Adaptive Antithetic Sampling for Variance Reduction

Hongyu Ren [*1] **Shengjia Zhao** [*1] **Stefano Ermon** [1]

Abstract

Variance reduction is crucial in stochastic estimation and optimization problems. Antithetic sampling reduces the variance of a Monte Carlo estimator by drawing correlated, rather than independent, samples. However, designing an effective correlation structure is challenging and application specific, thus limiting the practical applicability of these methods. In this paper, we propose a general-purpose adaptive antithetic sampling framework. We provide gradient-based and gradient-free methods to train the samplers such that they reduce variance while ensuring that the underlying Monte Carlo estimator is provably unbiased. We demonstrate the effectiveness of our approach on Bayesian inference and generative model training, where it reduces variance and improves task performance with little computational overhead.

1. Introduction

The problem of computing expectations that are too complex to be evaluated analytically is ubiquitous in machine learning. To address this difficulty, Monte Carlo estimation underlies the majority of modern machine learning algorithms. Instead of computing the exact expectation, Monte Carlo estimators draw samples from the underlying distribution and use them to compute an empirical mean. When the number of samples is sufficiently large, this empirical mean will approach the expectation according to the law of large numbers. Despite this guarantee, Monte Carlo estimators can still deviate substantially from the true expectation when the sample size is small.

The crux of Monte Carlo estimation is variance. According to Chebyshev inequality, the estimation error is bounded by the variance of the estimator. Thus, with lower variance we could achieve better estimation error guarantees. Increasing the number of samples is the most direct way to decrease the variance but at the cost of additional computation. Hence, a variety of variance reduction techniques have been proposed in the literature. Classic approaches include baselines (Weaver & Tao, 2001), control variates (Greensmith et al., 2004), importance sampling (Neal, 2001), rejection sampling, Rao-Blackwellization (Grisetti et al., 2007) etc. These techniques modify the underlying distribution, but still rely on the basic idea of taking *independent* and identically distributed (i.i.d.) samples and computing their empirical mean.

However, i.i.d. sampling is neither necessary nor optimal. Antithetic sampling methods (Hammersley & Morton, 1956) forgo the i.i.d. sampling construction. Given a fixed number of samples, antithetic sampling attempts to *jointly* select all of them at once so that they are more representative. An intuitive example is sampling without replacement: drawing k objects without replacement is guaranteed to provide a more complete picture of the underlying space (compared to i.i.d. sampling) because it avoids redundancy by construction. In fact, sampling without replacement provably reduces variance. The reduction, however, is modest.

Although a more effective, general-purpose antithetic sampling strategy is desirable, it is actually impossible. We will prove a no-free-lunch result: no antithetic sampling strategy can work better than sampling without replacement for a natural class of Monte Carlo estimation problems. Therefore, we cannot use a single antithetic sampling strategy and achieve significant variance reduction for every estimation problem; antithetic sampling must tailor to the specific problem at hand.

In this paper, we propose a general framework to learn an antithetic sampling distribution that automatically adapts to the underlying Monte Carlo estimation problem. The key idea is to construct a flexible, parametric family of joint probability distributions over a set of samples that is both learnable and guarantees the unbiasedness of the resulting estimator. We then propose learning methods that choose a joint distribution to minimize the variance of the estimator. We provide both a gradient-based learning scheme which

[*]Equal contribution [1]Stanford University. Correspondence to: Hongyu Ren <hyren@cs.stanford.edu>, Shengjia Zhao <sjzhao@stanford.edu>, Stefano Ermon <ermon@cs.stanford.edu>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

takes full advantage of recent advances in automatic differentiation, and a gradient free version when differentiation is not possible.

We demonstrate the effectiveness of our method on a diverse set of problems that benefit from low variance Monte Carlo estimation, including numerical computation of expectations, likelihood estimation for variational Bayesian inference, and training of generative adversarial networks by stochastic gradient descent. For all of these tasks, our method successfully reduces variance, which in turn leads to improved performance on standard metrics for each task. In addition, the computational overhead of our method is small. By amortizing the cost of training the antithetic sampler, we still achieve superior performance when baseline methods are given the same amount of wall-clock time.

2. Background

It is common in machine learning to evaluate expectations $\mu = \mathbb{E}_{p(x)}[f(x)]$ for some function f and distribution $p(x)$ on some sample space $x \in \mathcal{X}$. In addition, we often want to optimize the expectation $\mu(\phi) = \mathbb{E}_{p(x)}[f(x;\phi)]$ as a function of some parameters ϕ. To do so, first-order optimization methods such as gradient descent need to estimate the related expectation $\nabla\mu(\phi) = \mathbb{E}_{p(x)}[\nabla f(x;\phi)]$. Stochastic estimation and optimization problems of this form are ubiquitous in Bayesian inference, reinforcement learning, variational inference, risk minimization, etc (Bishop, 2006).

It is almost always too expensive to evaluate $\mu = \mathbb{E}_{p(x)}[f(x)]$ exactly (e.g., analytically). In this case, Monte Carlo estimation is widely used:

$$\mathbb{E}_{p(x)}[f(x)] \approx \frac{1}{m}\sum_{i=1}^{m} f(x_i) \stackrel{\text{def}}{=} \hat{\mu}_f(x_{1:m}) \tag{1}$$

where $x_{1:m} = x_1, \cdots, x_m \stackrel{\text{i.i.d}}{\sim} p(x)$ are m *independent*, identically distributed samples from $p(x)$. We will denote this i.i.d. distribution as $x_{1:m} \sim p(x_{1:m})$.

For any $m \geq 1$, $\hat{\mu}$ is an unbiased estimator of μ, meaning that $\mathbb{E}_{p(x_{1:m})}[\hat{\mu}_f(x_{1:m})] = \mu$. Chebychev's inequality provides (probabilistic) guarantees on how far $\hat{\mu}$ can be from μ:

$$\Pr\left[|\hat{\mu}_f(x_{1:m}) - \mu| \geq \epsilon\right] \leq \frac{\mathrm{Var}_{p(x_{1:m})}[\hat{\mu}_f(x_{1:m})]}{\epsilon^2} \tag{2}$$

Variance plays a paramount role in Monte Carlo estimation. The smaller the variance, the more concentrated $\hat{\mu}$ is around μ. For i.i.d. samples $x_{1:m} \sim p(x_{1:m})$

$$\mathrm{Var}_{p(x_{1:m})}[\hat{\mu}_f(x_{1:m})] = \frac{\mathrm{Var}_{p(x)}[f(x)]}{m}$$

so we can increase m to reduce variance. However this is usually expensive, as the computational cost increases (linearly) in m. It is therefore useful to design methods that reduce variance without increasing m.

