Resampled Proposal Distributions for Variational Inference and Learning

Aditya Grover [*1] **Ramki Gummadi** [*2] **Miguel Lazaro-Gredilla** [2] **Dale Schuurmans** [3] **Stefano Ermon** [1]

Abstract

Learning generative models of unlabelled data using black-box stochastic variational inference leads to learning objectives involving intractable expectations and hence, critically relies on efficient Monte Carlo evaluation of the gradients. This task is challenging when the approximate posterior is far from the true posterior, due to high variance in the gradient estimates. In this paper, we demonstrate that resampling proposed samples from the variational posterior which are assigned low likelihoods by the model (and hence, unrepresentative of the true posterior) can improve learning and trade-off extra computation for accuracy adaptively. We show that explicitly rejecting samples, while technically challenging to analyze due to the implicit nature of the resulting unnormalized proposal distribution, can have benefits over the competing state-of-the-art alternatives based on multi-sample objectives. We evaluate the proposed approach and demonstrate its effectiveness in comparison to state-of-the-art alternatives both via experiments on synthetic data and a benchmark density estimation task with sigmoid belief networks over the MNIST dataset.

1. Introduction

Black-box stochastic variational learning and inference in deep generative models with latent variables provides an effective mechanism for probabilistic reasoning over massive amounts of unlabelled data (Hoffman et al., 2013; Ranganath et al., 2013). Typically, inference in such models is *amortized* by introducing a recognition model that expresses the variational posterior over the latent variables conditioned on the observed data (Dayan et al., 1995; Gershman & Goodman, 2014). The generative and recognition models are commonly parameterized using deep neural networks which provides expressiveness, but leads to intractable expectations in the learning objective.

Unless the model and its latent variable space are appropriately reparametrizable (Kingma & Welling, 2014; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014), the general approach to evaluating and optimizing such intractable objectives involves Monte Carlo estimation of gradients using the recognition network as a proposal (Mnih & Rezende, 2016). A simple feed forward network, however, may not capture the full complexity of the posterior distribution, a difficulty which shows up in practice as high variance in the gradient estimates. While prior work has made significant progress on this issue, for example, (Mnih & Gregor, 2014; Titsias & Lázaro-Gredilla, 2015; Mnih & Rezende, 2016), it can often still be a formidable challenge even in simple cases, as we shall illustrate in this paper.

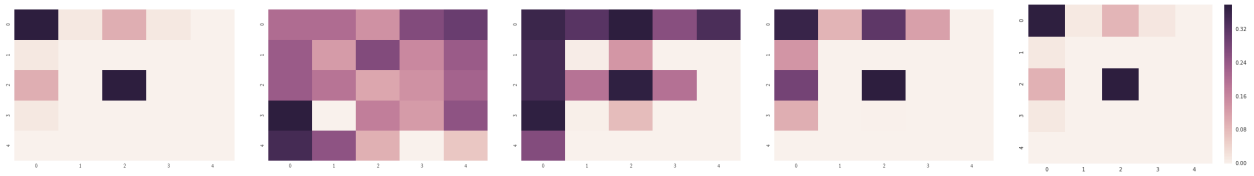
In this paper, we propose an accept-reject method that expands the class of distributions representable by the variational posterior. The probability of accepting a sample proposed by the recognition network in our framework depends on the likelihood assigned by the generative network. This leads to an implicit modification of the original variational posterior to a much richer family of approximating distributions that can be controlled based on the available computation. While our proposed framework and analysis is general, we focus on variational approximations to discrete distributions which are considerably more challenging since the reparameterization trick is inapplicable.

2. Resampling framework

Consider a generative model with a joint distribution $p_\theta(\mathbf{x}, \mathbf{h})$ parameterized by θ. Here, $\mathbf{x}$ and $\mathbf{h}$ denote the observed and latent variables respectively. Since the true posterior over the latent variables is intractable, we introduce a variational approximation to the posterior $r_\phi(\mathbf{h}|\mathbf{x})$ represented by a recognition network and parameterized by ϕ. The parameters of the generative model and the recognition network are learned jointly by optimizing an evidence lower bound (ELBO) on the marginal log-likelihood of the

[*]Equal contribution [1]Stanford University, California, USA [2]Vicarious FPC Inc., California, USA [3]University of Alberta, Alberta, Canada. Correspondence to: Aditya Grover <adityag@cs.stanford.edu>, Ramki Gummadi <ramki@vicarious.com>.

Presented at the ICML 2017 Workshop on Principled Approaches to Deep Learning, Sydney, Australia, 2017.



