Gradient Estimation for Binary Latent Variables via Gradient Variance Clipping

Russell Z. Kunes[1,2,4], **Mingzhang Yin**[4,5], **Max Land**[2], **Doron Haviv**[2], **Dana Pe'er**[2,3], **Simon Tavaré**[1,4]

[1]Department of Statistics, Columbia University
[2]Computational and Systems Biology, Memorial Sloan Kettering Cancer Center
[3]Howard Hughes Medical Institute
[4]Irving Institute of Cancer Dynamics, Columbia University
[5]Warrington College of Business, University of Florida

Abstract

Gradient estimation is often necessary for fitting generative models with discrete latent variables, in contexts such as reinforcement learning and variational autoencoder (VAE) training. The DisARM estimator achieves state of the art gradient variance for Bernoulli latent variable models in many contexts. However, DisARM and other estimators have potentially exploding variance near the boundary of the parameter space, where solutions tend to lie. To ameliorate this issue, we propose a new gradient estimator *bitflip*-1 that has lower variance at the boundaries of the parameter space. As bitflip-1 has complementary properties to existing estimators, we introduce an aggregated estimator, *unbiased gradient variance clipping* (UGC) that uses either a bitflip-1 or a DisARM gradient update for each coordinate. We theoretically prove that UGC has uniformly lower variance than DisARM. Empirically, we observe that UGC achieves the optimal value of the optimization objectives in toy experiments, discrete VAE training, and in a best subset selection problem.

Introduction

Many modern machine learning tasks rely on stochastic gradient estimators, where the estimand is the gradient of an expected value $\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z};\theta)}[f(\mathbf{z})]$ that is intractable to compute (Mohamed et al. 2020). For example, in reinforcement learning (RL) it is often of interest to compute the gradient of an expected reward with respect to the parameters of a distribution over actions, where the reward may be a black box function of discrete states and actions (Li 2017). In variational inference, the objective function is the evidence lower bound, expressed as an expected value of the log joint probability of latent variable and data under a variational distribution (Ranganath, Gerrish, and Blei 2014; Blei, Kucukelbir, and McAuliffe 2017). In many cases $\mathbf{z}$ is discrete; for example, in the design of biological sequences (Brookes, Park, and Listgarten 2019) or in models with spike and slab Bayesian priors (Moran et al. 2021).

When the latent variables $\mathbf{z}$ are discrete and high dimensional, there are several challenges in optimizing the mean-valued objective with respect to the distributional parameters θ. First, computing the exact expectation often requires an intractable number function evaluations due to an exponential number of summation terms (Titsias and Lázaro-Gredilla 2015). Moreover, the derivative of the function itself with respect to discrete variables is not well defined so the chain rule-based reparametrization trick (Kingma and Welling 2013) cannot be used.

A number of methods for estimating the gradient of expected values with respect to discrete random variables have been devised (Dong, Mnih, and Tucker 2020; Dimitriev and Zhou 2021; Dong, Mnih, and Tucker 2021; Yin, Yue, and Zhou 2019; Titsias and Lázaro-Gredilla 2015; Tucker et al. 2017; Grathwohl et al. 2017; Titsias and Shi 2022). A central role shared among the designs of useful gradient estimation is to control the bias and variance of the estimates. One line of research reduces the gradient variance in a trade-off of introducing bias. Widely used methods include continuous relaxations such as the Gumbel-softmax trick (Jang, Gu, and Poole 2016; Maddison, Mnih, and Teh 2016; Paulus et al. 2020), and the straight through gradient estimator (Bengio, Léonard, and Courville 2013; Yin et al. 2019), which have been successfully applied for learning latent representations of images (Razavi, Van den Oord, and Vinyals 2019) and text (Tran et al. 2019). Another line of work considers unbiased estimates that offer guarantees of convergence under conditions on the learning rate sequence (Ranganath, Gerrish, and Blei 2014; Robbins and Monro 1951). Some methods construct control variate baselines by continuous relaxation of the discrete distributions (Tucker et al. 2017; Grathwohl et al. 2017), by first-order Taylor expansions (Gu et al. 2015; Titsias and Shi 2022), or by Stein operators (Shi et al. 2022). Other methods reduce the estimator variance by applying antithetic sampling and coupled sampling (Yin and Zhou 2019; Dong, Mnih, and Tucker 2020; Dimitriev and Zhou 2021; Yin, Yue, and Zhou 2019; Yin et al. 2020; Kool, van Hoof, and Welling 2019). Our work proceeds in this direction of designing unbiased and low-variance gradient estimators for discrete optimization.

In this work, we notice that in the context of Bernoulli discrete latent variables, a number of existing unbiased methods have unfavorably high variance at the boundary of the parameter space (namely, near 0 and near 1) due to reliance on an importance weight that is necessary in order to maintain unbiasedness. To address this downside of existing estimators, we introduce an *unbiased gradient variance clip-*

ping (UGC) estimator that sidesteps this issue by conditionally using one of two types of gradient estimators. For a given coordinate, when values of the probability parameter θ are near $\frac{1}{2}$, UGC updates the parameter values in the direction of the DisARM gradient estimate. On the other hand, when values of the probability θ become close to the boundary, UGC transitions to using a novel gradient estimator, *bitflip*-1, that has *complementary* properties to existing estimators that require $O(1)$ function evaluations. Namely, rather than considering coordinate-wise independent samples of $\mathbf{z}$, bitflip-1 updates only a single coordinate of the parameter vector at a time, while holding other coordinates fixed to minimize variance. The result is that bitflip-1 has variance linear in the latent dimension K but without explicit dependence on the latent Bernoulli parameters. Our proposed estimator, UGC, has guaranteed uniformly lower variance than DisARM and is robust across practical problems where either DisARM or bitflip-1 alone may fail.

