Differentiable Antithetic Sampling for Variance Reduction in Stochastic Variational Inference

Mike Wu
Stanford University

Noah Goodman
Stanford University

Stefano Ermon
Stanford University

Abstract

Stochastic optimization techniques are standard in variational inference algorithms. These methods estimate gradients by approximating expectations with independent Monte Carlo samples. In this paper, we explore a technique that uses correlated, but more *representative*, samples to reduce variance. Specifically, we show how to generate *antithetic* samples with sample moments that match the population moments of an underlying proposal distribution. Combining a *differentiable* antithetic sampler with modern stochastic variational inference, we showcase the effectiveness of this approach for learning a deep generative model. An implementation is available at `https://github.com/mhw32/antithetic-vae-public`.

1 Introduction

A wide class of problems in science and engineering can be solved by gradient-based optimization of function expectations. This is especially prevalent in machine learning (Schulman et al., 2015), including variational inference (Ranganath et al., 2014; Rezende et al., 2014) and reinforcement learning (Silver et al., 2014). On the face of it, problems of this nature require solving an intractable integral. Most practical approaches instead use Monte Carlo estimates of expectations and their gradients. These techniques are unbiased but can suffer from high variance when sample size is small—one unlikely sample in the tail of a distribution can heavily skew the final estimate. A simple way to reduce variance is to increase the number of samples; however the computational cost grows quickly. We would like to reap the positive benefits of a larger sample size using as few samples as possible. *With a fixed computational budget, how do we choose samples?*

A large body of work has been dedicated to reducing variance in sampling, with the most popular in machine learning being reparameterizations for some continuous distributions (Kingma and Welling, 2013; Jang et al., 2016) and control variates to adjust for estimated error (Mnih and Gregor, 2014; Weaver and Tao, 2001). These techniques sample i.i.d. but perhaps it is possible to choose correlated samples that are more *representative* of their underlying distribution? Several such non-independent sampling approaches have been proposed in statistics. In this work we investigate *antithetics*, where for every sample we draw, we include a negatively correlated sample to minimize the distance between sample and population moments.

The key challenges in applying antithetic sampling to modern machine learning are (1) ensuring that antithetic samples are correctly distributed such that they provide unbiased estimators for Monte Carlo simulation, and (2) ensuring that sampling is differentiable to permit gradient-based optimization. We focus on stochastic variational inference and explore using antithetics for learning the parameters for a deep generative model. Critically, our method of antithetic sampling is differentiable and can be composed with reparametrizations of the underlying distributions to provide a fully differentiable sampling process. This yields a simple and low variance way to optimize the parameters of the variational posterior.

Concisely, our contributions are as follows:

- We review a method to to generate Gaussian variates with known sample moments, then apply it to antithetics, and generalize it to other families using deterministic transformations.
- We show that differentiating through the sampling computation improves variational inference.
- We show that training VAEs with antithetic samples improves learning across objectives, posterior

Proceedings of the 22[nd] International Conference on Artificial Intelligence and Statistics (AISTATS) 2019, Naha, Okinawa, Japan. PMLR: Volume 89.

families, and datasets.

2 Background

2.1 Variational Inference and Learning

Consider a generative model that specifies a joint distribution $p_\theta(x, z)$ over a set of observed variables $x \in \mathbb{R}^m$ and stochastic variables $z \in \mathbb{R}^d$ parameterized by θ. We are interested in the posterior distribution $p_\theta(z|x) = \frac{p_\theta(x|z)p(z)}{p(x)}$, which is intractable since $p(x) = \int_z p(x, z)dz$. Instead, we introduce a *variational posterior*, $q_\phi(z|x)$ that approximates $p_\theta(z|x)$ but is easy to sample from and to evaluate.

Our objective is to maximize the likelihood of the data (the "evidence"), $\log p_\theta(x)$. This is intractable so we optimize the evidence lower bound (ELBO) instead:

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log \frac{p_\theta(x, z)}{q_\phi(z|x)}] \tag{1}$$

The VAE (Kingma and Welling, 2013; Rezende et al., 2014) is an example of one such generative model where $p_\theta(x|z)$ and $q_\phi(z|x)$ are both deep neural networks used to parameterize a simple likelihood (e.g., Bernoulli or Gaussian).

Stochastic Gradient Estimation Since ϕ can impact the ELBO (though not the true marginal likelihood it lower bounds), we jointly optimize over θ and ϕ. The gradients of the ELBO objective are:

$$\nabla_\theta \text{ELBO} = \mathbb{E}_{q_\phi(z|x)}[\nabla_\theta \log p_\theta(x, z)] \tag{2}$$

$$\nabla_\phi \text{ELBO} = \nabla_\phi \mathbb{E}_{q_\phi(z|x)}[\log \frac{p_\theta(x, z)}{q_\phi(z|x)}] \tag{3}$$

Eqn. 2 can be directly estimated using Monte Carlo techniques. However, as it stands, Eqn. 3 is difficult to approximate as we cannot distribute the gradient inside the expectation. Luckily, if we constrain $q_\phi(z|x)$ to certain families, we can reparameterize.

Reparameterization Estimators Reparameterization refers to isolating sampling from the gradient computation graph (Kingma and Welling, 2013; Rezende et al., 2014). If we can sample $z \sim q_\phi(z|x)$ by applying a deterministic function $z = g_\phi(\epsilon) : \mathbb{R}^d \to \mathbb{R}^d$ to sampling from an unparametrized distribution, $\epsilon \sim R$, then we can rewrite Eqn. 3 as:

$$\nabla_\phi \text{ELBO} = \mathbb{E}_{\epsilon \in R}[\nabla_z \log \frac{p_\theta(x, z(\epsilon))}{q_\phi(z(\epsilon)|x)} \nabla_\phi g_\phi(\epsilon)] \tag{4}$$

which can now be estimated in the usual manner. As an example, if $q_\phi(z|x)$ is a Gaussian, $\mathcal{N}(\mu, \sigma^2)$ and we choose R to be $\mathcal{N}(0, 1)$, then $g(\epsilon) = \epsilon * \sigma + \mu$.

