DisARM: An Antithetic Gradient Estimator for Binary Latent Variables

Zhe Dong [1] **Andriy Mnih** [2] **George Tucker** [1]

1. Introduction

Training models with discrete latent variables is challenging due to the difficulty of estimating the gradients accurately. Much of the recent progress has been achieved by taking advantage of continuous relaxations of the system, which are not always available or even possible. The Augment-REINFORCE-Merge (ARM) estimator (Yin and Zhou, 2019) provides an alternative that, instead of relaxation, uses continuous augmentation. Applying antithetic sampling over the augmenting variables yields a relatively low-variance and unbiased estimator applicable to any model with binary latent variables. However, while antithetic sampling reduces variance, the augmentation process increases variance. We show that ARM can be improved by analytically integrating out the randomness introduced by the augmentation process, guaranteeing substantial variance reduction. Our estimator, *DisARM*, is simple to implement and has the same computational cost as ARM. We evaluate DisARM on several generative modeling benchmarks and show that it consistently outperforms ARM and a strong independent sample baseline in terms of both variance and log-likelihood.

2. Background

We consider the problem of optimizing

$$\mathbb{E}_{q_\theta(\mathbf{b})}\left[f_\theta(\mathbf{b})\right], \tag{1}$$

w.r.t. with the parameters θ of a factorial Bernoulli distribution $q_\theta(\mathbf{b})$. The gradient with respect to θ is

$$\begin{aligned}&\nabla_\theta E_{q_\theta(\mathbf{b})}\left[f_\theta(\mathbf{b})\right]\\ &= E_{q_\theta(\mathbf{b})}\left[f_\theta(\mathbf{b})\nabla_\theta \log q_\theta(\mathbf{b}) + \nabla_\theta f_\theta(\mathbf{b})\right].\end{aligned} \tag{2}$$

The second term can typically be estimated with a single Monte Carlo sample, so for notational clarity, we omit the dependence of f on θ in the following sections. Monte Carlo estimates of the first term can have large variance. Low-variance, unbiased estimators of the first term will be our focus.

Yin and Zhou (2019) use an antithetically coupled pair of samples to derive the ARM estimator. Antithetic sampling can reduce the variance of a Monte Carlo estimate if it induces negative covariance between the integrand evaluations (Owen, 2013). While we have no control over f, we can exploit properties of the score function $\nabla_\theta \log q_\theta(b)$. Buesing et al. (2016) show that for "location-scale" distributions, antithetically coupled samples have perfectly negatively correlated score functions, which suggests that using antithetic samples to estimate the gradient will be favorable. Unfortunately, the Bernoulli distribution is not a location-scale distribution, so this result is not immediately applicable.

However, the Bernoulli distribution can be reparameterized in terms of the Logistic distribution which is a location-scale distribution. In other words, when $z \sim \text{Logistic}(\alpha_\theta, 1)$, then $b = \mathbb{1}_{z>0} \sim \text{Bernoulli}(\sigma(\alpha_\theta))$, where $\sigma(x)$ is the Logistic function. We also have

$$\begin{aligned}&\mathbb{E}_{q_\theta(b)}\left[f(b)\nabla_\theta \log q_\theta(b)\right]\\ &= \nabla_\theta \mathbb{E}_{q_\theta(b)}\left[f(b)\right] = \nabla_\theta \mathbb{E}_{q_\theta(z)}\left[f(\mathbb{1}_{z>0})\right]\\ &= \mathbb{E}_{q_\theta(z)}\left[f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z)\right].\end{aligned}$$

This suggests sampling an antithetically coupled pair $(z, \tilde{z})$[1] and forming the estimator

$$\begin{aligned}&g_{\text{ARM}}(z, \tilde{z})\\ &= \tfrac{1}{2}\left(f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z) + f(\mathbb{1}_{\tilde{z}>0})\nabla_\theta \log q_\theta(\tilde{z})\right)\\ &= \tfrac{1}{2}(f(\mathbb{1}_{z>0}) - f(\mathbb{1}_{\tilde{z}>0}))\nabla_\theta \log q_\theta(z)\\ &= \tfrac{1}{2}(f(\mathbb{1}_{1-u<\sigma(\alpha_\theta)}) - f(\mathbb{1}_{u<\sigma(\alpha_\theta)}))\,(2u-1)\,\nabla_\theta \alpha_\theta\end{aligned} \tag{3}$$

where $u = \sigma(\alpha_\theta - z)$ and we use the fact that $\nabla_\theta \log q_\theta(z) = -\nabla_\theta \log q_\theta(\tilde{z})$ (Buesing et al., 2016) because the Logistic distribution is a location-scale distribution. This is the ARM estimator (Yin and Zhou, 2019). Notably, ARM only evaluates f at discrete values, so does not require a continuous relaxation. We expect such an estimator to have low variance because the learning signal is a difference of evaluations of f and Yin and Zhou (2019) empirically show that it performs comparably or outperforms

[1]Google Research, Brain Team, Mountain View, California, USA [2]DeepMind, London, United Kingdom. Correspondence to: Zhe Dong <zhedong@google.com>.

Negative Dependence and Submodularity for ML, ICML 2020 Workshop.

[1]In other words, drawing $\epsilon \sim \text{Logistic}(0, 1)$, then setting $z = \epsilon + \alpha_\theta$ and $\tilde{z} = -\epsilon + \alpha_\theta$.

previous methods. In the scalar setting, ARM is not useful because the exact gradient can be computed with 2 function evaluations, however, ARM can naturally be extended to the multi-dimensional setting with only 2 function evaluations

$$\tfrac{1}{2}(f(\mathbf{b}) - f(\tilde{\mathbf{b}}))\,(2\mathbf{u}-1)\,\nabla_\theta \alpha_\theta, \tag{4}$$

whereas the exact gradient requires exponentially many function evaluations.

3. DisARM

Requiring a reparameterization in terms of a continuous variable seems unnatural when the objective (Eq. 1) only depends on the discrete variable. The cost of this reparameterization is an increase in variance. In fact, the variance of $f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z)$ is at least as large as the variance of $f(b)\nabla_\theta \log q_\theta(b)$ because

$$f(b)\nabla_\theta \log q_\theta(b) = \mathbb{E}_{q_\theta(z|b)}\left[f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z)\right], \tag{5}$$

hence

$$\begin{aligned}&\mathrm{Var}(f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z))\\ &= \mathrm{Var}(f(b)\nabla_\theta \log q_\theta(b))\\ &+ \mathbb{E}_b\left[\mathrm{Var}_{z|b}(f(\mathbb{1}_{z>0})\nabla_\theta \log q_\theta(z))\right],\end{aligned}$$

i.e., an instance of conditioning (Owen, 2013). So, while ARM reduces variance via antithetic coupling, it also increases variance due to the reparameterization. It is not clear that this translates to an overall reduction in variance. In fact, as we show empirically, a two-independent-samples REINFORCE estimator with a leave-one-out baseline performs comparably or outperforms the ARM estimator (e.g., Table 1).