Background

Consider the problem of estimating the gradient:

$$\nabla_\theta \mathbb{E}_{p(\mathbf{z};\theta)}[f(\mathbf{z})] \tag{1}$$

where $\mathbf{z} = (z_1, \dots, z_K)$, $z_i \sim \text{Bernoulli}(\theta_i)$, $\theta_i \in [0,1]$, independently, $p(\mathbf{z};\theta) = \prod_{i=1}^K \text{Bernoulli}(\theta_i)$, and f is a potentially complicated and nonlinear function with domain on the lattice. This problem arises in discrete latent variable modeling and RL. To compute the exact gradient, we can replace the expectation in Equation (1) with the summation over all possible values of $\mathbf{z}$ which has 2^K summation terms. Computing the exact gradient thus requires an exponential number of evaluations which is infeasible to compute per iteration of gradient descent in high dimensional problems. Specifically we focus on the context of Bernoulli VAEs where $p_\lambda(\mathbf{x}_i|\mathbf{z}_i)$ is parameterized by a neural network, while $\mathbf{z} \in \{0,1\}^K$, and we fit an encoder network $q_\theta(\mathbf{z}|\mathbf{x})$ to maximize the evidence lower bound (ELBO):

$$\mathcal{L}(\theta,\lambda) = \mathbb{E}_q\big\{\log p_\lambda(\mathbf{x}|\mathbf{z}) + \log p(\mathbf{z}) - \log q_\theta(\mathbf{z}|\mathbf{x})\big\}.$$

The exact gradient of the objective function with respect to θ involves 2^k terms in general. As a result, we are forced to use a stochastic estimate of the gradient. Two methods are commonly applied for this task; score function gradient estimators (Ranganath, Gerrish, and Blei 2014), and the reparameterization trick (Kingma and Welling 2013).

The Score Function Gradient Estimator

The score function gradient estimator (also called Reinforce) is $\hat{g} := f(\mathbf{z})\nabla \log p(\mathbf{z};\theta)$. Its unbiasedness follows from the following computation, assuming the conditions of the dominated convergence theorem holds for f:

$$\begin{aligned}\nabla_\theta \mathbb{E}_{p(\mathbf{z};\theta)}[f(\mathbf{z})] &= \int f(\mathbf{z})\nabla_\theta p(\mathbf{z};\theta) d\mu(\mathbf{z}) \\ &= \int f(\mathbf{z})\nabla_\theta\big(\log p(\mathbf{z};\theta)\big)p(\mathbf{z};\theta) d\mu(\mathbf{z}) \\ &= \mathbb{E}\Big\{f(\mathbf{z})\nabla_\theta \log p(\mathbf{z};\theta)\Big\}.\end{aligned}$$

The estimator is generally applicable but in many cases has too high variance to be useful in practice. However, this estimator has proven useful in many situations with the inclusion of variance reduction techniques such as control variates (Ranganath, Gerrish, and Blei 2014; Tucker et al. 2017; Grathwohl et al. 2017).

Reparameterization Gradient Estimator

Another gradient estimator is by the reparameterization trick (Kingma and Welling 2013), which requires $\mathbf{z}$ to be expressable as a differentiable transformation of exogenous noise $\mathbf{z} = T(\theta,\epsilon)$, where $\epsilon \sim g(\cdot)$ is free of θ. The reparameterization gradient estimator has low variance but is less generally applicable (Naesseth et al. 2017). In the context of discrete random variables, it is necessary to apply a continuous relaxation to $\mathbf{z}$ and extend the domain of f to account for continuous input.

The ARM and DisARM Gradient Estimators

ARM (Yin and Zhou 2019) and DisARM (also called U2G) (Dong, Mnih, and Tucker 2020; Yin et al. 2020) are two methods for reducing the variance of the score function gradient estimator estimator for Bernoulli latent variables. As notation, α_θ will refer to the logits of the Bernoulli parameter, i.e. $\alpha_\theta := \log \frac{\theta}{1-\theta}$. The ARM estimator is motivated by a reparameterization. In one dimension, letting $b \sim \text{Logistic}(\alpha_\theta, 1)$ and $z = \mathbf{1}_{b>0}$; the desired gradient $\nabla_\theta \mathbb{E}\big[f(z)\big] = \nabla_\theta \mathbb{E}_b\big[f(\mathbf{1}_{b>0})\big] = \mathbb{E}_b\big[f(\mathbf{1}_{b>0})\nabla_\theta \log q_\theta(b)\big]$ where q_θ is the likelihood of the Logistic distribution with parameter α_θ. Logistic random variables with identical marginal distributions can be sampled by letting $\epsilon \sim \text{Logistic}(0,1)$ and setting $b = \epsilon + \alpha_\theta$ and $\tilde{b} = -\epsilon + \alpha_\theta$. This antithetic sampling produces an estimator with reduced variance:

$$\begin{aligned}\hat{g}_{\text{ARM}} &:= \frac{1}{2}\big(f(\mathbf{1}_{b>0})\nabla_\theta \log q_\theta(b) + f(\mathbf{1}_{\tilde{b}>0})\nabla_\theta \log q_\theta(\tilde{b})\big) \\ &= \frac{1}{2}\big(f(\mathbf{1}_{b>0}) - f(\mathbf{1}_{\tilde{b}>0})\big)\nabla_\theta \log q_\theta(b) \\ &= \big(f(z) - f(\tilde{z})\big)(u - \frac{1}{2})\nabla_\theta \alpha_\theta.\end{aligned}$$

Here, $\sigma(\cdot)$ is the sigmoid operation and, u is a uniform random variable defined by $\sigma(b - \alpha_\theta)$, and $z = \mathbf{1}_{1-u<\theta}$, $\tilde{z} = \mathbf{1}_{u<\theta}$. The procedure naturally extends to the multi-dimensional case giving the estimator $\hat{g}_{\text{ARM}} = \big((f(\mathbf{z}) - f(\tilde{\mathbf{z}}))(\mathbf{u} - \frac{1}{2})\big)\nabla_\theta \alpha_\theta$

The DisARM estimator takes a conditional expectation of the ARM estimator, conditioning on the values $(z,\tilde{z})$:

$$\begin{aligned}\hat{g}_{\text{DisARM}} &= \mathbb{E}_{p(b|z,\tilde{z})}\big[\hat{g}_{ARM}\big] \\ &= \frac{1}{2}\big(f(z) - f(\tilde{z})\big)(-1)^{\tilde{z}}\mathbf{1}_{z\neq\tilde{z}}\sigma(|\alpha_\theta|)\nabla_\theta \alpha_\theta\end{aligned}$$

This extends to the multi-dimensional case in an analogous way, requiring a constant number of function evaluations, and also further reduces the variance of ARM estimator by nature of Rao-Blackwellization.

