Variational Rejection Sampling

Aditya Grover* Stanford University

Ramki Gummadi* Vicarious

Miguel Lázaro-Gredilla Vicarious

Dale Schuurmans University of Alberta

Stefano Ermon Stanford University

Abstract

Learning latent variable models with stochastic variational inference is challenging when the approximate posterior is far from the true posterior, due to high variance in the gradient estimates. We propose a novel rejection sampling step that discards samples from the variational posterior which are assigned low likelihoods by the model. Our approach provides an arbitrarily accurate approximation of the true posterior at the expense of extra computation. Using a new gradient estimator for the resulting unnormalized proposal distribution, we achieve average improvements of 3.71 nats and 0.21 nats over state-of-the-art single-sample and multi-sample alternatives respectively for estimating marginal log-likelihoods using sigmoid belief networks on the MNIST dataset.

1 INTRODUCTION

Latent variable models trained using stochastic variational inference can learn complex, high dimensional distributions [Hoffman et al., 2013, Ranganath et al., 2014]. Learning typically involves maximization of a lower bound to the intractable log-likelihood of the observed data, marginalizing over the latent, unobserved variables. To scale to large datasets, inference is *amortized* by introducing a recognition model approximating the true posterior over the latent variables, conditioned on the observed data [Dayan et al., 1995, Gershman and Goodman, 2014]. The generative and recognition models are jointly trained and commonly parameterized using deep neural networks. While this provides flexibility, it also leads to expectations without any closed form expressions in the learning objective and corresponding gradients.

The general approach to stochastic optimization of such objectives involves Monte Carlo estimates of the gradients using the variational posterior (a.k.a. the recognition model) as a proposal distribution [Mnih and Rezende, 2016]. A simple feed forward network, however, may not capture the full complexity of the posterior, a difficulty which shows up in practice as high variance in the gradients estimated with respect to the parameters of the proposal distribution.

There is a vast body of prior work in variance reduction for stochastic optimization, including recent work focusing on variational methods for generative modeling. The standard approach is to use score function estimators with appropriate baselines [Glynn, 1990, Williams, 1992, Fu, 2006]. Many continuous distributions are also amenable to reparameterization, which transforms the original problem of taking gradients with respect to the parameters of the proposal to the simpler problem of taking gradients with respect to a deterministic function [Kingma and Welling, 2014, Rezende et al., 2014, Titsias and Lázaro-Gredilla, 2014]. Finally, a complementary technique for variance reduction is the use of multi-sample objectives which compute importance weighted gradient estimates based on multiple samples from the proposal [Burda et al., 2016, Mnih and Rezende, 2016]. We discuss these approaches in Section 2.

In this work, we propose a new class of estimators for variational learning based on rejection sampling. The *variational rejection sampling* approach modifies the sampling procedure into a two-step process: first, a proposal distribution (in our case, the variational posterior of a generative model) proposes a sample and then we explicitly accept or reject this sample based

∗ equal contribution.

Proceedings of the 21^{st} International Conference on Artificial Intelligence and Statistics (AISTATS) 2018, Lanzarote, Spain. PMLR: Volume 84.

on a novel differentiable accept-reject test. The test is designed to reject samples from the variational posterior that are assigned low likelihoods by the generative model, wherein the threshold for rejection can be controlled based on the available computation.

We show how this procedure leads to a modification of the original variational posterior to a richer family of approximating *resampled* proposal distributions. The modification is defined implicitly [Mohamed and Lakshminarayanan, 2016] since the only requirement from the original variational posterior is that it should permit efficient sampling. Hence, our soft accept-reject test provides a knob to smoothly interpolate between plain importance sampling with a fixed variational posterior (no rejections) to obtaining samples from the exact posterior in the limit (with potentially high rejection rate), thereby trading off statistical accuracy for computational cost. Further, even though the resampled proposal is unnormalized due to the introduction of an accept-reject test, we can surprisingly derive unbiased gradient estimates with respect to the model parameters that only require the unnormalized density estimates of the resampled proposal, leading to an efficient learning algorithm.

Empirically, we demonstrate that variational rejection sampling outperforms competing single-sample and multi-sample approaches by 3.71 nats and 0.21 nats respectively on average for estimating marginal log-likelihoods using sigmoid belief networks on the MNIST dataset.

2 BACKGROUND

In this section, we present the setup for stochastic optimization of expectations of arbitrary functions with respect to parameterized distributions. We also discuss prior work applicable in the context of variational learning. We use upper-case symbols to denote probability distributions and assume they admit densities on a suitable reference measure, denoted by the corresponding lower-case notation.

Consider the following objective:

$$L(\theta, \phi) = \mathbb{E}_{\mathbf{z} \sim Q_\phi} \left[f_{\theta,\phi}(\mathbf{z}) \right] \tag{1}$$

where θ and ϕ denote sets of parameters and Q_ϕ is a parameterized sampling distribution over $\mathbf{z}$ which can be discrete or continuous. We will assume that sampling $\mathbf{z}$ from Q_ϕ is efficient, and suppress subscript notation in expectations from $\mathbf{z} \sim Q_\phi$ to simply Q wherever the context is clear. We are interested in optimizing the expectation of a function $f_{\theta,\phi}$ with respect to the sampling distribution Q_ϕ using gradient methods. In general, $f_{\theta,\phi}$ and the density q_ϕ need not be differentiable with respect to θ and ϕ.

