ARMS: Antithetic-REINFORCE-Multi-Sample Gradient for Binary Variables

Alek Dimitriev [1] **Mingyuan Zhou** [1]

Abstract

Estimating the gradients for binary variables is a task that arises frequently in various domains, such as training discrete latent variable models. What has been commonly used is a REINFORCE based Monte Carlo estimation method that uses either independent samples or pairs of negatively correlated samples. To better utilize more than two samples, we propose ARMS, an Antithetic REINFORCE-based Multi-Sample gradient estimator. ARMS uses a copula to generate any number of mutually antithetic samples. It is unbiased, has low variance, and generalizes both DisARM, which we show to be ARMS with two samples, and the leave-one-out REINFORCE (LOORF) estimator, which is ARMS with uncorrelated samples. We evaluate ARMS on several datasets for training generative models, and our experimental results show that it outperforms competing methods. We also develop a version of ARMS for optimizing the multi-sample variational bound, and show that it outperforms both VIMCO and DisARM. The code is publicly available[1].

1. Introduction

At the heart of many optimization problems is optimizing an expectation of a function with respect to the parameters of the distribution, such as $\mathcal{E}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{b})]$, where $\boldsymbol{b} \sim p_{\boldsymbol{\phi}}(\boldsymbol{b})$. There are several approaches to estimate the gradient $\nabla_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{b})]$. Most commonly used is the reparameterization gradient (Kingma & Welling, 2014; Rezende et al., 2014). If f is differentiable, and $\boldsymbol{b}$ can be expressed as a differentiable deterministic transformation $\boldsymbol{b} = \mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})$ of a variable $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$ not dependent on $\boldsymbol{\phi}$, then:

$$\nabla_{\boldsymbol{\phi}}\mathcal{E}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{\epsilon}}\Big[f\big(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})\big)\Big] = \mathbb{E}_{\boldsymbol{\epsilon}}\Big[\nabla_{\boldsymbol{\phi}} f\big(\mathcal{T}_{\boldsymbol{\phi}}(\boldsymbol{\epsilon})\big)\Big].$$

If $\boldsymbol{b}$ is discrete, however, $\mathcal{T}_{\boldsymbol{\phi}}$ is not differentiable, and f is not always differentiable, $e.g.$, if it corresponds to the reward function in reinforcement learning. This has inspired work into replacing $\mathcal{T}_{\boldsymbol{\phi}}$ (and f) with a continuous relaxation for the discrete case, using the Gumbel-Softmax trick (Maddison et al., 2017; Jang et al., 2017). Some are biased (Bengio et al., 2013; Lorberbom et al., 2019), and others, such as REBAR (Tucker et al., 2017) and RELAX (Grathwohl et al., 2018), contain a debiasing term. However, a relaxation could require much more computation, $e.g.$, in best subset selection (Yin et al., 2020) where $f(\boldsymbol{b})$ is the loss function, and $\boldsymbol{b}$ is a (sparse) binary vector that selects a subset of features on which to train the model. Evaluating a relaxed version of f for any input would train the model with all features at considerable computational cost if the number of features is large.

A second, less well-known approach, is the measure valued gradient (Rosca et al., 2019; Mohamed et al., 2020). If possible, it expresses the d^{th} element of the gradient as a difference of two measures: $\nabla_{\boldsymbol{\phi}^d} p_{\boldsymbol{\phi}}(\boldsymbol{b}) = c^d(p_+^d(\boldsymbol{b}) - p_-^d(\boldsymbol{b}))$ This allows us to estimate $\nabla_{\boldsymbol{\phi}^d}\mathcal{E}(\boldsymbol{\phi})$ using as few as two f evaluations per dimension:

$$\nabla_{\boldsymbol{\phi}^d}\mathcal{E}(\boldsymbol{\phi}) = c^d\Big(\mathbb{E}_{\boldsymbol{b}\sim p_+^d}\big[f(\boldsymbol{b})\big] - \mathbb{E}_{\boldsymbol{b}'\sim p_-^d}\big[f(\boldsymbol{b})\big]\Big),$$

but this needs to be done for each dimension separately, which is not scalable for a large number of dimensions, requiring twice as many function evaluations as dimensions.

The most widely applicable approach is the score function estimator (Glynn, 1990; Fu, 2006), also known as REINFORCE (Williams, 1992). Using the log derivative trick $\nabla_{\boldsymbol{\phi}} p(\boldsymbol{b}) = p(\boldsymbol{b})\nabla_{\boldsymbol{\phi}} \ln p(\boldsymbol{b})$, it expresses the gradient as an expectation with respect to the original distribution:

$$\nabla_{\boldsymbol{\phi}}\mathcal{E}(\boldsymbol{\phi}) = \nabla_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{b}}\big[f(\boldsymbol{b})\big] = \mathbb{E}_{\boldsymbol{b}}\big[f(\boldsymbol{b})\nabla_{\boldsymbol{\phi}} \ln p(\boldsymbol{b})\big],$$

which results in a simple form, only requiring the ability to sample from the original distribution. Unlike the reparameterization trick, it is directly applicable to discrete variables, but usually suffers from high variance in the form presented above. Recent work has focused on various mechanisms for variance reduction, applied either to general variational inference tasks (Paisley et al., 2012; Ranganath et al., 2014; Ruiz et al., 2016; Kucukelbir et al., 2017) or focusing on discrete (Mnih & Gregor, 2014; Gu et al., 2016) cases. Leveraging the specific structure of the distribution $p_{\boldsymbol{\phi}}(\boldsymbol{b})$, while

[1]McCombs School of Business, The University of Texas at Austin, Austin, Texas 78712, USA. Correspondence to: Alek Dimitriev <alekdimi@utexas.edu>, Mingyuan Zhou <mingyuan.zhou@mccombs.utexas.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

[1]https://github.com/alekdimi/arms

preserving its unbiasedness, has also proven useful (Titsias & Lázaro-Gredilla, 2015).

Antithetic variates (Owen, 2013), which are negatively correlated samples, have recently gained popularity as a variance reduction method (Wu et al., 2019; Ren et al., 2019). The first estimator to use them for binary variables is the augment-REINFORCE-merge (ARM) estimator of Yin & Zhou (2019), which reparameterizes any binary variable using a uniform distribution that has an antithetic pair. Coupling the estimates for both uniform variables results in much lower variance. Another approach is the unbiased uniform gradient (U2G) estimator of Yin et al. (2020), which is independently discovered by Dong et al. (2020) and referred to as DisARM. These two equivalent estimators, which are derived using different methods, improve upon ARM by marginalizing out the continuous reparameterization.

Our main contribution is building upon the idea of using antithetic pairs and extending it to n samples that are jointly antithetic, resulting in a novel unbiased, low variance gradient estimator for binary variables, which we denote as the Antithetic REINFORCE Multi Sample (ARMS) gradient estimator. We also develop a version of ARMS for optimizing a multi-sample bound (Burda et al., 2016), which outperforms both VIMCO (Mnih & Rezende, 2016) and DisARM.

2. Background

Let $\boldsymbol{b} = (b^1, ..., b^m)$ be a vector of m independent Bernoulli variables with $b^d \sim p_{\phi^d}(b^d) = \text{Bern}(\sigma(\phi^d))$, where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function and $\boldsymbol{\phi} = (\phi^1, ..., \phi^m)$. This paper is focused on optimizing a factorized Bernoulli expectation with respect to the logits $\boldsymbol{\phi}$:

$$\mathcal{E}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{b})], \quad \boldsymbol{b} \sim p_{\boldsymbol{\phi}}(\boldsymbol{b}) = \prod_{d=1}^{m} p_{\boldsymbol{\phi}^d}(b^d). \tag{1}$$

Throughout the paper, we use superscripts to refer to dimensions of the Bernoulli vector and subscripts to refer to particular samples. This form arises, for example, in variational inference, where a mean-field approximation is commonly used for the latent space of a variational autoencoder. Note that f could depend on $\boldsymbol{\phi}$, but there are no difficulties in estimating $\mathbb{E}_{\boldsymbol{b}}[\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi}}(\boldsymbol{b})]$, so without loss of generality, we omit this gradient term for notational simplicity.