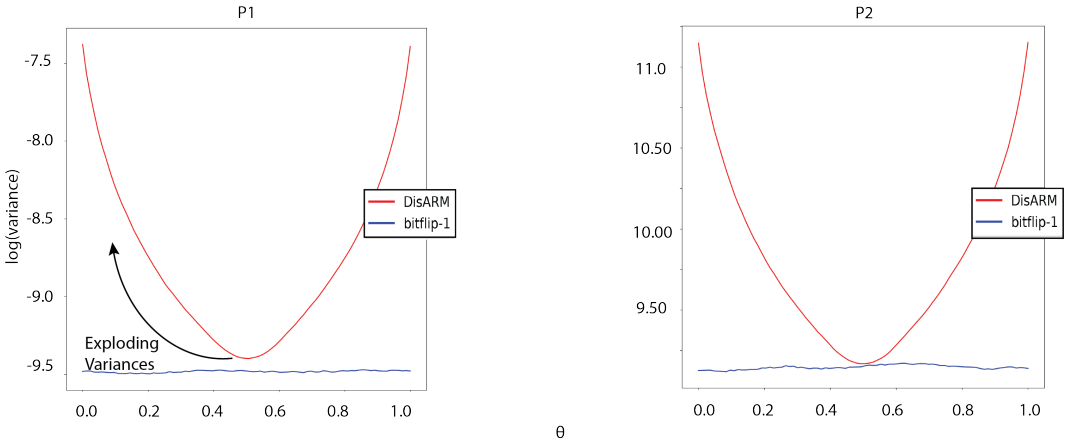

Figure 1: DisARM gradient variances potentially explode at the boundary of parameter space. *Left:* variance curves for P1, $f(z) = \sum_{i=1}^{K}(z_i - t)^2$ *Right:* variance curves for P2 $f(z) - (\sum_{i=1}^{K} z_i - t)^2$. In both cases $K = 20$, with $\theta_1 = \dots \theta_{19} = 0.5$ and θ_{20} varying on the x-axis.

Variance Properties of DisARM at the Boundary

Though DisARM is competitive compared to existing methods of gradient estimation, it has unfavorable variance at the boundaries of the parameter space. Reparameterizing the DisARM estimator in terms of probability θ gives:

$$\hat{g}_{DisARM,j} = \frac{1}{2}\big(f(\mathbf{z}) - f(\tilde{\mathbf{z}})\big)\frac{1}{\min(\theta_j, 1-\theta_j)}\mathbf{1}_{z_j \neq \tilde{z}_j}(-1)^{\tilde{z}_j}$$

where $\tilde{\mathbf{z}}_j$ satisfies $\mathbb{P}[z_j = 0, \tilde{z}_j = 1] = \mathbb{P}[z_j = 1, \tilde{z}_j = 0] = \min(\theta_j, 1-\theta_j)$, and $\mathbb{P}[z_j = \tilde{z}_j] = |1 - 2\theta_j|$.

We analyze the variance as the difference $\mathbb{E}[(\hat{g}_{DisARM,j})^2] - \mathbb{E}[(\hat{g}_{DisARM,j})]^2$. Without loss of generality, considering the case where $\theta_j < \frac{1}{2}$, the expected square $\mathbb{E}[(\hat{g}_{DisARM,j})^2]$ is:

$$\mathbb{E}\Big[\tfrac{1}{4}\big(f(\mathbf{z}) - f(\tilde{\mathbf{z}})\big)^2 \tfrac{1}{\theta_j^2}\mathbf{1}_{z_j \neq \tilde{z}_j}\Big] = \tfrac{1}{2\theta_j}\mathbb{E}\Big[\big(f(\mathbf{z}_1^{(j)}) - f(\tilde{\mathbf{z}}_0^{(j)})\big)^2\Big] \quad (2)$$

where $\mathbf{z}_1^{(j)}$ and $\tilde{\mathbf{z}}_0^{(j)}$ are defined by hard-coding the j'th element of $\mathbf{z}$ as 1 and 0 respectively and sampling remaining shared elements from their respective distributions. For unbiased gradient estimators, the term $\mathbb{E}[\hat{g}]^2 = \big(\mathbb{E}\big[f(\mathbf{z})|z_j = 1\big] - \mathbb{E}\big[f(\mathbf{z})|z_j = 0\big]\big)^2$ are the same. Therefore, Equation (2) suggests that DisARM suffers from large variances when $\theta_j \approx 1$ or $\theta_j \approx 0$ (see Figure 1). Another estimator competitive with DisARM is Reinforce-loo (Kool, van Hoof, and Welling 2019), expressed as $\frac{1}{2\theta_j(1-\theta_j)}\Big((f(z_{1,j}) - f(z_{2,j}))(z_{1,j} - \theta_j) + (f(z_{2,j}) - f(z_{1,j}))(z_{2,j} - \theta_j)\Big)$ where now $\mathbf{z}_1$ and $\mathbf{z}_2$ are sampled independently. Again, the presence of the $\frac{1}{2\theta_j(1-\theta_j)}$ weight induces high variances at the boundary. This motivates us to consider estimators with bounded variance at the boundary. However, we note that this problem might be ameliorated by parameterizing θ by logits $\theta = \frac{e^\phi}{1+e^\phi}$ with $\nabla_\phi \theta = \theta(1-\theta)$ as is commonly done in practice. Though this parameterization avoids explicit enforcement of the [0,1] constraint during optimization, solutions at the boundary cannot be reached exactly. In our simulations, we have observed slower convergence of this approach relative to projected gradient descent in a number of problem settings.

Unbiased Monte Carlo Estimate of the Gradient via Bit Flips

Note that the exact gradient is given by $E[f(\mathbf{z})|z_j = 1] - E[f(\mathbf{z})|z_j = 0]$. This suggests a simple estimation scheme: sample $\mathbf{z} \sim p_\theta$, and let $\tilde{\mathbf{z}}^{(j)}$ be the vector where the $j'th$ element of $\mathbf{z}$ is flipped. The single sample estimate is then $(-1)^{z_j}(f(\tilde{\mathbf{z}}^{(j)}) - f(\mathbf{z}))$. We can apply this to all elements of the gradient for a single sample $\mathbf{z}$ and retain the unbiasedness property. Since this requires $O(K)$ function evaluations with K as the dimension of variable $\mathbf{z}$, which may be too expensive in many settings, we define and analyze *bitflip*-1 as the randomized estimator given by sampling $\mathbf{z} \sim p_\theta$, sampling a random coordinate $j \sim \text{Categorical}(1, \cdots, K)$, and setting the estimate $\hat{g}_{\text{bitflip-1},j} := K * (-1)^{z_j}(f(\tilde{\mathbf{z}}^{(j)}) - f(\mathbf{z}))$, $\hat{g}_{\text{bitflip-1},-j} := 0$. Interestingly, the only dependence of $\hat{g}$ on θ is through the sampling procedure. We also point out that though the DisARM estimator is shown to be uniformly minimum variance among estimators that employ linear combinations of antithetic sampled Bernoulli variables ((Yin et al. 2020), Proposition 2), bitflip-1 cannot be expressed in this manner (and moreover, the coordinates are no longer independent) and so is not dominated. In fact, bitflip-1 is lower variance than DisARM whenever $\frac{1}{2\min(\theta, 1-\theta)} > K$.