Such objectives are intractable to even evaluate in general, but unbiased estimates can be obtained efficiently using Monte Carlo techniques. The gradients of the objective with respect to θ are given by:

$$\nabla_\theta L(\theta, \phi) = \mathbb{E}_Q \left[\nabla_\theta f_{\theta,\phi}(\mathbf{z}) \right].$$

As long as $f_{\theta,\phi}$ is differentiable with respect to θ, we can compute unbiased estimates of the gradients using Monte Carlo. There are two primary class of estimators for computing gradients with respect to ϕ which we discuss next.

Score function estimators. Using the fact that $\nabla_\phi q_\phi = q_\phi \nabla_\phi \log q_\phi$, the gradients with respect to ϕ can be expressed as:

$$\nabla_\phi L(\theta, \phi) = \mathbb{E}_Q \left[\nabla_\phi f_{\theta,\phi}(\mathbf{z}) \right] + \mathbb{E}_Q \left[f_{\theta,\phi}(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}) \right].$$

The first term can be efficiently estimated using Monte Carlo if $f_{\theta,\phi}$ is differentiable with respect to ϕ. The second term, referred to as the *score function estimator* or the likelihood-ratio estimator or REINFORCE by different authors [Fu, 2006, Glynn, 1990, Williams, 1992], requires gradients with respect to the log density of the sampling distribution and can suffer from large variance [Glasserman, 2013, Schulman et al., 2015]. Hence, these estimators are used in conjunction with control variates (also referred to as baselines). A control variate, c, is any constant or random variable (could even be a function of $\mathbf{z}$ if we can correct for its bias) positively correlated with $f_{\theta,\phi}$ that reduces the variance of the estimator without introducing any bias:

$$\mathbb{E}_Q \left[f_{\theta,\phi}(\mathbf{z}) \nabla_\phi \log q_\phi(\mathbf{z}) \right] = \mathbb{E}_Q \left[(f_{\theta,\phi}(\mathbf{z}) - c) \nabla_\phi \log q_\phi(\mathbf{z}) \right].$$

Reparameterization estimators. Many continuous distributions can be *reparameterized* such that it is possible to obtain samples from the original distribution by applying a deterministic transformation to a sample from a fixed distribution [Kingma and Welling, 2014, Rezende et al., 2014, Titsias and Lázaro-Gredilla, 2014]. For instance, if the sampling distribution is an isotropic Gaussian, $Q_\phi = \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$, then a sample $\mathbf{z} \sim Q_\phi$ can be equivalently obtained by sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and passing through a deterministic function, $\mathbf{z} = g_{\boldsymbol{\mu},\sigma}(\boldsymbol{\epsilon}) = \boldsymbol{\mu} + \sigma\boldsymbol{\epsilon}$. This allows exchanging the gradient and expectation, giving a gradient with respect to ϕ after reparameterization as:

$$\nabla_\phi L(\theta, \phi) = \mathbb{E}_{\boldsymbol{\epsilon} \sim S} \left[\nabla_{\mathbf{z}} f_{\theta,\phi}(\mathbf{z}) \nabla_\phi g_\phi(\boldsymbol{\epsilon}) \right]$$

where S is a fixed sampling distribution and $\mathbf{z} = g_\phi(\boldsymbol{\epsilon})$ is a deterministic transformation. Reparameterized gradient estimators typically have lower variance but are not widely applicable since they require g_ϕ and $f_{\theta,\phi}$

to be differentiable with respect to ϕ and $\mathbf{z}$ respectively unlike score function estimators [Glasserman, 2013, Schulman et al., 2015]. Recent work has tried to bridge this gap by reparameterizing continuous relaxations to discrete distributions (called Concrete distributions) that gives low variance, but biased gradient estimates [Maddison et al., 2017, Jang et al., 2017] and deriving gradient estimators that interpolate between score function estimators and reparameterization estimators for distributions that can be simulated using acceptance-rejection algorithms, such as the Gamma and Dirichlet distributions [Ruiz et al., 2016, Naesseth et al., 2017]. Further reductions in the variance of reparameterization estimators is possible as explored in recent work, potentially introducing bias [Roeder et al., 2017, Miller et al., 2017, Levy and Ermon, 2018].

2.1 Variational learning

We can cast variational learning as an objective of the form given in Eq. (1). Consider a generative model that specifies a joint distribution $p_\theta(\mathbf{x}, \mathbf{z})$ over the observed variables $\mathbf{x}$ and latent variables $\mathbf{z}$ respectively, parameterized by θ. We assume the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ over the latent variables is intractable, and we introduce a variational approximation to the posterior $q_\phi(\mathbf{z}|\mathbf{x})$ represented by a recognition network and parameterized by ϕ. The parameters of the generative model and the recognition network are learned jointly [Kingma and Welling, 2014, Rezende et al., 2014] by optimizing an evidence lower bound (ELBO) on the marginal log-likelihood of a datapoint $\mathbf{x}$:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_Q \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \quad \triangleq ELBO(\theta, \phi). \quad (2)$$

Besides reparameterization estimators that were introduced in the context of variational learning, there has been considerable research in the design of control variates (CV) for variational learning using the more broadly applicable score function estimators [Paisley et al., 2012]. In particular, Wingate and Weber [2013] and Ranganath et al. [2014] use simple scalar CV, NVIL proposed input-dependent CV [Mnih and Gregor, 2014], and MuProp combines input-dependent CV with deterministic first-order Taylor approximations to the mean-field approximation of the model [Gu et al., 2016]. Recently, REBAR used CV based on the Concrete distribution to give low variance, unbiased gradient updates [Tucker et al., 2017], which has been subsequently generalized to a more flexible parametric version in RELAX [Grathwohl et al., 2018].