2.1. Variance Reduction Techniques

Given the importance of Monte Carlo estimation and variance reduction, a significant number of techniques have been applied to diverse problems (L'Ecuyer & Owen, 2009). Example include stratified sampling (Neyman, 1934), control variate / baselines (Greensmith et al., 2004), importance sampling (Neal, 2001), rejection sampling (Grover et al., 2018), Rao-Blackwellization (Grisetti et al., 2007), etc.

In this paper, we build on antithetic sampling (Geweke, 1988; Wu et al., 2019), a classic variance reduction technique that has received less attention in the ML literature.

2.2. Antithetic Sampling

In the classic Monte Carlo estimator $\hat{\mu}_f(x_{1:m}) = 1/m\sum_{i=1}^{m} f(x_i)$, the samples $x_1, \cdots, x_m$ are sampled *independently* from $p(x)$. However, i.i.d. sampling is not necessary nor optimal. In fact, whenever they are sampled from a joint distribution $q(x_{1:m})$ that preserves the marginals, i.e. it satisfies $q(x_i) = p(x_i), \forall i = 1, \cdots, m$, our Monte Carlo estimator $\hat{\mu}_f(x_{1:m})$ is unbiased because

$$\mathbb{E}_{q(x_{1:m})}\left[\frac{1}{m}\sum_i f(x_i)\right] = \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_{q(x_i)}[f(x_i)] \tag{3}$$

$$= \mathbb{E}_{p(x)}[f(x)] \tag{4}$$

The core idea of antithetic sampling is to forgo i.i.d. sampling and use a joint distribution $q(x_{1:m})$ such that

$$\mathrm{Var}_{q(x_{1:m})}[\hat{\mu}_f(x_{1:m})] < \mathrm{Var}_{p(x_{1:m})}[\hat{\mu}_f(x_{1:m})]$$

For example, when $p(x)$ is symmetric (i.e. $p(x) = p(-x)$), such as a zero-mean Gaussian, a classical choice for $q(x_1, x_2)$ is given by the following process: sample $x_1 \sim p(x)$, then set $x_2 = -x_1$ (hence the name antithetic). It is easy to see that $q(x_1) = q(x_2) = p(x_1)$. This strategy works perfectly if f is an odd function, i.e. $f(-x) = -f(x)$, because we are guaranteed to have

$$\frac{1}{2}(f(x_1) + f(x_2)) = 0 = \mathbb{E}_{p(x)}[f(x)]$$

Only two samples are required to estimate the expectation exactly. On the other hand, if f is an even function $f(-x) = f(x)$, then this strategy backfires because $f(x_1) = f(x_2)$, so x_2 is redundant and antithetic sampling doubles the variance compared to i.i.d. sampling.

As the previous example indicates, the variance of our estimator under an antithetic distribution $q(x_{1:m})$ depends on

the function f. The natural question is: is there an antithetic sampling distribution that performs well for any f? We will now prove that the answer is yes and no: there are antithetic sampling distributions that always reduces variance, however, they only reduce variance by a very small amount in the worst-case.

2.3. No-Free-Lunch of Antithetics

We first suppose $x \in \mathcal{X}$ only takes a finite set of values. We define $q_{\text{sr}}(x_{1:m})$ as the following sampling without replacement distribution: draw $x_1 \sim p(x)$, and draw

$$x_2 \propto p(x)\mathbb{I}(x \neq x_1)$$
$$x_3 \propto p(x)\mathbb{I}(x \neq x_1)\mathbb{I}(x \neq x_2)$$
$$\cdots$$

Since $\forall i$, $q_{\text{sr}}(x_i) = p(x)$, sampling without replacement is an antithetic sampling method. In addition, it is guaranteed to improve variance (Sukhatme, 1957) compared to i.i.d. sampling.

Sampling without replacement always reduces variance, but only by a tiny amount. It is only an effective strategy if the probability of sampling repeated elements is large, which rarely happens in practice. In fact, no antithetic distribution can do much better, as shown by the following theorem.

Theorem 1 (No Free Lunch)**.** *Let $p(x)$ be a uniform distribution on $\mathcal{X}$, where $\mathcal{X}$ is a finite set. Let q be any antithetic distribution $q(x_1, x_2)$ where $x_1, x_2 \in \mathcal{X}$. Let $\mathcal{F}$ be the set of functions $\mathcal{X} \to \mathbb{R}$ such that $\text{Var}_{p(x_{1:2})}[\hat{\mu}_f(x_{1:2})] \neq 0$, then*

$$\max_{f \in \mathcal{F}} \frac{\text{Var}_{q(x_{1:2})}[\hat{\mu}_f(x_{1:2})]]}{\text{Var}_{p(x_{1:2})}[\hat{\mu}_f(x_{1:2})]} \geq 1 - \frac{1}{|\mathcal{X}| - 1} \tag{5}$$

For sampling without replacement, for any $f \in \mathcal{F}$

$$\frac{\text{Var}_{q_{\text{sr}}(x_{1:2})}[\hat{\mu}_f(x_{1:2})]]}{\text{Var}_{p(x_{1:2})}[\hat{\mu}_f(x_{1:2})]} = 1 - \frac{1}{|\mathcal{X}| - 1} \tag{6}$$

Theorem 1 proves that given *any* antithetic sampling distribution, there exists an estimation problem (a distribution and a worst case function f), such that it only improves variance by a tiny amount compared to i.i.d. sampling. In fact, for the uniform distribution $p(x)$, sampling without replacement is the mini-max optimal antithetic distribution (i.e., it is optimal for the worst-case function f). Therefore, the antithetic sampling method must be adapted to the function f. Manually designing good antithetic distributions requires detailed knowledge of f and $p(x)$, which is almost always not available when f is complex (e.g., the gradient of a deep neural network). An adaptive method is therefore necessary.

3. Gaussian-reparameterized Antithetics

To develop an *adaptive antithetic sampler*, our strategy is to define a family of antithetic samplers $q_\theta(x_{1:m})$, and choose the parameters θ to adapt to the specific distribution $p(x)$ and integrand f. We will now focus on the case where $\mathcal{X}$ is continuous.

The first roadblock is that for antithetic sampling to be unbiased, we need the marginals of $q_\theta(x_{1:m})$ to satisfy

$$q_\theta(x_i) = p(x_i), \quad \forall i = 1, \cdots, m \tag{7}$$

We now show that for a large set of distributions $p(x)$ it is possible to define a flexible family of parameterized distributions q_θ that satisfy this property by construction.