The expression of the gradient also suggests an interpretation of the DisARM estimator: that is, DisARM estimates $\mathbb{E}\big[f(\mathbf{z})|z_j = 1\big] - \mathbb{E}\big[f(\mathbf{z})|z_j = 0\big]$ with two samples and a multiplicative weight that ensures the unbiasedness property. Each of the two samples has j'th coordinate that is marginally Bernoulli(θ_j), with a joint distribution between

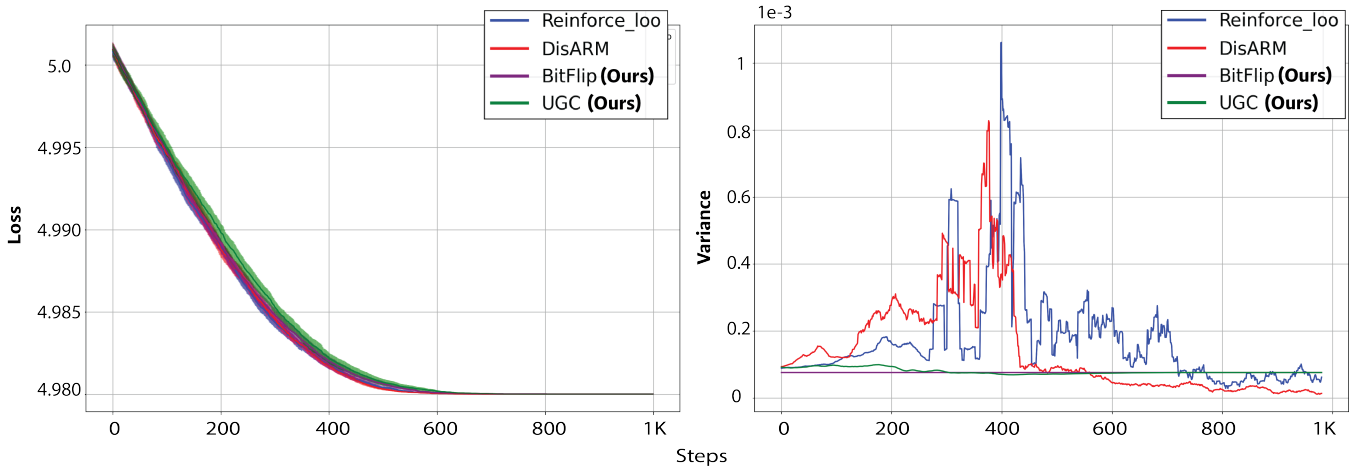

Figure 2: Performance on problem (P1) with $t = 0.499$, $K = 20$. All methods tested converge to the optimal solution in this setting, though DisARM and Reinforce-loo suffer higher gradient variances. *Left*: Training loss curves averaged over 10 trials for the problem (P1) with error bars $\pm\sigma/\sqrt{10}$; *Right*: Gradient variances averaged over 10 trials.

the two samples that gives us maximal amount of information about the gradient. If we are limited to two function evaluations, this suggests considering estimates of the form $f(\mathbf{z}) - f(\tilde{\mathbf{z}})$ for some $(\mathbf{z}, \tilde{\mathbf{z}})$ with the marginal distribution $z_j \sim \text{Bernoulli}(\theta_j)$. However, with just two function evaluations it makes sense to disregard terms where $z_j = \tilde{z}_j$ as it is not clear how to construct an estimator of $\mathbb{E}[f(\mathbf{z})|z_j = 1] - \mathbb{E}[f(\mathbf{z})|z_j = 0]$ in these cases.

All of this suggests considering estimators of the form:

$$\hat{g}_j := (-1)^{\tilde{z}_j}[f(\mathbf{z}) - f(\tilde{\mathbf{z}})] \times \frac{1}{p[z_j = 1, \tilde{z}_j = 0] + p[z_j = 0, \tilde{z}_j = 1]} \mathbf{1}_{z_j \neq \tilde{z}_j} \quad (3)$$

where the multiplicative term $\frac{1}{p[z_j=1,\tilde{z}_j=0]+p[z_j=0,\tilde{z}_j=1]}$ in Equation (3) ensures unbiasedness. This recovers DisARM when $p[z_j = 1, \tilde{z}_j = 0] = p[z_j = 0, \tilde{z}_j = 1] = \min(\theta, 1-\theta)$ and Reinforce-loo when $\mathbf{z} \perp\!\!\!\perp \tilde{\mathbf{z}}$. An important fact about DisARM is that it maximizes $p[z_j = 0, \tilde{z}_j = 1] + p[z_j = 1, \tilde{z}_j = 0]$, i.e. the coupling given by $p[z_j = 0, \tilde{z}_j = 1] = \min(\theta_j, 1-\theta_j)$ has the highest probability of differing values between $\tilde{z}_j$ and z_j subject to the marginal constraint that each random variable is $\text{Bernoulli}(\theta_j)$. This is due to the fact that $p[z_j = 0, \tilde{z}_j = 1] \leq \theta$ and $p[z_j = 0, \tilde{z}_j = 1] \leq 1 - \theta$ following the two constraints given by $p[z_j = 0] = p[z_j = 0, \tilde{z}_j = 1] + p[z_j = 0, \tilde{z}_j = 0]$ and $p[\tilde{z}_j = 1] = p[z_j = 1, \tilde{z}_j = 1] + p[z_j = 0, \tilde{z}_j = 1]$.