(a) Target distribution (b) $T, \alpha, \texttt{KL} : \infty, 1, 18$ (c) $10, 0.5, 3.1$ (d) $0, 0.2, 0.3$ (e) $-5, 0.01$,1e-3

Figure 1. The resampled posterior approximation (b-e) gets closer (in terms of KL divergence) to a target 2D discrete distribution (a) as we decrease the parameter T, which controls the acceptance probability α. The triples shown are $T, \alpha, \texttt{KL}$ divergence to target.

observed data $\mathbf{x}$,

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_r \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{h})}{r_\phi(\mathbf{h}|\mathbf{x})} \right] \triangleq ELBO(\theta, \phi). \quad (1)$$

In the resampling framework, we wish to express the variational posterior as $q_{\theta,\phi}(\mathbf{h}|\mathbf{x})$ defined implicitly using a proposal distribution $r_\phi(\mathbf{h}|\mathbf{x})$ (represented by a recognition network as before) and a factor corresponding to an acceptance probability that could depend on both the generative and the recognition parameters, as $a_{\theta,\phi}(\mathbf{h}|\mathbf{x}) \in (0, 1]$. Note that, unlike p, q, r, $a_{\theta,\phi}(\mathbf{h}|\mathbf{x})$ does not represent a distribution on the latent variable space $\mathbf{h}$, but simply a probability for each $\mathbf{h}$. This results in an approximate posterior, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x})$ that is proportional to $r_\phi(\mathbf{h}|\mathbf{x})a_{\theta,\phi}(\mathbf{h}|\mathbf{x})$. While there may be many possibilities for choosing $a_{\theta,\phi}(\mathbf{h}|\mathbf{x})$, in this paper, we instantiate a specific choice of the acceptance probability function in conjunction with a scalar "threshold", T (that could, in general, depend on $\mathbf{x}$). The resulting approximate posterior, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$ is defined in Algorithm 1.

Algorithm 1 Sampling definition of $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$, given $p_\theta(\mathbf{x}, \mathbf{h})$, $r_\phi(\mathbf{h}|\mathbf{x})$, and T.

while True **do**
  $h \leftarrow$ sample from proposal $r_\phi(\mathbf{h}|\mathbf{x})$.
  Compute negative acceptance log probability, λ as:

$$\lambda = \log(1 + e^{l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)}), \text{ where:}$$

$$l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \triangleq \log r_\phi(\mathbf{h}|\mathbf{x}) - \log p_\theta(\mathbf{x}, \mathbf{h}) - T$$

  Sample uniform: $u \sim U[0, 1]$.
  if $u < e^{-\lambda}$ **then**
    Output sample $\mathbf{h}$.
  end if
end while

Within a normalizing constant, the approximate posterior $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$ can be characterized as the product of the proposal distribution, $r_\phi(\mathbf{h}|\mathbf{x})$, with the acceptance probability, whose negative log likelihood was fixed to be $\log(1 + e^{l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)})$, i.e., the softplus function applied to $l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$. We summarize this observation below:

Proposition 1. *The approximate posterior, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$, from the sampler defined in Algorithm 1 is given by $\gamma_q(\mathbf{h}|\mathbf{x}, T)/Z_q(\mathbf{x}, T)$ (for fixed x and T) where $Z_q(\mathbf{x}, T) \triangleq \sum_{\mathbf{h}} \gamma_q(\mathbf{h}|\mathbf{x}, T)$ is an appropriate normalization constant and $\gamma_q(\mathbf{h}|\mathbf{x}, T)$ is defined via:*

$$\log \gamma_q(\mathbf{h}|\mathbf{x}, T) = \log r_\phi(\mathbf{h}|\mathbf{x}) - [l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)]^+ \quad (2)$$

where $l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) = \log r_\phi(\mathbf{h}|\mathbf{x}) - \log p_\theta(\mathbf{x}, \mathbf{h}) - T$ and $[*]^+$ denotes the softplus function, i.e., $\log(1 + e^*)$.

2.1. Approximation Quality Versus Runtime

Informally, the resampling scheme of Algorithm 1 enforces the following behavior: samples from the approximate posterior that disagree (as measured by the log-likelihoods) with the target posterior beyond a level implied by the corresponding threshold have an exponentially decaying probability of getting accepted, while leaving the remaining samples with negligible interference from resampling. When the proposed sample $\mathbf{h}$ from r_ϕ has a small enough value according to p_θ, it is likely that λ is large (and linear in the negative log-likelihood assigned by p_θ), resulting in a low acceptance probability. However, when the same is small, λ is close to 0, resulting in an acceptance probability close to 1. Therefore, a large value of T recovers the regular variational inference framework as a special case since the resulting sampler is identical to $r_\phi(\mathbf{h}|\mathbf{x})$, due to the lack of any rejections. On the other extreme, for a small value of T, we get the behavior of a rejection sampler with high computational cost that is also close to the target distribution in KL divergence. More formally, we have Theorem 1 which shows that the KL divergence can be improved monotonically by cranking down T. However, a smaller value of T would require more aggressive rejections and thereby, more computation.