The relationship in Eq. 5 suggests that it might be possible to perform a similar operation on the ARM estimator. Indeed, the key insight is to simultaneously condition on the pair $(b, \tilde{b}) = (\mathbb{1}_{z>0}, \mathbb{1}_{\tilde{z}>0})$. First, we derive the result for scalar b, then extend it to the multi-dimensional setting. Integrating out z conditional on $(b, \tilde{b})$, results in our proposed estimator

$$\begin{aligned}&g_{\mathrm{DisARM}}(b,\tilde{b}) \coloneqq \mathbb{E}_{q(z|b,\tilde{b})}\left[g_{\mathrm{ARM}}\right]\\ &= \tfrac{1}{2}\mathbb{E}_{q(z|b,\tilde{b})}\left[(f(\mathbb{1}_{z>0}) - f(\mathbb{1}_{\tilde{z}>0}))\nabla_\theta \log q_\theta(z)\right]\\ &= \tfrac{1}{2}(f(b) - f(\tilde{b}))\mathbb{E}_{q(z|b,\tilde{b})}\left[\nabla_\theta \log q_\theta(z)\right]\\ &= \tfrac{1}{2}(f(b) - f(\tilde{b}))\left((-1)^{\tilde{b}}\mathbb{1}_{b\neq\tilde{b}}\sigma(|\alpha_\theta|)\right)\nabla_\theta\alpha_\theta.\end{aligned} \tag{6}$$

See Appendix A for a detailed derivation. Note that $\mathbb{E}_{q(z|b,\tilde{b})}\left[\nabla_\theta \log q_\theta(z)\right]$ vanishes when $b = \tilde{b}$. While this does not matter for the scalar case, it will prove useful for the multi-dimensional case. We call the estimator DisARM because it integrates out the continuous randomness in ARM and only retains the *discrete* component. Similarly to above, we have that the variance of DisARM is upper bounded by the variance of ARM

$$\begin{aligned}\mathrm{Var}(g_{\mathrm{ARM}}) &= \mathrm{Var}(g_{\mathrm{DisARM}}) + \mathbb{E}_{b,\tilde{b}}\left[\mathrm{Var}_{z|b,\tilde{b}}(g_{\mathrm{ARM}})\right]\\ &\geq \mathrm{Var}(g_{\mathrm{DisARM}}).\end{aligned}$$

3.1. Multi-dimensional case

Now, consider the case where $\mathbf{b}$ is multi-dimensional. Although the distribution is factorial, f may be a complex nonlinear function. Focusing on a single dimension of α_θ, we have

$$\begin{aligned}&\nabla_{(\alpha_\theta)_i}\mathbb{E}_{q_\theta(\mathbf{b})}\left[f(\mathbf{b})\right] = \nabla_{(\alpha_\theta)_i}\mathbb{E}_{\mathbf{b}_i}\left[\mathbb{E}_{\mathbf{b}_{-i}}\left[f(\mathbf{b}_{-i},\mathbf{b}_i)\right]\right]\\ &= \mathbb{E}_{\mathbf{b}_i,\tilde{\mathbf{b}}_i}\left[\tfrac{1}{2}\left(\mathbb{E}_{\mathbf{b}_{-i}}\left[f(\mathbf{b}_{-i},\mathbf{b}_i)\right] - \mathbb{E}_{\mathbf{b}_{-i}}[f(\mathbf{b}_{-i},\tilde{\mathbf{b}}_i)]\right)\right.\\ &\left.\cdot\left((-1)^{\tilde{\mathbf{b}}_i}\mathbb{1}_{\mathbf{b}_i\neq\tilde{\mathbf{b}}_i}\sigma(|(\alpha_\theta)_i|)\right)\right],\end{aligned}$$

which follows from applying Eq. 6 where the function is now $\mathbb{E}_{\mathbf{b}_{-i}}\left[f(\mathbf{b}_{-i},\mathbf{b}_i)\right]$, and $\mathbf{b}_{-i}$ denotes the vector of samples obtained by leaving out ith dimension. Then, because expectations are linear, we can couple the inner expectations

$$\begin{aligned}&\mathbb{E}_{\mathbf{b}_i,\tilde{\mathbf{b}}_i}\left[\tfrac{1}{2}(\mathbb{E}_{\mathbf{b}_{-i}}\left[f(\mathbf{b}_{-i},\mathbf{b}_i)\right] - \mathbb{E}_{\mathbf{b}_{-i}}[f(\mathbf{b}_{-i},\tilde{\mathbf{b}}_i)])\right.\\ &\left.\left((-1)^{\tilde{\mathbf{b}}_i}\mathbb{1}_{\mathbf{b}_i\neq\tilde{\mathbf{b}}_i}\sigma(|(\alpha_\theta)_i|)\right)\right]\\ &= \mathbb{E}_{\mathbf{b}_i,\tilde{\mathbf{b}}_i}\left[\tfrac{1}{2}(\mathbb{E}_{\mathbf{b}_{-i},\mathbf{b}'_{-i}}[f(\mathbf{b}_{-i},\mathbf{b}_i) - f(\mathbf{b}'_{-i},\tilde{\mathbf{b}}_i)])\right.\\ &\left.\left((-1)^{\tilde{\mathbf{b}}_i}\mathbb{1}_{\mathbf{b}_i\neq\tilde{\mathbf{b}}_i}\sigma(|(\alpha_\theta)_i|)\right)\right],\end{aligned}$$

where we are free to choose any joint distribution on $(\mathbf{b}_{-i}, \mathbf{b}'_{-i})$ that maintains the marginal distributions. A natural choice satisfying this constraint is to draw $(\mathbf{b}, \tilde{\mathbf{b}})$ as an antithetic pair (independently for each dimension), then we can form the multi-dimensional DisARM estimator of ∇_{α_θ}

$$g_{\mathrm{DisARM},i} \coloneqq \tfrac{1}{2}(f(\mathbf{b}) - f(\tilde{\mathbf{b}}))(-1)^{\tilde{\mathbf{b}}_i}\mathbb{1}_{\mathbf{b}_i\neq\tilde{\mathbf{b}}_i}\sigma(|(\alpha_\theta)_i|).$$

Notably, whenever $\mathbf{b}_i = \tilde{\mathbf{b}}_i$, the gradient estimator vanishes exactly. In contrast, the multi-dimensional ARM estimator of $\nabla_{(\alpha_\theta)_i}$ (Eq. 4) vanishes only when $\mathbf{b} = \tilde{\mathbf{b}}$ in all dimensions, which occurs seldomly when $\mathbf{b}$ is high dimensional. The estimator for ∇_θ is obtained by summing over i:

$$g_{\mathrm{DisARM}}(\mathbf{b},\tilde{\mathbf{b}}) = \textstyle\sum_i\left(g_{\mathrm{DisARM},i}\cdot\nabla_\theta(\alpha_\theta)_i\right). \tag{7}$$

4. Experimental Results

Our goal was variance reduction to improve optimization, so we compare DisARM to the state-of-the-art methods: ARM (Yin and Zhou, 2019) and RELAX (Grathwohl et al.,

Code and additional information: `https://sites.google.com/view/disarm-estimator`.