2.1. LOORF

The Leave One Out REINFORCE (LOORF) estimator (Salimans & Knowles, 2014; Kool et al., 2019) is a simple REINFORCE baseline that utilizes n independent samples. If $\boldsymbol{b}_i \overset{iid}{\sim} p_{\boldsymbol{\phi}}(\boldsymbol{b})$, $i = 1, ..., n$, then an unbiased estimate of the gradient of $\nabla_\phi \mathcal{E}(\boldsymbol{\phi})$ is:

$$g_{\text{LOORF}}(\boldsymbol{b}) = \frac{1}{n}\sum_{i=1}^{n}\left(f(\boldsymbol{b}_i) - \frac{1}{n-1}\sum_{j\neq i} f(\boldsymbol{b}_j)\right)\nabla_{\boldsymbol{\phi}} \ln p_{\boldsymbol{\phi}}(\boldsymbol{b}_i)$$
$$= \frac{1}{n-1}\sum_{i=1}^{n}\left(f(\boldsymbol{b}_i) - \frac{1}{n}\sum_{i=1}^{n} f(b_j)\right)\nabla_{\boldsymbol{\phi}} \ln p_{\phi}(\boldsymbol{b}_i). \tag{2}$$

The latter form is simpler to implement because we can precompute the average once and subtract it for each sample. The proofs of the unbiasedness and equivalence of the two forms are known (Kool et al., 2019), but provided for completeness in Appendix A, along with all subsequent proofs.

2.2. ARM

The first binary estimator to use antithetic samples is ARM (Yin & Zhou, 2019), which accomplishes this by reparameterizing the gradient with the antithetic pair $(u, 1-u)$. We review the univariate case below. Using the observation that $\int_0^p (1-2u)du = p(1-p)$, they rewrite the analytical univariate gradient into the following form, which they denote as the augment-REINFORCE (AR) estimator:

$$\nabla_\phi \mathcal{E}(\phi) = \big(f(1) - f(0)\big)p(1-p)$$
$$= \mathbb{E}_u[f\big(\mathbb{1}_{u<p}(1-2u)\big] = \mathbb{E}_u[g_{\text{AR}}(u)], \ u \sim \text{Unif}(0,1),$$

where $\mathbb{1}_a$ is the indicator function having the value 1 if a is true and 0 otherwise. Since both $u, 1-u \sim \text{Unif}(0,1)$, they average $g_{\text{AR}}(u)$ and $g_{\text{AR}}(1-u)$, which is still unbiased, and arrive at the univariate ARM estimator:

$$\mathbb{E}_u[g_{\text{ARM}}(u)] = \mathbb{E}_u[g_{\text{AR}}(u) + g_{\text{AR}}(1-u)] =$$
$$= \frac{1}{2}\mathbb{E}_u\Big[f(\mathbb{1}_{u<p})(1-2u) + f(\mathbb{1}_{(1-u)<p})(2u-1)\Big]$$
$$= \mathbb{E}_u\left[\Big(f(\mathbb{1}_{u<p}) - f(\mathbb{1}_{u>1-p})\Big)\left(\frac{1}{2} - u\right)\right].$$

2.3. DisARM

Dong et al. (2020) observe that although ARM reduces variance by using an antithetic pair, it also increases the variance by using a continuous reparameterization. This leads them to condition on the discretized pair $(b, b') = (\mathbb{1}_{u<p}, \mathbb{1}_{1-u<p})$ and integrate out u to obtain DisARM:

$$g_{\text{DisARM}}(b, b') = \mathbb{E}_{u\sim p(u|b,b')}[g_{\text{ARM}}(u)]$$
$$= \tfrac{1}{2}(f(b) - f(b'))(b - b')\max(p, 1-p).$$

This is still unbiased due to the law of total expectation: $\mathbb{E}_u[g_{\text{ARM}}(u)] = \mathbb{E}_{b,b'}\big[\mathbb{E}_{u|b,b'}[g_{\text{ARM}}(u)]\big]$, and also has lower or equal variance due to the law of total variance:

$$\text{Var}(g_{\text{ARM}}(u)) = \mathbb{E}\big[\text{Var}_{u|b,b'}(g_{\text{ARM}}(u)\big] + \text{Var}\big(\mathbb{E}_{u|b,b'}[g_{\text{ARM}}(u)]\big)$$
$$\geq \text{Var}\big(\mathbb{E}_{u|b,b'}[g_{\text{ARM}}(u)]\big) = \text{Var}(g_{\text{DisARM}}(b, b')).$$

2.4. Copulas

A multivariate distribution $\boldsymbol{u} = (u_1, ..., u_n) \sim \mathcal{C}_n$, with marginal distributions $u_i \sim \text{Unif}(0,1)\ \forall i$ is called a copula. See Trivedi & Zimmer (2007) for a more detailed overview. We are interested in the case where there is strong negative dependence between each pair (u_i, u_j). The lower Fréchet–Hoeffding copula bound puts a limit on the negative dependence:

$$C(u_1, \ldots, u_n) \geq \max\Big\{1 - n + \sum_{i=1}^{n} u_i, 0\Big\}.$$

The lower bound is a CDF only when $n = 2$, and $C(u_1, u_2) = \max(u_1 + u_2 - 1, 0)$ corresponds to the pair $(u, 1-u), u \sim \text{Unif}(0,1)$. However for larger n this bound is not sharp (Trivedi & Zimmer, 2007), so there are no perfectly negatively dependent copulas in higher dimensions. To generate a copula sample, the probability integral transform is frequently used:

$$\begin{aligned} P(F_x(X) < u) &= P(F_x^{-1}(F_x(X)) < F_x^{-1}(u)) \\ &= P(X < F_x^{-1}(u)) = F_x(F_x^{-1}(u)) = u, \end{aligned} \quad (3)$$

where we assume F_x is a bijection to simplify the proof. If for each dimension of a multivariate sample $\boldsymbol{x} = (x_1, ..., x_n)$, it is distributed as $x_i \sim F_{x_i}(x)$, then a copula sample is obtained by applying the CDF function to each dimension $\boldsymbol{u} = \big(F_{x_1}(x_1), ..., F_{x_n}(x_n)\big)$.

3. ARMS

In the univariate case with $n = 2$ samples, antithetic pair estimators like (Dis)ARM have lower variance than LOORF, but we have observed that even on a toy example, they are outperformed by LOORF as the number of samples rises. This motivates us to find a way to generate n jointly antithetic samples, and use them to construct an unbiased estimator. For clarity, we first derive the univariate case of our jointly antithetic estimator, which is obtained by starting with two independent samples, extending this to two arbitrarily correlated Bernoulli samples, and then generalizing to n samples.

For two samples, LOORF has the following equivalent simple form, which we denote as the Product of Differences (PoD) estimator:

$$g_{\text{PoD}}(b, b') = \frac{1}{2}\Big(f(b) - f(b')\Big)\Big(\nabla_\phi \ln p_\phi(b) - \nabla_\phi \ln p_\phi(b')\Big). \quad (4)$$

Unless b and b' are independent, this estimator will be biased. However, when b and b' are both Bernoulli variables, we can obtain an unbiased estimator for any bivariate distribution, by using a multiplicative debiasing term. Let $(b, b') \sim \mathcal{B}_2(p)$ denote a sample from a bivariate Bernoulli distribution with marginals $b, b' \sim \text{Bern}(p)$ and $\text{Corr}(b, b') = \rho$. In this case, $g_{\text{PoD}}(b, b') = (f(b) - f(b')(b - b')/2$, whose expectation is expressed as:

$$\mathbb{E}_{b,b'}\big[g_{\text{PoD}}(b, b')\big] = \big(f(1) - f(0)\big)P(b = 1, b' = 0),$$

with the analytical gradient being $\nabla_\phi \mathcal{E}(\phi) = (f(1) - f(0))p(1-p)$. Therefore, multiplying the above expression with $p(1-p)/P(b = 1, b' = 0)$ results in an unbiased estimator for any dependence structure:

$$\begin{aligned} &\mathbb{E}_{b,b'}\left[\frac{p(1-p)}{P(b = 1, b' = 0)} g_{\text{PoD}}(b, b')\right] \\ &\quad = \frac{p(1-p)}{P(b = 1, b' = 0)}\big(f(1) - f(0)\big)P(b = 1, b' = 0) \\ &\quad = \big(f(1) - f(0)\big)p(1-p) = \nabla_\phi \mathcal{E}(\phi). \end{aligned}$$

Lastly, note that:

$$\begin{aligned} P(b = 1, b' = 0) &= p - P(b = 1, b' = 1) \\ &= 2(p - p(1-p)\rho - p^2) = p(1-p)(1-\rho), \end{aligned}$$

which simplifies the multiplicative term to $1/(1-\rho)$. We summarize the derivation with the following theorem, denoting it the Antithetic-REINFORCE-Two-Sample (ARTS) gradient estimator.

