4th Symposium on Advances in Approximate Bayesian Inference, 2021 1–14

Double Control Variates for Gradient Estimation in Discrete Latent Variable Models

Anonymous Authors
Anonymous Institution

Abstract

Stochastic gradient-based optimisation for discrete latent variable models is challenging due to the high variance of gradients. We introduce a variance reduction technique for score function estimators that makes use of *double control variates*. These control variates act on top of a main control variate, and try to further reduce the variance of the overall estimator. We develop a double control variate for the REINFORCE leave-one-out estimator using Taylor expansions. For training discrete latent variable models, such as variational autoencoders with binary latent variables, our approach adds no extra computational cost compared to standard training with the REINFORCE leave-one-out estimator. We apply our method to challenging high-dimensional toy examples and training variational autoencoders with binary latent variables. We show that our estimator can have lower variance compared to other state-of-the-art estimators.

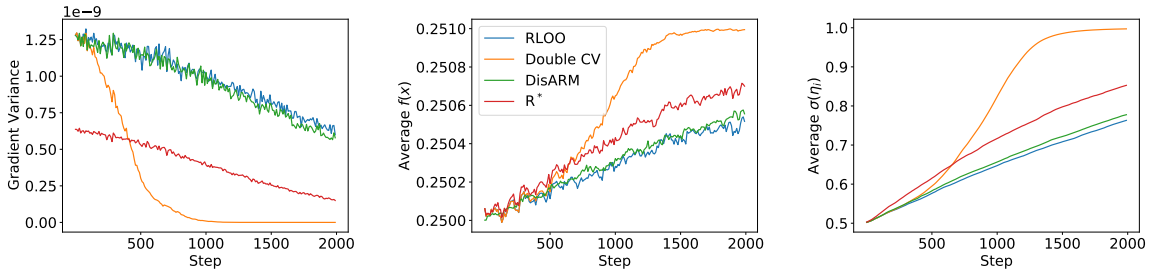
1. Background

Several problems in machine learning, such as variational inference and reinforcement learning, require the optimisation of an intractable expectation of an *objective function* $f(x)$ under a distribution $q_\eta(x)$ with tunable parameters η. Here $f(x)$ is a differentiable objective function. x is a D-dimensional vector. Since $f(x)$ can have a complex non-linear form, $\mathbb{E}_{q_\eta(x)}\left[f(x)\right]$ and its exact gradients are generally intractable. Several techniques apply stochastic optimisation based on unbiased Monte Carlo gradients by sampling from $q_\eta(x)$.

Pathwise or reparametrization gradients (Glasserman, 2003) have been shown to be effective for machine learning problems (Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014), but they are only applicable to continuous distributions. A very general class of gradient estimators that apply to both continuous and discrete variables is the score function or REINFORCE estimator (Glynn, 1990; Williams, 1992; Carbonetto et al., 2009; Paisley et al., 2012; Ranganath et al., 2014; Mnih and Gregor, 2014). However, these estimators suffer from high variance and reducing the variance remains an important open problem. Variance reduction techniques for REINFORCE estimators range from simple baselines (Ranganath et al., 2014; Mnih and Gregor, 2014) and Rao-blackwellization (Titsias and Lázaro-Gredilla, 2015; Tokui and Sato, 2017) to more advanced gradient-based control variates (Tucker et al., 2017; Grathwohl et al., 2018; Gu et al., 2016) and coupled sampling (Yin and Zhou, 2019; Dong et al., 2020; Yin et al., 2020; Dimitriev and Zhou, 2021). The score function estimator with a baseline b is given by $\frac{1}{K}\sum_{k=1}^{K}\left(f(x_k)-b\right)\nabla_\eta \log q_\eta(x_k), \quad x_k \sim q_\eta(x)$.

Another variance reduction method that has become prominent recently is the REINFORCE leave-one-out estimator (RLOO) (Salimans and Knowles, 2014; Kool et al., 2019;

© A. Authors.



(a) (b) (c)

Figure 1: Variance reduction for a toy high-dimensional maximization problem, following Tucker et al. (2017), with binary latent variables and fitting probabilities $\sigma(\eta_i)$ (where $\sigma(\eta_i) = 1$ is optimal); see Section 3.1. Panel (a) shows the gradient variances (estimated by 2000 samples) for four different estimators. Panel (b) the objective function that we want to maximize and (c) the average of the estimated $\sigma(\eta_i)$s. The proposed double control variate estimator is the most effective one.

Richter et al., 2020). Given $K \geq 2$ samples, it takes advantage of multiple evaluations of f to avoid learning the baseline b:

$$\frac{1}{K}\sum_{k=1}^{K}\left(f(x_k) - \frac{1}{K-1}\sum_{j\neq k} f(x_j)\right)\nabla_\eta \log q_\eta(x_k), \tag{1}$$

where each $\left(\frac{1}{K-1}\sum_{j\neq k} f(x_j)\right)\nabla_\eta \log q_\eta(x_k)$ acts as a sample-specific control variate. Despite its simplicity, this estimator performs very strongly for training discrete latent variables models (Dong et al., 2020; Richter et al., 2020). Presumably this is because the leave-one-out stochastic baselines can automatically adapt to the non-stationarity of the $f(x)$. Specifically, $f(x) := f_\theta(x)$ often contain additional *model parameters* θ updated at each optimization step[1], as for instance in variational autoencoders (VAEs) (Kingma and Welling, 2014; Rezende et al., 2014). Although θ is changing, the sample-specific baseline $\frac{1}{K-1}\sum_{j\neq k} f_\theta(x_j)$ always remains an unbiased estimate of $\mathbb{E}_{q_\eta(x)}[f_\theta(x)]$.

However, RLOO is still limited in how much variance reduction it can achieve.

Proposition 1 *Consider the estimator* $R^*(\eta) = \frac{1}{K}\sum_{k=1}^{K}\left(f(x_k) - \mathbb{E}f\right)\nabla_\eta \log q_\eta(x_k)$*, where* $\mathbb{E}f = \mathbb{E}_{q_\eta(x)}[f(x)]$ *is a constant baseline across all samples. Then,* $Var(RLOO) \geq Var(R^*)$*.*

According to Prop. 1, the performance of RLOO is bounded by R* which uses the mean $\mathbb{E}f$ (usually intractable in practice) as a constant baseline. Therefore, there is scope to further reduce the variance of this estimator.

In this work, we focus on the RLOO estimator and enhance it by adding extra control variates. We refer to the added baselines as *double control variates* since they co-exist with the main RLOO baseline, and are designed to have a complementary effect by reducing the variance of the initial RLOO estimator. For training latent variable models with discrete variables, our proposed estimator runs roughly at the same speed as the RLOO estimator.

1. For θ, it is straightforward to obtain low variance gradients.

2. Double Control Variates for REINFORCE LOO

We construct these new control variates along two directions:

(a) Since the main baseline $\frac{1}{K-1}\sum_{j\neq k} f(x_j)$ is stochastic and thus has variance, we can try to reduce the variance by adding a control variate for each stochastic term $f(x_j)$.

(b) We want to add a different type of control variate that depends on x_k which may have a complementary effect to the main RLOO baseline.