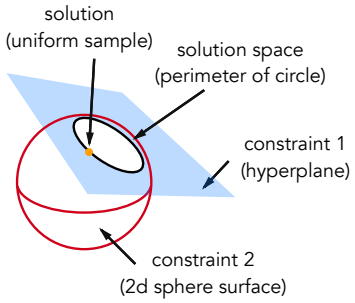


Figure 1: An illustration of Marsaglia's solution to the constrained sampling problem in two dimensions: build a $(k-1)$-dimensional sphere by intersecting a hyperplane and a k-dimensional sphere (each representing a contraint). Generating k samples is equivalent to uniformly sampling from the perimeter of the circle.

2.2 Antithetic Sampling

Normally, we sample i.i.d. from $q_\phi(z|x)$ and R to approximate Eqns. 2 and 4, respectively. However, drawing correlated samples could reduce variance in our estimation. Suppose we are given k samples $z_1, z_2, ..., z_k \sim q_\phi(z|x)$. We could choose a second set of samples $z_{k+1}, z_{k+2}, ..., z_{2k} \sim q_\phi(z|x, z_1, ..., z_k)$ such that z_{i+k} is somehow the "opposite" of z_i. Then, we can write down a new estimator using both sample sets. For example, Eqn. 2 can be approximated by:

$$\frac{1}{2k} \sum_{i=1}^{k} \nabla_\theta \log p_\theta(x, z_i) + \nabla_\theta \log p_\theta(x, z_{i+k}) \tag{5}$$

Assuming $z_{k+1}, ..., z_{2k}$ is marginally distributed according to $q_\phi(z|x)$, Eqn. 5 is unbiased. Moreover, if $q_\phi(z|x)$ is near symmetric, the variance of this new estimator will be cut significantly. But *what does "opposite" mean?* One idea is to define "opposite" as choosing z_{k+i} such that the moments of the combined sample set $z_1, ..., z_{2k}$ match the moments of $q_\phi(z|x)$. Intuitively, if z_i is too large, then choosing z_{k+i} to be too small can help rebalance the sample mean, reducing first order errors. Similarly, if our first set of samples is too condensed at the mode, then choosing antithetic samples with higher spread can stabilize the variance closer to its expectation. However, sampling $z_{k+1}, ..., z_{2k}$ with particular sample statistics in mind is a difficult challenge. To solve this, we first narrow our scope to Gaussian distributions, and later extend to other distribution families.

3 Generating Gaussian Variates with Given Sample Mean and Variance

We present the *constrained sampling problem*: given a Gaussian distribution with *population mean* μ and

population variance σ^2, we wish to generate k samples $x_1, ..., x_k \sim \mathcal{N}(\mu, \sigma^2)$ subject to the conditions:

$$\frac{1}{k}\sum_{i=1}^{k} x_i = \eta \tag{6}$$

$$\frac{1}{k}\sum_{i=1}^{k}(x_i - \eta)^2 = \delta^2 \tag{7}$$

where the constants η and δ^2 are given and represent the *sample mean* and *sample variance*. In other words, how can we draw samples from the correct marginal distribution conditioned on matching desired *sample* moments? For example, we might wish to match *sample* and *population* moments: $\eta = \mu$ and $\delta = \sigma$.

Over forty years, there have been a handful of solutions. We review the algorithm introduced by (Marsaglia and Good, 1980). In our experiments, we reference a second algorithm by (Pullin, 1979; Cheng, 1984), which is detailed in the supplement. We chose (Marsaglia and Good, 1980) due its simplicity, low computational overhead, and the fact that it makes the fewest random choices of proposed solutions.

Intuition Since $x_1, ..., x_k$ are independent, we can write the joint density function as follows:

$$p(x_1, ..., x_k) = (2\pi\sigma^2)^{-\frac{k}{2}} e^{-\frac{1}{2\sigma^2}\sum_i (x_i - \mu)^2} \tag{8}$$

We can interpret Eqn. 6 as a hyperplane and Eqn. 7 as the surface of a sphere in k dimensions. Let $\mathcal{X}$ be the set of all points $(x_1, ..., x_k) \in \mathbb{R}^k$ that satisfy the above constraints. Geometrically, we can view $\mathcal{X}$ as the intersection between the hyperplane and k-dimensional sphere, i.e., the surface of a $(k-1)$ dimensional sphere (e.g. a circle if $k = 2$).

We make the following important observation: the joint density (Eqn. 8) is constant for all points in $\mathcal{X}$. To see this, we can write the following:

$$\begin{aligned}\sum_i (x_i - \mu)^2 &= \sum_i (x_i - \eta)^2 \\ &\quad + 2(\eta - \mu)\sum_i (x_i - \eta) + k(\eta - \mu)^2 \\ &= \sum_i (x_i - \eta)^2 + k(\eta - \mu)^2 \\ &= k\delta^2 + k(\eta - \mu)^2\end{aligned}$$

where $\sum_i (x_i - \eta) = \sum_i (x_i) - k\eta = 0$ by Eqn. 6. Plugging this into the density function, rewrite Eqn. 8 as:

$$p(x_1, ..., x_k) = (2\pi\sigma^2)^{-\frac{k}{2}} e^{-\frac{1}{2\sigma^2}(k\delta^2 + k(\eta-\mu)^2)} \tag{9}$$

Critically, Eqn. 9 is independent of $x_1, ..., x_k$. For any $(\eta, \delta, \mu, \sigma)$, the density for every $x \in \mathcal{X}$ is constant. In other words, the conditional distribution of $x_1, ..., x_k$ given that $x_1, ..., x_k \in \mathcal{X}$ is the uniform distribution over $\mathcal{X}$. Surprisingly, it does *not* depend on μ or σ.

Therefore, to solve the constrained sampling problem, we need only be able to sample uniformly from the surface of a $(k-1)$ dimensional sphere.