In a parallel line of work, there is an increasing effort to learn models with more expressive posteriors. Major research in this direction focuses on continuous latent variable models, for *e.g.*, see Gregor et al. [2014, 2015], Salimans et al. [2015], Rezende and Mohamed [2015], Chen et al. [2017], Song et al. [2017], Grover and Ermon [2018] and the references therein. Closely related to the current work is Gummadi [2014], which originally proposed a resampling scheme to improve the richness of the posterior approximation and derived unbiased estimates of gradients for the KL divergence from arbitrary unnormalized posterior approximations. Related work for discrete latent variable models is scarce. Hierarchical models impose a prior over the discrete latent variables to induce dependencies between the variables [Ranganath et al., 2016], which can also be specified as an undirected model [Kuleshov and Ermon, 2017]. On the theoretical side, random projections of discrete posteriors have been shown to provide tight bounds on the quality of the variational approximation [Zhu and Ermon, 2015, Grover and Ermon, 2016, Hsu et al., 2016].

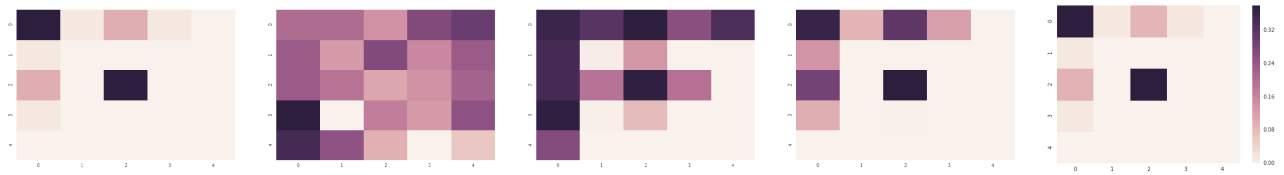
Multi-sample estimators. Multi-sample objectives improve the family of distributions represented by variational posteriors by trading off computational efficiency with statistical accuracy. Learning algorithms based on these objectives do not introduce additional parameters but instead draw multiple samples from the variational posterior to reduce the variance in gradient estimates as well as tighten the ELBO. A multi-sample ELBO is given as:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_{\mathbf{z}_1, \ldots, \mathbf{z}_k \sim Q_\phi} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p_\theta(\mathbf{x}, \mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})} \right] \quad (3)$$

Biased gradient estimators using similar objectives were first used by Raiko et al. [2015] for structured prediction. Burda et al. [2016] showed that a multi-sample ELBO is a tighter lower bound on the log-likelihood than the ELBO. Further, they derived unbiased gradient estimates for optimizing variational autoencoders trained using the objective in Eq. (3). VIMCO generalized this to discrete latent variable models with arbitrary Monte Carlo objectives using a score function estimator with per-sample control variates [Mnih and Rezende, 2016], which serves as a point of comparison in our experiments. Recently, Naesseth et al. [2018] proposed an importance weighted multi-sample objective for probabilistic models of dynamical systems based on sequential Monte Carlo.

3 THE VRS FRAMEWORK

To motivate variational rejection sampling (VRS), consider the ELBO objective in Eq. (2) for any fixed θ and $\mathbf{x}$. This is maximized when the variational posterior matches the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$. However,



(a) Target dist. (b) $T, a, \texttt{KL} : \infty, 1, 18$ (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 a. The triples shown are $T, a, \texttt{KL}$ divergence to target.

in practice, the approximate posterior could be arbitrarily far from the true posterior which we seek to mitigate by rejection sampling.

3.1 The Resampled ELBO (R-ELBO)

Consider an alternate sampling distribution for the variational posterior with the density defined below:

$$r_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) \propto q_\phi(\mathbf{z}|\mathbf{x}) a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) \tag{4}$$

where $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) \in (0, 1]$ is an acceptance probability function that could depend on θ, ϕ, and additional parameter(s) T. Unlike p, q, and r, note that $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ does not represent a density over the latent variables $\mathbf{z}$, but simply a function that maps each possible $\mathbf{z}$ to a number between 0-1 and hence, it denotes the probability of acceptance for each $\mathbf{z}$. In order to sample from $R_{\theta,\phi}$, we follow a two step sampling procedure defined in Algorithm 1. Hence, computing Monte Carlo expectations with respect to the modified proposal involves resampling from the original proposal due to an additional accept-reject step. We refer to such sampling distributions as *resampled* proposal distributions.

