piano-midi.de MIDI archive (Boulanger-Lewandowski et al., 2012). Each dataset is split into standard train, valid, and test sets and is represented as a sequence of 88-dimensional vectors denoting the notes active at the current timestep. We mean-centered the input data, and we modeled the output as a set of 88 factorized Bernoulli variables. We initialized the output biases of the VRNN to the training set statistics. For the results reported in Table 1, the Nottingham model used 64 units, the JSB Chorales model used 32 units, the MuseData model used 256 units, and the piano-midi.de model used 64 units. We report bounds on average log likelihood per timestep.

4.3.3. TIMIT

The TIMIT dataset is a standard benchmark for sequential models that contains 6300 utterances with an average duration of 3.1 seconds spoken by 630 different speakers. The 6300 utterances are divided into a training set of size 4620 and a test set of size 1680. We further divide the training set into a validation set of size 231 and a training set of size 4389, with the splits exactly as in (Fraccaro et al., 2016). Each TIMIT utterance is represented as a sequence of real-valued amplitudes which we split into a sequence of

		Nottingham		JSB Chorales		MuseData		Piano-MIDI.de	
N	Bound	Train	Test	Train	Test	Train	Test	Train	Test
4	ELBO	-3.03	-3.23	-5.38	-8.61	-5.42	-7.12	-7.06	-7.79
	IWAE	-3.02	-3.21	-5.23	-8.59	-5.22	-7.17	-7.18	-7.81
	FIVO	**-2.25**	**-2.86**	**-4.22**	**-6.95**	**-5.16**	**-6.55**	**-6.32**	**-7.72**
8	ELBO	-3.04	-3.60	-6.10	-8.60	-5.93	-7.11	-7.33	-7.83
	IWAE	-3.15	-3.30	-6.18	-7.53	-5.71	-7.10	-6.71	-7.81
	FIVO	**-1.98**	**-2.62**	**-5.10**	**-6.69**	**-5.47**	**-6.36**	**-6.22**	**-7.49**
16	ELBO	-3.39	-3.54	-6.10	-8.60	-6.18	-7.17	-7.23	-7.83
	IWAE	-2.18	-2.95	-4.60	-7.55	-5.74	-7.08	-7.04	-7.81
	FIVO	**-2.12**	**-2.58**	**-4.42**	**-6.60**	**-5.58**	**-6.09**	**-6.44**	**-7.19**

Table 2: Performance of the VRNN on the polyphonic music datasets trained with different bounds and numbers of particles.

		TIMIT			
		64 units		512 units	
N	Bound	Train	Test	Train	Test
4	ELBO	36,095	35,908	35,765	36,981
	IWAE	35,519	35,984	36,833	34,067
	FIVO	**39,636**	**40,211**	**40,940**	**41,834**
8	ELBO	35,617	35,612	38,467	37,902
	IWAE	35,822	36,835	37,161	38,074
	FIVO	**40,019**	**40,912**	**40,963**	**41,666**

Table 3: Performance of the VRNN on the TIMIT dataset trained with different bounds and numbers of particles.

200-dimensional frames, as in (Chung et al., 2015), (Fraccaro et al., 2016). Data preprocessing was limited to mean centering and variance normalization as in (Fraccaro et al., 2016). For TIMIT, the output distribution was a factorized Gaussian, and we report the average log likelihood bound per sequence.