Theorem 1. *Let $(b, b') \sim \mathcal{B}_2(\sigma(\phi))$ be a sample from a bivariate Bernoulli distribution with marginal distributions $b, b' \sim \text{Bern}(\sigma(\phi))$ and correlation $\rho = \text{Corr}(b, b')$. An unbiased estimator of $\nabla_\phi \mathbb{E}_b[f(b)]$ is:*

$$g_{\text{ARTS}}(b, b', \rho) = \Big(f(b) - f(b')\Big)\frac{b - b'}{2(1-\rho)}. \quad (5)$$

Using Theorem 1, we can easily derive DisARM/U2G, since it is just ARTS with a specific correlation. The antithetic pair used is:

$$(b, b') = (\mathbb{1}_{u<p}, \mathbb{1}_{(1-u)<p}),\ u \sim \text{Unif}(0,1),$$

in which case:

$$\begin{aligned} P(b = 1, b' = 0) &= P(u < p, 1 - u > p) \\ &= P(u < \min(p, 1-p)) = \min(p, 1-p). \end{aligned}$$

The multiplicative term is $p(1-p)/\min(p, 1-p) = \max(p, 1-p)$, which results in the DisARM/U2G estimator:

$$g_{\text{DisARM}}(b, b') = \frac{1}{2}\big(f(b) - f(b')\big)(b - b')\max(p, 1-p).$$

The next theorem shows that in the two sample case, the bivariate distribution with the lowest gradient variance uses the same pair of antithetic variables. The proof is given in Appendix A.

Theorem 2. *Let $b, b' \overset{iid}{\sim}$ Bern(p), and $(\tilde{b}, \tilde{b}') \sim \mathcal{B}(p)$. Then:*

$$\rho < 0 \implies Var(g_{\text{ARTS}}(\tilde{b}, \tilde{b}', \rho)) < Var(g_{\text{PoD}}(b, b')).$$

Furthermore, $Var(g_{\text{ARTS}}(\tilde{b}, \tilde{b}', \rho))$ is a decreasing function of ρ and achieves its minimum at:

$$\rho_{min} = -\min\left(\frac{p}{1-p}, \frac{1-p}{p}\right),$$

with a corresponding debiasing term $1/(1-\rho_{min}) = max(p, 1-p)$. If $(b, b') = (\mathbb{1}_{u<p}, \mathbb{1}_{1-u<p})$, where $u \sim$ Unif$(0,1)$, then $Corr(b, b') = \rho_{min}$.

We now extend ARTS to n samples. Let $\boldsymbol{b} = (b_1, ..., b_n) \sim \mathcal{B}_n(p)$ with marginals $b_i \sim$ Bern(p) and correlations $\rho_{ij} =$ Corr(b_i, b_j). If we compute $g_{\text{ARTS}}(b_i, b_j, \rho_{ij})$ for all $\binom{n}{2}$ pairs and take the average, we obtain an unbiased estimator because of the linearity of expectations:

$$\begin{aligned}
&\mathbb{E}_{b_1,...,b_n}\left[\frac{1}{n(n-1)}\sum_{i\neq j} g_{\text{ARTS}}(b_i, b_j, \rho)\right] \\
&= \frac{1}{n(n-1)}\sum_{i\neq j}\mathbb{E}\big[g_{\text{ARTS}}(b_i, b_j, \rho)\big] \\
&= \frac{1}{n(n-1)}\sum_{i\neq j}\nabla_\phi\mathcal{E}(\phi) = \nabla_\phi\mathcal{E}(\phi).
\end{aligned}$$

However the computational effort required is $\mathcal{O}(n^2)$, as opposed to LOORF, which is $\mathcal{O}(n)$. But if we restrict ourselves to a symmetric correlation structure, *i.e.*, $\rho_{ij} = \rho$ for $i \neq j$, we can compute the above average in $\mathcal{O}(n)$. To show this, we will make use of the following, proved in Appendix A:

$$\begin{aligned}
g_{\text{LOORF}}(\boldsymbol{b}) &= \frac{1}{n}\sum_{i=1}^{n}\left(f(b_i) - \frac{1}{n}\sum_{j=1}^{n} f(b_j)\right)\nabla_\phi \ln p(b_i) \\
&= \frac{1}{n(n-1)}\sum_{i\neq j}\frac{1}{2}\Big(f(b_i) - f(b_j)\Big)\Big(\nabla_\phi \ln p(b_i) - \nabla_\phi \ln p(b_j)\Big) \\
&= \frac{1}{n(n-1)}\sum_{i\neq j} g_{\text{PoD}}(b_i, b_j), \qquad (6)
\end{aligned}$$

as well as the fact that $g_{\text{ARTS}}(b_i, b_j, \rho) = g_{\text{PoD}}(b_i, b_j)/(1-\rho)$, to obtain the ARMS estimator:

$$\begin{aligned}
g_{\text{ARMS}}(\boldsymbol{b}, \phi, \rho) &= \frac{1}{n(n-1)}\sum_{i\neq j} g_{\text{ARTS}}(b_i, b_j, \rho) \\
&= \frac{1}{n(n-1)}\sum_{i\neq j}\frac{g_{\text{PoD}}(b_i, b_j)}{1-\rho} = \frac{g_{\text{LOORF}}(\boldsymbol{b})}{1-\rho}
\end{aligned}$$

where $\boldsymbol{b} = (b_1, ..., b_n)$. We summarize the derivation in Theorem 3. It is clear from Eq. 6 why LOORF outperforms (Dis)ARM when the number of samples increases. It is because computing LOORF for n samples is equivalent to averaging $n(n-1)/2$ independent pairs, whereas (Dis)ARM uses only $n/2$ antithetic pairs, and the antithetic variance reduction does not make up for using $n-1$ times fewer pairs.

Theorem 3. *Let $\tilde{\boldsymbol{b}} = (\tilde{b}_1, ..., \tilde{b}_n) \sim \mathcal{B}_n(\sigma(\phi))$ be a sample from an n-variate Bernoulli distribution with marginal distributions $\tilde{b}_1, ..., \tilde{b}_n \sim$ Bern$(\sigma(\phi))$ and pairwise correlation $\rho = Corr(\tilde{b}_i, \tilde{b}_j), i \neq j$. An unbiased estimator of $\nabla_\phi\mathbb{E}_{b\sim\text{Bern}(\sigma(\phi))}[f(b)]$ is:*

$$\begin{aligned}
&g_{\text{ARMS}}(\tilde{\boldsymbol{b}}, \phi, \rho) \\
&= \frac{1}{n-1}\sum_{i=1}^{n}\left(f(\tilde{b}_i) - \frac{1}{n}\sum_{j=1}^{n} f(\tilde{b}_j)\right)\frac{\tilde{b}_i - \sigma(\phi)}{1-\rho}. \qquad (7)
\end{aligned}$$

It is simple to show that ARMS generalizes both LOORF and DisARM. For n independent samples $\rho = 0$, so the debiasing term becomes $1/(1-\rho) = 1$ and the estimator reduces to LOORF. For $n = 2$ samples, it reduces to DisARM/U2G if we use an antithetic uniform pair as shown in Theorem 2.

Although Theorem 3 shows us how to form an unbiased estimator, it does so assuming we can sample n correlated Bernoulli variables, and that we know their common correlation ρ. Therefore, to use ARMS in practice, we must find a way to sample $(b_1, ..., b_n) \sim \mathcal{B}_n(p)$, and also be able to calculate $\rho =$ Corr(b_i, b_j). The next section outlines two copula based approaches that satisfy both conditions, although there are other ways, the exploration of which we leave for future work.