However, it is clear that the minimum variance coupling depends on f as we have:

$$E[\hat{g}^2] = \Big(\frac{1}{p[z_j = 0, \tilde{z}_j = 1] + p[z_j = 1, \tilde{z}_j = 0]}\Big) \quad (4)$$

$$\times\, \mathbb{E}\big((f(\mathbf{z}) - f(\tilde{\mathbf{z}}))^2 | z_j = 1, \tilde{z}_j = 0\big). \quad (5)$$

When f is continuous (in the sense that $|f(\mathbf{z}) - f(\tilde{\mathbf{z}})|$ is related to $d(\mathbf{z}, \tilde{\mathbf{z}})$ for a distance metric d) there is a trade-off between minimizing the first term in Equation (4) and second term in Equation (5). As $p[z_j = 0, \tilde{z}_j = 1]$ (and $p[z_j = 1, \tilde{z}_j = 0]$) increase, the expected function differences in Equation (5) are likely to be large. If f is such that the term in Equation (5) tends to be large, independently sampled $\mathbf{z}$ and $\tilde{\mathbf{z}}$ may even be lower variance than antithetic samples (Dong, Mnih, and Tucker 2020). DisARM updates the largest number of terms possible by maximizing the probabilities $p[z_j = 0, \tilde{z}_j = 1]$ and $p[z_j = 1, \tilde{z}_j = 0]$ and hence minimizes term in Equation (4), but insodoing may incur high variance through large values of Equation (5).

Variance Properties of *bitflip*-1

We assume the following natural continuity property of the function f:

Assumption 1 *Given four binary vectors $z, w, \tilde{z}, \tilde{w} \in \{0,1\}^K$, if $\{j : \tilde{z}_j \neq z_j\} \supset \{j : \tilde{w}_j \neq w_j\}$ and $w_i = z_i$ for all i such that $w_i = \tilde{w}_i$ and $z_i = \tilde{z}_j$, then $|f(w) - f(\tilde{w})| \leq |f(z) - f(\tilde{z})|$.*

In other words, given two binary strings we cannot make their function evaluations closer by introducing additional coordinates where they differ. Since each estimator considered is unbiased, it suffices to consider $\mathbb{E}[\hat{g}^2]$ for each gradient estimator $\hat{g}$.

Proposition 1 *(Variance of bitflip-1) Let $\hat{g}$ be an estimator in the family of estimators given by Equation* (3)*, which includes DisARM and Reinforce-loo gradient estimators. If Assumption 1 holds and if $\frac{1}{2\min(\theta_j, 1-\theta_j)} \geq K$:*

$$Var(\hat{g}_{\textit{bitflip-1}}) \leq Var(\hat{g})$$

We present an expanded version of this proposition and proof in the appendix. We also note that when $f(z)$ is separable, bitflip-1 has uniformly lower variance than DisARM in the following sense:

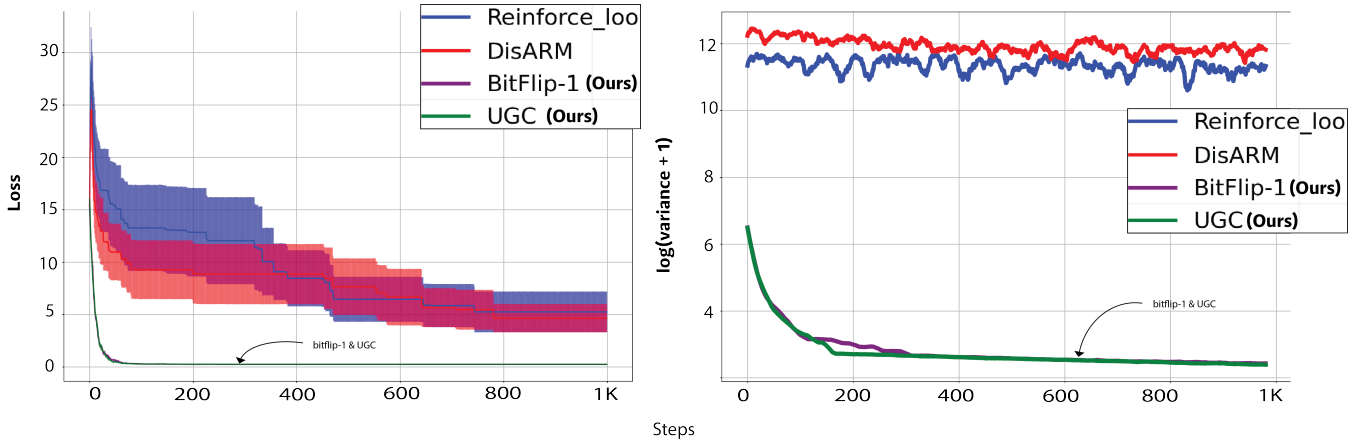

Figure 3: Performance on problem (P2) with $t = 0.499$, $K = 20$. DisARM and Reinforce-loo frequently fail to converge, while experiencing high gradient variances. *Left*: Training loss curves averaged over 10 trials for the problem (P2) with errors bars $\pm\sigma/\sqrt{10}$; *Right*: Average gradient variances over 10 trials.

Proposition 2 *Consider a member of the family of estimators given by Equation* (3)*, which includes DisARM and Reinforce-loo and denote this estimator* $\hat{g}$*. If* $f(\mathbf{z}) = \sum_{i=1}^{K} h(z_i)$*, then:*

$$\min_{\theta_1,\ldots,\theta_K} \max_{j=1,\ldots,K} Var(\hat{g}_j) \geq \max_{\theta_1,\ldots,\theta_K} \max_{j=1,\ldots,K} Var(\hat{g}_{bitflip,j})$$

Unbiased Gradient Variance Clipping

Though bitflip-1 has bounded variance for a given latent variable dimension K, it's variance grows linearly with K. Meanwhile, DisARM has variance growing with $\frac{1}{\min(\theta,1-\theta)}$ despite only depending on K implicitly through the function f. Motivated by these complementary behaviors and fact that θ_j and K are available, we can construct an estimator that dominates DisARM as follows.

$$\hat{g}_{UGC,j} = \begin{cases} \hat{g}_{bitflip-1,j} & \text{if } \min(\theta_j, 1-\theta_j) < \tau \\ \hat{g}_{DisARM,j} & \text{if } \min(\theta_j, 1-\theta_j) \geq \tau \end{cases} \quad (6)$$

where τ is a tuning parameter of the estimator. We denote this estimator by *unbiased gradient variance clipping* (UGC) as it replaces potentially high variance gradient estimates with bounded variance estimates without breaking unbiasedness of the estimate. A standard choice of τ is $\frac{1}{2K}$, motivated by the following result:

Proposition 3 *(Variance of UGC) Under assumption 1, when* $\tau \leq \frac{1}{2K}$, $Var(\hat{g}_{UGC,j}) \leq Var(\hat{g}_{DisARM,j})$ *for all coordinates* $j \in \{1, \ldots, K\}$

We find that UGC achieves better performance than bitflip-1 and DisARM on a number of tasks.