Marsaglia's Solution More precisely, we can generate the required samples $\mathbf{x} = (x_1, ..., x_k)$ from a point $\mathbf{z} = (z_1, ..., z_{k-1})$ uniformly distributed on the unit sphere in $\mathbb{R}^{k-1}$ centered at the origin by solving the linear system:

$$\mathbf{x} = k^{\frac{1}{2}}\delta\mathbf{z}B + \eta\mathbf{v} \tag{10}$$

where $\mathbf{v} = (1, 1, ..., 1)$ is a k dimensional vector of ones and B is a $(k-1)$ by k matrix such that the rows of B form an orthonormal basis with the null space of $\mathbf{v}$ i.e. we choose B where $BB^t = I$ and $B\mathbf{v}^t = 0$, which happens to satisfy our constraints:

$$\mathbf{x}\mathbf{v}^t = k\eta \tag{11}$$

$$(\mathbf{x} - \eta\mathbf{v})(\mathbf{x} - \eta\mathbf{v})^t = k\delta^2\mathbf{z}BB^t\mathbf{z}^t = k\delta^2 \tag{12}$$

As $\mathbf{z}$ is uniformly distributed over the unit $(k-1)$ sphere, Eqn. 11 and 12 guarantee that $\mathbf{x}$ is uniformly distributed in $\mathcal{X}$. We can generate $\mathbf{z}$ by sampling $(\epsilon_1, ..., \epsilon_{k-1}) \sim \mathcal{N}(0, 1)$ and setting $z_i = \epsilon_i / \sum_i \epsilon_i^2$. As in (Marsaglia and Good, 1980), we set B to RowNormalize(A) where A is defined as

$$\begin{bmatrix} 1-k & 1 & 1 & . & . & . & 1 & 1 & 1 \\ 0 & 2-k & 1 & . & . & . & 1 & 1 & 1 \\ 0 & 0 & 3-k & . & . & . & 1 & 1 & 1 \\ . & & & & & & & & . \\ . & & & & & & & & . \\ . & & & & & & & & . \\ 0 & 0 & 0 & . & . & . & -2 & 1 & 1 \\ 0 & 0 & 0 & . & . & . & 0 & -1 & 1 \end{bmatrix}$$

and RowNormalize divides each row vector in A by the sum of the elements in that row. We summarize the procedure in Alg. 1 and the properties of MarsagliaSample in Prop. 1.

Proposition 1. *For any $k > 2$, $\mu \in \mathbb{R}$ and $\sigma^2 > 0$, if $\eta \sim \mathcal{N}(\mu, \frac{\sigma^2}{k})$ and $\frac{(k-1)\delta^2}{\sigma^2} \sim \chi^2_{k-1}$ and $\boldsymbol{\epsilon} = \epsilon_1, ..., \epsilon_{k-1} \sim \mathcal{N}(0, 1)$ i.i.d., then the generated samples $x_1, ..., x_k =$ MarsagliaSample$(\boldsymbol{\epsilon}, \eta, \delta^2, k)$ are independent normal variates sampled from $\mathcal{N}(\mu, \sigma^2)$ such that $\frac{1}{k}\sum_i x_i = \eta$ and $\frac{1}{k}\sum_i (x_i - \eta)^2 = \delta^2$.*

Proof Sketch. We provide a full proof in the supplement. For a sketch, let $\mathbf{x} = (x_1, ..., x_k)$ such that $x_i \sim \mathcal{N}(\mu, \sigma^2)$ i.i.d. Compute sample statistics η, δ^2 from $\mathbf{x}$ as defined in Eqn. 13. Consider the joint distribution over samples and sample moments:

$$p(\mathbf{x}, \eta, \delta^2) = p(\eta, \delta^2)p(\mathbf{x}|\eta, \delta^2)$$

. We make two observations: first, η, δ^2, as defined, are drawn from $p(\eta, \delta^2)$. Second, as hinted above,

Algorithm 1: MARSAGLIASAMPLE

Data: i.i.d. samples $\epsilon_1, ..., \epsilon_{k-1} \sim \mathcal{N}(0,1)$;
Desired sample mean η and variance δ^2;
Number of samples $k \in \mathbb{N}$.
Result: A set of k samples $x_1, x_2, ..., x_k$ marginally distributed as $\mathcal{N}(\mu, \sigma^2)$ with sample mean η and sample variance δ^2.

$\gamma = \sqrt{n}\delta$;
$s = \sum_i \epsilon_i^2$;
for $i \leftarrow 1$ **to** k **do**
  $z_i = \epsilon_i[(k-i)(k-i+1)s]^{-1/2}$;
end
$x_1 = (1-k)\gamma z_1 + \eta$;
$x_k = \gamma \sum_{i=1}^{k-1} z_i + \eta$;
for $i \leftarrow 2$ **to** $k-1$ **do**
  $x_i = (\sum_{i=1}^{i-1} + (i-k)z_i)\gamma + \eta$;
end
Return $x_1, ..., x_k$;

$p(\mathbf{x}|\eta, \delta^2)$ is the uniform distribution over a $(k-1)$-sphere, which Marsaglia shows us how to sample from. Thus, any samples $\mathbf{x}' \sim p(\mathbf{x}|\eta = \eta, \delta^2 = \delta^2)$ will be distributed as $\mathbf{x}$ is (marginally), in other words i.i.d. Gaussian. □

As implied in Prop. 1, if we happen to know the population mean μ and variance σ^2 (as we do in variational inference), we could generate k i.i.d. Gaussian variates by sampling $\eta \sim \mathcal{N}(\mu, \frac{\sigma^2}{k})$ and $\frac{(k-1)\delta^2}{\sigma^2} \sim \chi^2_{k-1}$, and passing η, δ^2 to MARSAGLIASAMPLE.

4 Constrained Antithetic Sampling

We might be inclined to use MARSAGLIASAMPLE to directly generate samples with some fixed *deterministic* $\eta = \mu$ and $\delta = \sigma$. However, Prop. 1 holds only if the desired sample moments η, δ are *random* variables. If we choose them deterministically, we can no longer guarantee the correct marginal distribution for the samples, thus precluding their use for Monte Carlo estimates. Instead, what we can do is compute η and δ^2 from i.i.d. samples from $\mathcal{N}(\mu, \sigma^2)$, derive antithetic sample moments, and use MARSAGLIASAMPLE to generate a second set of samples distributed accordingly.