Algorithm 1 Sampler for $R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$

input $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$, $Q_\phi(\mathbf{z}|\mathbf{x})$
output $\mathbf{z} \sim R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$
while True **do**
  $\mathbf{z} \leftarrow$ sample from proposal $Q_\phi(\mathbf{z}|\mathbf{x})$.
  Compute acceptance probability $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$
  Sample uniform: $u \sim U[0, 1]$.
  if $u < a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ **then**
    Output sample $\mathbf{z}$.
  end if
end while

The resampled proposal defines a new evidence lower bound on the marginal log-likelihood of $\mathbf{x}$, which we refer to as the "resampled ELBO", or R-ELBO:

$$\log p_\theta(\mathbf{x}) \geq \mathbb{E}_R \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z}) Z_R(\mathbf{x}, T)}{q_\phi(\mathbf{z}|\mathbf{x}) a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)} \right] \triangleq \text{R-ELBO}(\theta, \phi). \tag{5}$$

where $Z_R(\mathbf{x}, T) = \mathbb{E}_Q[a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)]$ is the (generally intractable) normalization constant for the resampled proposal distribution. To make the resampling framework described above work, we need to define a suitable acceptance function and derive low variance Monte Carlo gradient estimators with respect to θ and ϕ for the R-ELBO which we discuss next.

3.2 Acceptance probability functions

The general intuition behind designing an acceptance probability function is that it should allow for the resampled posterior to come "close" to the target posterior $p_\theta(\mathbf{z}|\mathbf{x})$ (possibly at the cost of extra computation). While there could be many possible ways of designing such acceptance probability functions, we draw inspiration from rejection sampling [Halton, 1970].

In order to draw samples from a target distribution $\mathcal{T}(\mathbf{z})$, a rejection sampler first draws samples from an easy-to-sample distribution $\mathbf{z} \sim \mathcal{S}(\mathbf{z})$ with a larger-or-equal support, *i.e.*, $s(\mathbf{z}) > 0$ wherever $t(\mathbf{z}) > 0$. Then, provided we have a fixed, finite upper bound $M \in [1, \infty)$ on the likelihood ratio $t(\mathbf{z})/s(\mathbf{z})$, we can obtain samples from the target by accepting samples from $s(\mathbf{z})$ with a probability $\frac{t(\mathbf{z})}{Ms(\mathbf{z})}$. The choice of M guarantees that the acceptance probability is less than or equal to 1, and overall probability of any accepted sample $\mathbf{z}$ is proportional to $\frac{t(\mathbf{z})}{Ms(\mathbf{z})} s(\mathbf{z})$ which gives us $\mathbf{z} \sim \mathcal{T}(\mathbf{z})$ as desired. The constant M has to be large enough such that the acceptance probability does not exceed 1, but a very high value of M leads to an increase in computation due to a higher rejection rate.

If the target is only known up to a normalization constant, then rejection sampling can be used provided M is large enough to ensure that the acceptance probability never exceeds 1. However, we do not know in general how large M should be and even if we did, it would be computationally infeasible to actually use it in a practical algorithm. A natural approximation that departs from the typical rejection sampler would be to accept proposed samples with probability

$\min\left[1, \frac{t(\mathbf{z})}{Ms(\mathbf{z})}\right]$ for some M that is no longer guaranteed to dominate the likelihood ratios across the entire state space. In the setting of variational learning, the target corresponds to the true, but intractable posterior that can be specified up to a normalization constant as $p_\theta(\mathbf{z}|\mathbf{x}) \propto p_\theta(\mathbf{x}, \mathbf{z})$ for any fixed θ and $\mathbf{x}$. If $Q_\phi(\mathbf{z}|\mathbf{x})$ denotes the proposal distribution and $M(T)$ is any function of the threshold parameter T, the acceptance probability for the approximate rejection sampler is given by:

$$a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) = \min\left[1, \frac{p_\theta(\mathbf{x}, \mathbf{z})}{M(T)q_\phi(\mathbf{z}|\mathbf{x})}\right].$$

To get a fully differentiable approximation to the min operator, we consider:

$$\begin{aligned} a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) &= 1/\max\left[1, \frac{M(T)q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{x}, \mathbf{z})}\right] \\ &\approx 1/\left[1^t + \left(\frac{M(T)q_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{x}, \mathbf{z})}\right)^t\right]^{1/t} \end{aligned}$$

where the approximation in the last step holds for large t or when any of the two terms in the max expression dominates the other. For $t = 1$, we get the exponentiated negative softplus function which we will use to be the acceptance probability function in the remainder of this paper. We leave other approximations to future work. Letting $T = -\log M$, the log probability of acceptance is parameterized as:

$$\begin{aligned} \log a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) &= -\log[1 + \exp(l_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T))] \\ &= -[l_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)]^+ \end{aligned} \tag{6}$$

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

Informally, the resampling scheme of Algorithm 1 with the choice of acceptance probability function as in Eq. (6) 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 T have an exponentially decaying probability of getting accepted, while leaving the remaining samples with negligible interference from resampling.