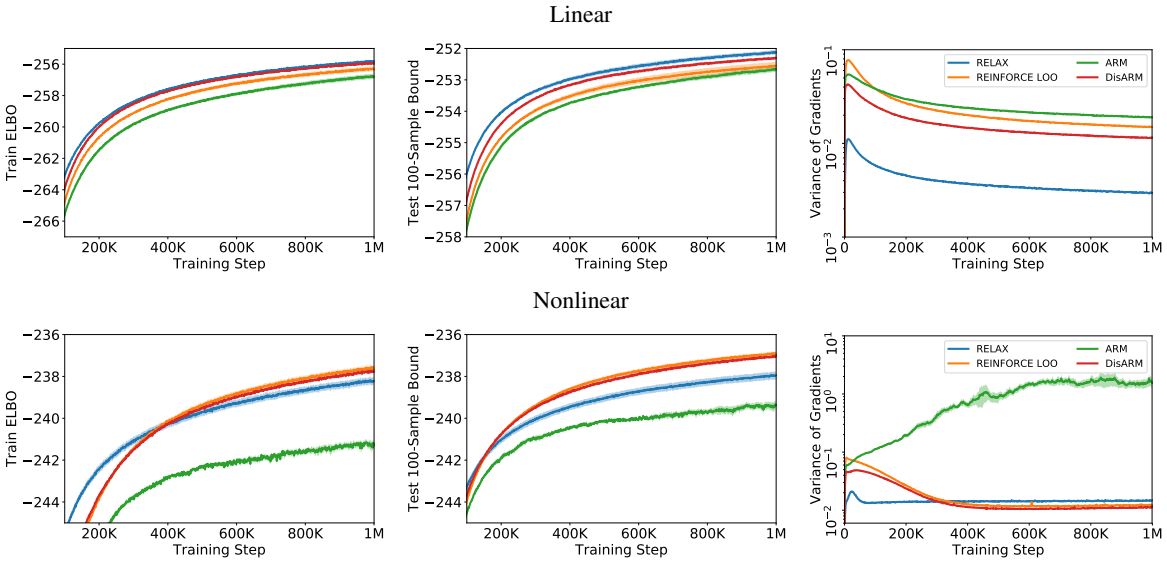

Figure 1: Training a Bernoulli VAE on FashionMNIST dataset by maximizing the ELBO. We plot the train ELBO (left column), test 100-sample bound (middle column), and the variance of gradient estimator (right column) for the linear (top row) and nonlinear (bottom row) models. We plot the mean and one standard error based on 5 runs from different random initializations. Results on MNIST and Omniglot were qualitatively similar (Appendix Figure 3).

2018) As we mentioned before, ARM and DisARM are more generally applicable than RELAX, however, we include it for comparison. We also include a two-independent-sample REINFORCE estimator with a leave-one-out baseline (REINFORCE LOO, Kool et al., 2019). This is a simple, but competitive method that has been omitted from previous works.

We train a VAE with Bernoulli latent variables, which is used as a gradient estimator benchmark for discrete latent variables. We evaluate the gradient estimators on three benchmark generative modeling datasets: MNIST, Fashion-MNIST and Omniglot. As our goal is optimization, we use dynamic binarization to avoid overfitting and we largely find that training performance mirrors test performance. We use the standard split into train, validation, and test sets. See Appendix C for further implementation details.

We use the same model architecture as Yin and Zhou (2019). Briefly, we considered linear and nonlinear models. The nonlinear model used fully connected neural networks with two hidden layers of 200 leaky ReLU units (Maas et al., 2013). Both models had a single stochastic layer of 200 Bernoulli latent variables. The models were trained with Adam (Kingma and Ba, 2015) using a learning rate 10^{-4} on mini-batches of 50 examples for 10^6 steps.

During training, we measure the training ELBO, the 100-sample bound on the test set, and the variance of the gradient estimator for the inference network averaged over parameters[2] and plot the results in Figure 1 for FashionMNIST and Appendix Figure 3 for MNIST and Omniglot. We report the final results in Table 1. We find a substantial performance gap between ARM and REINFORCE LOO, DisARM, or RELAX across all measures and configurations. We compared our implementation of ARM with the open-source implementation provided by Yin and Zhou (2019) and find that it replicates their results. Yin and Zhou (2019) evaluate performance on the *statically binarized* MNIST dataset, which is well known for overfitting and substantial overfitting is observed in their results. In such a situation, a method that performs worse at optimization may lead to better generalization. Additionally, they report the variance of the gradient estimator w.r.t. logits of the latent variables instead, which explains the discrepancy in the variance plots. Unlike the inference network parameter gradients, the logit gradients have no special significance as they are backpropagated into the inference network rather than used to update parameters directly. Finally, they use a different network architecture than the methods they compare to, so their results are not directly comparable to previously reported numbers. We use the same architecture across methods and implement the estimators in the same framework to ensure a fair comparison.

DisARM has reduced gradient estimator variance over REINFORCE LOO across all models and datasets. This translates to consistent improvements over REINFORCE LOO

[2]Estimated by approximating moments with an exponential moving average with decay rate 0.999.

Table 1: Mean variational lower bounds and the standard error of the mean computed based on 5 runs from different random initializations. The best performing method (up to the standard error) for each task is in bold.