Suppose x is univariate with density $p(x)$. Let $F(x)$ denote its cumulative density function (CDF), sampling $x \sim p(x)$ can be achieved by "inverting" the CDF via $u \sim \mathcal{U}[0, 1]$, $x = F^{-1}(u)$. Let Φ denote the CDF of the standard Gaussian distribution $\mathcal{N}(0, 1)$. By the same argument, it is easy to see that we can alternatively sample from $x \sim p(x)$ via the following process

$$\epsilon \sim \mathcal{N}(0, 1) \ , \ x = F^{-1}(\Phi(\epsilon)) := g(\epsilon) \tag{8}$$

where $g = F^{-1} \circ \Phi$. Therefore an expectation with respect to $p(x)$ can be converted into an expectation with respect to a standard Normal $\epsilon \sim \mathcal{N}(\epsilon; 0, 1)$ by the law of the unconscious statistician:

$$\mathbb{E}_{p(x)}[f(x)] = \mathbb{E}_{\mathcal{N}(\epsilon;0,1)}[f(g(\epsilon))]$$

More generally, when $p(\boldsymbol{x})$ is a multivariate density, we will assume that $\boldsymbol{x} \sim p(\boldsymbol{x})$ can be equivalently obtained as $\boldsymbol{x} = g(\boldsymbol{\epsilon})$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I_d)$, where I_d denotes an $d \times d$ identity matrix and g is a suitable function. This means that is possible to sample from $p(\boldsymbol{x})$ by transforming some simple random variables $\boldsymbol{\epsilon}$, e.g., obtained using random number generation primitives in a programming language. This is a very mild restriction on $p(\boldsymbol{x})$. Although the theory is general, in the experiments we will consider the most practically relevant case where $g(\cdot)$ can also be evaluated efficiently.

We can define a family of antithetic samplers that satisfy the marginalization property of Eq.(7) as follows.

Definition 1. *Let $\boldsymbol{x}$ be a continuous random vector with density $p(\boldsymbol{x})$ that can be sampled by $\boldsymbol{x} = g(\boldsymbol{\epsilon})$ for $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I_d)$. A Gaussian antithetic of order m for $p(\boldsymbol{x})$ is the family of distributions $q_\theta(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)$ defined implicitly by the following sampling procedure*

$$(\boldsymbol{\epsilon}_1, \cdots, \boldsymbol{\epsilon}_m) \sim \mathcal{N}(0, \Sigma_\theta)$$
$$\boldsymbol{x}_i = g(\boldsymbol{\epsilon}_i), \quad i = 1, \cdots, m$$

where $\Sigma_\theta \in \mathbb{R}^{md\times md}$ is positive-definite matrix parameterized by θ that is constrained to have $d\times d$ identity blocks on the diagonal:

$$\Sigma_\theta \in \mathbf{\Sigma}_{\text{unbiased}} \overset{\text{def}}{=} \{\Sigma \in \mathbb{R}^{md\times md}, \Sigma \succ 0, \Sigma_{\mathcal{I}\mathcal{I}} = I_d \quad \forall \mathcal{I} = (id+1, \cdots, id+d), i = 1, \cdots, m\} \tag{9}$$

When $x \sim p(x)$ is obtained by inverting CDFs as in equation (8), the family resembles a Gaussian copula (Durante & Sempi, 2010), a classic approach to build joint distributions with known, fixed marginals.

Our estimator for $\mu = \mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})]$ remains unchanged compared to Eq.(1), except we use antithetic (not i.i.d.) samples $\boldsymbol{x}_{1:m} \sim q_\theta(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)$

$$\hat{\mu}_f(\boldsymbol{x}_{1:m}) = \frac{1}{m}\sum_{i=1}^m f(\boldsymbol{x}_i) \tag{10}$$

$$= \frac{1}{m}\sum_{i=1}^m f\circ g(\boldsymbol{\epsilon}_i) := \hat{\mu}_{f\circ g}(\boldsymbol{\epsilon}_{1:m}) \tag{11}$$

A Gaussian-reparameterized antithetic distribution satisfies several desirable properties.

Proposition 1. *Let $q_\theta(\boldsymbol{x}_{1:m})$ be a Gaussian-reparameterized antithetic of order m for $p(\boldsymbol{x})$. Then for any k:*

1. *For any $\Sigma_\theta \in \mathbf{\Sigma}_{\text{unbiased}}$, the estimator (10) is unbiased*

$$\mathbb{E}_{q_\theta(\boldsymbol{x}_{1:m})}[\hat{\mu}_f(\boldsymbol{x}_{1:m})] = \mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})]$$

2. *If $\Sigma_\theta = I_{md}$, the Gaussian-reparameterized antithetic is equivalent to i.i.d sampling.*

3. *Given a Cholesky decomposition $\Sigma_\theta = L_\theta L_\theta^T$, we can sample from $q_\theta(\boldsymbol{x}_{1:m})$ by drawing m i.i.d. samples $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_m)^T$ from $\mathcal{N}(0, I_d)$, and $\boldsymbol{x}_{1:m} = L_\theta\boldsymbol{\delta}$.*

Proof. See Appendix □

Property 1 guarantees that for any choice of θ the corresponding estimator is unbiased, i.e., it will give the right answer in expectation. Property 2 guarantees that under mild conditions, there is a choice of θ where the performance is no worse than i.i.d. sampling. Property 3 states that drawing samples from $q_\theta(\boldsymbol{x}_{1:m})$ is not expensive. Given a $\Sigma_\theta \in \mathbb{R}^{md\times md}$, we only have to compute the Cholesky decomposition once, whose time complexity is $\mathcal{O}(m^3d^3)$. If we draw k batches of samples, the time complexity becomes $\mathcal{O}(m^3d^3 + km^2d^2)$. The cost can thus be amortized and is comparable to i.i.d sampling when $k \gg md$.

3.1. Parameterizing Gaussian Antithetics

To facilitate the design of optimization algorithms over $\Sigma_\theta \in \mathbf{\Sigma}_{\text{unbiased}}$, we provide an explicit parameterization of the feasible set $\mathbf{\Sigma}_{\text{unbiased}}$ in Definition 1. To simplify the notation, we will use a slightly more abstract definition than in the previous section. Specifically, we will consider the matrix $\Sigma \in \mathbb{R}^{md\times md}$ as a $m \times m$ matrix where each element belongs to the ring of $d \times d$ matrices.