More precisely, given a set of k independent normal variates $(x_1, ..., x_k) \sim \mathcal{N}(\mu, \sigma^2)$, we would like to generate a new set of k normal variates $(x_{k+1}, ..., x_{2k})$ such that the combined sample moments match the population moments, $\frac{1}{2k}\sum_{i=1}^{2k} x_i = \mu$ and $\frac{1}{2k}\sum_{i=1}^{2k}(x_i - \mu)^2 = \sigma^2$. We call the second set of samples $(x_{k+1}, ..., x_{2k})$ *antithetic* to the first set.

We compute sample statistics from the first set:

$$\eta = \frac{1}{k}\sum_{i=1}^{k} x_i \qquad \delta^2 = \frac{1}{k}\sum_{i=1}^{k}(x_i - \mu)^2 \tag{13}$$

Note that η, δ are random variables, satisfying $\eta \sim \mathcal{N}(\mu, \frac{\sigma^2}{k})$ and $\frac{(k-1)\delta^2}{\sigma^2} \sim \chi^2_{k-1}$. Ideally, we would want the second set to come from an "opposing" η' and δ'. To choose η' and δ', we leverage the *inverse CDF transform*: given the cumulative distribution function (CDF) for a random variable X, denoted F_X, we can define a uniform variate $Y = F_X(X)$. The *antithetic* uniform variable is then $Y' = 1 - Y$, which upon application of the inverse CDF function, is mapped back to a properly distributed antithetic variate $X' = F_X^{-1}(Y')$. Crucially, X and X' have the same marginal distribution, but are not independent.

Let F_η represent a Gaussian CDF and F_δ represent a Chi-squared CDF. We can derive η' and δ' as:

$$\eta' = F_\eta^{-1}(1 - F_\eta(\eta)) \tag{14}$$

$$\frac{(k-1)(\delta')^2}{\sigma^2} = F_\delta^{-1}\left(1 - F_\delta\left(\frac{(k-1)\delta^2}{\sigma^2}\right)\right) \tag{15}$$

Crucially, η', δ' chosen this way are random variables with the correct marginal distributions, i.e., $\eta' \sim \mathcal{N}(\mu, \frac{\sigma^2}{k})$ and $\frac{(k-1)(\delta')^2}{\sigma^2} \sim \chi^2_{k-1}$. knowing η', δ', it is straightforward to generate antithetic samples with MARSAGLIASAMPLE. We summarize the algorithm in Alg. 2 and its properties in Prop. 2.

Algorithm 2: ANTITHETICSAMPLE

Data: i.i.d. samples $(x_1, ..., x_k) \sim \mathcal{N}(\mu, \sigma^2)$;
i.i.d. samples $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_{k-1}) \sim \mathcal{N}(0,1)$;
Population mean μ and variance σ^2;
Number of samples $k \in \mathbb{N}$.
Result: A set of k samples $(x_{k+1}, x_{k+2}, ..., x_{2k})$ marginally distributed as $\mathcal{N}(\mu, \sigma^2)$ with sample mean η' and sample standard deviation δ'.

$v = k - 1$;
$\eta = \frac{1}{k}\sum_{i=1}^{k} x_i$;
$\delta^2 = \frac{1}{k}\sum_{i=1}^{k}(x_i - \eta)^2$;
$\eta' = F_\eta^{-1}(1 - F_\eta(\eta))$;
$\lambda = v\delta^2/\sigma^2$;
$\lambda' = F_\delta^{-1}(1 - F_\delta(\lambda))$;
$(\delta')^2 = \lambda'\sigma^2/v$;
$(x_{k+1}, ..., x_{2k}) =$
 MARSAGLIASAMPLE$(\boldsymbol{\epsilon}, \eta', (\delta')^2, k)$;
Return $(x_{k+1}, ..., x_{2k})$;

Proposition 2. *Given $k-1$ i.i.d samples $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_{k-1}) \sim \mathcal{N}(0,1)$, k i.i.d. samples $\boldsymbol{x} =$*

$(x_1, ..., x_k) \sim \mathcal{N}(\mu, \sigma^2)$, *let* $(x_{k+1}, ..., x_{2k}) =$ AntitheticSample$(\boldsymbol{x}, \boldsymbol{\epsilon}, \mu, \sigma^2, k)$ *be the generated antithetic samples. Then:*

1. $x_{k+1}, ..., x_{2k}$ *are independent normal variates sampled from* $\mathcal{N}(\mu, \sigma^2)$.
2. *The combined sample mean* $\frac{1}{2k}\sum_{i=1}^{2k} x_i$ *is equal to the population mean* μ.
3. *The sample variance of* $x_{k+1}, ..., x_{2k}$ *is anticorrelated with the sample variance of* $x_1, ..., x_k$.

Proof. The first property follows immediately from Prop. 1, as by construction η', δ' have the correct marginal distribution. Simple algebra shows that the inverse Gaussian CDF transform simplifies to $\eta' = 2*\mu - \eta$, giving the desired relationship $\eta/2 + \eta'/2 = \mu$. The third property follows from Eq. 15. □

Since both sets of samples share the same (correct) marginal distribution, $x_1, ..., x_{2k}$ can be used to obtain unbiased Monte Carlo estimates.

Proposition 3. *Given* $k-1$ *i.i.d samples* $\boldsymbol{\epsilon} = (\epsilon_1, ..., \epsilon_{k-1}) \sim \mathcal{N}(0, 1)$, k *i.i.d. samples* $\boldsymbol{x} = (x_1, ..., x_k) \sim \mathcal{N}(\mu, \sigma^2)$, *let* $(x_{k+1}, ..., x_{2k}) =$ AntitheticSample$(\boldsymbol{x}, \boldsymbol{\epsilon}, \mu, \sigma^2, k)$ *be the generated antithetic samples. Then* $\frac{1}{2k}\sum_{i=1}^{2k} f(x_i)$ *is an unbiased estimator of* $\mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[f(x)]$.