In the remaining of Section 2 we use $s(x) := \nabla_\eta \log q_\eta(x)$ to denote the score function. To accomplish both (a) and (b) simultaneously we start with the unbiased estimator $\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k)+\alpha b(x_k)\right]s(x_k) - \alpha\mathbb{E}_{q_\eta(x)}[b(x)s(x)]$, where we introduced a control variate $b(x_k)$, that depends on the current sample x_k and has analytic global correction $\mathbb{E}_{q_\eta(x)}[b(x)s(x)]$. Then, to create a double control variate estimator we treat $f(x)+\alpha b(x)$ as the "new effective objective function" and apply the leave-one-out procedure to it:

$$\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k)+\alpha b(x_k)-\frac{1}{K-1}\sum_{j\neq k}(f(x_j)+\alpha b(x_j))\right]s(x_k) - \alpha\mathbb{E}_{q_\eta(x)}[b(x)s(x)]. \tag{2}$$

The scalar α is a regression coefficient that can be further optimised to reduce the variance. In the above estimator we have highlighted with blue the first appearance $b(x_k)$, which can be thought of as a baseline paired with the value $f(x_k)$, and with red the second appearances $b(x_j)$ paired with the remaining values $f(x_j)$. Intuitively, $b(x_k)$ can be considered as targeting to reduce the variance of $f(x_k)$ and $b(x_j)$ the variance of $f(x_j)$.

In Sections 2.1 and 2.2 we describe two approaches to specify $b(x)$. For training latent variable models such as VAEs, the second will be the most practical since it adds no extra cost. The first method helps to introduce the idea and is based on a mean field argument.

2.1. Mean Field Approach

To specify $b(x)$ we can construct an approximation of $f(x)$ that correlates well with the exact value $f(x)$. While any surrogate of $f(x)$ with a tractable global correction could work, next we focus on the case when $f(x)$ is differentiable w.r.t. the input x and we use a first order Taylor expansion around the mean $\mu = \mathbb{E}_{q_\eta(x)}[x]$, so that $f(x) \approx f(\mu)+\nabla f(\mu)^\top(x-\mu)$. Since any constant term in $b(x)$ cancels out in (2), the constant $f(\mu)$ in the Taylor approximation can be dropped, yielding the double control variate $b(x) = \nabla f(\mu)^\top(x-\mu)$. By substituting this function in Eq. (2) we obtain the general estimator

$$\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k)+\alpha\nabla f(\mu)^\top(x_k-\mu)-\frac{1}{K-1}\sum_{j\neq k}\Big(f(x_j)+\alpha\nabla f(\mu)^\top(x_j-\mu)\Big)\right]s(x_k) \\ -\alpha\mathbb{E}_{q_\eta(x)}[s(x)(x-\mu)^\top]\nabla f(\mu), \tag{3}$$

where $\mathbb{E}_{q_\eta(x)}[s(x)(x-\mu)^\top]$ will typically have an analytical form. For binary latent variables $x \in \{0,1\}^d$ and a factorised Bernoulli distribution of the form $q_\eta(x) = \prod_{i=1}^{d}\mu_i^{x_i}(1-\mu_i)^{1-x_i}$, $\mu_i = \sigma(\eta_i)$. $\mathbb{E}_{q_\eta(x)}[s(x)(x-\mu)^\top] = \mathrm{diag}(\mu\circ(1-\mu))$ and the global correction term simplifies to $-\alpha\mu\circ(1-\mu)\circ\nabla f(\mu)$, where $\circ$ denotes element-wise vector product.

2.2. An Estimator without Extra Gradient Evaluations

The estimator in Eq. (3) requires a backpropagation operation to compute the gradient $\nabla f(\mu)$, which adds extra computational cost compared to standard RLOO. Next, we wish to develop an alternative estimator that avoids this extra cost for certain problems. For many applications, such as VAEs, the function $f(x)$ depends on model parameters θ (typically different than η) that we update at each optimisation iteration by computing the gradients $\{\nabla_\theta f(x_j)\}_{j=1}^K$. Then, from the same backpropagation operations is easy to also return the gradients w.r.t. the latent vectors, i.e. to compute $\{\nabla f(x_j)\}_{j=1}^K$. We would like to utilize these latter gradients to define the double control variate $b(x)$.

Starting from $b(x) = \nabla f(\mu)^\top (x - \mu)$, we first want to modify $b(x_k)$ by replacing $\nabla f(\mu)$ with some new gradient computed from $\{\nabla f(x_j)\}_{j=1}^K$. We cannot use the full average because this will lead to $(\frac{1}{K}\sum_{j=1}^K \nabla f(x_j))^\top (x_k - \mu)$ which has an intractable global correction due to the intractable term $\mathbb{E}_{q_\eta(x_k)}[\nabla f(x_k)^\top (x_k - \mu)\nabla_\eta \log q_\eta(x_k)]$. However, we can use the leave-one-out gradient, i.e. by leaving out $\nabla f(x_k)$, which gives

$$b_k(x_{1:K}) = \left(\frac{1}{K-1}\sum_{j\neq k} \nabla f(x_j)\right)^\top (x_k - \mu), \tag{4}$$

This has a tractable correction term $\mathbb{E}_{q_\eta(x_k)}[b_k(x_{1:K})\nabla \log q_\eta(x_k)]$.

Proposition 2 *For $b_k(x_{1:K})$ from (4) we obtain the following unbiased gradient estimator*

$$\frac{1}{K}\sum_{k=1}^K \left[f(x_k) + \alpha b_k(x_{1:K}) - \frac{1}{K-1}\sum_{j\neq k}(f(x_j) + \alpha b_j(x_{1:K}))\right] \\ \times s(x_k) - \alpha \mathbb{E}_{q_\eta(x)}[s(x)(x-\mu)^\top]\left(\frac{1}{K}\sum_{k=1}^K \nabla f(x_k)\right). \tag{5}$$

The proof of unbiasedness is given in the appendix. We choose the regression coefficient α by minimizing the total variance. If $g(\alpha)$ denotes the stochastic gradient and $\bar{g} = \mathbb{E}[g(\alpha)]$ the exact gradient where the latter does not depend on α, the total variance is $\mathrm{Tr}[\mathbb{E}(g(\alpha) - \bar{g})(g(\alpha) - \bar{g})^\top] = \mathbb{E}[||g(\alpha)||^2] + const$. Thus, in practice at each optimisation iteration we can perform a gradient step towards minimizing the empirical variance $||g(\alpha)||^2$.

3. Experiments

3.1. Toy Learning Problem

We consider a generalization of the artificial problem considered by Tucker et al. (2017), where the goal is to maximize $\mathcal{E}(\eta) = \mathbb{E}_{q_\eta(x)}[D^{-1}\sum_{i=1}^D (x_i - p_0)^2]$, where $q_\eta(x) = \prod_{i=1}^D \sigma(\eta_i)^{x_i}(1 - \sigma(\eta_i))^{1-x_i}$, $p_0 = 0.499$ and the optimal solution is $\sigma(\eta_i) = 1$ for all $i = 1, \ldots, D$. While Tucker et al. (2017) considered $D = 1$, here we additionally consider a more difficult high-dimensional case with $D = 200$. We compare three methods: (i) RLOO, (ii) DisARM and our proposed double control variates estimator (Double CV) from Eq. (5). We use $K = 2$ samples for all methods. Also we include in the comparison R* which is tractable in this toy

	Bernoulli Likelihoods			Gaussian Likelihoods		
	MNIST	Fashion-MNIST	Omniglot	MNIST	Fashion-MNIST	Omniglot
RLOO	-103.11 ± 0.16	-241.53 ± 0.24	-116.83 ± 0.05	668.07 ± 0.40	179.52 ± 0.23	443.51 ± 0.93
DoubleCV	$\mathbf{-102.45 \pm 0.13}$	$\mathbf{-240.96 \pm 0.17}$	$\mathbf{-116.22 \pm 0.08}$	$\mathbf{676.87 \pm 1.18}$	$\mathbf{186.35 \pm 0.64}$	$\mathbf{446.95 \pm 0.63}$
DisARM	-102.56 ± 0.09	-241.02 ± 0.20	-116.36 ± 0.05	668.03 ± 0.61	182.65 ± 0.47	446.22 ± 1.38
RELAX	$\mathbf{-101.86 \pm 0.11}$	$\mathbf{-240.63 \pm 0.16}$	$\mathbf{-115.79 \pm 0.06}$	$\mathbf{688.58 \pm 0.52}$	$\mathbf{196.38 \pm 0.66}$	$\mathbf{462.30 \pm 0.91}$