Experiments

Toy Experiments

In Tucker et al. (2017), the authors optimize the objective $\mathbb{E}_\theta[(z-t)^2]$ where z is a single Bernoulli random variable with a parameter θ and t is set to either 0.49 or 0.499. The optimizer of this problem is $\theta = 0$, with values of t closer to 0.5 representing harder problems. As bitflip-1 computes the exact gradient for univariate latent variable z, we extend this problem to two multivariate problems:

$$(P1) : \min \mathbb{E}[\textstyle\sum_{k=1}^{K}(z_k - t)^2]; \quad (P2) : \min \mathbb{E}[(\textstyle\sum_{k=1}^{K} z_k - t)^2]$$

In problem (P1), due to the separability of the objective, bitflip-1 computes the exact gradient multiplied by K and updates a random component (Figure 2). Problem (P2) is harder in the sense that it contains many interaction terms and the exact gradient is expensive to compute for moderate K. Figure 3 shows results for $K = 20$ and $t = 0.499$ (with other results in the appendix). Notably, for $(P2)$ both the Reinforce-loo baseline and DisARM fail to converge to the optimum. This occurs due to the fact that these gradients can often be in the wrong direction due to noise and then are unable to estimate high magnitude gradients at $\theta = 1$. When $\theta \approx 1$, UGC will switch to using bitflip gradients and can move away from the suboptimal $\theta = 1$.

L_0 Best Subset Regression

Fitting linear regression with a sparsity penalty has become a ubiquitous task across many domains (Tibshirani 2011). Such regression estimators frequently are computed by minimizing squared error subject to a constrain on the L_1 norm of the regression coefficients β. The non-convex problem of optimizing subject to constraint on the L_0 norm has received less attention due to computational challenges but is addressed in Yin et al. (2020). Specifically, they consider the following estimator of β under the linear regression assumptions $y \sim \mathcal{N}(x^\top \beta, \sigma^2)$:

$$\min_{\beta} \frac{1}{n} ||\mathbf{y} - \mathbf{X}\beta||_2^2 + \lambda ||\beta||_0$$

This optimization problem penalizes the cardinality of the coefficient vector β, rather than its L_1 norm and so more directly encodes the assumption that the true coefficient vector is sparse. In Yin et al. (2020), the authors show that this

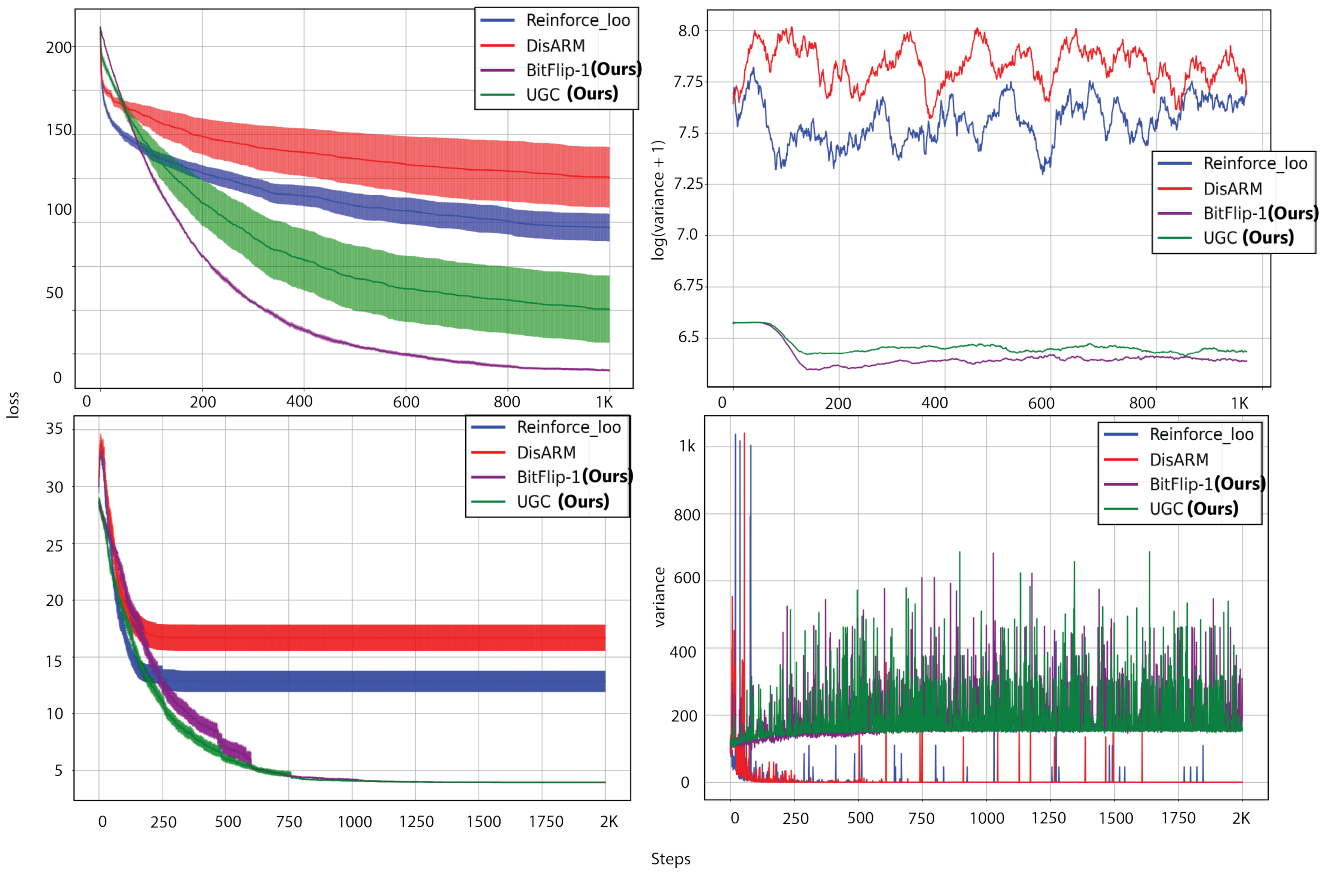

Figure 4: Performance on the gradient based subset optimization problem for linear regression. $p = 200$, $n = 60$, $\Sigma = I$, $|S| = 3$, *top*: SNR $= \beta^\top \beta / \sigma^2 = 3.8125$, parameterization by $\phi = \log(\theta/(1-\theta))$*bottom*: SNR $= \beta^\top \beta / \sigma^2 = 1.694$. Parameterization by θ, with projected gradient descent onto $[0, 1]$. *Left*: Training loss curves for the best subset optimization problem, averaged over 10 random samples of the data with error bars $\pm\sigma/\sqrt{10}$. *Right*: Average gradient variances across 10 random samples of the data. Though bitflip-1 and UGC are higher variance in the second example, we note that this is because the gradient variance is not well defined when $\theta \in \{0, 1\}$ and so Monte Carlo variances are artificially close to 1. DisARM and Reinforce-loo do not converge to the correct part of parameter space.