When the proposed sample $\mathbf{z}$ from Q_ϕ is assigned a small likelihood by p_θ, the random variable $l_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ is correspondingly large with high probability (and linear in the negative log-likelihood assigned by p_θ), resulting in a low acceptance probability. Conversely, when p_θ assigns a high likelihood to $\mathbf{z}$, we get a higher acceptance probability. Furthermore, a large value of the scalar bias T results in an acceptance probability of 1, recovering the regular variational inference setting as a special case. 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 decreasing 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, $\mathrm{KL}(R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)\|P_\theta(\mathbf{z}|\mathbf{x}))$ is monotonically increasing in T where $R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ is the resampled proposal distribution with the choice of acceptance probability function in Eq. (6). Furthermore, the behavior of the sampler in Algorithm 1 interpolates between the following two extremes:*

- *As $T \to +\infty$, $R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ is equivalent to $Q_\phi(\mathbf{z}|\mathbf{x})$, with perfect sampling efficiency for the accept-reject step i.e., $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) \to 1$.*
- *As $T \to -\infty$, $R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)$ is equivalent to $P_\theta(\mathbf{z}|\mathbf{x})$, with the sampling efficiency of a plain rejection sampler i.e., $a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T) \to 0\ \forall \mathbf{z}$.*

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. As T is reduced, Figure 1 demonstrates progressive improvement in the posterior quality 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. In summary, we can express the R-ELBO as:

$$\text{R-ELBO}(\theta, \phi) = \log p_\theta(\mathbf{x}) - \mathrm{KL}(R_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)\|P_\theta(\mathbf{z}|\mathbf{x})). \tag{7}$$

Theorem 1 and Eq. (7) give the following corollary.

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

With an appropriate acceptance probability function, we can therefore traverse the computational-statistical trade off for maximum likelihood estimation by adaptively tuning the threshold T based on available computation.

3.3 Gradient estimation

The resampled proposal distribution in Eq. (4) is unnormalized with an intractable normalization constant, $Z_R(\mathbf{x}, T) = \mathbb{E}_Q\left[a_{\theta,\phi}(\mathbf{z}|\mathbf{x}, T)\right]$. The presence of an intractable normalization constant seems challenging for both evaluation and stochastic optimization of

the R-ELBO. Even though the constant cannot be computed in closed form,[1] we can nevertheless compute Monte Carlo estimates of its gradients, as we show in Lemma 1 in the appendix. The resulting R-ELBO gradients are summarized below in Theorem 2.

Theorem 2. *Let* $\texttt{COV}_R(A(\mathbf{z}), B(\mathbf{z}))$ *denote the covariance of the two random variables* $A(\mathbf{z})$ *and* $B(\mathbf{z})$*, where* $\mathbf{z} \sim R_{\theta,\phi}$*. Then:*

- *The R-ELBO gradients with respect to* θ*:*

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

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

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

- *The R-ELBO gradients with respect to* ϕ*:*

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

where $\sigma(*)$ *denotes the sigmoid function applied to* $*$*, i.e.,* $\sigma(*) = 1/(1+\exp(-*))$.

In the above expressions, the gradients are expressed as the covariance of two random variables that are a function of the latent variables sampled from the approximate posterior $R_{\theta,\phi}(\mathbf{z}|\mathbf{x},T)$. Hence, we only need samples from $R_{\theta,\phi}$ for learning, which can be done using Algorithm 1 followed by Monte Carlo estimation analogous to estimation of the usual ELBO gradients.

4 LEARNING ALGORITHM

A practical implementation of variational rejection sampling as shown in Algorithm 2 requires several algorithmic design choices that we discuss in this section.

4.1 Threshold selection heuristic

The monotonicity of KL divergence in Theorem 1 suggests that it is desirable to choose a value of T as low as computationally feasible for obtaining the highest accuracy. However, the quality of the approximate posterior, $Q_\phi(\mathbf{z}|\mathbf{x})$ for a fixed parameter ϕ could vary significantly across different examples $\mathbf{x}$. This would require making T dependent on $\mathbf{x}$. Although learning T in a parametric way is one possibility, in this work, we restrict attention to a simple estimation based approach that reduces the design choice to a single hyperparameter that can be adjusted to trade extra computation for accuracy. For each fixed

Algorithm 2 Variational Rejection Sampling

input Network architectures for $p_\theta(\mathbf{x},\mathbf{z}), q_\phi(\mathbf{z}|\mathbf{x})$; quantile hyperparameter $\gamma \in (0,1)$; initial parameters, θ_0, ϕ_0; threshold update frequency, F; quantile estimation sample count N; Covariance estimate sample count $S \geq 2$, SGD based optimizer, OPT; Dataset $\{\mathbf{x}_k\}_{k=1}^K$, Number of epochs: N.
output Final estimates, θ, ϕ.

1: Initialize $\theta \leftarrow \theta_0$; $\phi \leftarrow \phi_0$; $T(\mathbf{x}) = +\infty\ \forall\ \mathbf{x}$.
2: **for** $e \in \{1,\ldots,N\}$ **do**
3: **if** $e \bmod F = 0$ **then**
4: **for** each $\mathbf{x}$ in dataset **do**
5: Sample $Z^N \leftarrow \{\mathbf{z}_1,\ldots,\mathbf{z}_N\} \sim q_\phi(\mathbf{z}|\mathbf{x})$.
6: $T(\mathbf{x}) \leftarrow \hat{T}_\gamma^N(\mathbf{x},\theta,\phi)$, the Monte Carlo estimate of Eq. (8) based on samples Z^N.
7: **end for**
8: **end if**
9: **for** each $\mathbf{x}$ in dataset **do**
10: Draw S independent samples $\{\mathbf{z}_1,\ldots,\mathbf{z}_S\} \sim R_{\theta,\phi}(\mathbf{z}|\mathbf{x},T(\mathbf{x}))$ using Algorithm 1.
11: Use Theorem 2 and Eq. (9) with $\{\mathbf{z}_1,\ldots,\mathbf{z}_S\}$ to estimate gradients, $\hat{g}_\theta, \hat{g}_\phi$.
12: Update $\theta, \phi \leftarrow OPT(\theta,\phi,\hat{g}_\theta,\hat{g}_\phi)$.
13: **end for**
14: **end for**