Train ELBO				
Dynamic MNIST	REINFORCE LOO	ARM	DisARM	RELAX
Linear	-116.57 ± 0.15	-117.66 ± 0.04	$\mathbf{-116.30 \pm 0.08}$	-115.93 ± 0.15
Nonlinear	$\mathbf{-102.45 \pm 0.12}$	-107.32 ± 0.28	$\mathbf{-102.56 \pm 0.19}$	-102.53 ± 0.15
Fashion MNIST				
Linear	-256.33 ± 0.14	-256.80 ± 0.16	$\mathbf{-255.97 \pm 0.07}$	-255.83 ± 0.03
Nonlinear	$\mathbf{-237.66 \pm 0.11}$	-241.30 ± 0.10	$\mathbf{-237.77 \pm 0.08}$	-238.23 ± 0.17
Omniglot				
Linear	-121.66 ± 0.10	-122.45 ± 0.10	$\mathbf{-121.15 \pm 0.12}$	-120.79 ± 0.09
Nonlinear	$\mathbf{-115.26 \pm 0.15}$	-118.76 ± 0.05	$\mathbf{-115.08 \pm 0.11}$	-116.56 ± 0.15
Test 100-sample bound				
Dynamic MNIST	REINFORCE LOO	ARM	DisARM	RELAX
Linear	$\mathbf{-109.25 \pm 0.09}$	-109.70 ± 0.05	$\mathbf{-109.13 \pm 0.04}$	-108.76 ± 0.06
Nonlinear	$\mathbf{-97.41 \pm 0.09}$	-101.15 ± 0.39	$\mathbf{-97.52 \pm 0.11}$	-97.76 ± 0.11
Fashion MNIST				
Linear	-252.55 ± 0.12	-252.66 ± 0.07	$\mathbf{-252.30 \pm 0.05}$	-252.13 ± 0.06
Nonlinear	$\mathbf{-236.94 \pm 0.09}$	-239.37 ± 0.15	$\mathbf{-237.02 \pm 0.07}$	-237.95 ± 0.16
Omniglot				
Linear	-117.70 ± 0.10	-118.01 ± 0.06	$\mathbf{-117.39 \pm 0.09}$	-117.10 ± 0.08
Nonlinear	$\mathbf{-114.39 \pm 0.21}$	-116.56 ± 0.07	$\mathbf{-114.26 \pm 0.14}$	-116.28 ± 0.26

with linear models and comparable performance on the nonlinear models across all datasets. For linear networks, RELAX achieves lower gradient estimator variance and better performance. However, this does not hold for nonlinear networks. For nonlinear networks across three datasets, RELAX initially has lower variance gradients, but DisARM overtakes it as training proceeds. Furthermore, training the model on a P100 GPU was nearly twice as slow for RELAX, while ARM, DisARM and REINFORCE LOO trained at the same speed. This is consistent with previous findings (Yin and Zhou, 2019).

5. Multi-sample Variational Bounds

We also derive a local version of DisARM designed for optimizing the multi-sample variational bound in Appendix B. In Appendix D.2, we show that it outperforms VIMCO (Mnih and Rezende, 2016), the current state-of-the-art gradient estimator.

6. Discussion

We have introduced DisARM, an unbiased, low-variance gradient estimator for Bernoulli random variables based on antithetic sampling. Our starting point was the ARM estimator (Yin and Zhou, 2019), which reparameterizes Bernoulli variables in terms of Logistic variables and estimates the REINFORCE gradient over the Logistic variables using antithetic sampling. Our key insight is that the ARM estimator involves unnecessary randomness because it operates on the augmenting Logistic variables instead of the original Bernoulli ones. In other words, ARM is competitive despite rather than because of the Logistic augmentation step, and its low variance is completely due to the use of antithetic sampling. We derive DisARM by integrating out the augmenting variables from ARM using a variance reduction technique known as conditioning. As a result, DisARM has lower variance than ARM and consistently outperforms it. Then, we extended DisARM to the multi-sample objective and showed that it outperformed the state-of-the-art method. Given DisARM's generality and simplicity, we expect it to be widely useful.

While relaxation-based estimators (e.g., REBAR and RELAX) can outperform DisARM in some cases, DisARM is always competitive and more generally applicable as it does not rely on a continuous relaxation. In the future, it would be interesting to investigate how to combine the strengths of DisARM with those of relaxation-based estimators in a single estimator. Finally, ARM has been extended to categorical variables (Yin et al., 2019) and future work could extend DisARM similarly.

References

Buesing, L., Weber, T., and Mohamed, S. (2016). Stochastic gradient estimation with finite differences. In *NIPS2016 Workshop on Advances in Approximate Inference*.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2016). Importance weighted autoencoders. In *Proceedings of the 4th International Conference on Learning Representations*.

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*.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*.

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

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

Mnih, A. and Rezende, D. (2016). Variational inference for monte carlo objectives. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 2188–2196.

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

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. In *Advances in Neural Information Processing Systems 30*.

Yin, M., Yue, Y., and Zhou, M. (2019). ARSM: Augment-REINFORCE-swap-merge estimator for gradient backpropagation through categorical variables. In *Proceedings of the 36th International Conference on Machine Learning*.

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

A. DisARM Derivation

To finish the derivation of Eq. 6, we need to compute

$$\begin{aligned}
&\mathbb{E}_{q(z|b,\tilde{b})}\left[\nabla_\theta \log q_\theta(z)\right] \\
&= \mathbb{E}_{q(z|b,\tilde{b})}\left[1 - \frac{2\exp(-(z-\alpha_\theta))}{1+\exp(-(z-\alpha_\theta)}\right]\nabla_\theta \alpha_\theta \\
&= \mathbb{E}_{q(u|b,\tilde{b})}\left[2u-1\right]\nabla_\theta\alpha_\theta = \left(2\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right]-1\right)\nabla_\theta\alpha_\theta,
\end{aligned}$$

where we have used the change of variables $z = \log(u) - \log(1-u) + \alpha_\theta$. This is a common reparameterization of a Logistic variable in terms of a Uniform variable, so when $z \sim \text{Logistic}(\alpha_\theta, 1)$, then $u \sim \text{Uniform}(0,1)$. Thus, the joint distribution $q(u, b, \tilde{b})$ is generated by sampling $u \sim \text{Uniform}(0,1)$ and setting $b = \mathbb{1}_{z>0} = \mathbb{1}_{1-u<\sigma(\alpha_\theta)}$ and $\tilde{b} = \mathbb{1}_{\tilde{z}>0} = \mathbb{1}_{u<\sigma(\alpha_\theta)}$. Conditioning on $b, \tilde{b}$ imposes constraints on the value of u, hence $q(u|b,\tilde{b})$ is a truncated Uniform variable. To understand $\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right]$, it suffices to enumerate the possibilities:

- $b=0,\ \tilde{b}=0$ implies $\sigma(\alpha_\theta) < u < \sigma(-\alpha_\theta)$, which is symmetric around $\frac{1}{2}$, so $\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right] = \frac{1}{2}$.
- $b=1,\ \tilde{b}=1$ implies $\sigma(-\alpha_\theta) < u < \sigma(\alpha_\theta)$, which is symmetric around $\frac{1}{2}$, so $\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right] = \frac{1}{2}$.
- $b=0,\ \tilde{b}=1$ implies $u < \min(\sigma(-\alpha_\theta), \sigma(\alpha_\theta)) = 1-\sigma(|\alpha_\theta|)$. Thus,
$$\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right] = \frac{1-\sigma(|\alpha_\theta|)}{2}.$$
- $b=1,\ \tilde{b}=0$ implies $u > \max(\sigma(-\alpha_\theta), \sigma(\alpha_\theta)) = \sigma(|\alpha_\theta|)$. Thus,
$$\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right] = \frac{1+\sigma(|\alpha_\theta|)}{2}.$$