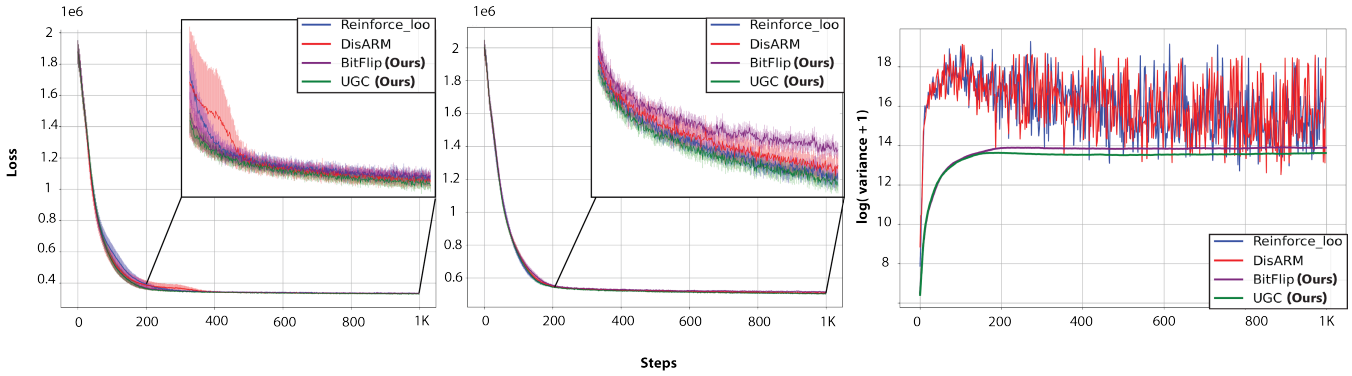

Figure 5: Performance on the gaussian mixture model problem fit via discrete VAEs. Cluster means are sampled $N(0, 8^2)$ per simulation. *Right*: Training loss curves for the gaussian mixture model problem ($\sigma = 2.0$), averaged over 10 random samples of the data with error bars $\pm\sigma/\sqrt{10}$. *Middle*:Training loss curves for $\sigma = 4.0$ *Right*: Average gradient variances ($\sigma = 4.0$) across 10 random samples of the data. Through the experiment, the true number of clusters is 6, the number of features is 20, and the hidden dimension is 10.

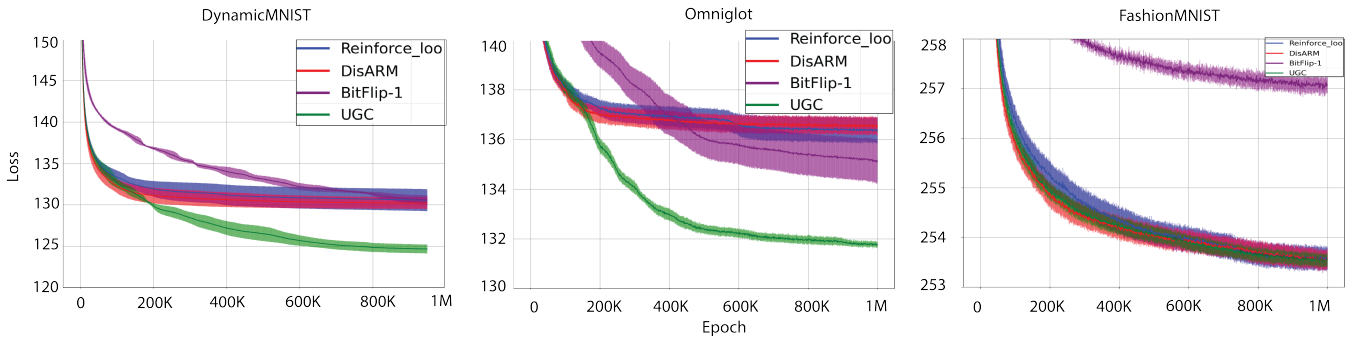

Figure 6: Performance on the binarized discrete VAE fit to DynamicMNIST, FashionMNIST and Omniglot datasets over 5 random seeds, with error bars given by $\pm\sigma/\sqrt{5}$. The binary latent variable is 30 dimensional with 1-layer encoder and decoder networks. UGC achieves better convergence than alternative estimators.

	Gradient estimator			
SNR	bitflip-1	UGC	DisARM	Rein.-loo
15.25	**0.0 (0.0)**	**0.0 (0.0)**	0.05 (0.02)	0.04 (0.01)
3.81	**0.0 (0.0)**	**0.0 (0.0)**	0.06 (0.03)	0.04 (0.01)
1.69	**0.01 (0.01)**	**0.01 (0.01)**	0.06 (0.03)	0.05 (0.02)
0.95	0.04 (0.01)	**0.03 (0.01)**	0.06 (0.02)	0.05 (0.01)

Table 1: False Positive Rate (FPR) of best subset selection.

	Gradient estimator			
SNR	bitflip-1	UGC	DisARM	Rein.-loo
15.25	**0.96 (0.1)**	**0.96 (0.1)**	0.56 (0.26)	0.6 (0.36)
3.81	**1.0 (0.0)**	**1.0 (0.0)**	0.66 (0.26)	0.53 (0.16)
1.69	0.83 (0.27)	**0.87 (0.22)**	0.43 (0.26)	0.50 (0.31)
0.95	0.43 (0.30)	**0.67 (0.21)**	0.40 (0.29)	0.43 (0.21)

Table 2: True Positive Rate (TPR) of best subset selection.

problem can be approximately solved with the gradient estimator DisARM via the equivalent optimization problem:

$$\min_{\theta} \mathbb{E}_{z\sim\theta}\big[\min_{\beta} \frac{1}{n}||\mathbf{y} - \mathbf{X}(\mathbf{z}\odot\beta)||_2^2 + \lambda||\mathbf{z}||_0\big],$$

where $\odot$ means Hadamard product. The solutions of the second problem are guaranteed to occur at the boundaries of the parameter space and coincide with the solution of the original regression problem. As the solutions occur at the boundary, this scenario is one where bitflip-1 and UGC perform well, shown in Figure 4. Specifically, in low signal-to-noise ratio (SNR) settings in Tables 1 and 2, other gradient estimators cannot reliably recover the correct solution.