Let $\mathbb{M}$ be the ring of $d \times d$ matrices, and e be its identity element. We rewrite the definition of $\mathbf{\Sigma}_{\text{unbiased}}$ in Eq.(9) as:

$$\mathbf{\Sigma}_{\text{unbiased}} \overset{\text{def}}{=} \{\Sigma \in \mathbb{M}^{m\times m}, \Sigma \succ 0, \Sigma_{ii} = e, \forall i\} \tag{12}$$

To find a parameterization for $\mathbf{\Sigma}_{\text{unbiased}}$ we let $\epsilon > 0$ be any real number. Given any $\theta \in \mathbb{M}^{m\times m}$, we turn it into an element of $\mathbf{\Sigma}_{\text{unbiased}}$ with the function ψ defined by

$$\tilde{\Sigma} = \epsilon I + \theta\theta^T, \quad \theta \in \mathbb{M}^{m\times m}$$
$$\psi(\theta) = \text{diag}(\tilde{\Sigma})^{-1/2}\tilde{\Sigma}\text{diag}(\tilde{\Sigma})^{-T/2} \tag{13}$$

where $\text{diag}(\tilde{\Sigma})$ sets all entries of $\tilde{\Sigma}$ to zero except the diagonal, and $A^{-1/2}$ is obtained from the Cholesky decomposition $A = A^{1/2}A^{T/2}$ for a positive semidefinite matrix A.

The following proposition shows this parameterization is both correct (i.e., every parameter $\theta \in \mathbb{M}^{m\times m}$ maps into $\mathbf{\Sigma}_{\text{unbiased}}$) and lossless (i.e., every $\Sigma \in \mathbf{\Sigma}_{\text{unbiased}}$ can be obtained from a $\theta \in \mathbb{M}^{m\times m}$).

Theorem 2. *For any $\epsilon > 0$, the map ψ defined in Eq.(13) is a surjection from $\mathbb{M}^{m\times m}$ into $\mathbf{\Sigma}_{\text{unbiased}}$.*

Therefore we can optimize θ over the space of $m \times m$ matrices $\mathbb{M}^{m\times m}$, and use $\psi(\theta)$ to produce the corresponding $\Sigma_\theta \in \mathbf{\Sigma}_{\text{unbiased}}$ that we need for our antithetic distribution.

4. Learning an Antithetic Distribution

In the previous section we defined a family of antithetic distributions $\mathcal{N}(0, \Sigma)$ where $\Sigma_\theta \in \mathbf{\Sigma}_{\text{unbiased}}$. Proposition (1) guarantees that it can perform as well as i.i.d., but the question is of course if we can do *better*. In this section, we discuss how to find the optimal $\theta \in \mathbb{M}^{m\times m}$ (equivalently, find the best $\Sigma \in \mathbf{\Sigma}_{\text{unbiased}}$) by optimization, leveraging the parameterization of $\mathbf{\Sigma}_{\text{unbiased}}$ via ψ from Theorem 2.

Given that the antithetic estimator is unbiased, a natural optimization criterion is for our estimator $\hat{\mu}_f(\boldsymbol{x}_{1:m})$ to have small variance:

$$\min_{\theta\in\mathbb{M}^{m\times m}} \text{Var}_{q_\theta(\boldsymbol{x}_{1:m})}\left[\hat{\mu}_f(\boldsymbol{x}_{1:m})\right]$$
$$= \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\psi(\theta))}\left[(\hat{\mu}_{f\circ g}(\boldsymbol{\epsilon}_{1:m}) - \mu)^2\right]$$

where the equality follows from the definition $\boldsymbol{x}_i = g(\boldsymbol{\epsilon}_i)$. When f is vector-valued (e.g., a gradient) we can minimize the trace of the covariance

$$\min_{\theta \in \mathbb{M}^{m\times m}} \operatorname{tr}\left(\operatorname{Cov}_{q_\theta(\boldsymbol{x}_{1:m})}\left[\hat{\mu}_f(\boldsymbol{x}_{1:m})\right]\right) = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\psi(\theta))}\left[\left\|\hat{\mu}_{f\circ g}(\boldsymbol{\epsilon}_{1:m}) - \mu\right\|^2\right] \tag{14}$$

Eq.(14) involves the value of μ, which is generally not known. An estimate of μ is helpful but not required. This is because by Proposition (1), μ does not depend on θ, so we can rewrite (14) as

$$\min_{\theta \in \mathbb{M}^{m\times m}} \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\psi(\theta))}\left[\left\|\hat{\mu}_{f\circ g}(\boldsymbol{\epsilon}_{1:m})\right\|^2\right] - \|\mu\|^2$$

μ is thus a additive constant with no effect on the minimization of Eq.(14). We can, in fact, choose any constant in place of μ. Nonetheless, μ (or a good approximation to μ) is an effective control variate (Greensmith et al., 2004) and reduces the variance when estimating Eq.(14) from samples.

We can use a bootstrap method to estimate μ: we sample multiple antithetic batches $\boldsymbol{x}_{1:m}{}^{(1)}, \cdots, \boldsymbol{x}_{1:m}{}^{(k)} \sim q_\theta(\boldsymbol{x}_{1:m})$; for each antithetic batch we can compute $\hat{\mu}_f(\boldsymbol{x}_{1:m}{}^{(i)})$. We can average over these estimate

$$\tilde{\mu} = \frac{1}{k}\sum_i \hat{\mu}_f(\boldsymbol{x}_{1:m}{}^{(i)}) \tag{15}$$

as an approximation to μ. Instead of minimizing $\|\hat{\mu}_f(\boldsymbol{x}_{1:m}) - \mu\|^2$, we minimize $\|\hat{\mu}_f(\boldsymbol{x}_{1:m}) - \tilde{\mu}\|$. Intuitively, this choice is also appealing because we are encouraging the empirical average for a batch of size m to approach the average of a larger batch of size mk.

Notation: For convenience of notation we abbreviate $\|\hat{\mu}_{f\circ g}(\boldsymbol{\epsilon}_{1:m}) - \mu\|^2$ as $\phi(\boldsymbol{\epsilon}_{1:m})$. We make its dependence on f, g and μ (which is replaced with $\tilde{\mu}$) implicit. Our goal is to minimize $\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\psi(\theta))}\left[\phi(\boldsymbol{\epsilon}_{1:m})\right]$.

There are several methods to minimize Eq.(14). In particular, we use different methods depending on whether $f \circ g$ is differentiable with respect to $\boldsymbol{\epsilon}$. If it is differentiable, then $\phi(\boldsymbol{\epsilon}_{1:m})$ will also be differentiable with respect to each $\boldsymbol{\epsilon}_i$, and we can use gradient based optimization. Otherwise we will have to use gradient free optimizers.