$\mathbf{x}$, let $\mathcal{L}_{\theta,\phi}(\mathbf{x})$ denote the probability distribution of the scalar random variable, $-\log p_\theta(\mathbf{x},\mathbf{z}) + \log q_\phi(\mathbf{z}|\mathbf{x})$, where $\mathbf{z} \sim Q_\phi(\mathbf{z}|\mathbf{x})$. Let $\mathcal{Q}_\mathcal{L}$ denote the quantile function[2] for any given 1-D distribution $\mathcal{L}$. For each quantile parameter $\gamma \in (0,1]$, we consider a heuristic family of threshold parameters given $\mathbf{x}, \phi, \theta$, defined as:

$$T_\gamma(\mathbf{x},\theta,\phi) \triangleq \mathcal{Q}_{\mathcal{L}_{\theta,\phi}(\mathbf{x})}(\gamma). \tag{8}$$

For example, for $\gamma = 0.5$, this is the median of $\mathcal{L}_{\theta,\phi}(\mathbf{x})$. Eq. (8) implies that the acceptance probability stays roughly in the range of γ for most samples. This is due to the fact that the negative log of the acceptance probability, defined in Eq. (6) as $[l_{\theta,\phi}(\mathbf{z}|\mathbf{x},T)]^+$ is positive approximately with probability $1-\gamma$, an event which is likely to result in a rejection. In Algorithm 2, we compute a Monte Carlo estimate for the threshold and denote the resulting value using N samples as $\hat{T}_\gamma^N(\mathbf{x},\theta,\phi)$ (Lines 3-8). This estimation is done independently from the SGD updates, once every F epochs, to save computational cost, and also implies that T is not continuously updated as a function of θ, ϕ. Technically speaking, this introduces a slight bias in the gradients through their dependence on T, but we ignore this correction since it only happens once every few epochs.

[1] Note that Monte Carlo estimates of the partition function can be obtained efficiently for evaluation.

[2] Recall that for a given CDF $\mathcal{F}(x)$, the quantile function is its 'inverse', namely $\mathcal{Q}(p) = \inf\{x \in \mathbb{R} : p \leq \mathcal{F}(x)\}$.

4.2 Computing covariance estimates

To compute an unbiased Monte Carlo estimate of the covariance terms in the gradients, we need to subtract the mean of at least one random variable while forming the product term. In order to do this in Algorithm 2 (Lines 10-11), 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, similar to the local learning signals proposed in Mnih and Rezende [2016]. This requires generating $S \geq 2$ samples from $R_{\theta,\phi}$ simultaneously at each step to be able to compute each gradient. More precisely, the leave-one-out unbiased Monte Carlo estimator for the covariance of two random variables A, B is defined as follows. Let $(a_1, b_1), \ldots, (a_S, b_S) \sim (A, B)$ be S independent samples from the joint pair (A, B), and let $\hat{m}_A$ denote the sample mean for A: $\hat{m}_A \triangleq \frac{1}{S}\sum_{i=1}^{S} a_i$. Then the covariance estimate is given by:

$$\widehat{\texttt{COV}}_R(A, B) \triangleq \frac{1}{S-1}\sum_{i=1}^{S}(a_i - \hat{m}_A)b_i. \tag{9}$$

4.3 Hyperparameters and overall algorithm

In summary, Algorithm 2 involves the following hyperparameters: S, the number of samples for estimating covariance; γ, the quantile used for setting thresholds; F, the number of epochs between updating $T(\mathbf{x})$.

5 EXPERIMENTAL EVALUATION

We evaluated variational rejection sampling (VRS) against competing methods on a diagnostic synthetic experiment and a benchmark density estimation task.

5.1 Diagnostic experiments

In this experiment, we consider a synthetic setup that involves fitting an approximate posterior candidate from a constrained family to a fixed target distribution that clearly falls outside the approximating family. We restrict attention to training a 1-D parameter, ϕ exclusively (*i.e.*, we do not consider optimization over θ), and for the non-amortized case (*i.e.*, conditioning on $\mathbf{x}$ is not applicable). The target distribution is 1-D, with support on non-negative integers, $z \in \{0, 1, \ldots\}$ and denoted as $P(z)$. This distribution, visualized in Figure 2, is obtained by removing the mass on the first c integers of a Poisson distribution with rate $\lambda^* > 0$. More details are given in the Appendix. The approximate proposal is parameterized as $Q_\phi \triangleq \texttt{Poi}(e^\phi)$, where ϕ is an unconstrained scalar, and denotes a (unmodified) Poisson distribution with the (non-negative) rate parameter, e^ϕ. Note that for

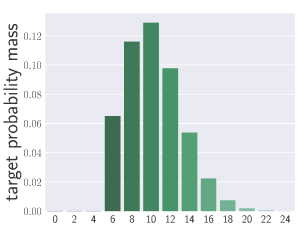


Figure 2: Target distribution, P.