Proof. Let $q(x_1, \cdots, x_k, x_{k+1}, \cdots, x_{2k})$ denote the joint distribution of the $2k$ samples. Note the two groups of samples $(x_1, ..., x_k)$ and $(x_{k+1}, ..., x_{2k})$ are not independent. However,

$$\mathbb{E}_{(x_1,\cdots,x_k,x_{k+1},\cdots,x_{2k})\sim q}\left[\frac{1}{2k}\sum_{i=1}^{2k} f(x_i)\right] =$$

$$\frac{1}{2k}\sum_{i=1}^{2k}\mathbb{E}_{x_i\sim q_i(x_i)}\left[f(x_i)\right] = \mathbb{E}_{x\sim\mathcal{N}(\mu,\sigma^2)}[f(x)]$$

because by assumption and Prop. 2, each x_i is marginally distributed as $\mathcal{N}(\mu, \sigma^2)$. □

4.1 Approximate Antithetic Sampling

If F_η and F_δ were well-defined and invertible, we could use Alg. 2 as is, with its good guarantees. On one hand, since η is normally distributed, the inverse CDF transform simplifies to:

$$\eta' = 2 * \mu - \eta \tag{16}$$

However, there is no general closed form expression for F_δ^{-1}. Our options are then to either use a discretized table of probabilities or approximate the inverse CDF. Because we desire differentiability, we choose to use a normal approximation to F_δ.

Antithetic Hawkins-Wixley Canal (2005) surveys a variety of normal approximations, all of which are a linear combination of χ^2 variates to a power root. We choose to use (Hawkins and Wixley, 1986) as (P1) it is an even power, (P2) it contains only 1 term involving a random variate, and (P3) is shown to work better for smaller degrees of freedom (smaller sample sizes). We derive a closed form for computing δ' from δ by combining the normal approximation with Eqn. 16. We denote this final transform as the *antithetic Hawkins-Wixley transform*:

$$\lambda' = v(2(1 - \frac{3}{16v} - \frac{7}{512v^2} + \frac{231}{8192v^3}) - (\frac{\lambda}{v})^{1/4})^4 \tag{17}$$

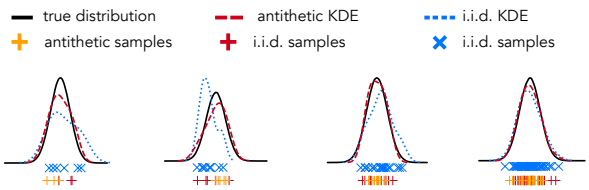
where $\lambda \sim \chi^2_v$ with v being the degree of freedom. Therefore, if we set $\lambda = (k-1)\delta^2/\sigma^2 \sim \chi^2_{k-1}$ and $v = k - 1$, then we can derive $(\delta')^2 = \lambda'\sigma^2/(k-1)$ where λ' is computed as in Eqn. 17, whose derivation can be found in the supplementary material.

P1 is important as odd degree approximations e.g. (Wilson and Hilferty, 1931) can result in a negative value for λ' under small k. P2 is required to derive a closed form as most linear combinations do not factor. P3 is desirable for variational inference.

To update Alg. 2, we swap the fourth line with Eqn. 16 and the sixth line with Eqn. 17. The first property in Prop. 2 and therefore also Prop. 3 do not hold anymore: the approximate AntitheticSample has bias that depends on the approximation error in Eqn. 17. In practice, we find the approximate AntitheticSample to be effective. From now on, when we refer to AntitheticSample, we refer to the approximate version. See supplement for a written algorithm. We refer to Fig. 2 for an illustration of the impact of antithetics: sampling i.i.d. could result in skewed sample distributions that over-emphasize the mode or tails, especially when drawing very few samples. Including antithetic samples helps to "stabilize" the sample distribution to be closer to the true distribution.

5 Generalization to Other Families

Marsaglia and Good (1980)'s algorithm is restricted to distribution families that can be transformed to a unit sphere (primarily Gaussians), as are many similar algorithms (Cheng, 1984; Pullin, 1979). However, we can explore "generalizing" AntitheticSample to a wider class of families by first antithetically sampling in a Gaussian distribution, then transforming its samples to samples from another family using a deterministic function, $g : \mathbb{R}^d \to \mathbb{R}^d$. Although we are not explicitly matching the moments of the derived distributions, we expect that transformations of more representative samples in an initial distribution may be



(a) $k = 8$ (b) $k = 10$ (c) $k = 20$ (d) $k = 50$

Figure 2: The effect of AntitheticSample in 1 dimension. We vary the number of samples k, and plot the true distribution (solid black line), a kernel density estimate (KDE) of the empirical distribution (dotted blue line) of $2k$ i.i.d. samples (blue), and a KDE of the empirical distribution (dashed red line) of k i.i.d. samples (red) pooled with k antithetic samples (orange). This snapshot was taken from the first epoch of training an AntiVAE on dynamic MNIST.

more representative in the transformed distribution. We now discuss a few candidates for $g(\cdot)$.

5.1 One-Liners

Devroye (1996) presents a large suite of "one line" transformations between distributions. We focus on three examples starting from a Gaussian to (1) Log Normal, (2) Exponential, and (3) Cauchy. Many additional transformations (e.g. to Pareto, Gumbel, Weibull, etc.) can be used in a similar fashion. Let F_x refer to the CDF of a random variable x. See supplementary material for derivations.

Log Normal $g(z) = e^z$ where $z \sim \mathcal{N}(\mu, \sigma^2)$.

Exponential Let $F_x(x) = 1 - \exp^{\lambda x}$ where $\lambda \in \mathbb{R}^d$ is a learnable parameter. Then $F_x^{-1}(y) = -\frac{1}{\lambda}\log y$. Thus, $g(u, \lambda) = -\frac{1}{\lambda}\log u$ where $u \in U(0, 1)$.