Theorem 1. *For fixed θ, ϕ, the KL divergence between the approximate and true posteriors, $\texttt{KL}(q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \| p_\theta(\mathbf{h}|\mathbf{x}))$ is monotone in T. Furthermore, the behavior of the sampler in Algorithm 1 interpolates between the following two extremes:*

- *As $T \to +\infty$, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$ approximates $r_\phi(\mathbf{h}|\mathbf{x})$ with perfect sampling efficiency.*

- *As $T \to -\infty$, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$ approximates $p_\theta(\mathbf{h}|\mathbf{x})$, with the sampling efficiency equivalent to a naive rejection sampler.*

This phenomenon is illustrated in Figure 1 where we approximate an example 2D discrete target distribution on a 5×5 grid, with a uniform proposal distribution plus resampling. With no resampling ($T = \infty$), the approximation is far from the target. Figure 1 demonstrates progressive improvement in the posterior quality as T is reduced (both visually as well as via an estimate of the KL divergence from approximation to the target), along with an increasing computation cost reflected in the lower acceptance probabilities.

2.2. The Resampled ELBO (R-ELBO)

We may now consider optimizing the evidence lower-bound on the log-likelihood corresponding to the implicit resampled posterior, $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$, rather than the usual objective corresponding to the original unmodified proposal distribution, $r_\phi(\mathbf{h}|\mathbf{x})$. To avoid confusion with the latter, we refer to this modified tighter objective function as the "resampled ELBO" or R-ELBO, defined below,

$$\log p_\theta(\mathbf{x}) \geq \text{R-ELBO} \triangleq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{h}) Z_q(\mathbf{x}, T)}{\gamma_q(\mathbf{h}|\mathbf{x}, T)} \right] \tag{3}$$

$$\log p_\theta(\mathbf{x}) \geq \text{R-ELBO} \triangleq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{h}) Z_q(\mathbf{x}, T)}{r_\phi(\mathbf{h}|\mathbf{x}) a_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)} \right] \tag{4}$$

$$\log p_\theta(\mathbf{x}) \geq \text{R-ELBO} \triangleq \mathbb{E}_q \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{h}) Z_q(\mathbf{x}, T)}{r_\phi(\mathbf{h}|\mathbf{x}) a_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)} \right] \tag{5}$$

where γ_q and Z_q were defined in Proposition 1. Alternatively, we can express the R-ELBO as,

$$\text{R-ELBO} = \log p_\theta(\mathbf{x}) - \text{KL}(q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \| p_\theta(\mathbf{h}|\mathbf{x})). \tag{6}$$

Using Eq. (6) and Theorem 1, we get the corollary below.

Corollary 1. *The R-ELBO gets tighter by decreasing T (but more expensive to compute).*

Even though the R-ELBO expression has an unknown constant Z_q, its gradients can be written as the covariance of two random variables that are a function of the latent variables sampled from the approximate posterior $q_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)$. Hence, we only need access to samples from q for learning, which can be done using Monte Carlo analogous to the usual ELBO gradients. For this, a generalization of the usual ELBO gradients to arbitrary unnormalized proposal distributions is necessary, which is done via Lemma 1 in the appendix. The resulting R-ELBO gradients are summarized below in Theorem 2.

Theorem 2. *Let $\text{COV}_q(A(\mathbf{h}), B(\mathbf{h}))$ denote the covariance of the two random variables $A(\mathbf{h})$ and $B(\mathbf{h})$, where $\mathbf{h}$ is sampled from the distribution q. Then, the R-ELBO gradients with respect to θ and ϕ are given by:*

- The R-ELBO gradients with respect to θ are given by,

  $$\nabla_\phi \text{R-ELBO}(\theta, \phi) = \text{COV}_q \left(A_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T), B_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \right)$$

  where the covariance is between the following r.v.,

  $$A_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \triangleq \log p_\theta(\mathbf{x}, \mathbf{h}) - \log r_\phi(\mathbf{h}|\mathbf{x}) - [l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)]^+$$
  $$B_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \triangleq (1 - \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T))) \nabla_\phi \log r_\phi(\mathbf{h}|\mathbf{x}))$$

- The R-ELBO gradients with respect to ϕ are given by,

  $$\nabla_\theta \text{R-ELBO}(\theta, \phi) = \mathbb{E}_q \left[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{h}) \right] - \text{COV}_q(A_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T), \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{h}))$$

To compute an unbiased Monte Carlo estimate of the covariance, we need to subtract the mean of at least one random variable while forming the product term. To do this, we process a fixed batch of (accepted) samples per gradient update, and for each sample, use all-but-one to compute the mean estimate to be subtracted, a trick reminiscent of, and inspired by the local learning signals proposed in Mnih & Rezende (2016) (see Sec. 2.5.3 in the paper for details).