4.1. Gradient Based Variance Minimization

Suppose $\nabla_{\boldsymbol{\epsilon}} f \circ g(\boldsymbol{\epsilon})$ can be computed, then we can also compute the gradient of $\nabla_{\boldsymbol{\epsilon}_i}\phi(\boldsymbol{\epsilon}_{1:m})$. Then we can minimize Eq.(14) by reparameterization (Kingma & Welling, 2013). For any parameter $\theta \in \mathbb{M}^{m\times m}$, because $\psi(\theta)$ is positive definite, we can compute its Cholesky decomposition (proved in Appendix) as $L(\theta) = \psi(\theta)^{1/2}$. Then for $\boldsymbol{\delta} \sim \mathcal{N}(0, I)$, $L(\theta)\boldsymbol{\delta}$ will be distributed as $\mathcal{N}(0, \psi(\theta))$. Therefore we can rewrite our optimization objective Eq.(14) as

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\delta};0,I)}[\phi(L(\theta)\boldsymbol{\delta})] \tag{16}$$

which we can minimize with stochastic gradient descent.

$$\nabla_\theta \mathbb{E}_{\mathcal{N}(\boldsymbol{\delta};0,I)}[\phi(L(\theta)\boldsymbol{\delta})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\delta};0,I)}[\nabla_\theta \phi(L(\theta)\boldsymbol{\delta})]$$

4.2. Gradient Free Variance Optimization

When the gradient $\nabla_{\boldsymbol{\epsilon}} f \circ g(\boldsymbol{\epsilon})$ is not available (e.g., a black box function or g is not differentiable), we can use the reinforce estimator.

$$\nabla_\theta \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\Sigma_\theta)}[\phi(\boldsymbol{\epsilon}_{1:m})] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}_{1:m};0,\Sigma_\theta)}\left[\phi(\boldsymbol{\epsilon}_{1:m})\nabla_\theta \log \mathcal{N}(\boldsymbol{\epsilon}_{1:m}; 0, \Sigma_\theta)\right] \tag{17}$$

If computing $f \circ g$ is expensive, then we can use the importance sampled reinforce estimator. Let $r(\boldsymbol{\epsilon}_{1:m})$ be any distribution such that $\phi(\boldsymbol{\epsilon}_{1:m})\mathcal{N}(\boldsymbol{\epsilon}_{1:m}; 0, \Sigma_\theta)/r(\boldsymbol{\epsilon}_{1:m})$ is finite, then the gradient of Eq.(17) is

$$\mathbb{E}_{r(\boldsymbol{\epsilon}_{1:m})}\left[\frac{\phi(\boldsymbol{\epsilon}_{1:m})\mathcal{N}(\boldsymbol{\epsilon}_{1:m}; 0, \Sigma_\theta)}{r(\boldsymbol{\epsilon}_{1:m})}\nabla_\theta \log \mathcal{N}(\boldsymbol{\epsilon}_{1:m}; 0, \Sigma_\theta)\right] \tag{18}$$

We can draw a set of samples $\boldsymbol{\epsilon}_{1:m}{}^{(i)}$ from $r(\boldsymbol{\epsilon}_{1:m})$ and store the values $\phi(\boldsymbol{\epsilon}_{1:m}{}^{(i)})$ for each i. We can use this same set of samples to repeated compute the gradient Eq.(18) and reuse the $\phi(\boldsymbol{\epsilon}_{1:m}{}^{(i)})$ we computed.

5. Reducing Computation Cost for Adaptive Antithetic

Even though we can find a good antithetic distribution by optimizing Eq.(14), this is itself a stochastic optimization problem that can be expensive. The time invested in finding a good q_θ needs to payoff in terms of reduced number of samples required. In this section we propose several strategies to reduce the computational overhead of adaptive antithetic sampling in practice. Combining these approaches, we are able to achieve superior wall-clock time in our experiments compared with i.i.d. and other baselines despite of the overhead of training q_θ.

5.1. Reduction of Parameter Count

Our parameterization of the feasible space by ψ has several desirable properties as shown in Theorem 2. However, $\theta \in \mathbb{M}^{m\times m}$ (or equivalently $\mathbb{R}^{md\times md}$) has m^2d^2 learnable parameters, and becomes infeasible to use for large m or d.

To reduce m, we do not perform antithetic sampling on the entire batch. Instead we define antithetic distributions

over micro-batches of size k, and i.i.d. sample m/k micro-batches as follows:

$$\underbrace{\boldsymbol{x}_{1:k}}_{\text{micro batch 1}}, \underbrace{\boldsymbol{x}_{k+1:2k}}_{\text{micro batch 2}}, \cdots \underbrace{\boldsymbol{x}_{m-k+1:m}}_{\text{micro batch m/k}} \overset{\text{i.i.d}}{\sim} q_\theta(\boldsymbol{x}_{1:k}) \quad (19)$$

Then because we only have to parameterize k dimensional Gaussian antithetic distributions, $\theta \in \mathbb{M}^{k\times k}$. This reduces number of parameters to k^2d^2.

We can further reduce the number of trainable parameters by choosing $\theta \in \mathbb{M}^{k\times k'}$ for some $k' < k$. Even for very small k', the model still has many free parameters because each element belongs to the matrix ring $\mathbb{M}$. In experiments we observe that we can even find good Σ with $k' = 2$. For most of our experiments on high dimensional problems, we choose $k = 8$, and $\theta \in \mathbb{M}^{k\times 2}$. The number of parameters will be $2kd^2$.

To reduce d, we note that Theorem 2 (Appendix A) is still true if instead of using the ring $\mathbb{M}$ in Eq.(12,13), we use the subring of block diagonal matrices $\bar{\mathbb{M}}$. That is, given any partition of $\{1, \cdots, d\}$, we use the ring consisting of matrices of the following form

$$A \in \mathbb{R}^{d\times d}, A_{ij} = 0 \text{ if } i, j \text{ belong to different partitions}$$

We show in the appendix that ψ is still a surjection into $\boldsymbol{\Sigma}_{\text{unbiased}}$ under this new ring. The previous strategies are sufficient for our experiments, but this strategy could be necessary for even higher dimensional problems.

5.2. Amortization of the Learning Cost

Monte Carlo estimates are widely used in stochastic optimization frameworks. In these applications we are not estimating $\mathbb{E}_{p(\boldsymbol{x})}[f(\boldsymbol{x})]$ for a single f, rather we have a *sequence* of related estimation problems

$$\mathbb{E}_{p(\boldsymbol{x})}[f_1(\boldsymbol{x})], \cdots, \mathbb{E}_{p(\boldsymbol{x})}[f_T(\boldsymbol{x})]$$

For example, for stochastic gradient descent, $f_t(\boldsymbol{x})$ is the gradient at the t-th iteration. Learning the antithetic distribution for every f_t is clearly infeasible, however, assuming f_t and f_{t+1} are similar, any q_θ that achieves small variance on the estimation of $\mathbb{E}_{p(\boldsymbol{x})}[f_t(\boldsymbol{x})]$ is also likely to achieve small variance on the estimation of $\mathbb{E}_{p(\boldsymbol{x})}[f_{t+1}]$. Then what we can do is to choose a subset of time steps $\{t_1, \cdots\} \subset \{1, \cdots, T\}$ and update q_θ by Eq.(14) only on these time steps. The cost of each update can be amortized over $[t_i, t_{i+1}]$. In particular, if updates are exponentially sparse (e.g., an update every $t_i = 2^i$ steps), then the overhead of finding q^* is almost negligible. We show in our experiments that this leads to very good performance on deep learning tasks trained with stochastic gradient descent.