Cauchy Let $F_x(x) = \frac{1}{2} + \frac{1}{\pi}\arctan(\frac{x-x_0}{\gamma})$ where $x_0 \in \mathbb{R}^d, \gamma \in \mathbb{R}^d$ are learnable parameters. Then $F_x^{-1}(y) = \gamma(\tan(\pi y) + x_0)$. Given $u \in U(0, 1)$, we define $g(u, x_0, \gamma) = \gamma(\tan(\pi u) + x_0)$.

5.2 Deeper Flows

One liners are an example of a simple flow where we know how to score the transformed sample. If we want more flexible distributions, we can apply normalizing flows (NF). A *normalizing flow* (Rezende and Mohamed, 2015) applies T invertible transformations $h^{(t)}, t = 1, ..., T$ to samples $z^{(0)}$ from a simple distribution, leaving $z^{(T)}$ as a sample from a complex distribution. A common normalizing flow is a linear-time transformation: $g(z) = z + u(h(w^T z + b))$ where $w \in \mathbb{R}^d, u \in \mathbb{R}^d, b \in \mathbb{R}$ are learnable parameters, and h is a non-linearity. In variational inference, flows enable us to parameterize a wider set of posterior families.

We can also achieve flexible posteriors using volume-preserving flows (VPF), of which Tomczak and Welling (2016) introduced the Householder transformation: $g(z) = (I - 2\frac{v \cdot v^T}{\|v\|^2})z$ where $v \in \mathbb{R}^d$ is a trainable parameter. Critically, the Jacobian-determinant is 1.

Algorithm 3: AntiVAE Inference

Data: A observation x; number of samples $k \geq 6$; a variational posterior $q(z|x)$ e.g. a d-dimensional Gaussian, $\mathcal{N}^d(\mu, \sigma^2)$.
Result: Samples $z_1^d, ..., z_k^d \sim q_{\mu,\sigma}(z|x)$ that match moments.

μ^d, σ^d =InferenceNetwork(x);
μ = Flatten(μ^d);
σ = Flatten(σ^d);
$\epsilon_1, ..., \epsilon_{kd/2} \sim \mathcal{N}(0, 1)$;
$\boldsymbol{\xi} = \xi_1, ..., \xi_{\frac{kd}{2}-1} \sim \mathcal{N}(0, 1)$;
for $i \leftarrow 1$ **to** $kd/2$ **do**
  $y_i = \epsilon_i * \sigma + \mu$;
end
$\boldsymbol{y} = (y_1, ..., y_{kd/2})$;
$y_{\frac{kd}{2}+1}, ..., y_{kd}$ =AntitheticSample$(\boldsymbol{y}, \boldsymbol{\xi}, \mu, \sigma)$;
$\boldsymbol{z} = (y_1, ..., y_{kd})$;
$z_1^d, ..., z_k^d$ = UnFlatten$(\boldsymbol{z})$;
Return $z_1^d, ..., z_k^d$;

6 Differentiable Antithetic Sampling

Finally, we can use AntitheticSample to approximate the ELBO for variational inference.

For a given observation $x \in p_{\text{data}}(x)$ from an empirical dataset, we write the antithetic gradient estimators as:

$$\nabla_\theta \text{ELBO} \approx \frac{1}{2k}\sum_{i=1}^{k}[\nabla_\theta \log p_\theta(x, z_i) + \nabla_\theta \log p_\theta(x, z_{i+k})] \tag{18}$$

$$\nabla_\phi \text{ELBO} \approx \frac{1}{2k}\sum_{i=1}^{k}[\nabla_z \log \frac{p_\theta(x, z_i(\epsilon))}{q_\phi(z_i(\epsilon)|x)}\nabla_\phi g_\phi(\epsilon_i) + \nabla_z \log \frac{p_\theta(x, z_{i+k}(\epsilon))}{q_\phi(z_{i+k}(\epsilon)|x)}\nabla_\phi g_\phi(\epsilon_{i+k})] \tag{19}$$

where $(\epsilon_1, ..., \epsilon_k) \sim \mathcal{N}(0, 1)$, $\boldsymbol{\xi} = (\xi_1, ..., \xi_{k-1})$, $\boldsymbol{z} = (z_1, ..., z_k) \sim q_\phi(z|x)$, and $(z_{k+1}, ..., z_{2k}) =$ AntitheticSample$(\boldsymbol{z}, \boldsymbol{\xi}, \mu, \sigma^2, k)$. Optionally, $\boldsymbol{z} =$ Transform$(\boldsymbol{z}, \alpha)$ where Transform denotes any sample transformation(s) with parameters α.

Alternative variational bounds have been considered recently, including an importance-weighted estimator of the ELBO, or IWAE (Burda et al., 2015). Antithetic sampling can be applied in a similar fashion, as also shown in (Shu et al., 2019).

Importantly, AntitheticSample is a special instance of a reparameterization estimator. Aside from samples from a parameter-less distribution (unit Gaussian), AntitheticSample is completely deterministic, meaning that it is differentiable with respect to the population moments μ and σ^2 by any modern auto-differentiation library. Allowing backpropagation through AntitheticSample means that any free parameters are aware of the sampling strategy. Thus, including antithetics will change the optimization trajectory, resulting in a different variational posterior than if we had used i.i.d samples alone. In Sec. 8, we show experimentally that *most of the benefit* of differentiable antithetic sampling comes from being differentiable.

Alg. 3 summarizes inference in a VAE using differentiable antithetic sampling (denoted by AntiVAE[1]). To the best of our knowledge, the application of antithetic sampling to stochastic optimization, especially variational inference is novel. Both the application of (Marsaglia and Good, 1980) to drawing antithetics and the extension of AntitheticSample to other distribution families by transformation is novel. This is also the first instance of differentiating through an antithetic sample generator.