Combining the cases, we have that

$$2\mathbb{E}_{q(u|b,\tilde{b})}\left[u\right] - 1 = (-1)^{\tilde{b}}\mathbb{1}_{b\neq\tilde{b}}\sigma(|\alpha_\theta|).$$

B. Multi-sample Variation Bounds

B.1. Background

Objectives of the form Eq. 1 are often used in variational inference for discrete latent variable models. For example, to fit the parameters of a discrete latent variable model $p_\theta(x, \mathbf{b})$, we can lower bound the log marginal likelihood $\log p_\theta(x) \geq \mathbb{E}_{q_\theta(\mathbf{b}|x)}\left[\log p_\theta(x,\mathbf{b}) - \log q_\theta(\mathbf{b}|x)\right]$, where $q_\theta(\mathbf{b}|x)$ is a variational distribution. Burda et al. (2016) introduced an improved multi-sample variational bound that

reduces to the ELBO when $K = 1$ and converges to the log marginal likelihood as $K \to \infty$

$$\mathcal{L} \coloneqq \mathbb{E}_{\prod_k p_\theta(\mathbf{b}^k)} \left[\log \tfrac{1}{K} \sum_k w(\mathbf{b}^k) \right],$$

where $w(\mathbf{b}) = \frac{p(\mathbf{b},x)}{q(\mathbf{b}|x)}$. We omit the dependence of w on θ because it is straightforward to account for.

In this case, Mnih and Rezende (2016) introduced a gradient estimator, VIMCO, that uses specialized control variates that take advantage of the structure of the objective

$$\sum_k \left(\log \tfrac{1}{K} \sum_j w(\mathbf{b}^j) - \log \tfrac{1}{K-1} \sum_{j \neq k} w(\mathbf{b}^j) \right) \cdot \nabla_\theta \log q_\theta(\mathbf{b}^k|x),$$

which is unbiased because the second term has zero expectation: $\mathbb{E}_{\prod_k q_\theta(\mathbf{b}^k|x)}[\cdot] = 0$.

B.2. DisARM Extension to Multi-sample Variation Bounds

We could naïvely apply DisARM to the multi-sample objective, however, our preliminary experiments did not suggest this improved performance over VIMCO. However, we can obtain an estimator similar to VIMCO (Mnih and Rezende, 2016) by applying DisARM to the multi-sample objective *locally*, once for each sample. Recall that in this setting, our objective is the multi-sample variational lower bound (Burda et al., 2016)

$$\begin{aligned}\mathcal{L} &\coloneqq \mathbb{E}_{\prod_k q_\theta(\mathbf{b}^k)} \left[\log \frac{1}{K} \sum_k w(\mathbf{b}^k) \right] \\ &= \mathbb{E}_{\prod_k q_{\theta^k}(\mathbf{b}^k)} \left[\log \frac{1}{K} \sum_k w(\mathbf{b}^k) \right],\end{aligned}$$

where to simplify notation, we introduced dummy variables $\theta^k = \theta$, so that $\nabla_\theta \mathcal{L} = \sum_k \frac{\partial \mathcal{L}}{\partial \theta^k}$. Now, let $f_{\mathbf{b}^{-k}}(\mathbf{d}) = \log \frac{1}{K} \left(\sum_{\mathbf{c} \in \mathbf{b}^{-k}} w(\mathbf{c}) + w(\mathbf{d}) \right)$ with $\mathbf{b}^{-k} \coloneqq (\mathbf{b}^1, \ldots, \mathbf{b}^{k-1}, \mathbf{b}^{k+1}, \ldots, \mathbf{b}^K)$, so that

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \theta^k} &= \frac{\partial \mathcal{L}}{\partial \theta^k} \mathbb{E}_{\mathbf{b}^k} \left[\mathbb{E}_{\mathbf{b}^{-k}} \left[f_{\mathbf{b}^{-k}}(\mathbf{b}^k) \right] \right] \\ &= \frac{\partial \mathbb{E}_{\mathbf{b}^k} \left[\mathbb{E}_{\mathbf{b}^{-k}} \left[f_{\mathbf{b}^{-k}}(\mathbf{b}^k) \right] \right]}{\partial \alpha_{\theta^k}} \frac{\partial \alpha_{\theta^k}}{\partial \theta^k}.\end{aligned}$$

Then by applying Eq. 7 to $\mathbb{E}_{\mathbf{b}^{-k}}[f_{\mathbf{b}^{-k}}]$, we have that $\left(\frac{\partial \mathcal{L}}{\partial \alpha_{\theta^k}} \mathbb{E}_{\mathbf{b}^k} \left[\mathbb{E}_{\mathbf{b}^{-k}} \left[f_{\mathbf{b}^{-k}}(\mathbf{b}^k) \right] \right] \right)_i$ is

$$\mathbb{E}_{\mathbf{b}^k, \tilde{\mathbf{b}}^k} \left[\tfrac{1}{2} \left(\mathbb{E}_{\mathbf{b}^{-k}} \left[f_{\mathbf{b}^{-k}}(\mathbf{b}^k) \right] - \mathbb{E}_{\mathbf{b}^{-k}} \left[f_{\mathbf{b}^{-k}}(\tilde{\mathbf{b}}^k) \right] \right) \left(\mathbb{1}_{\mathbf{b}_i^k \neq \tilde{\mathbf{b}}_i^k} (-1)^{\tilde{\mathbf{b}}_i^k} \sigma(|(\alpha_{\theta^k})_i|) \right) \right].$$

We can form an unbiased estimator by drawing K antithetic pairs $\mathbf{b}^1, \tilde{\mathbf{b}}^1, \ldots, \mathbf{b}^K, \tilde{\mathbf{b}}^K$ and forming

$$\frac{1}{4} \left(f_{\mathbf{b}^{-k}}(\mathbf{b}^k) - f_{\mathbf{b}^{-k}}(\tilde{\mathbf{b}}^k) + f_{\tilde{\mathbf{b}}^{-k}}(\mathbf{b}^k) - f_{\tilde{\mathbf{b}}^{-k}}(\tilde{\mathbf{b}}^k) \right) \left(\mathbb{1}_{\mathbf{b}_i^k \neq \tilde{\mathbf{b}}_i^k} (-1)^{\tilde{\mathbf{b}}_i^k} \sigma(|(\alpha_\theta)_i|) \right), \tag{8}$$

for the gradient of the ith dimension and kth sample. Conveniently, we can compute $w(\mathbf{b}^1), w(\tilde{\mathbf{b}}^1), \ldots, w(\mathbf{b}^K), w(\tilde{\mathbf{b}}^K)$ once and then compute the estimator for all k and i without additional evaluations of w. As a result, the computation associated with this estimator is the same as for VIMCO with $2K$ samples, and thus we use it as a baseline comparison in our experiments. We could average over further configurations to reduce the variance of our estimate of $\mathbb{E}_{\mathbf{b}^{-k}}[f_{\mathbf{b}^{-k}}]$, however, we leave evaluating this to future work.