3. Related work

For continuous distributions, there are several works that attempt to improve the variational approximation to the posterior. A few prominent ones include normalizing flow models (Rezende et al., 2014; Kingma et al., 2016), Hamiltonian variational inference (Salimans et al., 2015), auxiliary generative models (Maaløe et al., 2016). Variance reduction is largely achieved through the reparameterization trick for location-scale family of distributions (Kingma & Welling, 2014; Rezende et al., 2014; Titsias & Lázaro-Gredilla, 2014). Recently, Naesseth et al. (2017) proposed to use an accept-reject method to extend the reparameterization trick to the gamma and Dirichlet distributions.

Prior work for the case of discrete variational distributions is relatively scarce. NVIL (Mnih & Gregor, 2014) uses REINFORCE (Williams, 1992) with baselines to reduce the variance in gradient estimates. On the theoretical side, random projections of the posterior distribution have been shown to provide tight bounds on the quality of the variational approximation (Grover & Ermon, 2016). Hierarchical variational models impose a prior over the latent variables to induce dependencies between the variables (Ranganath et al., 2016). Recently, the concrete distribution was proposed to obtain low variance gradients through a continuous relaxation of the discrete distribution using Gumbel variables (Maddison et al., 2016; Jang et al., 2016).

A common theme for learning both discrete and continuous variational distributions is the use of multi-sample objectives. Such objectives were first proposed by Raiko et al. (2015) for structured prediction. Burda et al. (2016) showed that the multi-sample objectives are in fact tighter lower bounds on the log-likelihood and used similar objectives for training variational autoencoders with continuous latent units. Finally, VIMCO extended the same to discrete latent variable models (Mnih & Rezende, 2016). Similar to our proposed framework, multi-sample objectives permit a computational-statistical trade-off by varying the number of samples used to compute the Monte Carlo estimate. We compare the two approaches in detail in the experiments.

4. Experiments

The setup and hyperparameter details for the experiments beyond those mentioned below are described in Appendix.

4.1. Diagnostic experiments

Consider a discrete 1-D target distribution, $p_{\lambda^*,c,\epsilon}(h)$, illustrated in Figure 2, with support $h \in \{0, 1, \ldots\}$, obtained by forcing a negligible mass, $\epsilon \to 0$, $0 \leq h < c$ on $\texttt{Poi}(\lambda^*)$, a Poisson distribution with rate $\lambda^* > 0$. We focus solely on the dynamics of learning a variational parameter ϕ, and consider $\theta \triangleq \lambda^*, c, \epsilon$ as fixed for simplicity. The approximate proposal is parameterized as $r_\phi \triangleq \texttt{Poi}(e^\phi)$, where ϕ is an unconstrained scalar, and denotes a (unmodified) Poisson distribution with the (non-negative) rate parameter, e^ϕ. Note that for $\texttt{Poi}(e^\phi)$ to explicitly represent a small mass on $h < c$ would require $\phi \to \infty$. As a result, $\{r_\phi\}$ does not contain candidates close to the target distribution in the sense of KL divergence, even while it may be possible to approximate well with a simple resampling modification that transforms the raw proposal r_ϕ into a better candidate.

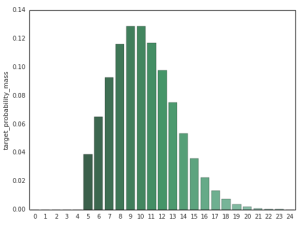


Figure 2. Target distribution, p, with rate $\lambda^* = 10, c = 5, \epsilon = 1\text{e}{-20}$.