7 Experiments

We compare performance of the VAE and AntiVAE on seven image datasets: static MNIST (Larochelle and Murray, 2011), dynamic MNIST (LeCun et al., 1998), FashionMNIST (Xiao et al., 2017), OMNIGLOT (Lake et al., 2015), Caltech 101 Silhouettes (Marlin et al., 2010), Frey Faces[2], and Histopathology patches (Tomczak and Welling, 2016). See supplement for details.

In both VAE and AntiVAE, $q_\phi(z|x)$ and $p_\theta(x|z)$ are two-layer MLPs with 300 hidden units, Xavier initialization (Glorot and Bengio, 2010), and ReLU. By default, we set $d = 40$ and $k = 8$ (i.e. 4 antithetic samples) and optimize either the ELBO or IWAE. For grayscale images, $p_\theta(x|z)$ parameterize a discretized logistic distribution as in (Kingma et al., 2016; Tomczak and Welling, 2017). The log variance from $p_\theta(x|z)$ is clamped between -4.5 and 0.0 (Tomczak and Welling, 2017). We use Adam (Kingma and Ba, 2014) with a fixed learning rate of $3 \cdot 10^{-4}$ and a mini-batch of 128. We train for 500 epochs. Test marginal log likelihoods are estimated via importance sampling using 100 i.i.d. samples. See supplement for additional experiments where we vary architectures, measure runtimes, report variance over many runs, and more.

[1]For some experiments, we use Cheng's algorithm instead of Marsaglia's. We refer to this as AntiVAE (Cheng).

[2]https://cs.nyu.edu/ roweis/data.html

8 Results

Fig. 3 and Table 1 show test log likelihoods (over 5 runs). We summarize findings below:

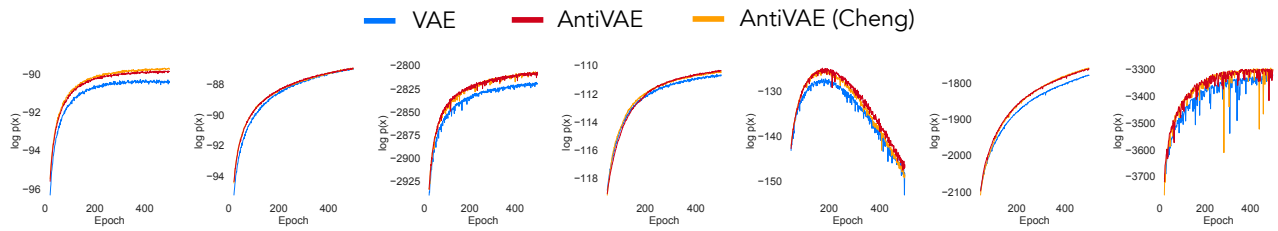
VAE vs AntiVAE AntiVAE consistently achieves higher log likelihoods, usually by a margin of 2 to 5 log units. With FashionMNIST/Histopathology, the margin grows to as much as 30 log units. In the 3 cases that AntiVAE performs worse than VAE, the log-likelihoods are almost equal (≤ 1 log unit). In Fig. 3b, we see a case where, even when the final performance is equivalent, AntiVAE learns faster. We find similar behavior using a tighter IWAE bound or other posterior families defined by one liners and flows. With the latter, we see improvements of up to 25 log units. A better sampling strategy is effective regardless of the choice of objective and distributional family.

As k increases, the effect of antithetic sampling diminishes. Fig. 4a illustrates that as the number of samples $k \to \infty$, posterior samples will match the true moments of $q_\phi(z|x)$ regardless of the sampling strategy. But as $k \to 0$, the effectiveness grows quickly. We expect best performance at small (but not too small) k where the normal approximation (Eqn. 17) is decent and the value of antithetics is high.

As d increases, the effect of antithetic sampling grows. Fig. 4b illustrates that the importance of sampling strategy increases as the dimensionality grows due to an exponential explosion in the volume of the sample space. With higher dimensionality, we find antithetic sampling to be more effective.

Backpropagating through antithetic sampling greatly improves performance. From Fig. 4c, 4d, we see that most of the improvement from antithetics relies on differentiating through AntitheticSample. This is sensible as the model can adjust parameters if it is aware of the sampling strategy, leading to better optima. Even if we do not backpropagate through sampling (draw antithetic samples from $\mathcal{N}(0, 1)$ followed by standard reparameterization), we will still find modest improvement over i.i.d. sampling.

We believe differentiability encourages initial samples to be more diverse. To test this, we measure the variance of the first $k/2$ samples (1) without antithetics, (2) with non-differentiable antithetics, and (3) with

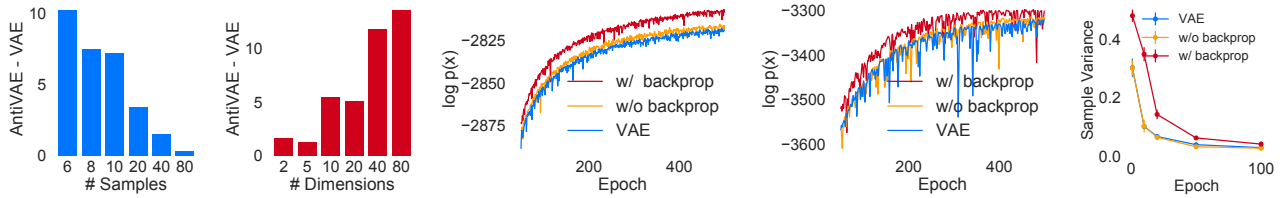


(a) static (b) dynamic (c) Fashion (d) Omniglot (e) Caltech (f) Frey (g) Hist.

Figure 3: A comparison of test log likelihoods over 500 epochs between VAE and AntiVAE. Transforming samples to match moments seems to have different degrees of effectiveness depending on the data domain. However, we find that the test ELBO with AntiVAE is almost always greater or equal to that of the VAE. This behavior is not sensitive to hyperparameters e.g. learning rate or MLP hidden dimension. For each subplot, we start plotting from epoch 20 to 500. We cannot resample observations in Caltech101, leading to overfitting.