Table 1: Training nonlinear binary latent VAEs with $K = 2$ (except for RELAX) on MNIST, Fashion-MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

	Bernoulli Likelihoods			Gaussian Likelihoods		
	MNIST	Fashion-MNIST	Omniglot	MNIST	Fashion-MNIST	Omniglot
RLOO	-100.50 ± 0.22	-239.03 ± 0.15	-114.75 ± 0.07	687.83 ± 0.50	195.27 ± 0.24	460.23 ± 1.42
DoubleCV	$\mathbf{-99.89 \pm 0.12}$	-238.98 ± 0.18	$\mathbf{-114.56 \pm 0.06}$	$\mathbf{691.51 \pm 0.75}$	$\mathbf{199.01 \pm 0.60}$	463.03 ± 0.94
DisARM	-100.67 ± 0.07	-239.20 ± 0.15	-115.05 ± 0.07	683.28 ± 0.89	192.96 ± 0.29	458.38 ± 0.88
ARMS	-100.07 ± 0.08	$\mathbf{-238.50 \pm 0.13}$	-114.57 ± 0.06	687.26 ± 1.21	197.25 ± 0.48	$\mathbf{463.30 \pm 0.86}$
RELAX	-101.86 ± 0.11	-240.63 ± 0.16	-115.79 ± 0.06	688.58 ± 0.52	196.38 ± 0.66	462.30 ± 0.91

Table 2: Training a nonlinear binary latent VAE with $K = 4$ (except for RELAX) on MNIST, Fashion-MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

example. Fig. 1 compares the methods in terms of variance, the objective function and the average value of the D probabilities $\sigma(\eta_i)$. Fig. 6 shows further comparison for the $D = 1$ case, as in Tucker et al. (2017). We observe that Double CV gradients have smaller variance which results in much faster optimisation convergence.

3.2. Variational Autoencoders with Binary Latent Variables

3.2.1. Experimental setup

We consider training nonlinear variational autoencoders (Kingma and Welling, 2014; Rezende et al., 2014) with binary latent variables. We conduct separate experiments for binary output data $y \in \{0, 1\}^d$ and continuous data $y \in \mathbb{R}^d$. For binary data we use the standard Bernoulli likelihood. For continuous data we centered data between $[-1, 1]$ and consider a Gaussian likelihood of the form $p_\theta(y|x) = \mathcal{N}(y|m_\theta(x), \Sigma)$, where $m_\theta(x)$ is a decoder mean function that depends on the latent variable x and Σ is a learnable diagonal covariance matrix. We consider the datasets MNIST, Fashion-MNIST and Omniglot. For all three datasets we use both the dynamically binarized versions and their original continuous versions. More details and the results for linear VAEs are included in the appendix.

We compared the following estimators: RLOO, DisARM and the proposed Double CV method where all three estimators use K samples. We experimented with $K = 2$ and $K = 4$. For $K = 4$ we also compare to the state-of-the-art ARMS estimator recently proposed by Dimitriev and Zhou (2021). Besides, we include in the comparison the RELAX estimator that combines concrete relaxation (Tucker et al., 2017) with a learned control variate (Grathwohl

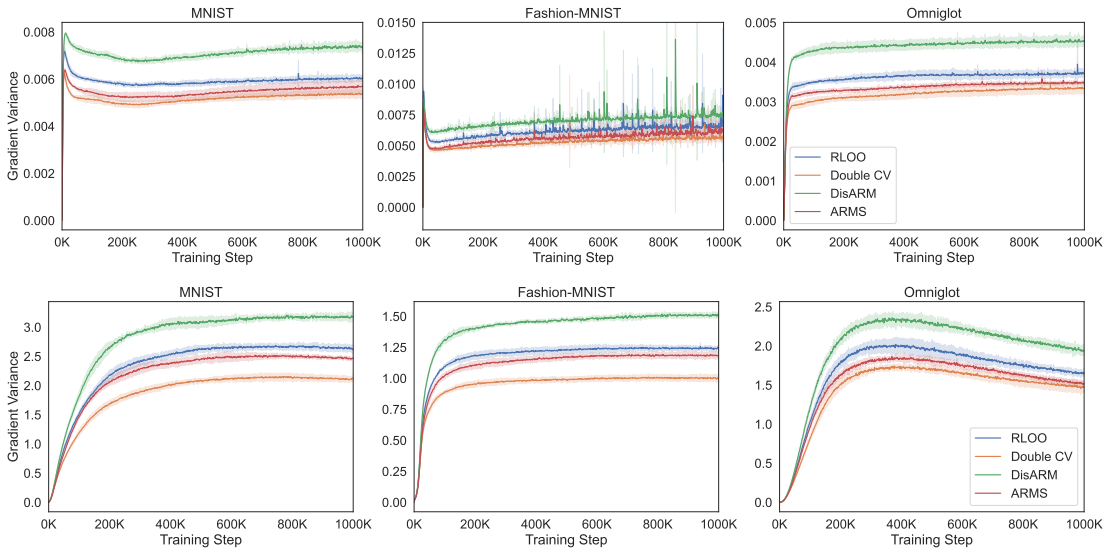


Figure 2: Variance of gradient estimates in training nonlinear binary latent variational autoencoders with $K = 4$ on MNIST, Fashion-MNIST, and Omniglot. *Top:* Using Bernoulli likelihoods and dynamically binarized datasets. *Bottom:* Using Gaussian likelihoods and non-binarized datasets.

et al., 2018). We point out that RLOO, DisARM, Double CV, and ARMS (when $K = 4$) have roughly the same running time on a P100 GPU while RELAX is computationally more expensive and is roughly twice slower than the other four estimators with $K = 4$ (see Table 3). Also note that RELAX is less generally applicable since it assumes the existence of a concrete relaxation for x.

3.2.2. Results

Table 1 shows the training ELBO for binarized and continuous datasets when training the VAE by different estimators with $K = 2$. We can observe that Double CV consistently outperforms RLOO in all experiments, while having approximately the same running time. Double CV also outperforms DisARM in all cases for both Bernoulli and Gaussian likelihoods. Furthermore, Fig. 3 plots the gradient variance and the training ELBO for the binarized datasets as a function of the training steps. Similarly, Fig. 4 shows the corresponding results for the non-binarized (continuous) datasets where a Gaussian likelihood is used. We observe that the Double CV estimator can have lower variance than RLOO and DisARM. Also, while RELAX performs better than the other methods it is less generally applicable and more expensive.

For $K = 4$, the final training ELBO values are reported in Table 2 and the variances of the different estimators are plotted in Fig. 2. We can observe that Double CV consistently has lower variance than other estimators and it outperforms ARMS in terms of training ELBO in most cases. It also significantly outperforms RELAX. Note that, even with $K = 4$, Double CV is still nearly twice faster than RELAX.

References

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.

Carbonetto, P., King, M., and Hamze, F. (2009). A stochastic approximation method for inference in probabilistic graphical models. In *Advances in Neural Information Processing Systems*, volume 22.

Dimitriev, A. and Zhou, M. (2021). ARMS: antithetic-reinforce-multi-sample gradient for binary variables. In *International Conference on Machine Learning*, volume 139, pages 2717–2727.