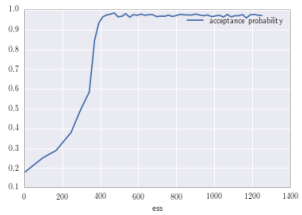
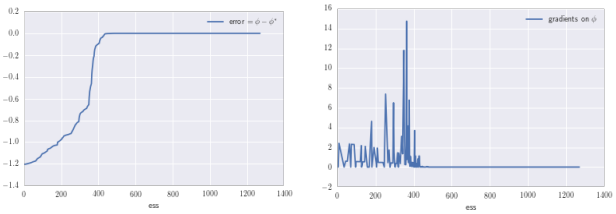


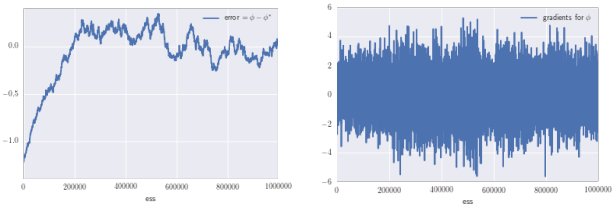
Figure 3. Acceptance probability at each SGD iteration

In Figures 3 and 4 we illustrate the performance of our approach for an example setting (details in appendix). Figure 3 shows the efficiency of the sampler automatically improving as the learning progresses. Figure 4a shows the difference between the current parameter ϕ and the known optimal $\phi^* = \log \lambda^*$ quickly converging to 0 as learning proceeds. As a benchmark, we also evaluated VIMCO, which optimizes a multi-sample version of the ELBO. Figure 5 suggests that the signal in gradients is too low (i.e., high variance in gradient estimates), as a possible cause of the observed behavior with VIMCO, which was persistent with much smaller learning rates and large sample sizes compared to resampling. One explanation is that the VIMCO gradient update for ϕ has a term that assigns the same average weight to the entire batch of samples, both good and bad ones (see Eq.(8) in Mnih & Rezende (2016)). By contrast, Algorithm 1 discards rejected sample proposals from contributing to the gradients explicitly. Yet another qualitative aspect that distinguishes our approach from typical multi-sample objectives is that Algorithm 1 can adapt the effective sample size dynamically based on current sample quality, as opposed to being fixed in advance.



(a) Error: $\phi - \phi^*$ (b) Gradients for ϕ.

Figure 4. Resampling learning dynamics. The x-axis shows the effective sample size (ess), which includes both accepted and rejected samples at each SGD iteration.



(a) Error: $\phi - \phi^*$. (b) Gradients for ϕ.

Figure 5. VIMCO learning dynamics. The x-axis shows the effective sample size (ess), which is equal to k times the number of iterations at each SGD iteration.

Table 1. Test NLL (in nats) for MNIST. NVIL results from Mnih & Gregor (2014), VIMCO results from Mnih & Rezende (2016).

SBN Architecture	200-200-200
NVIL	96.7
VIMCO (k=5)	92.8
VIMCO (k=10)	92.6
Resampled-SBN	**91.9**

4.2. MNIST benchmark experiments

We preformed density estimation by training sigmoid belief networks using the R-ELBO objective on the binarized MNIST dataset. We compare the proposed Resampled-SBN with NVIL (Mnih & Gregor, 2014) and VIMCO (Mnih & Rezende, 2016), described in Section 3. The results are shown in Table 1. To keep computation budget roughly the same when comparing against VIMCO, we report results corresponding to thresholds where the average acceptance probability of the Resampled-SBN is between 0.05 and 0.10. As we can see, Resampled-SBN performs favorably compared to the other competing methods.

5. Conclusion

We proposed a resampling framework for variational inference and learning in generative models that is theoretically principled and allows for flexible trade-off between computation and statistical accuracy by improving the quality of the variational approximation made by any parameterized model. We demonstrated the practical benefits of our framework over competing alternatives based on multi-sample objectives. In the future, we would want to extend this for tasks beyond density estimation and also to exploit the factorization structure in sparse graphical models/layered networks for better efficiency. Yet another direction involves combining the proposed approach with multi-sample objectives.

References

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

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

Gershman, Sam and Goodman, Noah D. Amortized inference in probabilistic reasoning. In *Proceedings of the Thirty-Sixth Annual Conference of the Cognitive Science Society*, 2014.

Grover, Aditya and Ermon, Stefano. Variational Bayes on Monte Carlo steroids. In *NIPS*, 2016.

Hoffman, Matthew D, Blei, David M, Wang, Chong, and Paisley, John. Stochastic variational inference. *JMLR*, 14(1):1303–1347, 2013.

Jang, Eric, Gu, Shixiang, and Poole, Ben. Categorical reparameterization with Gumbel-softmax. In *ICLR*, 2016.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. In *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.

Maaløe, Lars, Sønderby, Casper Kaae, Sønderby, Søren Kaae, and Winther, Ole. Auxiliary deep generative models. In *ICML*, 2016.

Maddison, Chris J, Mnih, Andriy, and Teh, Yee Whye. The concrete distribution: A continuous relaxation of discrete random variables. In *ICLR*, 2016.