C. Experimental Details

Input images to the networks were centered with the pixel mean of the training dataset. For the nonlinear network activations, we used leaky rectified linear units (LeakyReLU) (Maas et al., 2013) activations with 0.3 negative slope as in (Yin and Zhou, 2019). The parameters of the inference and generation networks were optimized with Adam (Kingma and Ba, 2015) using learning rate 1×10^{-4}. The logits for the prior distribution $p(b)$ were optimized using SGD with learning rate 1×10^{-2}. For RELAX, we initialize the trainable temperature and scaling factor of the control variate to 0.1 and 1.0, respectively. The learned control variate in RELAX was a single layer neural network with 137 LeakyReLU units. The control variate parameters were also optimized with Adam using learning rate 1×10^{-4}.

D. Additional Experimental Results

D.1. Learning a Toy Model

As a simple illustrative problem, introduced by Tucker et al. (2017), we apply DisARM to maximize $\mathbb{E}_{b \sim \text{Bernoulli}(\sigma(\phi))} \left[(b - p_0)^2 \right]$ with $p_0 \in \{0.49, 0.499, 0.4999\}$, and compare its performance to ARM and REINFORCE LOO in Figure 2[3]. DisARM exhibits lower variance than REINFORCE LOO and ARM, especially for the more difficult versions of the problem as p_0 approaches 0.5.

[3] Yin and Zhou (2019) show that ARM outperforms RELAX on this task, so we omit it.

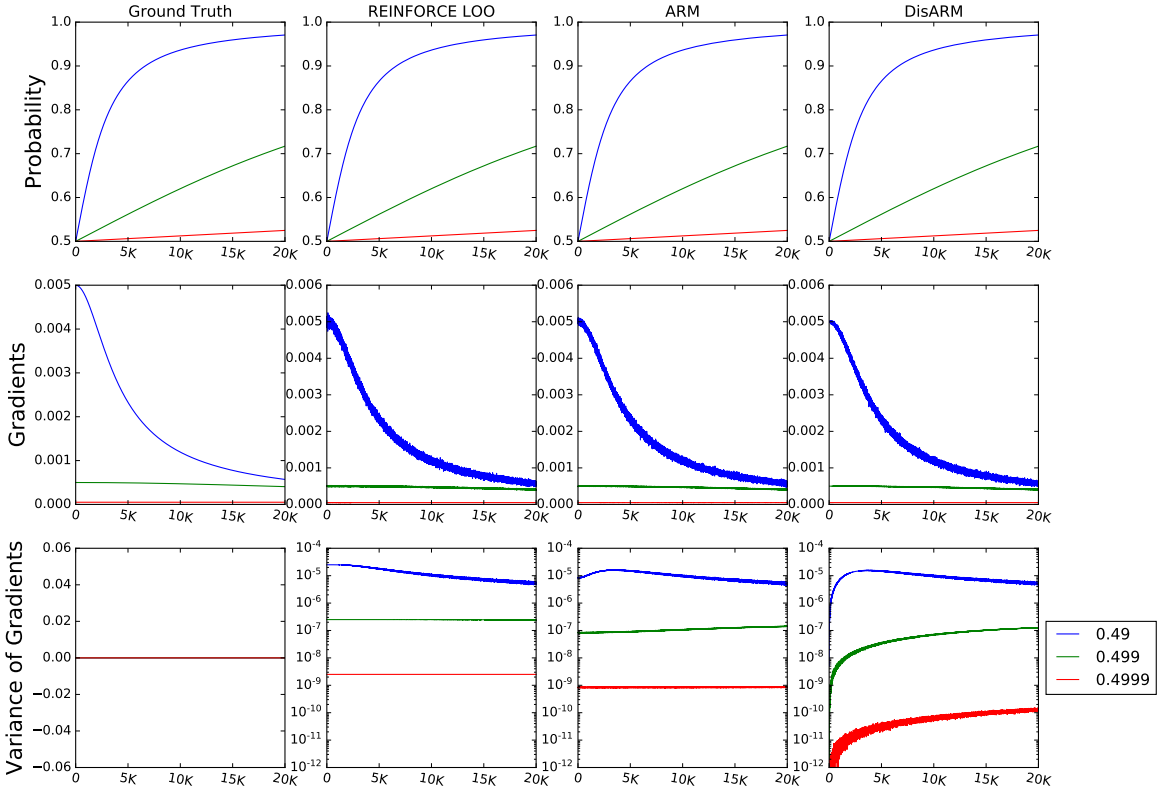

Figure 2: Comparing gradient estimators for the toy problem (Appendix D.1). We plot the trace of the estimated Bernoulli probability $\sigma(\phi)$, the estimated gradients, and the variance of the estimated gradients. The variance is measured based on 5000 Monte-Carlo samples at each iteration.