Dong, Z., Mnih, A., and Tucker, G. (2020). DisARM: An antithetic gradient estimator for binary latent variables. In *Advances in Neural Information Processing Systems*, volume 33, pages 18637–18647. Curran Associates, Inc.

Dong, Z., Mnih, A., and Tucker, G. (2021). Coupled gradient estimators for discrete latent variables. *arXiv preprint arXiv:2106.08056*.

Geffner, T. and Domke, J. (2018). Using large ensembles of control variates for variational inference. In *Advances in Neural Information Processing Systems*, volume 31.

Glasserman, P. (2003). *Monte Carlo methods in financial engineering*, volume 53. Springer Science & Business Media.

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

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

Gu, S., Levine, S., Sutskever, I., and Mnih, A. (2016). MuProp: Unbiased backpropagation for stochastic neural networks. In *International Conference on Learning Representations*.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. International Conference for Learning Representations.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. In *International Conference on Learning Representations*.

Kool, W., Hoof, H. V., and Welling, M. (2019). Buy 4 reinforce samples, get a baseline for free! In *DeepRLStructPred@ICLR*.

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

Mnih, A. and Rezende, D. J. (2016). Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*.

Owen, A. B. (2013). *Monte Carlo theory, methods and examples.*

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational Bayesian inference with stochastic search. In *International Conference on Machine Learning.*

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black box variational inference. In *International Conference on Artificial Intelligence and Statistics*, page 814–822.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning.*

Richter, L., Boustati, A., Nüsken, N., Ruiz, F., and Akyildiz, O. D. (2020). VarGrad: A low-variance gradient estimator for variational inference. In *Advances in Neural Information Processing Systems*, volume 33, pages 13481–13492. Curran Associates, Inc.

Robbins, H. and Monro, S. (1951). A Stochastic Approximation Method. *The Annals of Mathematical Statistics*, 22(3):400–407.

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

Titsias, M. K. and Lázaro-Gredilla, M. (2014). Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning.*

Titsias, M. K. and Lázaro-Gredilla, M. (2015). Local expectation gradients for black box variational inference. *Advances in Neural Information Processing Systems*, 28:2638–2646.

Tokui, S. and Sato, I. (2017). Evaluating the variance of likelihood-ratio gradient estimators. In *International Conference on Machine Learning*, pages 3414–3423.

Tucker, G., Mnih, A., Maddison, C. J., and Sohl-Dickstein, J. (2017). REBAR: low-variance, unbiased gradient estimates for discrete latent variable models. In *International Conference on Learning Representations.*

Weaver, L. and Tao, N. (2001). The optimal reward baseline for gradient-based reinforcement learning. In *Uncertainty in Artificial Intelligence*, pages 538–545.

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

Yin, M., Ho, N., Yan, B., Qian, X., and Zhou, M. (2020). Probabilistic best subset selection via gradient-based optimization.

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

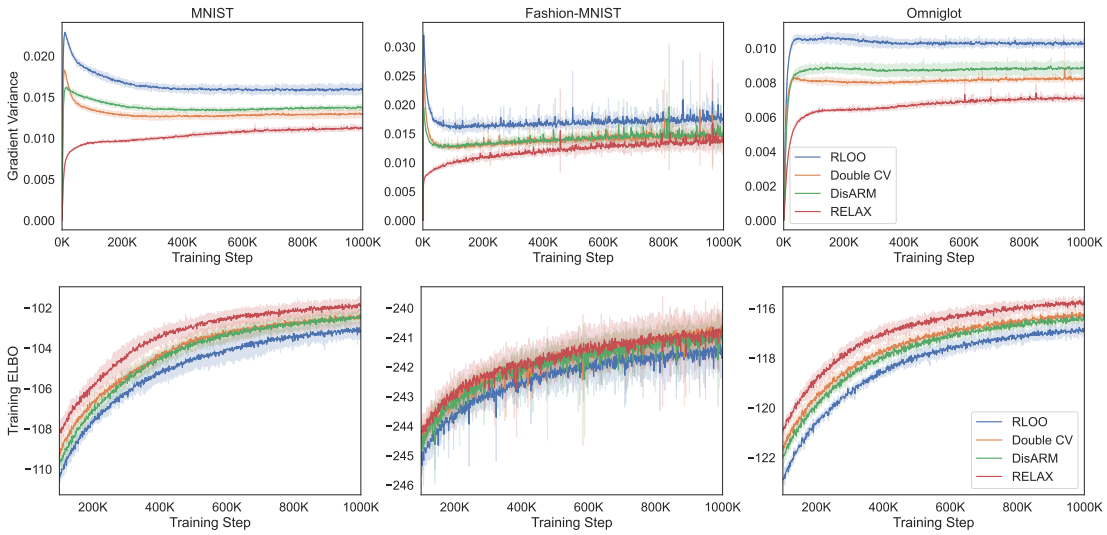


Figure 3: Training nonlinear binary latent VAEs with Bernoulli likelihoods with $K = 2$ (except for RELAX) on dynamically binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.

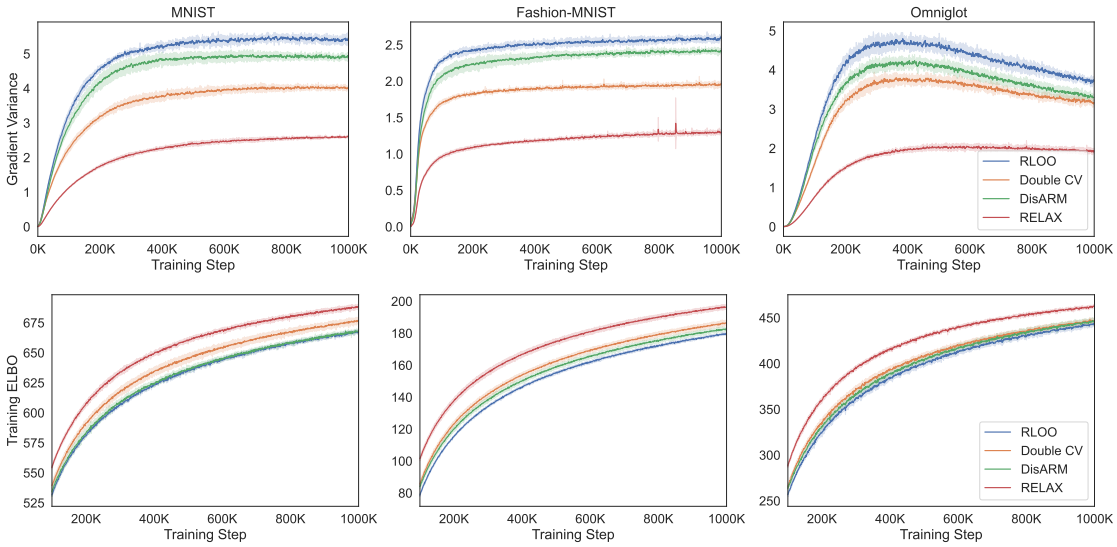


Figure 4: Training nonlinear binary latent VAEs with Gaussian likelihoods with $K = 2$ (except for RELAX) on non-binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.