Gaussian Mixture Model

We investigate the capability of a discrete VAE fit with each gradient estimator to identify Gaussian mixtures. Specifically we generate samples from a $20-$dimensional Gaussian mixture model distribution with 6 components by first sampling component means from a $N(0, 8^2)$ distribution, then sampling data conditional on component means from a Normal distribution with variance σ^2, with σ^2 being the parameter controlling the signal to noise ratio. Though each esimator achieves comparable convergence rate for multiple signal to noise ratios, bitflip-1 and UGC have markedly lower variance throughout training as shown in Figure 5.

Discrete Variational Autoencoder Training

We replicate the discrete VAE architecture and experimental setup on binarized DynamicMNIST, Omniglot and FashionMNIST datasets (Yin and Zhou 2019; Dong, Mnih, and Tucker 2020). Interestingly, we note that DisARM exhibits fast convergence early on in training but later in training is unable to make progress, while bitflip-1 proceeds slowly during initial training but reaches a better final optimum. This observation validates our analysis that DisARM might encounter high variance at the boundary of parameter space where the algorithm converges to. As shown in Figure 6, UGC achieves the best of both worlds: after switching to bitflip-1 derived gradients, it reaches a better solution than both methods.

Discussion

We have presented a method for producing low variance gradient estimates at the boundary of the parameter space for Bernoulli latent variable models. Noticing that existing methods suffer high variance gradients near the boundary of $[0, 1]$, we introduce a combined estimator, UGC, that uses DisARM gradients near the middle of $[0, 1]$ and bitflip-1 gradients near the boundary. We expect our approach to be useful for fitting various kinds of sparse latent variable models; for example, for fitting variational autoencoders with spike-and-slab priors via mean field variational inference (Moran et al. 2021). Our empirical results hopefully open the door to a number of theoretical questions. Future work may define classes of discrete functions and estimators where we can find optimal gradient estimators subject to constraint on the number of function evaluations.

References

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

Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518): 859–877.

Brookes, D.; Park, H.; and Listgarten, J. 2019. Conditioning by adaptive sampling for robust design. In *International conference on machine learning*, 773–782. PMLR.

Dimitriev, A.; and Zhou, M. 2021. Arms: Antithetic-reinforce-multi-sample gradient for binary variables. In *International Conference on Machine Learning*, 2717–2727. PMLR.

Dong, Z.; Mnih, A.; and Tucker, G. 2020. DisARM: An antithetic gradient estimator for binary latent variables. *Advances in neural information processing systems*, 33: 18637–18647.

Dong, Z.; Mnih, A.; and Tucker, G. 2021. Coupled gradient estimators for discrete latent variables. *Advances in Neural Information Processing Systems*, 34: 24498–24508.

Grathwohl, W.; Choi, D.; Wu, Y.; Roeder, G.; and Duvenaud, D. 2017. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*.

Gu, S.; Levine, S.; Sutskever, I.; and Mnih, A. 2015. Muprop: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*.

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

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

Kool, W.; van Hoof, H.; and Welling, M. 2019. Buy 4 reinforce samples, get a baseline for free! *ICLR 2019 workshop: Deep RL Meets Structured Prediction*.

Li, Y. 2017. Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Mohamed, S.; Rosca, M.; Figurnov, M.; and Mnih, A. 2020. Monte Carlo Gradient Estimation in Machine Learning. *Journal of machine learning research: JMLR*, 21(132): 1–62.

Moran, G. E.; Sridhar, D.; Wang, Y.; and Blei, D. M. 2021. Identifiable variational autoencoders via sparse decoding. *arXiv preprint arXiv:2110.10804*.

Naesseth, C.; Ruiz, F.; Linderman, S.; and Blei, D. 2017. Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics*, 489–498. PMLR.

Paulus, M.; Choi, D.; Tarlow, D.; Krause, A.; and Maddison, C. J. 2020. Gradient estimation with stochastic softmax tricks. *Advances in Neural Information Processing Systems*, 33: 5691–5704.

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

Razavi, A.; Van den Oord, A.; and Vinyals, O. 2019. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32.

Robbins, H.; and Monro, S. 1951. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Shi, J.; Zhou, Y.; Hwang, J.; Titsias, M. K.; and Mackey, L. 2022. Gradient Estimation with Discrete Stein Operators. *arXiv:2202.09497*.

Tibshirani, R. 2011. Regression shrinkage and selection via the lasso: a retrospective. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3): 273–282.

Titsias, M.; and Lázaro-Gredilla, M. 2015. Local expectation gradients for black box variational inference. *Advances in neural information processing systems*, 28.

Titsias, M.; and Shi, J. 2022. Double Control Variates for Gradient Estimation in Discrete Latent Variable Models. In *International Conference on Artificial Intelligence and Statistics*, 6134–6151. PMLR.

Tran, D.; Vafa, K.; Agrawal, K.; Dinh, L.; and Poole, B. 2019. Discrete flows: Invertible generative models of discrete data. *Advances in Neural Information Processing Systems*, 32.

Tucker, G.; Mnih, A.; Maddison, C. J.; Lawson, J.; and Sohl-Dickstein, J. 2017. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *Advances in Neural Information Processing Systems*, 30.

Yin, M.; Ho, N.; Yan, B.; Qian, X.; and Zhou, M. 2020. Probabilistic best subset selection via gradient-based optimization. *arXiv preprint arXiv:2006.06448*.

Yin, M.; Yue, Y.; and Zhou, M. 2019. ARSM: Augment-REINFORCE-swap-merge estimator for gradient backpropagation through categorical variables. In *International Conference on Machine Learning*, 7095–7104. PMLR.

Yin, M.; and Zhou, M. 2019. ARM: Augment-REINFORCE-merge gradient for stochastic binary networks. In *International Conference on Learning Representations*.

Yin, P.; Lyu, J.; Zhang, S.; Osher, S.; Qi, Y.; and Xin, J. 2019. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*.