Mnih, Andriy and Gregor, Karol. Neural variational inference and learning in belief networks. In *ICML*, 2014.

Mnih, Andriy and Rezende, Danilo J. Variational inference for monte carlo objectives. In *ICML*, 2016.

Naesseth, Christian, Ruiz, Francisco, Linderman, Scott, and Blei, David. Reparameterization gradients through acceptance-rejection sampling algorithms. In *AISTATS*, 2017.

Raiko, Tapani, Berglund, Mathias, Alain, Guillaume, and Dinh, Laurent. Techniques for learning binary stochastic feedforward neural networks. In *ICLR*, 2015.

Ranganath, Rajesh, Gerrish, Sean, and Blei, David M. Black box variational inference. In *AISTATS*, 2013.

Ranganath, Rajesh, Tran, Dustin, and Blei, David. Hierarchical variational models. In *ICML*, 2016.

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

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

Titsias, Michalis and Lázaro-Gredilla, Miguel. Doubly stochastic variational Bayes for non-conjugate inference. In *ICML*, 2014.

Titsias, Michalis and Lázaro-Gredilla, Miguel. Local expectation gradients for black box variational inference. In *NIPS*, 2015.

Williams, Ronald J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Appendix

A. Proofs of theoretical results

A.1. Proposition 1

Proof. The probability of an accepted sample is proportional to the product of proposing it and accepting it, which is given by $r_\phi(\mathbf{h}|\mathbf{x})e^{-[l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+}$. Taking logarithms finishes the proof. □

A.2. Theorem 1

Proof. We can explicitly write down the acceptance probability function as,

$$\begin{aligned} a_{\theta,\phi}(\mathbf{h}|\mathbf{x},T) &= e^{-[l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+} \\ &= \frac{e^T p_\theta(\mathbf{x},\mathbf{h})}{e^T p_\theta(\mathbf{x},\mathbf{h}) + r_\phi(\mathbf{h}|\mathbf{x})} \end{aligned}$$

From the above equation, it is easy to see that as $T \to \infty$, we get an acceptance probability close to 1, resulting in an approximate posterior close to the original proposal, $r_\phi(\mathbf{h}|\mathbf{x})$, whereas with $T \to -\infty$, the acceptance probability degenerates to a standard rejection sampler with acceptance probability close to $e^T \frac{p_\theta(\mathbf{x},\mathbf{h})}{r_\phi(\mathbf{h}|\mathbf{x})}$, but with potentially untenable efficiency. Intermediate values of T can interpolate between these two extremes.

To prove monotonicity, we first derive the partial derivative of the KL divergence with respect to T as a covariance of two random variables that are monotone transformations of each other. To get the derivative, we use the fact that the gradient of the KL divergence is the negative of the ELBO gradient derived in Theorem 1. Recall that the ELBO and the KL divergence add up to a constant independent of T, and that the expressions for the gradients with respect to T and ϕ are functionally the same. We have,

$$\nabla_T \mathtt{KL}(q\|p) = -\mathtt{COV}_q\left(A(\mathbf{h}), \nabla_T \log \gamma_q(\mathbf{h})\right),$$

where,

$$\begin{aligned} A(\mathbf{h}) &= \log p_\theta(\mathbf{x},\mathbf{h}) - \log \gamma_q(\mathbf{h}) \\ &= \log p_\theta(\mathbf{x},\mathbf{h}) - \log r_\phi(\mathbf{h}|\mathbf{x}) \\ &\quad + [\log r_\phi(\mathbf{h}|\mathbf{x}) - \log p_\theta(\mathbf{x},\mathbf{h}) - T]^+ \\ &= [l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+ - l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T) - T. \end{aligned}$$

For the second term in the covariance, we can use the expression from Proposition 1 to write,

$$\begin{aligned} \nabla_T \log \gamma_q(\mathbf{h}) &= -\nabla_T [l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+ \\ &= -\sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T))\nabla_T l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T) \\ &= \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)), \end{aligned}$$

where $\sigma(x) \triangleq 1/(1+e^{-x})$ is the sigmoid function. Putting the two terms together, we have,

$$\begin{aligned} \nabla_T \mathtt{KL}(q\|p) = -\mathtt{COV}_q([l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+ \\ - l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T) - T,\ \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T))). \end{aligned}$$

To prove that the two random variables, $[l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)]^+ - l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T) - T$ and $\sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T))$ are a monotone transformation of each other, we can use the identity $[x]^+ - x = \log(1+e^x) - x = -\log\sigma(x)$ to rewrite the final expression for the gradient of the KL divergence as,

$$\nabla_T \mathtt{KL}(q\|p) = \mathtt{COV}_q\left(\log\sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T)) + T,\ \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x},T))\right)$$

The inequality follows from the fact that the covariance of a random variable and a monotone transformation (the logarithm in this case) is non-negative. □

A.3. Theorem 2

Before proving Theorem 2, we first state and prove an important lemma.