Appendix A. Related Work

Our proposed gradient estimators follow the general form of unbiased REINFORCE estimators (Williams, 1992; Glynn, 1990; Carbonetto et al., 2009; Paisley et al., 2012; Ranganath et al., 2014; Mnih and Gregor, 2014), which unlike reparametrization or pathwise gradients (Kingma and Welling, 2014; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014), are applicable also to discrete latent variables. The double control variates we develop build on top of the RLOO estimator (Kool et al., 2019; Salimans and Knowles, 2014; Richter et al., 2020); see also the VIMCO method of Mnih and Rezende (2016) who also used a leave-one-out procedure. RLOO was shown to be a competitive estimator for challenging problems such as training VAEs with binary or categorical latent variables (Dong et al., 2020; Richter et al., 2020; Dong et al., 2021). As shown by our experiments, our enhancement of RLOO with double control variates leads to further variance reduction, and without increasing the computational cost when training VAEs.

In our current framework, the double control variates are constructed by using the gradients of the objective function $f_\theta(x)$. These gradients are also used by other unbiased gradient techniques based on control variates, such as the MuProp estimator (Gu et al., 2016), the concrete relaxation methods REBAR (Tucker et al., 2017) and RELAX (Grathwohl et al., 2018). Our method differs significantly since our control variates act on top of the sample-specific RLOO baseline $\frac{1}{K-1}\sum_{j\neq k} f_\theta(x_j)$, i.e., they try to have complementary effect to this existing control variate. This means that our estimators preserve RLOO's property of capturing the non-stationarity of $f_\theta(x)$, since the leave-one-out baseline always tracks the expected value $\mathbb{E}[f_\theta(x)]$ as θ evolves. In contrast, previous gradient-based estimators use *stand-alone* global control variates. For instance, the baseline in MuProp (Gu et al., 2016) is constructed using only $f_\theta(\mu)$ and x_k, which can be a poor tracker of the expected value $\mathbb{E}[f_\theta(x)]$. Unlike MuProp, REBAR (Tucker et al., 2017) and RELAX (Grathwohl et al., 2018) are much more effective, however they are more expensive than our method — they require differentiating f_θ three times, while our method can work with just two, and they are less generally applicable since they assume a concrete relaxation for x.

Other recent REINFORCE type of estimators for discrete latent variables are based on coupled sampling (Owen, 2013), such as antithetic sampling (Yin and Zhou, 2019; Dong et al., 2020; Yin et al., 2020; Dimitriev and Zhou, 2021). For instance, the recent DisARM estimator independently proposed by Dong et al. (2020) and Yin et al. (2020) was shown to give state-of-the-art results for binary latent-variable models with $K = 2$ antithetic samples.

Appendix B. Proofs

B.1. Proof of Proposition 1

The RLOO estimator can be written as

$$\underbrace{\frac{1}{K}\sum_{k=1}^{K}\left(f(x_k)-\mathbb{E}f\right)\nabla_\eta \log q_\eta(x_k)}_{\mathrm{R}^*}+\underbrace{\frac{1}{K}\sum_{k=1}^{K}\left(\mathbb{E}f-\frac{1}{K-1}\sum_{j\neq k} f(x_j)\right)\nabla_\eta \log q_\eta(x_k)}_{E} \tag{6}$$

where R^* is the REINFORCE estimator with baseline $\mathbb{E}f$ and E is a residual term of zero mean. To prove the Proposition we will use $Var(\mathrm{RLOO}) = Var(\mathrm{R}^* + E) = Var(\mathrm{R}^*) +$

$Var(E) + 2Cov(\mathrm{R}^*, E)$. Then, it suffices to show that $Cov(\mathrm{R}^*, E) = 0$. We have

$$Cov(\mathrm{R}^*, E) = \frac{1}{K^2} \sum_{k=1}^{K} \sum_{k'=1}^{K} \mathbb{E}\left[(f(x_k) - \mathbb{E}f)(\mathbb{E}f - f_{-k'})\nabla_\eta \log q_\eta(x_k) \nabla_\eta \log q_\eta(x_{k'})^\top\right]$$

where we used $f_{-k'} = \frac{1}{K-1}\sum_{j\neq k'} f(x_j)$ for short. For all terms in the double sum such that $k = k'$ the expectation

$$\mathbb{E}\left[(f(x_k) - \mathbb{E}f)(\mathbb{E}f - f_{-k})\nabla_\eta \log q_\eta(x_k) \nabla_\eta \log q_\eta(x_k)^\top\right] = 0$$

because the zero-mean random variable $\mathbb{E}f - f_{-k}$ is independent from the remaining product (since it does not contain the sample x_k). For all cross terms $k \neq k'$ the whole product $(f(x_k) - \mathbb{E}f)(\mathbb{E}f - f_{-k'})\nabla_\eta \log q_\eta(x_k)$ does not contain the sample $x_{k'}$. Therefore this product is independent from $\nabla_\eta \log q_\eta(x_{k'})$ and thus each cross term is zero because of the score function property $\mathbb{E}[\nabla_\eta \log q_\eta(x_{k'})] = 0$. This shows that $Cov(\mathrm{R}^*, E) = 0$ which completes the proof.

B.2. Proof of Prop. 2

The estimator can be written as

$$\begin{aligned}
&\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k) - \frac{1}{K-1}\sum_{j\neq k} f(x_j)\right]\nabla_\eta \log q_\eta(x_k) \\
&+ \alpha\frac{1}{K}\sum_{k=1}^{K}\left(b_k(x_{1:K}) - \frac{1}{K-1}\sum_{j\neq k} b_j(x_{1:K})\right)\nabla_\eta \log q_\eta(x_k) \\
&- \alpha\mathbb{E}_{q(x)}[\nabla_\eta \log q_\eta(x) \times (x-\mu)^\top]\left(\frac{1}{K}\sum_{k=1}^{K}\nabla f(x_k)\right),
\end{aligned} \tag{7}$$

where $b_k(x_{1:K}) = \left(\frac{1}{K-1}\sum_{j\neq k}\nabla f(x_j)\right)^\top (x_k - \mu)$ and $b_j(x_{1:K}) = \left(\frac{1}{K-1}\sum_{m\neq j}\nabla f(x_m)\right)^\top (x_j - \mu)$. It suffices to show that the expectation of the second line is minus the correction term at the third line. The expectation of each term $b_j(x_{1:K})\nabla_\eta \log q_\eta(x_k)$ for $j \neq k$ is zero because the zero-mean term $x_j - \mu$ is always independent from the rest terms in the product. Then, we need to examine only the expectation of

$$\frac{1}{K}\sum_{k=1}^{K} b_k(x_{1:K})\nabla_\eta \log q_\eta(x_k) = \frac{1}{K(K-1)}\sum_{k=1}^{K}\nabla_\eta \log q_\eta(x_k)(x_k - \mu)^\top \sum_{j\neq k}\nabla f(x_j).$$

Then observe that the expectation of $\nabla_\eta \log q_\eta(x_k) \times (x_k - \mu)^\top$ is the same for every sample x_k, so the above reduces to

$$\mathbb{E}_{q_\eta(x)}[\nabla_\eta \log q_\eta(x) \times (x-\mu)^\top]\frac{1}{K(K-1)}\sum_{k=1}^{K}\sum_{j\neq k}\nabla f(x_j)$$

from which the result follows since $\sum_{k=1}^{K}\sum_{j\neq k}\nabla f(x_j) = (K-1)\sum_{k=1}^{K}\nabla f(x_k)$.