3.1. Copula Sampling for Multivariate Bernoulli

A copula sample can be transformed into a multivariate Bernoulli using the reparameterization $b \sim$ Bern$(p) \iff b = \mathbb{1}_{u<p}$, $u \sim$ Unif$(0,1)$. Furthermore, symmetry in the bivariate copula CDFs:

$$P(u_i < p, u_j < p) = P(u_k < p, u_l < p), \quad \forall i \neq j, k \neq l,$$

implies symmetry for the Bernoulli correlations $\rho_{ij} = \rho_{kl}$, because $\mathbb{E}[b_i b_j] = P(b_i = 1, b_j = 1) = P(u_i < p, u_j < p)$. As a consequence, if we use a copula for sampling, evaluating the bivariate CDF will be required to calculate ρ and different copulas will produce different correlations for each $p \in [0,1]$. Ideally, we want this correlation to be as low as possible for all p. For any distribution, the lower limit for a common correlation between n identically distributed variables is $\rho = -1/(n-1)$, which follows from rearranging the non-negativity of the variance equation: Var$(\sum_{i=1}^{n} b_i) = n$Var$(b_1) + n(n-1)\rho$Var$(b_1) \geq 0$. We propose two different copulas, both of which start with

Algorithm 1 Antithetic Dirichlet copula sampling

Input: Copula dimension n
Sample $v_i \sim \text{Unif}(0,1)$, $i = 1 \dots n$
Set: $d_i = \ln(v_i) / \sum_{j=1}^{n} \ln(v_j)$
Invert back to uniform: $\tilde{u}_i = 1 - (1 - d_i)^{n-1}$
return: $(\tilde{u}_1, ..., \tilde{u}_n)$, $(1 - \tilde{u}_1, ..., 1 - \tilde{u}_n)$

maximally negatively correlated variables, and preserve most of the correlation when transformed to uniform variables.

3.1.1. Dirichlet Copula

The first of two sampling approaches for ARMS is a Dirichlet copula, since a sample from the Dirichlet distribution exhibits perfect negative dependence, which means the correlation between any pair of elements in the vector is the lower bound $1/(n-1)$. This makes it a suitable candidate, provided we can transform the marginal distributions to uniform, and can compute the bivariate CDF. Both conditions are possible if we restrict ourselves to the uniform Dirichlet density $\boldsymbol{d} \sim \text{Dir}(\mathbf{1}_n)$. In this case, the marginal CDFs of $\boldsymbol{d}$ and $\mathbf{1} - \boldsymbol{d}$ have closed-form solutions $F_{d_i}(x) = 1 - (1-x)^{n-1}$ and $F_{1-d_i}(x) = (1-x)^{n-1}$, respectively (Ng et al., 2011). Applying Eq. 3 results in two different copula samples:

$$\tilde{\boldsymbol{u}} = (\tilde{u}_1, ..., \tilde{u}_n), \quad \tilde{u}_i = 1 - (1 - d_i)^{n-1}$$
$$\tilde{\boldsymbol{u}}' = (\tilde{u}'_1, ..., \tilde{u}'_n), \quad \tilde{u}'_i = (1 - d_i)^{n-1} = 1 - \tilde{u}_i,$$

with the full procedure summarized in Algorithm 1. This satisfies the sampling requirement, since given $\tilde{\boldsymbol{u}}$, a multivariate Bernoulli sample is $\tilde{\boldsymbol{b}} = (\tilde{b}_1, ..., \tilde{b}_n)$, where $\tilde{b}_i = \mathbb{1}_{\tilde{u}_i < p}$. The bivariate CDF, necessary for calculating $\rho = \text{Corr}(\tilde{b}_i, \tilde{b}_j)$, also has a closed-form solution:

$$\begin{aligned} P(\tilde{u}_i < p, \tilde{u}_j < p) &= P(d_i < q, d_j < q) \\ &= 2p - 1 + \max(0, 2(1-p)^{1/(n-1)} - 1)^{n-1} \\ P(\tilde{u}'_i < p, \tilde{u}'_j < p) &= \max(0, 2p^{1/(n-1)} - 1)^{n-1}, \end{aligned} \tag{8}$$

where $q = 1 - (1-p)^{1/(n-1)}$. We summarize the Dirichlet copula approach with the following theorem, with the full derivation given in Appendix B.

Theorem 4. *Let $\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{u}}'$ be the samples produced by the Dirichlet copula in Algorithm 1. If:*

$$\tilde{\boldsymbol{b}} = (\tilde{b}_1, ..., \tilde{b}_n), \quad \tilde{b}_i = \mathbb{1}_{\tilde{u}_i < p}$$
$$\tilde{\boldsymbol{b}}' = (\tilde{b}'_1, ..., \tilde{b}'_n), \quad \tilde{b}'_i = \mathbb{1}_{\tilde{u}'_i < p},$$

then $\tilde{\boldsymbol{b}}, \tilde{\boldsymbol{b}}' \sim \mathcal{B}_n(p)$ are samples from an n-variate Bernoulli distribution with common pairwise correlations:

$$\rho = \frac{\max(0, 2(1-p)^{\frac{1}{n-1}} - 1)^{n-1} - (1-p)^2}{p(1-p)}$$
$$\rho' = \frac{\max(0, 2p^{1/(n-1)} - 1)^{n-1} - p^2}{p(1-p)}.$$

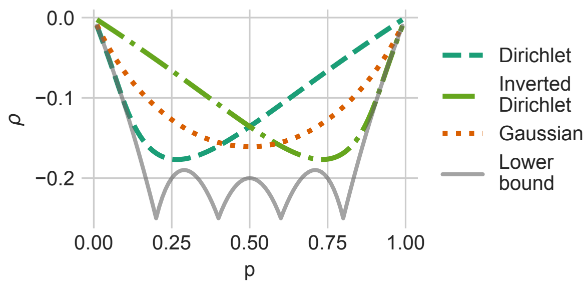

Figure 1. Correlations $\rho = \text{Corr}(b_i, b_j)$ for $p \in [0, 1]$, when using different copulas to sample a five dimensional multivariate Bernoulli $\boldsymbol{b} \sim \mathcal{B}_5(p)$, along with the bound in (Hoernig, 2018).

Algorithm 2 Antithetic Gaussian copula sampling

Input: copula dimension n
For $i, j = 1, ..., n$, set:

$$(\boldsymbol{\rho})_{ij} = \begin{cases} -1/(n-1) & , i \neq j \\ 1 & , \text{otherwise} \end{cases}$$

Sample $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\rho})$
Invert back to uniform: $\tilde{u}_i = \Phi(x_i) \quad i = 1 \dots n$
return: $\tilde{u}_1, ..., \tilde{u}_n$

Lastly, although $\text{Corr}(\tilde{u}_i, \tilde{u}_j) = \text{Corr}(\tilde{u}'_i, \tilde{u}'_j)$, their bivariate CDFs are different, and we observed that if $p > 0.5$, then $\text{Corr}(\tilde{b}_i, \tilde{b}_j) < \text{Corr}(\tilde{b}'_i, \tilde{b}'_j)$ (and vice versa). Therefore, in our experiments we use $\tilde{\boldsymbol{u}}$ when $p > 0.5$ and $\tilde{\boldsymbol{u}}'$ when $p < 0.5$. We illustrate the different Bernoulli correlations of each copula as a function of p in Fig. 1, along with a lower bound of the lower bound for symmetric Bernoulli variables (Hoernig, 2018).

3.1.2. Antithetic Gaussian Copula

An alternative sampling method for n dependent uniform variables is the widely used Gaussian copula, Let $\boldsymbol{x} \sim \mathcal{N}(0, \boldsymbol{\rho})$ denote a sample from an n-variate Gaussian, and denote $\Phi(x)$ to be the univariate Gaussian CDF. From Eq 3, by applying the CDF to each dimension of $\boldsymbol{x}$, the result is a Gaussian copula sample: $\boldsymbol{u} = (\Phi(x_1), ... \Phi(x_n))$. Since we are interested in mutually antithetic samples, we use the correlation matrix $\rho_{ij} = -1/(n-1), i \neq j$. We summarize the sampling procedure in Algorithm 2. The antithetic

Gaussian copula can also be recovered from the Dirichlet copula, where $\forall i, \alpha_i = \alpha$, and $\alpha \to \infty$, in which case the distribution becomes a multivariate Gaussian with the same common negative correlation. Although neither the univariate nor bivariate CDF have a closed form, all commonly used deep learning packages contain numerical approximations for both, which we make use of in our experiments.