Lemma 1. *Suppose $p(x) = \gamma_p(x)/Z_p$ and $q(x) = \gamma_q(x)/Z_q$ are two unnormalized distributions, where only q depends on ϕ (the recognition network parameters), but both p and q can depend on θ.* [1] *Let $\mathcal{A}(x) \triangleq \log\gamma_p(x) - \log\gamma_q(x)$. Then the variational lower bound objective (on $\log Z_p$) and its gradients with respect to the parameters θ, ϕ are given by,*

$$\begin{aligned} ELBO(\theta,\phi) &\triangleq \mathbb{E}_q\left[\mathcal{A}(x)\right] + \log Z_q \\ \nabla_\phi ELBO(\theta,\phi) &= \mathtt{COV}_q\left(\mathcal{A}(x), \nabla_\phi \log\gamma_q(x)\right) \\ \nabla_\theta ELBO(\theta,\phi) &= \mathbb{E}_q\left[\nabla_\theta \log\gamma_p(x)\right] \\ &\quad + \mathtt{COV}_q\left(\mathcal{A}(x), \nabla_\theta \log\gamma_q(x)\right). \end{aligned}$$

Note that the covariance is the expectation of the product of (at least one) mean-subtracted version of the two random variables. Further, we can also write, $\mathtt{KL}(q||p) = \log\left(\mathbb{E}_q\left[e^{-\bar{\mathcal{A}}(x)}\right]\right)$, where $\bar{\mathcal{A}}(x) \triangleq \mathcal{A}(x) - \mathbb{E}_q\left[\mathcal{A}(x)\right]$ is the mean subtracted version of the learning signal, $\mathcal{A}(x)$.

Proof. The equation for the ELBO follows from the definition. For the gradients, we can write, $\nabla_\phi ELBO(\theta,\phi) = D_2 - D_1 + D_3$, where

$$\begin{aligned} D_1 &= \nabla_\phi \mathbb{E}_q\left[\log\gamma_q(x)\right] \\ D_2 &= \nabla_\phi \mathbb{E}_q\left[\log\gamma_p(x)\right] \\ D_3 &= \nabla_\phi \log Z_q \end{aligned}$$

[1] The dependence for q on θ can happen via some resampling mechanism that is allowed to, for example, evaluate γ_p on the sample proposals before making its accept/reject decisions, as in our case.

where,

$$
\begin{aligned}
D_1 &= \nabla_\phi \mathbb{E}_q \left[\log \gamma_q(x)\right] \\
&= \sum_x \nabla_\phi \left[q(x) \log \gamma_q(x)\right] \\
&= \sum_x \left(\frac{q(x)}{\gamma_q(x)} \nabla_\phi \gamma_q(x) + \log \gamma_q(x)\, \nabla_\phi q(x) \right) \\
&= \frac{1}{Z_q} \nabla_\phi Z_q + \sum_x q(x) \log \gamma_q(x) \nabla_\phi \log q(x) \\
&= D_3 + \mathbb{E}_q \left[\log \gamma_q(x) \nabla_\phi \log q(x)\right] \\
D_2 &= \nabla_\phi \mathbb{E}_q \left[\log \gamma_p(x)\right] \\
&= \nabla_\phi \sum_x q(x) \log \gamma_p(x) \\
&= \sum_x \log \gamma_p(x) \nabla_\phi q(x) \\
&= \sum_x \log \gamma_p(x) q(x) \nabla_\phi \log q(x) \\
&= \mathbb{E}_q \left[\log \gamma_p(x) \nabla_\phi \log q(x)\right]
\end{aligned}
$$

which implies,

$$
\begin{aligned}
\nabla_\phi ELBO(\theta, \phi) &= D_2 - (D_1 - D_3) \\
&= \mathbb{E}_q \left[\left(\log \gamma_p(x) - \log \gamma_q(x)\right) \nabla_\phi \log q(x)\right].
\end{aligned}
$$

Next, observe that $\mathbb{E}_q \left[\nabla_\phi \log q(x)\right] = 0$, Therefore, using the fact that the expectation of the product of two random variables is the same as their covariance when at least one of the two random variables has a zero mean, we get $\nabla_\phi ELBO(\theta, \phi) = \texttt{COV}_q \left(\mathcal{A}(x), \nabla_\phi \log q(x)\right)$. Next note that we can add an arbitrary constant to either random variable without changing the covariance, therefore this is equal to $\texttt{COV}_q \left(\mathcal{A}(x), \nabla_\phi \log q(x) - \nabla_\phi \log Z_q\right) = \texttt{COV}_q \left(\mathcal{A}(x), \nabla_\phi \log \gamma_q(x)\right)$.