B.3. The "half" Double Control Variate Estimators

One question is whether we need both $b(x_k)$ and $b(x_j)$ or we could keep one of them, i.e. to use an "$b(x_k)$ only" or "$b(x_j)$ only" estimator. It is straightforward to express these latter unbiased estimators, as follows. The "$b(x_k)$ only" estimator is given by

$$\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k)+\alpha b(x_k)-\frac{1}{K-1}\sum_{j\neq k}f(x_j)\right]\nabla_\eta \log q_\eta(x_k)-\alpha\mathbb{E}_{q_\eta(x)}[b(x)\nabla_\eta \log q_\eta(x)]. \tag{8}$$

and the "$b(x_j)$ only" by

$$\frac{1}{K}\sum_{k=1}^{K}\left[f(x_k)-\frac{1}{K-1}\sum_{j\neq k}(f(x_j)+\alpha b(x_j))\right]\nabla_\eta \log q_\eta(x_k). \tag{9}$$

It is easy to show that both estimators are unbiased. However, in practice these estimators can be much less effective in terms of variance reduction than their Double CV combination. In Figure 5 we apply these two estimators to the toy learning problem with $D = 10$. Both estimators are significantly outperformed by the full Double CV estimator. Notably, the "$b(x_k)$ only" estimator could outperform R* since it uses a baseline that depends on the current sample x_k, while "$b(x_j)$ only" reduces the variance of the RLOO control variate but remains bounded by R*.

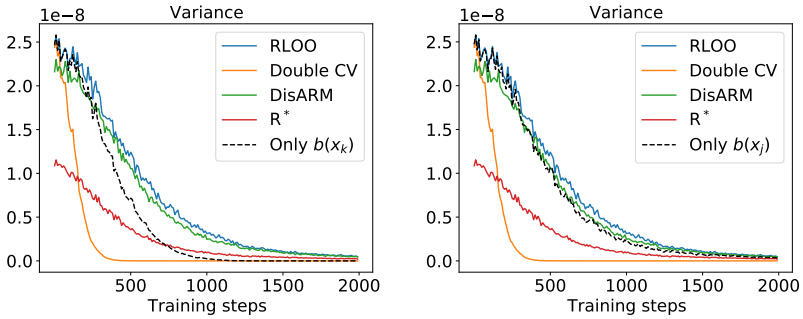


Figure 5: *Left:* Variance of the "only $b(x_k)$" estimator where only the half part of the double control variate is used. *Right:* The corresponding plot for the "only $b(x_j)$" estimator where the other half part of the double control variate is used. The full double control variate estimator (Double CV), RLOO, DisARM and R* are included for comparison. The experiment corresponds to the toy problem with $D = 10$ and $b(x)$ was chosen according to Eq. (4), i.e. the full Double CV estimator is from (5).

Appendix C. Additional Results and Experimental Details

C.1. Toy Experiment with $D = 1$

For completeness, we include the results of a simpler version of the toy experiment described in Section 3.1, where we set $D = 1$. This is the setting used in several previous works (Tucker

et al., 2017; Grathwohl et al., 2018; Yin and Zhou, 2019; Dong et al., 2020). The variances of the gradient estimators and the training curves of $\sigma(\eta)$ are plotted in Fig. 6.

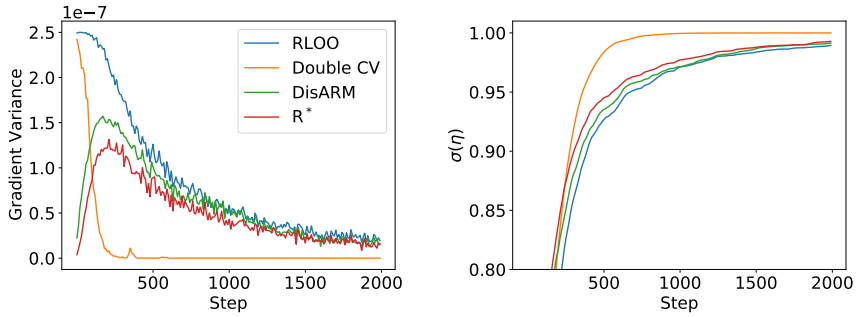
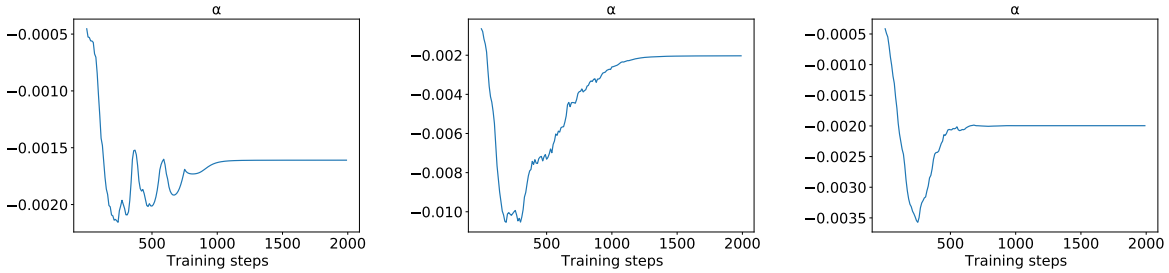


Figure 6: *Left:* Variance of the gradient estimators for the toy problem with $D = 1$. *Right:* The estimated value $\sigma(\eta)$ across iterations (optimal value is 1).



$D = 1$ $D = 10$ $D = 200$

Figure 7: The evolution of the estimated regression coefficient α during optimisation for the toy learning problem.

C.2. Training Binary Latent VAEs

C.2.1. Experimental Details

We follow the VAE models used in Yin and Zhou (2019); Dong et al. (2020). The VAE model uses fully connected neural networks with two hidden layers of 200 LeakyReLU activation units with the coefficient 0.3. All models are trained using Adam (Kingma and Ba, 2014) with learning rate 10^{-3} for the binarized data, while for the continuous data we used smaller learning rate 10^{-4}. In all experiments α was trained with learning rate 10^{-3}. For all experiments we use a uniform factorized Bernoulli prior over the $D = 200$ dimensional latent variable x. The model was trained by maximizing the ELBO using an amortised factorised variational Bernoulli distribution.

C.2.2. Time comparison

In Fig. 3 We report the per-step running time of RLOO, Double CV, DisARM, ARMS estimators when $K = 4$ and compare to RELAX. RELAX is almost twice slower.

	RLOO	Double CV	DisARM	ARMS	RELAX
Time (sec/step)	0.0035	0.0036	0.0031	0.0037	0.0080

Table 3: Time per step when training a Bernoulli VAE with $K = 4$ (except for RELAX) on dynamically binarized Fashion-MNIST.

C.2.3. Full results of training ELBOs

Here we include the full results of final training ELBOs from the experiment in Section 3.2. Table 4 and Table 5 extend Table 1 to include the linear VAE results trained under the same setting. Table 6 and Table 7 extend Table 2 to include the linear VAE results trained under the same setting. The linear VAE has 200 dimensional latent variable x and use a single fully-connected layer to produce the logits (for Bernoulli likelihoods) or the mean (for Gaussian likelihoods) of the distribution of y.

C.2.4. Additional figures for nonlinear VAEs

In Fig. 8 we plot the average training ELBOs as a function of training steps from the $K = 4$ experiment in Section 3.2.

C.2.5. Additional figures for linear VAEs

We plot the gradient variance and average training ELBOs of training linear VAEs in Figures 9,10,11, and 12.