3.2. ARMS for the Multivariate Case

Let $\boldsymbol{b} \sim p(\boldsymbol{b} \,|\, \boldsymbol{\phi}) = \prod_{d=1}^{m} p_{\phi^d}(b^d)$ denote a m-dimensional factorized Bernoulli sample with probabilities $\sigma(\boldsymbol{\phi}) = (\sigma(\boldsymbol{\phi}_1), ..., \sigma(\boldsymbol{\phi}_m))$, such that $b^d \sim \text{Bern}(\sigma(\boldsymbol{\phi}^d)) = p_{\boldsymbol{\phi}^d}(b)$, and let $\boldsymbol{b}^{-d} = (b_1, ..., b_{d-1}, b_{d+1}, ..., b_m)$. We sample n correlated Bernoulli variables for each dimension:

$$\tilde{\boldsymbol{b}}^d = (\tilde{b}_1^d, ..., \tilde{b}_n^d) \sim \mathcal{B}_n(\sigma(\boldsymbol{\phi}^d)), \quad d = 1, ..., m,$$

with $\boldsymbol{\rho}^d = \text{Corr}(\tilde{\boldsymbol{b}}_i^d, \tilde{\boldsymbol{b}}_j^d), i \neq j$ being the pairwise correlations. Focusing on the d^{th} dimension, we have:

$$\begin{aligned} &\nabla_{\boldsymbol{\phi}^d} \mathbb{E}_{p_{\boldsymbol{\phi}}(\boldsymbol{b})}\left[f(\boldsymbol{b})\right] = \mathbb{E}_{\boldsymbol{b}^{-d}}\left[\nabla_{\boldsymbol{\phi}^d} \mathbb{E}_{b^d}\left[f(\boldsymbol{b}^{-d}, b^d)\right]\right] \\ &= \mathbb{E}_{\boldsymbol{b}^{-d}}\Bigg[\mathbb{E}_{\tilde{\boldsymbol{b}}^d}\Bigg[\frac{1}{n-1}\sum_{i=1}^{n}\Bigg(f(\boldsymbol{b}^{-d}, \tilde{b}_i^d) \\ &\qquad - \frac{1}{n}\sum_{j=1}^{n} f(\boldsymbol{b}^{-d}, \tilde{b}_j^d)\Bigg)\frac{\tilde{b}_i^d - \sigma(\boldsymbol{\phi}^d)}{1-\boldsymbol{\rho}^d}\Bigg]\Bigg]. \end{aligned}$$

Replacing $\boldsymbol{b}^{-d}$ by the already sampled $\tilde{\boldsymbol{b}}_i^{-d}$, we have:

$$\begin{aligned} &\mathbb{E}_{\tilde{\boldsymbol{b}}^1,...,\tilde{\boldsymbol{b}}^m}\Bigg[\frac{1}{n-1}\sum_{i=1}^{n}\Bigg(f(\tilde{\boldsymbol{b}}_i^{-d}, \tilde{b}_i^d) \\ &\qquad - \frac{1}{n}\sum_{j=1}^{n} f(\tilde{\boldsymbol{b}}_j^{-d}, \tilde{b}_j^d)\Bigg)\frac{\tilde{b}_i^d - \sigma(\boldsymbol{\phi}^d)}{1-\boldsymbol{\rho}^d}\Bigg] \\ &= \mathbb{E}_{\tilde{\boldsymbol{b}}^1,...,\tilde{\boldsymbol{b}}^m}\Bigg[\frac{1}{n-1}\sum_{i=1}^{n}\Bigg(f(\tilde{\boldsymbol{b}}_i) \\ &\qquad - \frac{1}{n}\sum_{j=1}^{n} f(\tilde{\boldsymbol{b}}_j)\Bigg)\frac{\tilde{b}_i^d - \sigma(\boldsymbol{\phi}^d)}{1-\boldsymbol{\rho}^d}\Bigg], \end{aligned}$$

where $\tilde{\boldsymbol{b}}_i = (\tilde{\boldsymbol{b}}_i^1, ..., \tilde{\boldsymbol{b}}_i^m)$, i.e., the i^{th} sample from each dimension. The multivariate ARMS estimator is then:

$$\begin{aligned} &g_{\text{ARMS}}(\tilde{\boldsymbol{b}}_1, ..., \tilde{\boldsymbol{b}}_n, \boldsymbol{\phi}, \boldsymbol{\rho}) \\ &= \frac{1}{n-1}\sum_{i=1}^{n}\left(f(\tilde{\boldsymbol{b}}_i) - \frac{1}{n}\sum_{j=1}^{n} f(\tilde{\boldsymbol{b}}_j)\right)\frac{\tilde{\boldsymbol{b}}_i - \sigma(\boldsymbol{\phi})}{1-\boldsymbol{\rho}}, \end{aligned}$$

where just like the univariate case, we only need n evaluations of f regardless of the number of dimensions m.

3.3. ARMS for the Multi Sample Variational Bound

In variational inference, a commonly optimized objective of the form of Eq. 1 is the ELBO (Jordan et al., 1998), a tractable lower bound of the marginal likelihood:

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x})}[\ln p_{\boldsymbol{\theta}}(\boldsymbol{b}, \boldsymbol{x}) - \ln q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x})] \leq \ln p(\boldsymbol{x}).$$

If we use multiple independent samples $\boldsymbol{b}_1, ..., \boldsymbol{b}_n$, with $q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x}) = \prod_{k=1}^{n} q_{\boldsymbol{\phi}}(\boldsymbol{b}_k \,|\, \boldsymbol{x})$, Burda et al. (2016) show that we can form a tighter bound:

$$\mathcal{L}_n = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x})}\left[\ln\left(\frac{1}{n}\sum_{k=1}^{n} r(\boldsymbol{b}_k)\right)\right], \; r(\boldsymbol{b}_k) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{b}_k, \boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{b}_k \,|\, \boldsymbol{x})},$$

and proved that $\mathcal{L}_{\text{ELBO}} \leq \mathcal{L}_n \leq \mathcal{L}_{n+1} \leq \ln p(\boldsymbol{x})$. The multi sample bound depends non-linearly on each sample due to the logarithm inside the expectation, which means replacing the n i.i.d. samples with dependent ones changes the objective. For example, if we let $\boldsymbol{b}_1 = ... = \boldsymbol{b}_n = \boldsymbol{b}$, then instead of $\mathcal{L}_n$, we obtain:

$$\mathbb{E}_{\boldsymbol{b} \sim q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x})}\left[\ln \frac{1}{n}\sum_{i=1}^{n} r(\boldsymbol{b})\right] = \mathbb{E}_{\boldsymbol{b} \sim q_{\boldsymbol{\phi}}(\boldsymbol{b} \,|\, \boldsymbol{x})}\Big[\ln r(\boldsymbol{b})\Big] = \mathcal{L}_1.$$

However, we can still use ARMS to create a local baseline for each sample if we draw n additional correlated samples. Let $\boldsymbol{b}_1, .., \boldsymbol{b}_n$ denote the i.i.d. samples and $\tilde{\boldsymbol{b}}_1, .., \tilde{\boldsymbol{b}}_n$ the correlated samples. Define $f(\boldsymbol{b}) = \ln\left(\frac{1}{n}\sum_{k=1}^{n} r(\boldsymbol{b}_k)\right)$, and $f_{\boldsymbol{b}_{-k}}(\boldsymbol{b}_k) = \ln\Big(\frac{1}{n}\big(\sum_{l \neq k} r(\boldsymbol{b}_l) + r(\boldsymbol{b}_k)\big)\Big)$. Because the samples are i.i.d., the gradient can be written as:

$$\begin{aligned} &\nabla_{\boldsymbol{\phi}} \mathbb{E}_{\boldsymbol{b}}[f(\boldsymbol{b})] = \mathbb{E}_{\boldsymbol{b}}\Big[f(\boldsymbol{b})\nabla_{\boldsymbol{\phi}} \ln \prod_{k=1}^{n} q_{\boldsymbol{\phi}}(\boldsymbol{b}_k|\boldsymbol{x})\Big] = \\ &= \sum_{k=1}^{n} \mathbb{E}_{\boldsymbol{b}}\Big[f(\boldsymbol{b})\nabla_{\boldsymbol{\phi}} \ln q_{\boldsymbol{\phi}}(\boldsymbol{b}_k|\boldsymbol{x})\Big] = \\ &= \sum_{k=1}^{n} \mathbb{E}_{\boldsymbol{b}_{-k}}\Big[\mathbb{E}_{\boldsymbol{b}_k}\Big[f_{\boldsymbol{b}_{-k}}(\boldsymbol{b}_k)\nabla_{\boldsymbol{\phi}} \ln q_{\boldsymbol{\phi}}(\boldsymbol{b}_k|\boldsymbol{x})\Big]\Big] \\ &= \sum_{k=1}^{n} \mathbb{E}_{\boldsymbol{b}_{-k}}\Bigg[\mathbb{E}_{\tilde{\boldsymbol{b}}_1,..,\tilde{\boldsymbol{b}}_n}\Bigg[\frac{1}{n-1}\sum_{i=1}^{n}\Big(f_{\boldsymbol{b}_{-k}}(\tilde{\boldsymbol{b}}_i) \\ &\qquad - \frac{1}{n}\sum_{j=1}^{n} f_{\boldsymbol{b}_{-k}}(\tilde{\boldsymbol{b}}_j)\Big)\frac{\tilde{\boldsymbol{b}}_i - \sigma(\boldsymbol{\phi})}{1-\boldsymbol{\rho}}\Bigg]\Bigg], \end{aligned}$$

where in the last line we replaced the inner expectation, which is just REINFORCE w.r.t. $\boldsymbol{b}_k$, with the ARMS estimator. Note that we can precompute $r(\boldsymbol{b}_1), ..., r(\boldsymbol{b}_n)$ and $r(\tilde{\boldsymbol{b}}_1), ..., r(\tilde{\boldsymbol{b}}_n)$ for a total of $2n$ function evaluations, which is the same number required by DisARM for optimizing the same n sample bound, and the number required by VIMCO for the $2n$ sample bound, which is the what we compare to in the experiments.

4. Experimental Results

First, to illustrate the benefit of jointly antithetic samples, we compare ARMS, using either a Dirichlet (ARMS-D)

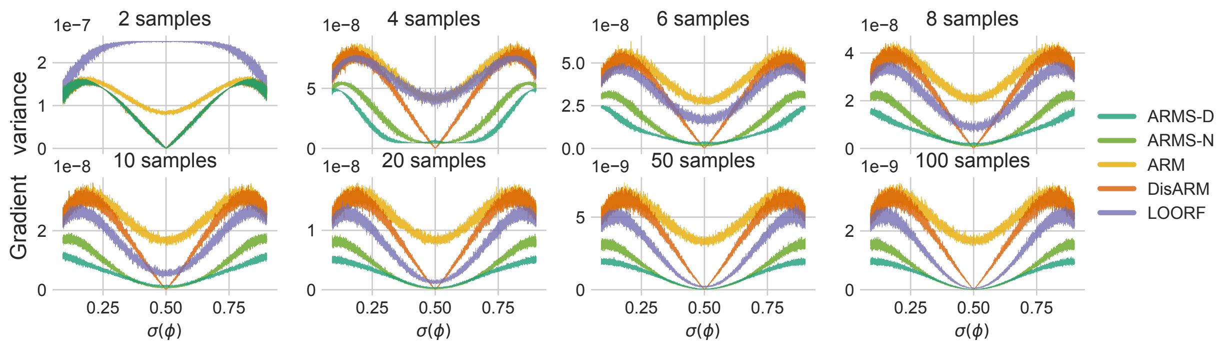

Figure 2. Variance of the gradients of each estimator for the toy problem across a range of values of ϕ. The top and bottom row correspond to using $n \in \{2, 4, 6, 8\}$ and $n \in \{10, 20, 50, 100\}$ samples, respectively, at each gradient step. The variance is estimated using 1000 Monte Carlo samples per step.

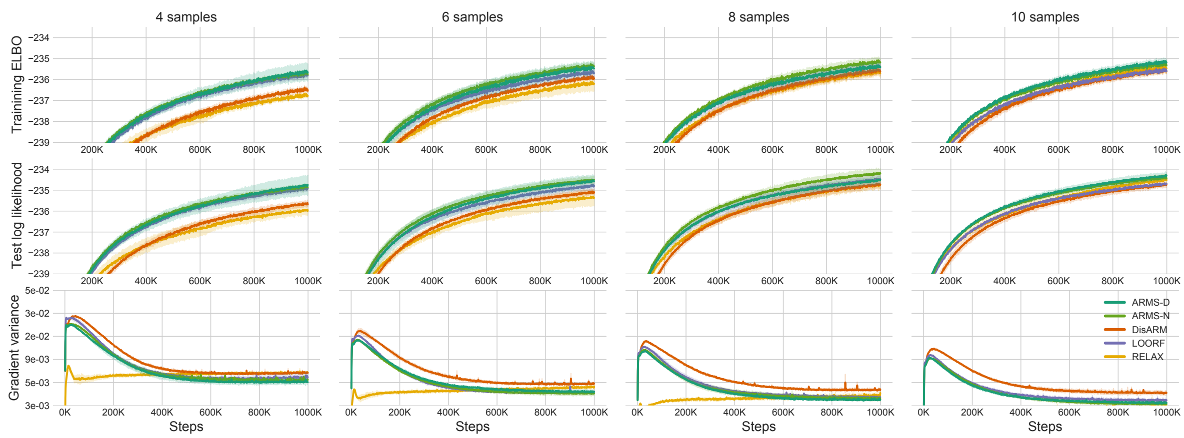

Figure 3. Training a nonlinear discrete VAE on Fashion MNIST using the ELBO. Columns correspond to $n \in \{4, 6, 8, 10\}$ samples used per step, respectively. Rows correspond to the training ELBO, test log likelihood, and the variance of the gradient updates averaged over all parameters. Results for different datasets and other networks can be found in Appendix D.

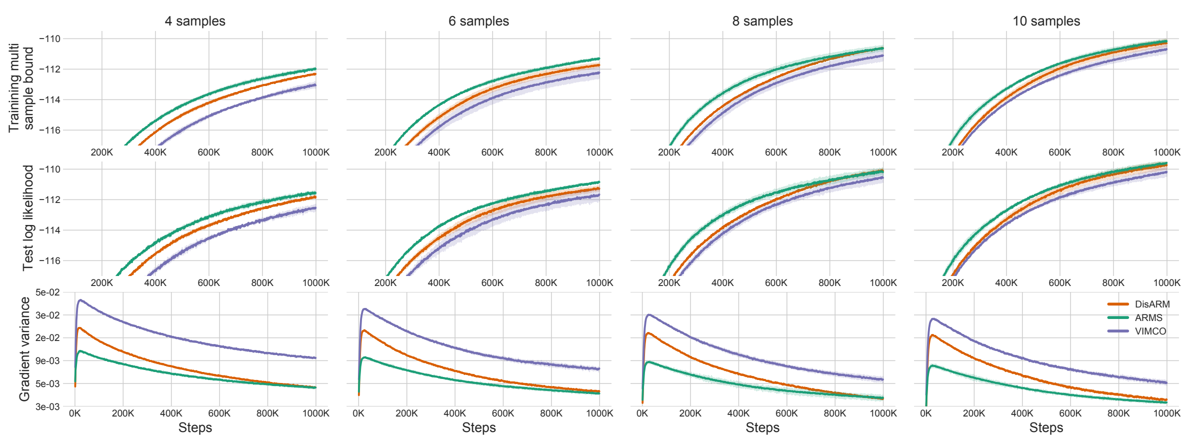

Figure 4. Training a linear discrete VAE on Dynamic MNIST using the multi sample bound. Columns correspond to $n \in \{4, 6, 8, 10\}$ samples used per step, respectively. Rows correspond to the training multi sample bound, test log likelihood, and the variance of the gradient updates averaged over all parameters. Results for different datasets and other networks can be found in Appendix D.

or Gaussian copula (ARMS-N), to ARM, DisARM, and LOORF on the following toy problem (Tucker et al., 2017), where the task is to maximize:

$$\mathcal{E}(\phi) = \mathbb{E}_b[(b - 0.499)^2], \quad b \sim \text{Bern}(\sigma(\phi)),$$

with the optimal solution being $\sigma(\phi) = 1$. We run the analysis for $n \in \{2, 4, 6, 8, 10, 20, 50, 100\}$ samples per gradient step, using an even number to ensure a fair comparison to methods that can only use pairs. Figure 2 shows the vari-

Table 1. Final training ELBO of VAEs using different estimators. Results are reported on three datasets: Dynamic MNIST, Fashion MNIST, and Omniglot, and for 4, 6, 8, and 10 samples, with the best performing methods in bold.