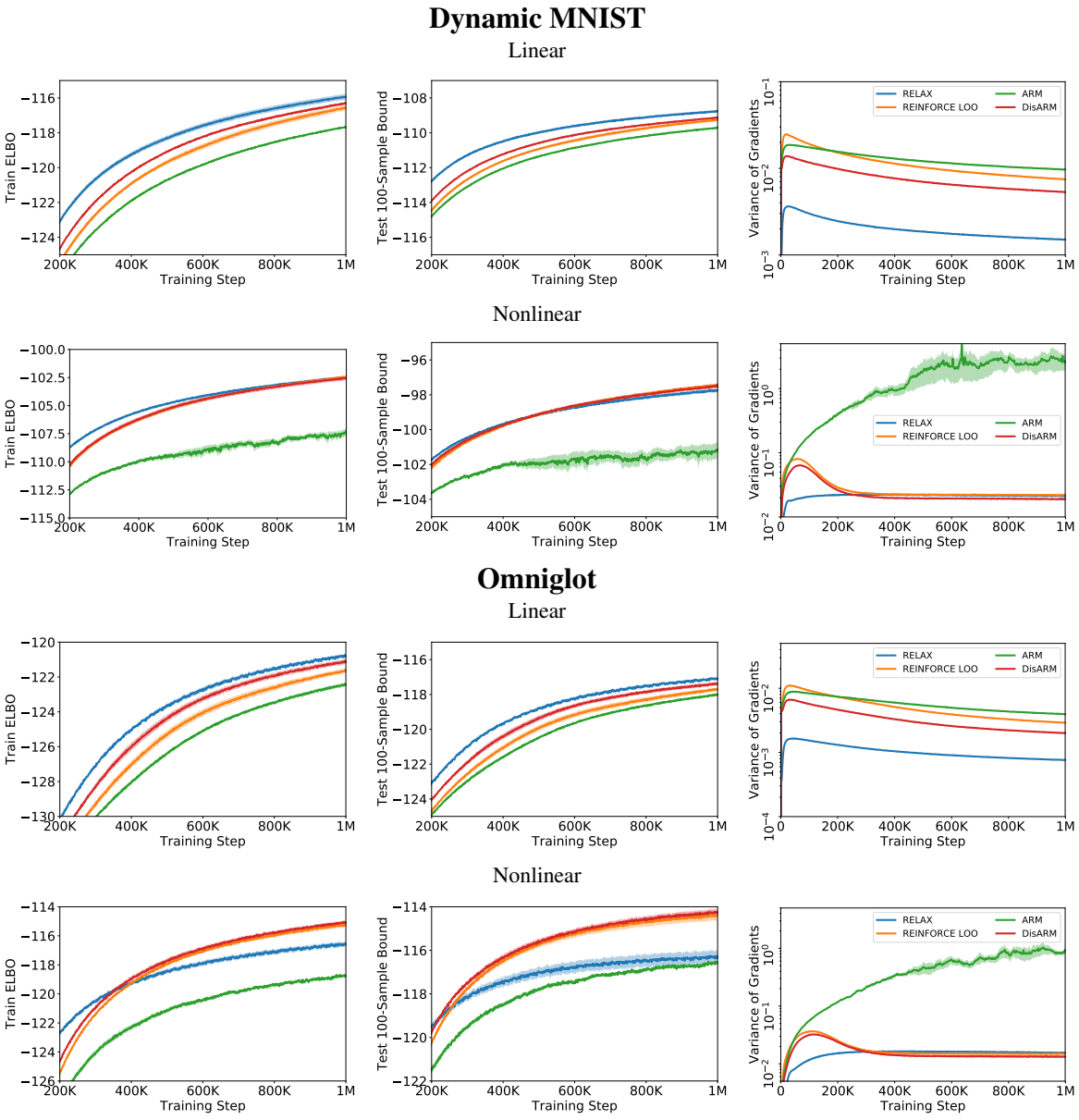

Figure 3: Experimental results for training a Bernoulli VAE with ELBO using RELAX (blue), REINFORCE LOO (orange), ARM (green) and DisARM (red) to update the parameters. We applied the estimators on MNIST and Omniglot with dynamic binarization. We evaluate the ELBO on training set (left column), 100-sample bound on test set (middle column) and the variance of gradients (right column) for linear (top row) and nonlinear (bottom row) models. The mean and standard error (shaded area) are estimated given 5 trials with different random initializations.

D.2. Training a Bernoulli VAE with Multi-sample Bounds

To ensure a fair comparison on computational grounds, we compare the performance of models trained using DisARM with K pairs of antithetic samples to models trained using VIMCO with $2K$ independent samples. For all of the performance results, we use the $2K$-sample bound, which favors VIMCO because this is precisely the objective it maximizes.

In order for a comparison of gradient estimator variances to be meaningful, the estimators must be unbiased estimates of the same gradient. So for the variance comparison, we compare DisARM with K pairs to averaging two independent VIMCO estimators with K samples so that they use the same amount of computation. Furthermore, we compute the variance estimates along the same model trajectory (generated by VIMCO updates).

As shown in Appendix Figure 4, Figure 5, Figure 6 and Appendix Table 2, DisARM consistently improves on VIMCO across different datasets, network settings, and number of samples/pairs.

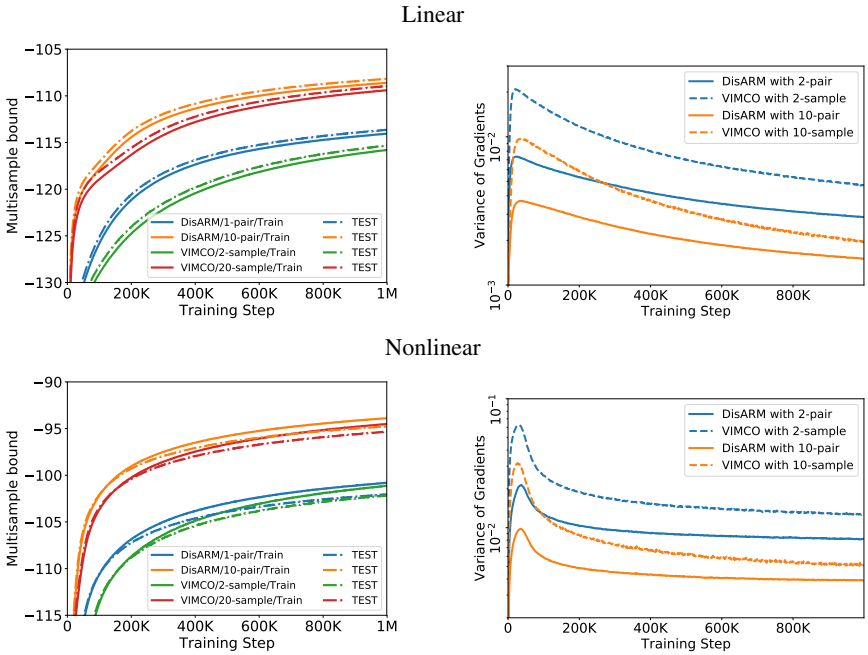

Figure 4: Training a Bernoulli VAE by maximizing the multi-sample variational bound with DisARM and VIMCO on **Dynamic MNIST**. We report the training and test multi-sample bound and the variance of the gradient estimators for the linear (top row) and nonlinear (bottom row) models.