	RLOO	Double CV	DisARM	RELAX
MNIST:				
Linear	-113.06 ± 0.05	-112.82 ± 0.07	$\mathbf{-112.72 \pm 0.07}$	$\mathbf{-112.18 \pm 0.07}$
Nonlinear	-103.11 ± 0.16	$\mathbf{-102.45 \pm 0.13}$	-102.56 ± 0.09	$\mathbf{-101.86 \pm 0.11}$
Fashion-MNIST:				
Linear	-257.38 ± 0.17	$\mathbf{-256.21 \pm 0.17}$	-257.01 ± 0.06	$\mathbf{-255.16 \pm 0.17}$
Nonlinear	-241.53 ± 0.24	$\mathbf{-240.96 \pm 0.17}$	-241.02 ± 0.20	$\mathbf{-240.63 \pm 0.16}$
Omniglot:				
Linear	-119.63 ± 0.05	-119.52 ± 0.02	$\mathbf{-119.42 \pm 0.03}$	$\mathbf{-119.16 \pm 0.02}$
Nonlinear	-116.83 ± 0.05	$\mathbf{-116.22 \pm 0.08}$	-116.36 ± 0.05	$\mathbf{-115.79 \pm 0.06}$

Table 4: Training binary latent VAEs with $K = 2$ (except for RELAX) on dynamically binarized MNIST, Fashion-MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

	RLOO	Double CV	DisARM	RELAX
MNIST				
Linear	503.01 ± 0.22	504.33 ± 0.98	$\mathbf{504.43 \pm 0.93}$	$\mathbf{513.38 \pm 0.52}$
Nonlinear	668.07 ± 0.40	$\mathbf{676.87 \pm 1.18}$	668.03 ± 0.61	$\mathbf{688.58 \pm 0.52}$
Fashion-MNIST				
Linear	29.75 ± 0.40	31.08 ± 0.24	$\mathbf{31.71 \pm 0.20}$	$\mathbf{37.54 \pm 0.30}$
Nonlinear	179.52 ± 0.23	$\mathbf{186.35 \pm 0.64}$	182.65 ± 0.47	$\mathbf{196.38 \pm 0.66}$
Omniglot				
Linear	245.73 ± 0.33	245.97 ± 1.02	$\mathbf{247.70 \pm 0.85}$	$\mathbf{255.69 \pm 0.70}$
Nonlinear	443.51 ± 0.93	$\mathbf{446.95 \pm 0.63}$	446.22 ± 1.38	$\mathbf{462.30 \pm 0.91}$

Table 5: Training binary latent VAEs with Gaussian likelihoods using $K = 2$ (except for RELAX) on non-binarized MNIST, Fashion-MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

	RLOO	Double CV	DisARM	ARMS
MNIST:				
Linear	-111.89 ± 0.09	$\mathbf{-111.79 \pm 0.09}$	-112.01 ± 0.06	-111.87 ± 0.02
Nonlinear	-100.50 ± 0.22	$\mathbf{-99.89 \pm 0.12}$	-100.67 ± 0.07	-100.07 ± 0.08
Fashion-MNIST:				
Linear	-254.59 ± 0.16	$\mathbf{-254.52 \pm 0.23}$	-255.01 ± 0.10	-254.67 ± 0.20
Nonlinear	-239.03 ± 0.15	-238.98 ± 0.18	-239.20 ± 0.15	$\mathbf{-238.50 \pm 0.13}$
Omniglot:				
Linear	-118.89 ± 0.02	-118.95 ± 0.02	-118.97 ± 0.01	$\mathbf{-118.87 \pm 0.02}$
Nonlinear	-114.75 ± 0.07	$\mathbf{-114.56 \pm 0.06}$	-115.05 ± 0.07	-114.57 ± 0.06

Table 6: Training binary latent VAEs with $K = 4$ on dynamically binarized MNIST, Fashion MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

	RLOO	Double CV	DisARM	ARMS
MNIST:				
Linear	$\mathbf{516.65 \pm 0.54}$	515.79 ± 0.71	512.47 ± 0.72	514.55 ± 0.71
Nonlinear	687.83 ± 0.50	$\mathbf{691.51 \pm 0.75}$	683.28 ± 0.89	687.26 ± 1.21
Fashion-MNIST:				
Linear	36.70 ± 0.41	36.61 ± 0.34	34.90 ± 0.52	$\mathbf{37.56 \pm 0.43}$
Nonlinear	195.27 ± 0.24	$\mathbf{199.01 \pm 0.60}$	192.96 ± 0.29	197.25 ± 0.48
Omniglot:				
Linear	257.43 ± 0.16	257.88 ± 0.69	254.99 ± 0.69	$\mathbf{258.22 \pm 0.18}$
Nonlinear	460.23 ± 1.42	463.03 ± 0.94	458.38 ± 0.88	$\mathbf{463.30 \pm 0.86}$

Table 7: Training binary latent VAEs with Gaussian likelihoods using $K = 4$ on non-binarized MNIST, Fashion-MNIST, and Omniglot. We report the average ELBO on the training set over 5 independent runs.

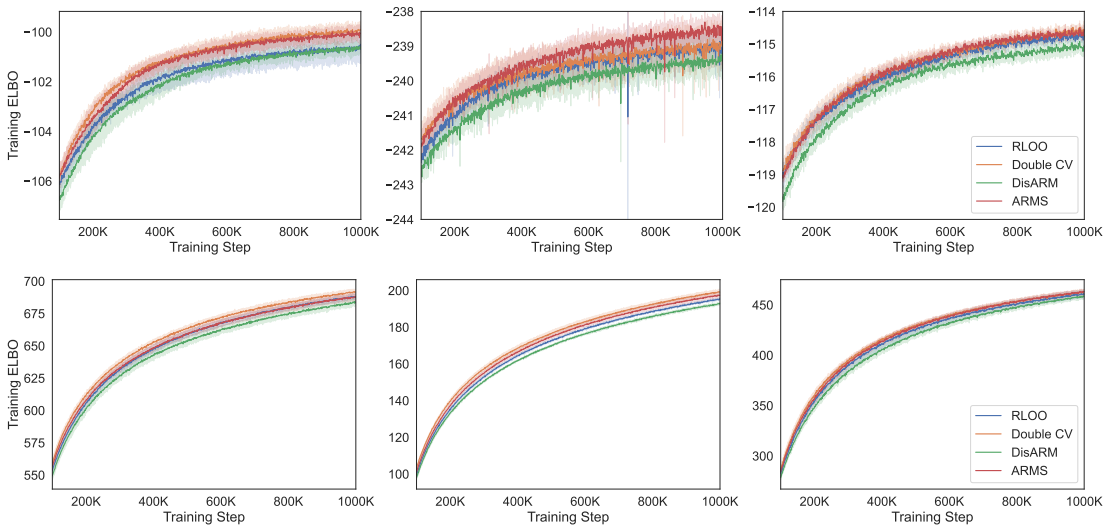


Figure 8: Average training ELBOs for nonlinear binary latent VAEs trained by different estimators with $K = 4$ on MNIST, Fashion-MNIST, and Omniglot. *Top:* Using Bernoulli likelihoods and dynamically binarized datasets. *Bottom:* Using Gaussian likelihoods and non-binarized datasets.