		Samples	ARMS-D	ARMS-N	LOORF	DisARM	RELAX
Dynamic MNIST	Linear	4	-112.13 ± 0.10	**-111.96 ± 0.09**	-112.32 ± 0.04	-113.26 ± 0.05	-112.98 ± 0.25
		6	-111.03 ± 0.02	**-110.89 ± 0.07**	-110.99 ± 0.07	-112.11 ± 0.03	-111.46 ± 0.06
		8	**-110.30 ± 0.04**	-110.62 ± 0.06	-110.42 ± 0.04	-111.78 ± 0.07	-110.58 ± 0.01
		10	**-110.08 ± 0.05**	-110.14 ± 0.09	-110.17 ± 0.04	-111.08 ± 0.11	-110.17 ± 0.09
	Nonlinr	4	**-98.65 ± 0.16**	-98.97 ± 0.13	**-98.62 ± 0.05**	-100.45 ± 0.16	-100.52 ± 0.08
		6	-98.53 ± 0.13	**-97.87 ± 0.01**	-98.14 ± 0.18	-99.28 ± 0.11	-99.17 ± 0.17
		8	**-97.90 ± 0.12**	**-97.89 ± 0.10**	-98.14 ± 0.21	-98.69 ± 0.21	-98.80 ± 0.02
		10	-97.64 ± 0.06	**-97.32 ± 0.11**	-97.50 ± 0.29	-98.62 ± 0.12	-98.69 ± 0.07
Fashion MNIST	Linear	4	**-252.56 ± 0.11**	-252.69 ± 0.06	-252.71 ± 0.09	-254.02 ± 0.05	-253.53 ± 0.06
		6	-251.94 ± 0.13	**-251.73 ± 0.05**	-252.03 ± 0.08	-252.97 ± 0.06	-252.31 ± 0.14
		8	**-251.32 ± 0.11**	**-251.11 ± 0.23**	-251.41 ± 0.10	-252.57 ± 0.05	-251.36 ± 0.08
		10	-251.29 ± 0.02	**-251.08 ± 0.08**	-251.26 ± 0.03	-251.75 ± 0.21	**-251.16 ± 0.06**
	Nonlinr	4	**-235.65 ± 0.12**	-235.75 ± 0.06	-235.80 ± 0.07	-236.54 ± 0.06	-236.77 ± 0.03
		6	-235.47 ± 0.19	**-235.36 ± 0.08**	-235.70 ± 0.13	-235.94 ± 0.05	-236.20 ± 0.25
		8	-235.41 ± 0.10	**-235.19 ± 0.14**	-235.40 ± 0.13	-235.62 ± 0.16	-235.70 ± 0.14
		10	**-235.18 ± 0.11**	-235.32 ± 0.05	-235.59 ± 0.01	-235.60 ± 0.09	-235.46 ± 0.19
Omniglot	Linear	4	**-118.25 ± 0.08**	**-118.27 ± 0.05**	-118.41 ± 0.07	-119.24 ± 0.17	-118.75 ± 0.08
		6	**-117.62 ± 0.01**	**-117.62 ± 0.04**	-117.75 ± 0.08	-118.47 ± 0.12	-117.90 ± 0.03
		8	**-117.60 ± 0.05**	-117.66 ± 0.12	-117.74 ± 0.10	-118.41 ± 0.10	-117.71 ± 0.02
		10	**-117.03 ± 0.09**	**-116.99 ± 0.04**	-117.21 ± 0.08	-117.70 ± 0.01	-117.13 ± 0.05
	Nonlinr	4	-112.09 ± 0.27	**-112.03 ± 0.12**	-112.20 ± 0.26	-113.24 ± 0.16	-114.08 ± 0.35
		6	-111.50 ± 0.06	**-111.39 ± 0.10**	**-111.26 ± 0.15**	-112.30 ± 0.05	-113.71 ± 0.13
		8	**-110.91 ± 0.04**	-111.01 ± 0.06	**-110.85 ± 0.35**	-111.82 ± 0.09	-113.64 ± 0.10
		10	**-110.66 ± 0.05**	-110.79 ± 0.26	-110.79 ± 0.20	-111.33 ± 0.19	-114.00 ± 0.10

ance of the gradient of each estimator as the optimization progresses from $\sigma(\phi) = 0.1$ to $\sigma(\phi) = 0.9$, which takes approximately 20,000 steps. Both ARMS-D and ARMS-N have significantly lower variance for almost all values of ϕ, but the Dirichlet copula appears to have lower variance, except for a narrow range near $\sigma(\phi) = 0.5$. Interestingly, the variance reduction for both is still large even for $n = 10$ samples, when the correlation between each Bernoulli pair in the sample cannot be larger than $-1/9$.

4.1. ELBO Based Discrete VAE

Our experimental setup follows the one in Yin & Zhou (2019) and Dong et al. (2020), and all VAE experiments are built on top of the available DisARM code. For this task, we compare ARMS-D and ARMS-N to DisARM/U2G, LOORF, and RELAX. We omit ARM, since it is outperformed by DisARM/U2G, as shown in both Yin et al. (2020) and Dong et al. (2020). All estimators optimize the ELBO of a variational autoencoder (VAE) (Kingma & Welling, 2014), a commonly used method for generative modelling. The comparison is done on three different benchmark datasets: dynamically binarized MNIST, Fashion MNIST, and Omniglot, with each dataset split into the training, validation, and test sets. For each dataset we use either a linear or nonlinear encoder decoder pairs, with $n \in \{4, 6, 8, 10\}$ samples per gradient step. When $n = 2$, both copulas reduce to DisARM/U2G, for which our experiments mirrored the results found in Dong et al. (2020), so we omit this case. All results are reported based on five independent runs.

The implementation details are the following: each VAE contains a stochastic binary layer with 200 units, with two types of encoder-decoder pairs used: linear or nonlinear. The nonlinear network has two hidden layers of 200 units each, using LeakyReLU (Maas et al., 2013) activations with a coefficient of 0.3. Adam (Kingma & Ba, 2015) with a learning rate of $1e^{-4}$ is used to optimize the network parameters, and SGD with learning rate $1e^{-2}$ for the prior distribution logits. The optimization is run for 10^6 steps with mini batches of size 50. For RELAX, the scaling factor is initialized to 1, the temperature to 0.1, and the control variate is a neural network with one hidden layer of 137 units using LeakyReLU activations. The only data preprocessing involves subtracting the global mean of the dataset from each image before it is input to the encoder. All the models were trained on a K40 Nvidia GPU and Intel Xeon E5-2680 processor.

In Fig. 3, we plot the train ELBO, test log likelihood, and variance, for a nonlinear network trained on Fashion MNIST. The variance at each step is an average of the gradient variance of all networks parameters. The same visualization

Table 2. Final train multi sample bound of a Discrete VAE trained with different estimators. Results are averaged over five runs, with the best performing methods in bold.

Dataset	Network	Samples	ARMS	DisARM	VIMCO
Dynamic MNIST	Linear	4	**-112.31 ± 0.05**	-113.26 ± 0.05	-113.07 ± 0.14
		6	**-111.73 ± 0.04**	-112.11 ± 0.03	-112.23 ± 0.27
		8	**-110.62 ± 0.13**	-111.78 ± 0.07	-111.10 ± 0.27
		10	**-110.30 ± 0.07**	-111.08 ± 0.11	-110.69 ± 0.20
	Nonlinr	4	**-98.00 ± 0.02**	-100.45 ± 0.16	-98.07 ± 0.06
		6	**-96.63 ± 0.15**	-99.28 ± 0.11	-96.91 ± 0.12
		8	**-96.17 ± 0.09**	-98.69 ± 0.21	-96.49 ± 0.04
		10	**-95.27 ± 0.10**	-98.62 ± 0.12	-95.77 ± 0.03
Fashion MNIST	Linear	4	**-252.24 ± 0.18**	-254.02 ± 0.05	-252.75 ± 0.06
		6	**-251.40 ± 0.01**	-252.97 ± 0.06	-251.64 ± 0.19
		8	**-250.68 ± 0.15**	-252.57 ± 0.05	-250.78 ± 0.04
		10	**-250.54 ± 0.04**	-251.75 ± 0.21	-250.67 ± 0.12
	Nonlinr	4	**-234.69 ± 0.10**	-236.54 ± 0.06	-234.71 ± 0.20
		6	**-233.68 ± 0.15**	-235.94 ± 0.05	**-233.72 ± 0.45**
		8	**-233.05 ± 0.02**	-235.62 ± 0.16	-233.40 ± 0.08
		10	**-232.74 ± 0.05**	-235.60 ± 0.09	-233.21 ± 0.07
Omniglot	Linear	4	**-119.23 ± 0.04**	**-119.24 ± 0.17**	-120.16 ± 0.06
		6	**-118.50 ± 0.23**	**-118.47 ± 0.12**	-119.07 ± 0.08
		8	**-118.40 ± 0.09**	-118.02 ± 0.08	**-118.49 ± 0.11**
		10	**-118.19 ± 0.03**	-117.70 ± 0.01	-118.35 ± 0.14
	Nonlinr	4	**-111.56 ± 0.10**	-113.24 ± 0.16	-111.72 ± 0.18
		6	**-110.51 ± 0.06**	-112.30 ± 0.05	-110.62 ± 0.08
		8	**-109.76 ± 0.10**	-111.82 ± 0.09	-109.84 ± 0.01
		10	**-109.35 ± 0.01**	-111.33 ± 0.19	-109.53 ± 0.06