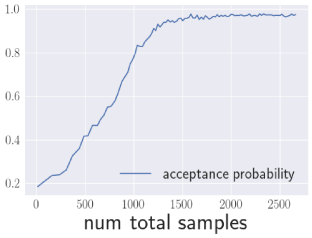
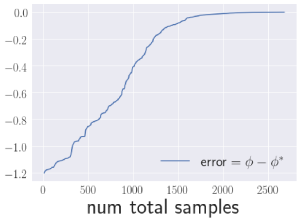
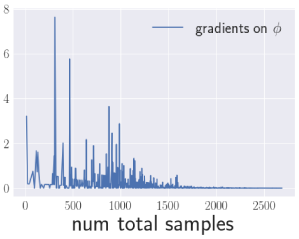


Figure 3: Acceptance probability vs. SGD iteration

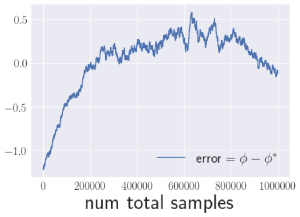


(a) Error: $\phi - \phi^*$

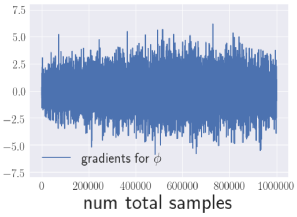


(b) Gradients for ϕ.

Figure 4: VRS learning dynamics. The x-axis shows the number of total samples (both accepted and rejected) at each SGD iteration.



(a) Error: $\phi - \phi^*$.



(b) Gradients for ϕ.

Figure 5: VIMCO learning dynamics. The x-axis shows the number of total samples, which is equal to k times the number of iterations at each SGD iteration.

$\texttt{Poi}(e^\phi)$ to explicitly represent a small mass on $z < c$ would require $\phi \to \infty$, but this would be a bad fit for points just above c. As a result, $\{Q_\phi\}$ does not contain candidates close to the target distribution in the sense of $\texttt{KL}$ divergence, even a simple resampling modification could transform the raw proposal Q_ϕ into a better candidate approximation, R.

In Figures 3 and 4 we illustrate the dynamics of SGD using VRS gradients for approximating P. To keep the analysis simple, the threshold T was kept fixed at a constant value during learning. Figure 3 shows the efficiency of the sampler improving as the learning progresses due to a better fit to the target distribution. Figure 4a shows the difference between the current parameter ϕ and $\phi^* = \log \lambda^*$ from the target distribution, quickly converging to 0 as learning proceeds.

As a benchmark, we evaluated the dynamics based on VIMCO gradients [Mnih and Rezende, 2016]. Figure 5 suggests that the signal in gradients is too low (*i.e.*,

Table 1: Test NLL (in nats) for MNIST comparing VRS with published results. Lower is better.

(a) Baseline results from Tucker et al. [2017]

Model / Architecture	200	200-200
NVIL ($k = 1$)	112.5	99.6
MuProp	111.7	99.07
REBAR ($\lambda = 0.1$)	111.7	99
REBAR	111.6	99.8
Concrete ($\lambda = 0.1$)	111.3	102.8
VRS (IS, $\gamma = 0.95$)	106.97	96.38
VRS (RS, $\gamma = 0.95$)	106.89	96.30
VRS (IS, $\gamma = 0.9$)	106.71	**96.26**
VRS (RS, $\gamma = 0.9$)	**106.63**	96.36

(b) Baseline results from Mnih and Rezende [2016]

Model / Architecture	200-200-200
NVIL ($k = 1$)	95.2
NVIL ($k = 2$)	93.6
NVIL ($k = 5$)	93.7
NVIL ($k = 10$)	93.4
NVIL ($k = 50$)	96.2
RWS ($k = 2$)	94.6
RWS ($k = 5$)	93.4
RWS ($k = 10$)	93.0
RWS ($k = 50$)	92.5
VIMCO ($k = 2$)	93.5
VIMCO ($k = 5$)	92.8
VIMCO ($k = 10$)	92.6
VIMCO ($k = 50$)	91.9
VRS (IS, $\gamma = 0.95$)	92.01
VRS (RS, $\gamma = 0.95$)	91.93
VRS (IS, $\gamma = 0.9$)	92.09
VRS (RS, $\gamma = 0.9$)	**91.69**

high variance in gradient estimates). This behavior was persistent even with much smaller learning rates and large sample sizes compared to VRS gradients. 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 and Rezende [2016]). In contrast, Algorithm 1 discards rejected samples from contributing to the gradients explicitly. Yet another qualitative aspect that distinguishes VRS gradients from importance weighted multi-sample objective gradients is that Algorithm 1 can dynamically adapt the amount of additional computation spent in resampling based on sample quality, as opposed to being fixed in advance.

5.2 Generative modeling

We trained sigmoid belief networks (SBN) on the binarized MNIST dataset [LeCun et al., 2010]. Following prior work, the SBN benchmark architectures for this task consist of several linear layers of 200 hidden units, and the recognition model has the same architecture in the reverse direction. Training such models is sensitive to choices of hyperparamters, and hence we directly compare VRS with published baselines in Table 1. The hyperparameter details for SBNs trained with VRS are given in the Appendix.