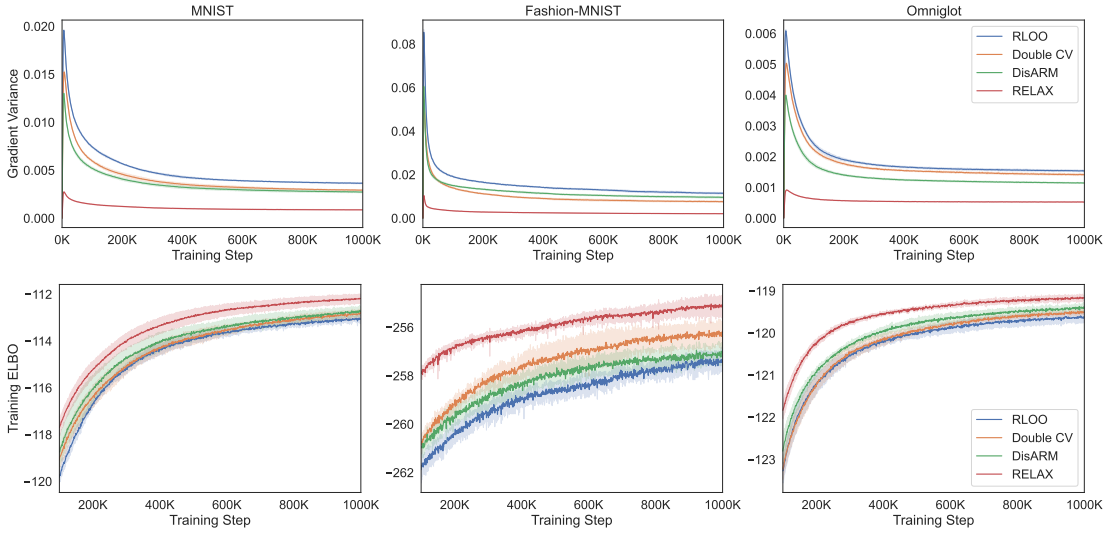


Figure 9: Training linear binary latent VAEs with Bernoulli likelihoods with $K = 2$ (except for RELAX) on dynamically binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.

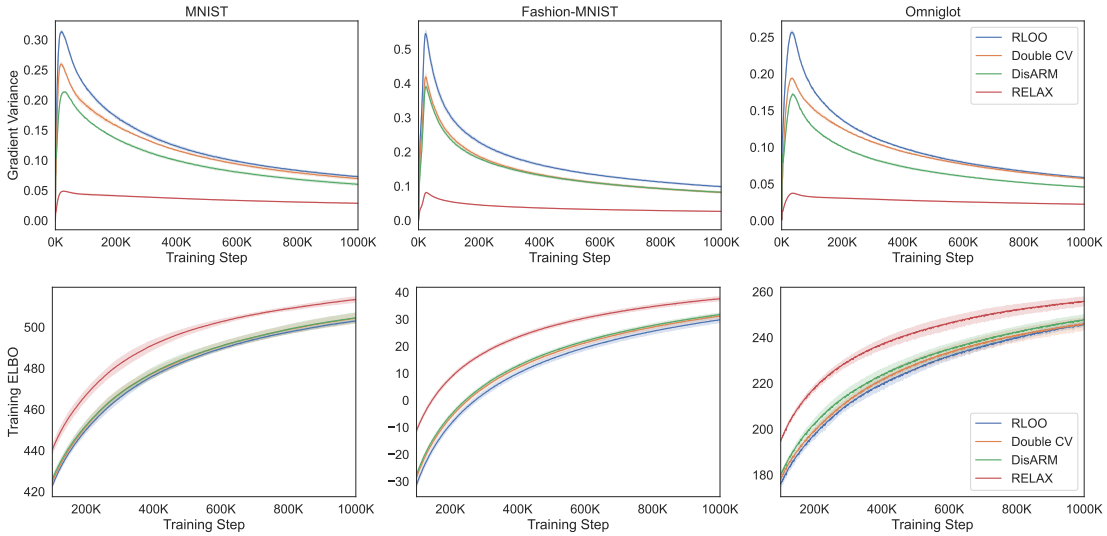


Figure 10: Training linear binary latent VAEs with Gaussian likelihoods with $K = 2$ (except for RELAX) on non-binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.

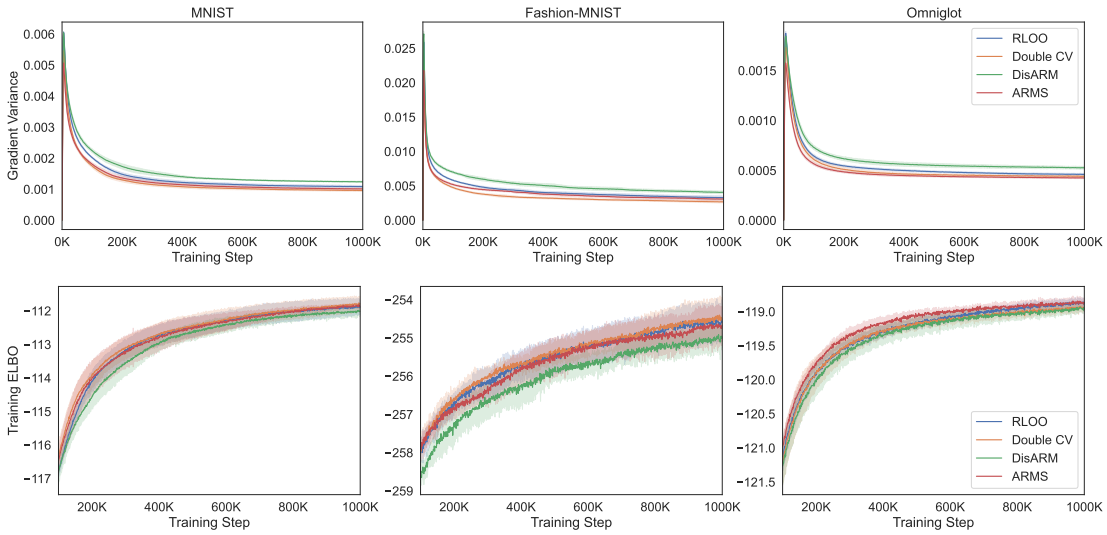


Figure 11: Training linear binary latent VAEs with Bernoulli likelihoods with $K = 4$ (except for RELAX) on dynamically binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.

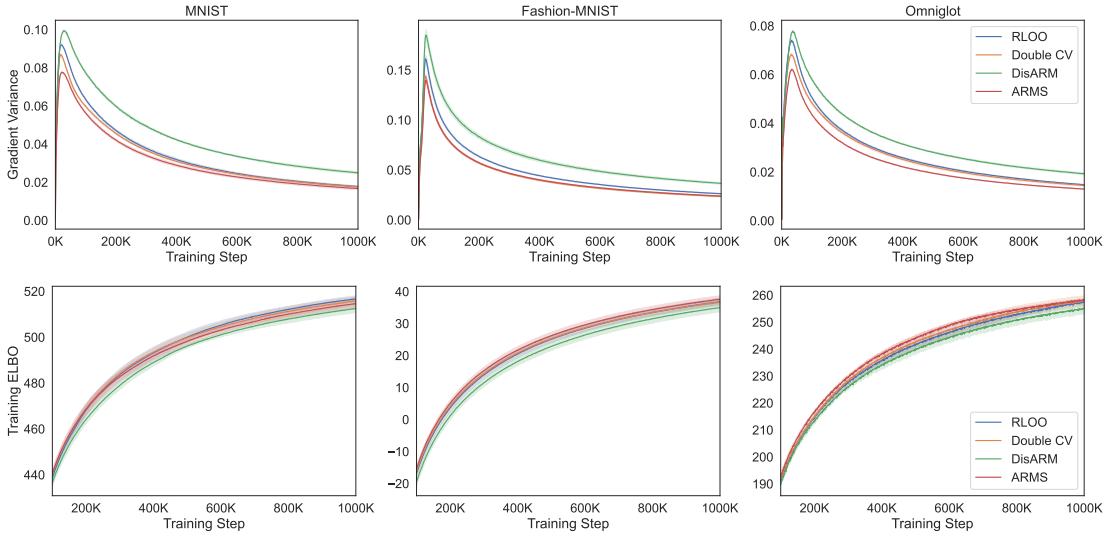


Figure 12: Training linear binary latent VAEs with Gaussian likelihoods with $K = 4$ (except for RELAX) on non-binarized MNIST, Fashion-MNIST, and Omniglot. *Top:* Variance of gradient estimates. *Bottom:* Average ELBO on training examples.