for the other datasets and linear networks can be found in Appendix D. In Table 1, we report the average of five final training ELBO results for all estimators on the three datasets, both types of networks, and different number of samples. The corresponding test log likelihood table can be found in Appendix D, with the performance gaps between methods being similar. ARMS consistently outperforms the state of the art and using either the Dirichlet and Gaussian copula appears to have no discernible difference. For linear networks, RELAX is competitive, thought not for nonlinear networks. LOORF generally appears to be a strong overlooked estimator that closely tails the performance of ARMS, indicating that it is beneficial to use all samples when creating a baseline, unlike pair based methods.

4.2. Multi Sample Bound Based Discrete VAE

For the multi sample bound objective, we compare the multi sample version of ARMS using a Dirichlet copula to VIMCO (Mnih & Rezende, 2016), an estimator tailored to this objective, and the multi sample version of DisARM (Dong et al., 2020). To ensure a fair comparison, all three estimators use the same number of function evaluations, which means VIMCO optimizes the n-sample bound, whereas ARMS and DisARM optimize the $\frac{n}{2}$-sample bound. All the experimental details are otherwise identical to Section 4.1. We show training plots over time in Fig. 4. To be able to compare the variance, we average two n-sample bound VIMCO estimates instead, so that the objective optimized is the same. The average of five final train multi sample bounds are shown in Table 2, for each dataset, both types of networks and different number of samples. Similarly to the ELBO case, ARMS outperforms both VIMCO and DisARM, regardless of the dataset, the type of network, or the number of samples.

5. Conclusion

To optimize the parameters of Bernoulli variables in expectation-based objectives, we proposed ARMS, an unbiased, low-variance gradient estimator based on n jointly antithetic samples. Also presented are two ways of generating n jointly antithetic Bernoulli samples, based on a Dirichlet copula and a Gaussian copula, respectively. For $n = 2$ samples, both copulas produce the same antithetic used by DisARM/U2G. ARMS also generalizes LOORF, which can be obtained by using n independent samples. As shown by the experiments, when training a variational autoencoder using either the ELBO or the multi-sample bound, ARMS outperforms both estimators based on multiple independent samples, and the ones based on pairs of antithetic samples. There are several potential avenues of future work. We showcase two different ways of sampling n jointly antithetic Bernoulli variables using either a Dirichlet or Gaussian copula, but there could be better ways, not necessarily based on copulas. Another promising direction is extending ARMS to categorical variables, which has been done for ARM to arrive at ARSM (Yin et al., 2019). Lastly, we optimized the multi sample bound with i.i.d. samples for fair comparison, but using antithetic samples is also a lower bound of the marginal likelihood, albeit a different bound.

Acknowledgements

The authors thank the reviewers for their thoughtful feedback and the authors of DisARM for publicly releasing their code. The authors acknowledge the support of Grants IIS-1812699 and ECCS-1952193 from the U.S. National Science Foundation, the support of a gift fund from ByteDance Inc., and the Texas Advanced Computing Center (TACC) for providing HPC resources that have contributed to the research results reported within this paper.

References

Bengio, Y., Léonard, N., and Courville, A. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *4th International Conference* *on Learning Representations, ICLR*, 2016.

Dong, Z., Mnih, A., and Tucker, G. Disarm: An antithetic gradient estimator for binary latent variables. In *Advances in Neural Information Processing Systems 33*, 2020.

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

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

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. In *6th International Conference on Learning Representations, ICLR*, 2018.

Gu, S., Levine, S., Sutskever, I., and Mnih, A. Muprop: Unbiased backpropagation for stochastic neural networks. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR*, 2016.

Hoernig, S. On the minimum correlation between symmetrically distributed random variables. *Operations Research Letters*, 46(4):469–471, 2018.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net, 2017.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. In *Learning in graphical models*, pp. 105–161. Springer, 1998.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR*, 2014.

Kool, W., van Hoof, H., and Welling, M. Buy 4 REINFORCE samples, get a baseline for free! In *Workshop, Deep Reinforcement Learning Meets Structured Prediction, ICLR*, 2019.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1): 430–474, 2017.

Lorberbom, G., Jaakkola, T. S., Gane, A., and Hazan, T. Direct optimization through arg max for discrete variational auto-encoder. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 6200–6211, 2019.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*. Citeseer, 2013.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR*. OpenReview.net, 2017.

Mnih, A. and Gregor, K. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pp. 1791–1799. PMLR, 2014.

Mnih, A. and Rezende, D. J. Variational inference for monte carlo objectives. In *Proceedings of the 33nd International Conference on Machine Learning, ICML*, volume 48, pp. 2188–2196. JMLR.org, 2016.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *J. Mach. Learn. Res.*, 21:132:1–132:62, 2020.

Ng, K. W., Tian, G.-L., and Tang, M.-L. *Dirichlet and related distributions: Theory, methods and applications*, volume 888. John Wiley & Sons, 2011.

Owen, A. B. Monte carlo theory, methods and examples. ,, 2013.

Paisley, J. W., Blei, D. M., and Jordan, M. I. Variational bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning, ICML*, 2012.

Ranganath, R., Gerrish, S., and Blei, D. Black box variational inference. In *Artificial intelligence and statistics*, pp. 814–822. PMLR, 2014.

Ren, H., Zhao, S., and Ermon, S. Adaptive antithetic sampling for variance reduction. In *International Conference on Machine Learning*, pp. 5420–5428. PMLR, 2019.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Rosca, M., Figurnov, M., Mohamed, S., and Mnih, A. Measure-valued derivatives for approximate bayesian inference. *4th workshop on Bayesian Deep Learning, NeurIPS*, 2019.

Ruiz, F. J. R., Titsias, M. K., and Blei, D. M. The generalized reparameterization gradient. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29*, pp. 460–468, 2016.

Salimans, T. and Knowles, D. A. On using control variates with stochastic approximation for variational bayes and its connection to stochastic linear regression. *arXiv preprint arXiv:1401.1022*, 2014.

Titsias, M. and Lázaro-Gredilla, M. Local expectation gradients for black box variational inference. In *Advances in neural information processing systems*, pp. 2620–2628. Citeseer, 2015.

Trivedi, P. K. and Zimmer, D. M. *Copula modeling: an introduction for practitioners*. ., 2007.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, D., and Sohl-Dickstein, J. REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems 30*, pp. 2627–2636, 2017.

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

Wu, M., Goodman, N., and Ermon, S. Differentiable antithetic sampling for variance reduction in stochastic variational inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2877–2886. PMLR, 2019.

Yin, M. and Zhou, M. ARM: augment-reinforce-merge gradient for stochastic binary networks. In *7th International Conference on Learning Representations, ICLR*, 2019.

Yin, M., Yue, Y., and Zhou, M. ARSM: augment-reinforce-swap-merge estimator for gradient backpropagation through categorical variables. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97, pp. 7095–7104. PMLR, 2019.

Yin, M., Ho, N., Yan, B., Qian, X., and Zhou, M. Probabilistic Best Subset Selection by Gradient-Based Optimization. *arXiv e-prints*, 2020.