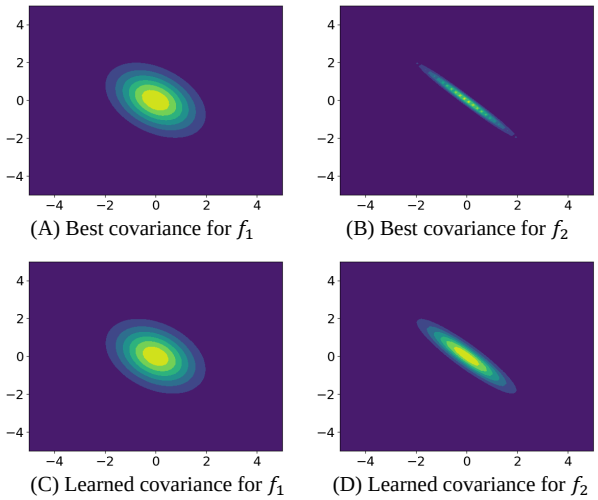
6. Experiments

We test our adaptive Gaussian antithetic on three tasks. The first one is a controlled synthetic task, where we verify that our model demonstrates expected behavior. The second task is Bayesian inference, where we reduce variance and achieve better estimation of the posterior. The third task is improving stochastic gradient descent training of generative adversarial networks, where we reduce variance and obtain quantitative improvements in terms of inception scores for the same wall-clock time.

6.1. Simple Estimation Problems

In this section we estimate several simple functions to visualize the behavior of our model. In both problems we use a one dimensional $\mathcal{X}$ and a one dimensional function $f : \mathcal{X} \to \mathbb{R}$. We choose $p(x)$ as $\mathcal{N}(0, 1)$, and antithetic batch size $m = 2$. $q_\theta(x_1, x_2)$ is a two dimensional Gaussian we can easily visualize by contour plots.

The two functions we choose are $f_1 = e^x + 2x\sin(x)$ and $f_2 = x^3$. In Figure 1(A, B) we plot the optimal antithetic distribution $q^*(x_1, x_2)$ that minimizes variance. This distribution is found by grid search over all possible parameters (up to precision 10^{-2}). For f_1 the antithetic distribution found by our model is identical to the optimal antithetic distribution (found by grid search). For f_2, the optimal sampling distribution has singular covariance: $x_1 = -x_2$ with probability 1 under $q^*(x_1, x_2)$. Eq.(13) only parameterizes Gaussians with full rank covariances, so we cannot exactly represent $q^*(x_1, x_2)$. Nonetheless our method learns a good approximation to q^*.



(A) Best covariance for f_1 (B) Best covariance for f_2

(C) Learned covariance for f_1 (D) Learned covariance for f_2

Figure 1: **Top:** Best covariance for optimal antithetic Gaussian $q^*(x_1, x_2)$ found by grid search. **Bottom:** Covariance found by our adaptive learning method.

6.2. Application to Variational Bayes

In Bayesian estimation problems we often have an unobserved variable $\boldsymbol{z} \in \mathcal{Z}$, and an observed variable $\boldsymbol{x} \in \mathcal{X}$. Given a likelihood $p(\boldsymbol{x}|\boldsymbol{z})$ and a prior $p(\boldsymbol{z})$, the objective is to either estimate $\log p(\boldsymbol{x})$ or the posterior $p(\boldsymbol{z}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x})}$. For both applications, the major challenge is to compute $p(\boldsymbol{x})$ or $\log p(\boldsymbol{x})$.

A typical way to estimate $\log p(\boldsymbol{x})$ is to use importance sampling. Let $q(\boldsymbol{z})$ be any distribution on $\mathcal{Z}$. We can obtain the evidence lower bound (ELBO) to the log likelihood

$$\log p(\boldsymbol{x}) = \log \mathbb{E}_{p(\boldsymbol{z})}[p(\boldsymbol{x}|\boldsymbol{z})] = \log \mathbb{E}_{q(\boldsymbol{z})}\left[\frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right]$$
$$\geq \mathbb{E}_{q(\boldsymbol{z})}\left[\log \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right] := \mathcal{L}(\boldsymbol{x})$$

The bound is exact if $q(\boldsymbol{z}) = p(\boldsymbol{z}|\boldsymbol{x})$, so we can maximize $\mathcal{L}(\boldsymbol{x})$ over a set of candidate importance sampling distributions $q(\boldsymbol{z})$ to obtain the tightest bound. This procedure is usually known as stochastic variational inference (SVI) (Hoffman et al., 2013). However, $q(\boldsymbol{z})$ is usually a simple distribution for computational reasons, so $q(\boldsymbol{z})$ seldom approximates $p(\boldsymbol{z}|\boldsymbol{x})$ well. As a result, $\mathcal{L}(\boldsymbol{x})$ is usually a loose bound.

To address this limitation, (Burda et al., 2015) proposed using multi-variable importance sampling

$$\mathcal{L}^{is}(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}_1)\cdots q(\boldsymbol{z}_m)}\left[\log \frac{1}{m}\sum_i \frac{p(\boldsymbol{x},\boldsymbol{z}_i)}{q(\boldsymbol{z}_i)}\right] \quad (20)$$

where it is guaranteed that $\mathcal{L}(\boldsymbol{x}) \leq \mathcal{L}^{is}(\boldsymbol{x}) \leq \log p(\boldsymbol{x})$. In fact we do not have to use a factorized distribution $q(\boldsymbol{z}_1)\cdots, q(\boldsymbol{z}_m)$, which would correspond to i.i.d. sampling. Instead, we can use a correlated joint distribution $q_\theta(\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m)$, as long as it has the right marginals.

$$\mathcal{L}^{\text{anti}}(\boldsymbol{x}) = \mathbb{E}_{q_\theta(\boldsymbol{z}_1,\cdots,\boldsymbol{z}_m)}\left[\log \frac{1}{m}\sum_i \frac{p(\boldsymbol{x},\boldsymbol{z}_i)}{q(\boldsymbol{z}_i)}\right] \quad (21)$$