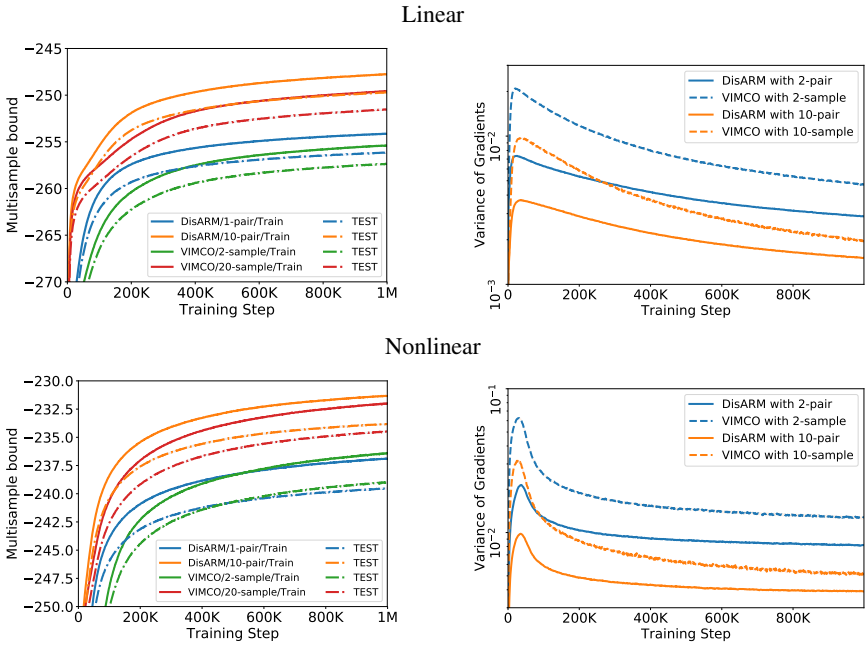

Figure 5: Training a Bernoulli VAE by maximizing the multi-sample variational bound with DisARM and VIMCO on **FashionMNIST**. We report the training and test multi-sample bound and the variance of the gradient estimators for the linear (top row) and nonlinear (bottom row) models.

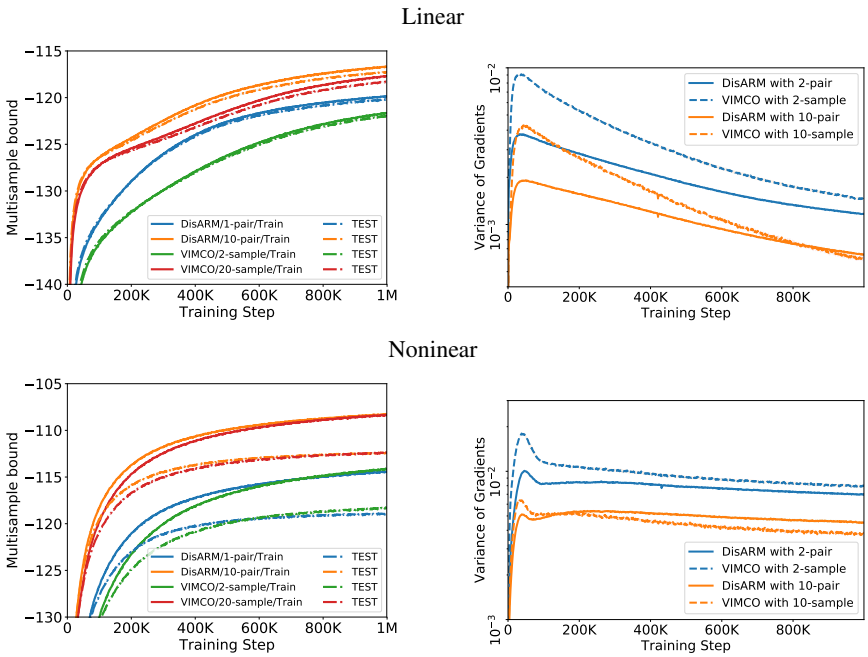

Figure 6: Training a Bernoulli VAE by maximizing the multi-sample variational bound with DisARM and VIMCO on **Omniglot**. We report the training and test multi-sample bound and the variance of the gradient estimators for the linear (top row) and nonlinear (bottom row) models.

Table 2: Comparison of multi-sample bounds. We report the variational lower bound averaged over 5 trials with different random initializations and the standard error of the mean. We bolded the best performing method (up to standard error) for each task. For VIMCO K-samples, we report the K-sample bound and for DisARM K-pairs, we report the $2K$-sample bound for a fair comparison to VIMCO on computational grounds although DisARM is optimizing the K-sample bound.

Train multi-sample bound				
Dynamic MNIST	DisARM 1-pair	VIMCO 2-samples	DisARM 10-pairs	VIMCO 20-samples
Linear	$\mathbf{-114.06 \pm 0.13}$	-115.80 ± 0.08	$\mathbf{-108.61 \pm 0.08}$	-109.40 ± 0.07
Nonlinear	$\mathbf{-100.80 \pm 0.11}$	-101.14 ± 0.10	$\mathbf{-93.89 \pm 0.06}$	-94.52 ± 0.05
Fashion MNIST				
Linear	$\mathbf{-254.15 \pm 0.09}$	-255.41 ± 0.10	$\mathbf{-247.77 \pm 0.08}$	-249.60 ± 0.11
Nonlinear	-236.91 ± 0.10	$\mathbf{-236.41 \pm 0.10}$	$\mathbf{-231.34 \pm 0.06}$	-232.01 ± 0.08
Omniglot				
Linear	$\mathbf{-119.89 \pm 0.06}$	-121.66 ± 0.08	$\mathbf{-116.70 \pm 0.03}$	-117.68 ± 0.07
Nonlinear	-114.45 ± 0.06	$\mathbf{-114.18 \pm 0.07}$	$\mathbf{-108.29 \pm 0.04}$	$\mathbf{-108.37 \pm 0.05}$
Test multi-sample bound				
Dynamic MNIST	DisARM 1-pair	VIMCO 2-samples	DisARM 10-pairs	VIMCO 20-samples
Linear	$\mathbf{-113.63 \pm 0.13}$	-115.31 ± 0.07	$\mathbf{-108.18 \pm 0.08}$	-108.97 ± 0.08
Nonlinear	$\mathbf{-102.03 \pm 0.10}$	$\mathbf{-102.15 \pm 0.11}$	$\mathbf{-94.78 \pm 0.07}$	-95.34 ± 0.06
Fashion MNIST				
Linear	$\mathbf{-256.14 \pm 0.10}$	-257.35 ± 0.12	$\mathbf{-249.71 \pm 0.10}$	-251.52 ± 0.13
Nonlinear	-239.53 ± 0.10	$\mathbf{-238.99 \pm 0.11}$	$\mathbf{-233.82 \pm 0.08}$	-234.47 ± 0.09
Omniglot				
Linear	$\mathbf{-120.23 \pm 0.07}$	-121.99 ± 0.08	$\mathbf{-117.29 \pm 0.04}$	-118.29 ± 0.07
Nonlinear	-118.96 ± 0.07	$\mathbf{-118.36 \pm 0.11}$	$\mathbf{-112.43 \pm 0.07}$	$\mathbf{-112.42 \pm 0.07}$