With regards to key baseline hyperparameters in Table 1, Concrete and REBAR specify a temperature controlling the degree of relaxation (denoted by λ), whereas multi-sample estimators based on importance weighting specify the number of samples, k to trade-off computation for statistical accuracy. The relevant parameter, γ in our case is not directly comparable but we report results for values of γ where empirically, the average number of rejections per training example was somewhere between drawing $k = 5$ and $k = 20$ samples for an equivalent importance weighted objective for both $\gamma = 0.95$ and $\gamma = 0.9$ (with the latter requiring more computation). Additionally, we provide two estimators for evaluating the test lower bound for VRS. For the importance sampled (IS) version, we simply evaluate the ELBO using importance sampling with the original posterior, Q_ϕ. The resampled (RS) version, on the other hand uses the resampled proposal, $R_{\theta,\phi}$ with the partition function, Z_R estimated as a Monte Carlo expectation.

From the results, we observe that VRS outperforms other methods, including multi-sample estimators with k as high as 50 that require much greater computation than the VRS models considered. Generally speaking, the RS estimates are better than the corresponding IS estimates, and decreasing γ improves performance (at the cost of increased computation).

6 CONCLUSION

We presented a rejection sampling framework for variational learning in generative models that is theoretically principled and allows for a 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 for variational learning of discrete distributions.

In the future, we plan to generalize VRS while exploiting factorization structure in the generative model based on intermediate resampling checkpoints [Gummadi, 2014]. Many baseline methods in our experiments are complementary and could benefit from VRS. Yet another direction involves the applications to stochastic optimization problems arising in reinforcement learning and structured prediction.

ACKNOWLEDGEMENTS

We are thankful to Zhengyuan Zhou and Daniel Levy for helpful comments on early drafts. This research has been supported by a Microsoft Research PhD fellowship in machine learning for the first author, NSF grants #1651565, #1522054, #1733686, Toyota Research Institute, Future of Life Institute, and Intel.

References

Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *International Conference on Learning Representations*, 2016.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In *International Conference on Learning Representations*, 2017.

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

Michael C Fu. Gradient estimation. *Handbooks in operations research and management science*, 13:575–616, 2006.

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

Paul Glasserman. *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media, 2013.

Peter W Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *International Conference on Learning Representations*, 2018.

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

Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *International Conference on Machine Learning*, 2015.

Aditya Grover and Stefano Ermon. Variational Bayes on Monte Carlo steroids. In *Advances in Neural Information Processing Systems*, 2016.

Aditya Grover and Stefano Ermon. Boosted generative models. In *AAAI Conference on Artificial Intelligence*, 2018.

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. MuProp: Unbiased backpropagation for stochastic neural networks. In *International Conference on Learning Representations*, 2016.

R. Gummadi. Resampled belief networks for variational inference. In *NIPS Workshop on Advances in Variational Inference*, 2014.

John H Halton. A retrospective and prospective survey of the Monte Carlo method. *SIAM review*, 12(1):1–63, 1970.

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

Lun-Kai Hsu, Tudor Achim, and Stefano Ermon. Tight variational bounds via random projections and I-projections. In *Artificial Intelligence and Statistics*, 2016.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Volodymyr Kuleshov and Stefano Ermon. Neural variational inference and learning in undirected graphical models. In *Advances in Neural Information Processing Systems*, 2017.

Yann LeCun, Corinna Cortes, and Christopher JC Burges. MNIST handwritten digit database. *http://yann. lecun. com/exdb/mnist*, 2010.

Daniel Levy and Stefano Ermon. Deterministic policy optimization by combining pathwise and score function estimators for discrete action spaces. In *AAAI Conference on Artificial Intelligence*, 2018.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The Concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

Andrew C Miller, Nicholas J Foti, Alexander D'Amour, and Ryan P Adams. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*, 2017.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, 2014.

Andriy Mnih and Danilo J Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, 2016.

Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.

Christian Naesseth, Francisco Ruiz, Scott Linderman, and David Blei. Reparameterization gradients through acceptance-rejection sampling algorithms. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Christian A Naesseth, Scott W Linderman, Rajesh Ranganath, and David M Blei. Variational sequential Monte Carlo. In *International Conference on Artificial Intelligence and Statistics*, 2018.

John Paisley, David Blei, and Michael Jordan. Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning*, 2012.

Tapani Raiko, Mathias Berglund, Guillaume Alain, and Laurent Dinh. Techniques for learning binary stochastic feedforward neural networks. In *International Conference on Learning Representations*, 2015.

Rajesh Ranganath, Sean Gerrish, and David M Blei. Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, 2014.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, 2014.

Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: An asymptotically zero-variance gradient estimator for variational inference. In *Advances in Neural Information Processing Systems*, 2017.

Francisco Ruiz, Michalis Titsias, and David Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems*, 2016.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, 2015.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, 2015.

Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-nice-mc: Adversarial training for MCMC. In *Advances in Neural Information Processing Systems*, 2017.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*, 2014.

George Tucker, Andriy Mnih, Chris J Maddison, and Jascha Sohl-Dickstein. REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, 2017.

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

David Wingate and Theophane Weber. Automated variational inference in probabilistic programming. *arXiv preprint arXiv:1301.1299*, 2013.

Michael Zhu and Stefano Ermon. A hybrid approach for probabilistic inference using random projections. In *International Conference on Machine Learning*, 2015.