Model	stat. MNIST	dyn. MNIST	FashionMNIST	Omniglot	Caltech	Frey	Hist.
VAE	-90.44	-86.96	-2819.13	-110.65	-127.26	-1778.78	-3320.37
AntiVAE	-89.74	-86.94	-2807.06	**-110.13**	**-124.87**	-1758.66	-3293.01
AntiVAE (Cheng)	**-89.70**	**-86.93**	**-2806.71**	-110.39	-125.19	**-1758.29**	**-3292.72**
VAE+IWAE	-89.78	-86.71	-2797.02	**-109.32**	-123.99	-1772.06	-3311.23
AntiVAE+IWAE	**-89.71**	**-86.62**	**-2793.01**	-109.48	**-123.35**	**-1771.47**	**-3305.91**
VAE ($\log \mathcal{N}$)	**-149.47**	-145.13	-2891.75	-164.01	-269.51	-1910.11	-3460.18
AntiVAE ($\log \mathcal{N}$)	-149.78	**-141.76**	**-2882.11**	**-163.55**	**-266.82**	**-1895.15**	**-3454.54**
VAE (Exp.)	**141.95**	-140.91	-2971.00	-159.92	-200.14	-2176.83	-3776.48
AntiVAE (Exp.)	141.98	**-140.58**	**-2970.12**	**-158.15**	**-197.47**	**-2156.93**	**-3770.33**
VAE (Cauchy)	-217.69	-217.53	-3570.53	-187.34	-419.78	-2404.24	-3930.40
AntiVAE (Cauchy)	**-215.89**	**-217.12**	**-3564.80**	**-186.02**	**-417.0**	**-2395.07**	**-3926.95**
VAE+10-NF	-90.07	-86.93	-2803.98	-110.03	-128.62	-1780.61	-3328.68
AntiVAE+10-NF	**-89.77**	**-86.57**	**-2801.90**	**-109.43**	**-127.23**	**-1777.26**	**-3303.00**
VAE+10-VPF	-90.59	-86.99	-2802.65	-110.19	-128.87	-1789.18	-3312.30
AntiVAE+10-VPF	**-90.00**	**-86.59**	**-2797.05**	**-109.04**	**126.72**	**-1787.18**	**-3305.42**

Table 1: Test log likelihoods between the VAE and AntiVAE under different objectives and posterior families (a higher number is better). Architecture and hyperparameters are consistent across models. AntiVAE (Cheng) refers to drawing antithetic sampling using an alternative algorithm to Marsaglia (see supplement). Results show the average over 5 independent runs with different random seeds. For measurements of variance, see supplement.



(a) FashionMNIST (b) FashionMNIST (c) FashionMNIST (d) Histopathology (e) Histopathology

Figure 4: (a) With more samples, the difference in $\log p(x)$ between AntiVAE and VAE approaches 0. (b) The benefit of antithetics varies directly with dimensionality. (c) Backpropagating through AntitheticSample is responsible for most of the improvement over i.i.d. sampling. However, even without it, antithetics outperforms VAE. (d) Similar observation in Histopathology. (e) Differentiable antithetics encourages sample diversity.

differentiable antithetics. Fig. 4e shows that samples in (3) have consistently higher variance than (1) or (2).

AntiVAE runtimes are comparable. We measure an average 0.004 sec. increase in wallclock time per step when adding in antithetics.

9 Conclusion

We present a differentiable antithetic sampler for variance reduction. We show its benefits for a family of VAEs. We hope to apply it to reinforcement learning using pathwise derivatives (Levy and Ermon, 2018).

Acknowledgements

This research was supported by NSF (#1651565, #1522054, #1733686), ONR, AFOSR (FA9550-19-1-0024), FLI, and SocioNeticus (part of the DARPA SocialSim program). MW is supported by NSF GRFP.

References

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

L. Canal. A normal approximation for the chi-square distribution. *Computational statistics & data analysis*, 48(4):803–808, 2005.

R. C. Cheng. Generation of inverse gaussian variates with given sample mean and dispersion. *Applied statistics*, pages 309–316, 1984.

L. Devroye. Random variate generation in one line of code. In *Proceedings of the 28th conference on Winter simulation*, pages 265–272. IEEE Computer Society, 1996.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

D. M. Hawkins and R. Wixley. A note on the transformation of chi-squared variables to normality. *The American Statistician*, 40(4):296–298, 1986.

E. Jang, S. Gu, and B. Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

H. Larochelle and I. Murray. The neural autoregressive distribution estimator. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 29–37, 2011.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

D. Levy and S. Ermon. Deterministic policy optimization by combining pathwise and score function estimators for discrete action spaces. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

B. Marlin, K. Swersky, B. Chen, and N. Freitas. Inductive principles for restricted boltzmann machine learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 509–516, 2010.

G. Marsaglia and I. Good. C69. generating a normal sample with given sample mean and variance. *Journal of Statistical Computation and Simulation*, 11 (1):71–74, 1980.

A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

D. Pullin. Generation of normal variates with given sample mean and variance. *Journal of Statistical Computation and Simulation*, 9(4):303–309, 1979.

R. Ranganath, S. Gerrish, and D. Blei. Black box variational inference. In *Artificial Intelligence and Statistics*, pages 814–822, 2014.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

J. Schulman, N. Heess, T. Weber, and P. Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pages 3528–3536, 2015.

R. Shu, H. H. Bui, j. Whang, and S. Ermon. Training variational autoencoders with buffered stochastic variational inference. In *AISTATS*, 2019.

D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller. Deterministic policy gradient algorithms. In *ICML*, 2014.

J. M. Tomczak and M. Welling. Improving variational auto-encoders using householder flow. *arXiv preprint arXiv:1611.09630*, 2016.

J. M. Tomczak and M. Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

L. Weaver and N. Tao. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in*

artificial intelligence, pages 538–545. Morgan Kaufmann Publishers Inc., 2001.

E. B. Wilson and M. M. Hilferty. The distribution of chi-square. *Proceedings of the National Academy of Sciences*, 17(12):684–688, 1931.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.