The derivation for the gradient with respect to θ is analogous, except for D_2, which has an additional term $\mathbb{E}_q \left[\nabla_\theta \log \gamma_p(x)\right]$ which did not appear in the gradient with respect to ϕ because of our assumption on the lack of dependence of $\log \gamma_p(x)$ on the recognition parameters ϕ. For the identity on the KL divergence, we have,

$$
\begin{aligned}
\texttt{KL}(q||p) &= \log Z_p - \log Z_q + \mathbb{E}_q \left[\log \gamma_q(x) - \log \gamma_p(x)\right] \\
&= \log \left(\sum_x \frac{\gamma_p(x)}{Z_q} \right) + \mathbb{E}_q \left[\log \gamma_q(x) - \log \gamma_p(x)\right] \\
&= \log \left(\mathbb{E}_q \left[\frac{\gamma_p(x)}{\gamma_q(x)} \right] \right) + \mathbb{E}_q \left[\log \gamma_q(x) - \log \gamma_p(x)\right] \\
&= \log \left(\mathbb{E}_q \left[e^{-\mathcal{A}(x)} \right] \right) + \mathbb{E}_q \left[\mathcal{A}(x)\right] \\
&= \log \left(\mathbb{E}_q \left[e^{-\bar{\mathcal{A}}(x)} \right] \right).
\end{aligned}
$$

□

Using the above lemma, we provide a proof for Theorem 2 below.

Proof. We apply the result of Theorem 1, which computes the ELBO corresponding to the two unnormalized distributions on the latent variable space $\mathbf{h}$ (for fixed $\mathbf{x}, T$), with $\log \gamma_p(.) \triangleq \log p_\theta(\mathbf{h}, \mathbf{x})$ and $\log \gamma_q(.) \triangleq \log r_\phi(\mathbf{h}|\mathbf{x}) - [l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)]^+$. This gives: ∇_ϕR-ELBO$(\theta, \phi) = \texttt{COV}_q \left(A_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T), \nabla_\phi \log \gamma_q(\mathbf{h})\right)$. We can then evaluate $\nabla_\phi \log \gamma_q(\mathbf{h}) = \left(1 - \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T))\right) \nabla_\phi \log r_\phi(\mathbf{h}|\mathbf{x})$, where $\sigma()$ is the sigmoid function. Note that this is a consequence of the fact that the derivative of the softplus, $\log(1+e^x)$, is the sigmoid function, $1/(1+e^{-x})$. Similarly for the θ gradient, we get,

$$
\begin{aligned}
\nabla_\theta \text{R-ELBO}(\theta, \phi) = \mathbb{E}_q &\left[\nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{h})\right] \\
&+ \texttt{COV}_q \left(A_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T), \nabla_\theta \log \gamma_q(\mathbf{h})\right)
\end{aligned}
$$

where,

$$
\begin{aligned}
\nabla_\theta \log \gamma_q(\mathbf{h}) &= \nabla_\theta \left[l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)\right]^+ \\
&= \sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)) \nabla_\theta l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T) \\
&= -\sigma(l_{\theta,\phi}(\mathbf{h}|\mathbf{x}, T)) \nabla_\theta \log p_\theta(\mathbf{x}, \mathbf{h}).
\end{aligned}
$$

□

B. Experimental details

B.1. Synthetic

The target distribution was set with an optimal parameter $\phi^* = \log(10.0)$ (i.e. the rate parameter is 10.0), and $c = 5$. The optimizer used was SGD with momentum with and mass 0.5. For resampling, plots show results with learning rate set to 0.01 and $T = 50$ For VIMCO, plots show results with learning rate set to 0.005 and $k = 100$.

B.2. MNIST

We consider a 50,000/10,000/10,000 train/validation/test split of the binarized MNIST dataset. For a direct comparison with prior work, both the generative and recognition networks have the same architecture of stochastic layers. No additional deterministic layers were used. The batch size was 50, the optimizer used is Adam with a learning rate of 3e-4. We updated the resampling thresholds after every 100,000 iterations to correspond to acceptance of the top 95 percentile, i.e. only the bottom 5% of the samples were rejected. The lower bounds on the test set are calculated based on importance sampling with 25 samples.