$\mathcal{L}^{anti}(\boldsymbol{x})$ still lower bounds $\log p(\boldsymbol{x})$ as long as q_θ is an antithetic distribution for $q(\boldsymbol{z})$ because

$$\mathcal{L}^{\text{anti}}(\boldsymbol{x}) \leq \log \mathbb{E}_{q_\theta(\boldsymbol{z}_1,\cdots,\boldsymbol{z}_m)}\left[\frac{1}{m}\sum_i \frac{p(\boldsymbol{x},\boldsymbol{z}_i)}{q(\boldsymbol{z}_i)}\right]$$
$$= \log \frac{1}{m}\sum_i \mathbb{E}_{q(\boldsymbol{z})}\left[\frac{p(\boldsymbol{x},\boldsymbol{z})}{q(\boldsymbol{z})}\right] = \log p(\boldsymbol{x})$$

Therefore we can optimize over Gaussian-reparameterized antithetic distributions q_θ to maximize $\mathcal{L}^{\text{anti}}(\boldsymbol{x})$ because it is a lower bound to $\log p(\boldsymbol{x})$ for any choice of q_θ.

To find a good antithetic distribution q_θ we train on a small set of $\boldsymbol{x}$, and apply the learned q_θ to previously unseen inputs $\boldsymbol{x}$. We find that q_θ generalizes well across different $\boldsymbol{x}$. Therefore, the cost of learning q_θ is negligible when it is amortized over multiple estimation problems for $\log p(\boldsymbol{x})$.

6.2.1. Dataset and Evaluation

We first train a variational autoencoder on MNIST and Omniglot for the pretrained model $q(\boldsymbol{z}|\boldsymbol{x}), p(\boldsymbol{x}|\boldsymbol{z})$. We evaluate whether our antithetic sampler can achieve a tighter bound $\mathcal{L}^{\text{anti}}$ on $\log p(\boldsymbol{x})$, where $\boldsymbol{x}$ are images in the test set. As our baselines, we use i.i.d. sampling to achieve lower bound $\mathcal{L}^{is}(\boldsymbol{x})$ as in Eq.(20). We also compared with negative sampling introduced in section 2.2.

Since the objective $\mathcal{L}^{\text{anti}}$ is a function of $\boldsymbol{x}$, we also evaluate the generalization of our model by training the antithetic sampler on a small set (100) of $\boldsymbol{x}$ but evaluate on a large number (1000) of samples, such that both the variance of this estimation and the additional computation overhead is negligible.

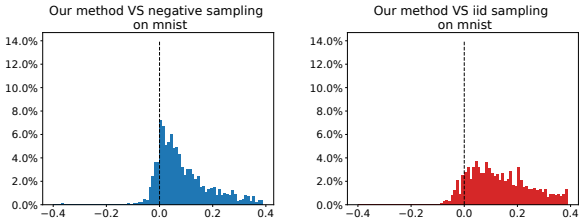


Figure 2: For each sample $\boldsymbol{x}$ in the test set, we compute the difference between our lower bound $\mathcal{L}^{\text{anti}}$ and lower bound obtained by baseline methods. A positive difference indicates that our method is better. **Left:** Histogram of the difference between our method and negative antithetic. **Right:** Histogram of the difference between our method and i.i.d. sampling $\mathcal{L}^{\text{is}}$. Negative bins only account for 5% of all data. This means that for almost all samples $\boldsymbol{x}$ in the test set, our method obtains a larger (tighter) lower bound to $\log p(\boldsymbol{x})$ compared to both baselines.

6.2.2. Results

Our results for MNIST are shown in Figure 2. The quantitative results for MNIST and Omniglot are shown in the Appendix. We consistently obtain larger (better) lower bound compared to baseline methods. In particular, our method shows strong generalization ability, although it is trained on a small number of $\boldsymbol{x}$, we still achieve good performance (large $\mathcal{L}^{\text{anti}}(\boldsymbol{x})$) for almost every $\boldsymbol{x}$ in the test set. If we amortize the cost of learning q_θ over approximately 1000 estimation tasks, our improvement comes with negligible overhead.

6.3. Application to GAN Training

Let $\mathcal{X} \subset \mathbb{R}^k$ be the space of images with k pixels, and $\mathcal{Z} \subset \mathbb{R}^d$ be the space of latent vectors. We are given an

empirical distribution (i.e. set of images) $p_{\text{data}}(\boldsymbol{x})$ on $\mathcal{X}$, and a simple distribution $p(\boldsymbol{z})$ on $\mathcal{Z}$ (e.g. Gaussian). Generative Adversarial Net (GAN) (Goodfellow et al., 2014) and its variants (Radford et al., 2015; Arjovsky et al., 2017; Gulrajani et al., 2017; Mao et al., 2017) has two sets of functions: a set of generator functions $\{G : \mathcal{Z} \to \mathcal{X}\}$ that "generates" images by mapping latent vectors in $\mathcal{Z}$ into images $\mathcal{X}$, and a set of discriminator functions $\{D : \mathcal{X} \to \mathbb{R}\}$ that maps an image $\mathcal{X}$ to a real number $\mathbb{R}$.

Training a generative adversarial net consists of two objectives: find a generator that generates images that are indistinguishable to the discriminator, and find a discriminator that better distinguishes generator images $G(\boldsymbol{z}), \boldsymbol{z} \sim p(\boldsymbol{z})$, and input images $p_{\text{data}}(\boldsymbol{x})$. A typical training objective is

$$\min_G \max_D \mathcal{L}(G, D) = \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[D(\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[D(G(\boldsymbol{z}))]$$

To optimize this objective the typical approach is by joint stochastic gradient descent.

$$\begin{aligned}
\nabla_G \mathcal{L}(G, D) &= -\mathbb{E}_{p(\boldsymbol{z})}[\nabla_G D(G(\boldsymbol{z}))] \\
\nabla_D \mathcal{L}(G, D) &= \mathbb{E}_{p_{\text{data}}(\boldsymbol{x})}[\nabla_D D(\boldsymbol{x})] - \mathbb{E}_{p(\boldsymbol{z})}[\nabla_D D(G(\boldsymbol{z}))] \\
G_{\text{new}} &= G_{\text{old}} - \nabla_G \mathcal{L}(G_{\text{old}}, D_{\text{old}}) \\
D_{\text{new}} &= D_{\text{old}} - \nabla_D \mathcal{L}(G_{\text{old}}, D_{\text{old}})
\end{aligned}$$

However computation of both $\nabla_G \mathcal{L}(G, D)$ and $\nabla_D \mathcal{L}(G, D)$ require Monte Carlo estimation of expectations. Usually this is achieved by i.i.d. samples of $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m \sim_{\text{i.i.d.}} p(\boldsymbol{z})$, and samples (that may not be i.i.d.) $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m \sim p_{\text{data}}(\boldsymbol{x})$.

$$\begin{aligned}
\nabla_G \mathcal{L}(G, D) &\approx -\frac{1}{m} \sum_i \nabla_G D(G(\boldsymbol{z}_i)) \\
\nabla_D \mathcal{L}(G, D) &\approx \frac{1}{m} \sum_i \nabla_D D(\boldsymbol{x}_i) - \frac{1}{m} \sum_i \nabla_D D(G(\boldsymbol{z}_i))
\end{aligned}$$

It has been empirically observed that large variance in the estimation of $\nabla_G \mathcal{L}(G, D)$ and $\nabla_D \mathcal{L}(G, D)$ hurts training. (Brock et al., 2018; Chavdarova et al., 2018) decrease the variance of gradient estimation to significantly improve training outcome (measured by FID/inception score). In particular, (Brock et al., 2018) reduces variance by using a larger batch size m; (Chavdarova et al., 2018) uses stochastic variance reduced gradient (SVRG). However, SVRG is computationally expensive, and it is usually more efficient to increase batch size m instead.

Our method can be naturally applied to this setup. Instead of increasing batch size, we antithetically sample $\boldsymbol{x}_{1:m}$. We use exponentially infrequent updates to q_θ described in Section 5.2, while for each update, we optimize with Eq.(14) until convergence. The intuition is that as training progresses and the model converges, training becomes increasingly stable; it is more likely that a q_θ that achieves low variance for $\nabla_G \mathcal{L}(G, D)$, can also achieve low variance for $\nabla_G \mathcal{L}(G_{\text{new}}, D_{\text{new}})$. Overall, our overhead is small enough, such that our method has more efficient wall clock time compared to increasing batch size and other baselines.

Dataset and Evaluation Metrics We train WGAN-GP (Gulrajani et al., 2017) on MNIST dataset and FashionMNIST. For comparison we use i.i.d. sampled $\boldsymbol{z}_1, \cdots, \boldsymbol{z}_m$, and negative sampling described in Section 2.2.

To evaluate the performance of the learned generator G we use inception score (Salimans et al., 2016). Let $n \in N$ be the space of labels (e.g. digit class). We train a classifier $r(n|\boldsymbol{x})$ that maps each image $\boldsymbol{x} \in \mathcal{X}$ to a distribution on the label space. Consider the joint distribution $r(\boldsymbol{x}, n) = r(\boldsymbol{x})r(n|\boldsymbol{x})$ where $r(\boldsymbol{x})$ is defined by $\boldsymbol{x} = G(\boldsymbol{z}), \boldsymbol{z} \sim p(\boldsymbol{z})$, inception score computes $\exp D_{\text{KL}}(r(n|\boldsymbol{x}) \| r(n))$.

Results The results for MNIST and FashionMNIST are shown in Figure 3 in the Appendix. Given the same batch size m our method out-performs baseline methods on the variance of gradient estimation and inception score. When m is small, our method provides a marginal improvement, which does not justify the overhead. However, as soon as the batch size is large enough (e.g. 64), our method yields a significant improvement. Even when we take into account computation overhead of learning q_θ, our method still outperforms baseline methods by a large margin.

7. Conclusion

Variance reduction is a key challenge whenever Monte Carlo estimators are used in practice. In this paper, we investigated the use of antithetic sampling, a classic variance reduction technique that is currently not widely used in machine learning. We provided a general framework to automatically learn a good antithetic distribution based on the novel Gaussian-reparameterized antithetic family. Our approach provides provably unbiased estimates and strikes a good balance between flexibility and ease of training with gradient-based or gradient-free methods. Although there is a computational cost associated with learning a good antithetic family, we demonstrated empirically that it can be amortized and the variance reduction it affords pays off in terms of wall-clock time. Antithetic sampling can be easily combined with other techniques such as importance sampling, control variates, etc. Exploring synergies between these orthogonal strategies is an exciting direction for future work. Our methods have limitations with high dimensional random variables, we take it as future work to further reduce the number of parameters in order to increase scalability.

Acknowledgements

This research was supported by Toyota Research Institute, NSF (#1651565, #1522054, #1733686), ONR (N00014-19-1-2145), AFOSR (FA9550-19-1-0024), Amazon AWS, and JP Morgan.

References

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Bishop, C. M. *Pattern recognition and machine learning*. springer, 2006.

Brock, A., Donahue, J., and Simonyan, K. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

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

Chavdarova, T., Stich, S., Jaggi, M., and Fleuret, F. Stochastic variance reduced gradient optimization of generative adversarial networks. 07 2018.

Durante, F. and Sempi, C. Copula theory: an introduction. In *Copula theory and its applications*, pp. 3–31. Springer, 2010.

Geweke, J. Antithetic acceleration of monte carlo integration in bayesian inference. *Journal of Econometrics*, 38 (1-2):73–89, 1988.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Greensmith, E., Bartlett, P. L., and Baxter, J. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

Grisetti, G., Stachniss, C., and Burgard, W. Improved techniques for grid mapping with rao-blackwellized particle filters. *IEEE transactions on Robotics*, 23(1):34–46, 2007.

Grover, A., Gummadi, R., Lazaro-Gredilla, M., Schuurmans, D., and Ermon, S. Variational rejection sampling. In *Proc. 21st International Conference on Artificial Intelligence and Statistics*, 2018.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

Hammersley, J. and Morton, K. A new monte carlo technique: antithetic variates. In *Mathematical proceedings of the Cambridge philosophical society*, volume 52, pp. 449–475. Cambridge University Press, 1956.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

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

L'Ecuyer, P. and Owen, A. B. *Monte Carlo and Quasi-Monte Carlo Methods 2008*. Springer, 2009.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., and Paul Smolley, S. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, 2017.

Neal, R. M. Annealed importance sampling. *Statistics and computing*, 11(2):125–139, 2001.

Neyman, J. On the two different aspects of the representative method: the method of stratified sampling and the method of purposive selection. *Journal of the Royal Statistical Society*, 97(4):558–625, 1934.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Sukhatme, P. V. *Sampling theory of surveys with applications*. The Indian Society Of Agricultural Statistics; New Delhi, 1957.

Weaver, L. and Tao, N. The optimal reward baseline for gradient-based reinforcement learning. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 538–545. Morgan Kaufmann Publishers Inc., 2001.

Wu, M., Goodman, N., and Ermon, S. Differentiable antithetic sampling for variance reduction in stochastic variational inference. 2019.