CARMS: Categorical-Antithetic-REINFORCE Multi-Sample Gradient Estimator

Anonymous Author(s)
Affiliation
Address
email

Abstract

Accurately backpropagating the gradient through categorical variables is a challenging task that arises in various domains, such as training discrete latent variable models. To this end, we propose CARMS, an unbiased estimator for categorical random variables based on multiple mutually negatively correlated (jointly antithetic) samples. CARMS combines REINFORCE with copula based sampling to avoid duplicate samples and reduce the variance, while keeping the estimator unbiased using a simple multiplicative term. It generalizes both the ARMS antithetic estimator for binary variables, which is CARMS for two categories, as well as LOORF/VarGrad, the leave-one-out REINFORCE estimator, which is CARMS with independent samples. We evaluate CARMS on several benchmark datasets on a generative modeling task, as well as a structured output prediction task, and find it to outperform competing methods including a strong self-control baseline. The code is available in the supplementary material.

1 Introduction

When optimizing an expectation based objective of the form $\mathbb{E}_{z\sim q_\phi(z)}[f(z)]$, we sometimes require the gradients with respect to the parameters ϕ of the distribution. This is challenging for discrete variables, because the commonly used reparameterization gradient does not directly work, unless the discrete distribution is approximated by a continuous and reparameterizable one [Jang et al., 2017, Maddison et al., 2017]. A significant part of this field has thus been score function (REINFORCE) based estimators [Glynn, 1990, Williams, 1992, Fu, 2006], which are general and do not require differentiability of f. In this paper, we focus on the case when z is a high dimensional categorical variable with logits ϕ. An common example of this form is the evidence lower bound (ELBO) [Jordan et al., 1998], which arises in variational inference, which is used for training variational autoencoders [Kingma and Welling, 2014, Rezende et al., 2014]. Because their latent space consists of a large number of categorical variables, they require Monte Carlo gradients with respect to the parameters of the stochastic distribution, but have excellent performance, as a categorical VAE has in practice achieved state of the art zero-shot image generation [Ramesh et al., 2021].

Our main contribution is a novel unbiased and low variance gradient estimator for categorical variables. The Categorical-Antithetic-REINFORCE-Multi-Sample (CARMS) estimator uses a copula to generate any number of antithetic (mutually negatively correlated) [Owen, 2013] categorical samples, and constructs an unbiased estimator by combining them into a baseline for variance reduction. This approach is inspired by the ARMS estimator [Dimitriev and Zhou, 2021], which also uses multiple antithetic samples, but only works for binary variables. For two categories, CARMS reduces to it, while for independent samples, CARMS reduces to the leave-one-out-REINFORCE (LOORF) estimator. Our approach achieves higher ELBO with VAE models than the state of the art, as well as higher log likelihood for conditional image completion.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

Related work One widely used group of gradient estimators for categorical variables is based on trading off bias for lower variance. This includes the straight through (ST) estimator [Bengio et al., 2013], the direct argmax [Lorberbom et al., 2019], as well as the concurrently developed equivalent Gumbel-Softmax (GS) [Jang et al., 2017] or Concrete [Maddison et al., 2017] that uses a continuous relaxation. These were further improved by combining them with REINFORCE to obtain unbiased estimators, which includes REBAR [Tucker et al., 2017], as well as RELAX [Grathwohl et al., 2018], which uses a free form neural network.

In practice, REINFORCE is almost always augmented by baselines, e.g. in variational inference [Mnih and Gregor, 2014, Ranganath et al., 2014, Paisley et al., 2012, Ruiz et al., 2016, Kucukelbir et al., 2017]. MuProp [Gu et al., 2016] uses a first order mean field Taylor approximation, but requires f to be differentiable. Other estimators apply Rao-Blackwellization, $e.g.$, to one latent variable at a time [Titsias and Lázaro-Gredilla, 2015] or to the reparameterized Dirichlet vector in ARSM [Dong et al., 2021]. Besides REINFORCE and reparameterization type gradients, there is also the measure valued gradient [Rosca et al., 2019] and finite differences [Fu, 2006], but are less common in practice. A comprehensive review can be found in Mohamed et al. [2020].

The more recent approaches for categorical variables are unbiased and REINFORCE based, building on the idea of using multiple samples to construct a baseline for variance reduction. One such baseline is LOORF [Kool et al., 2019a], originally introduced in Salimans and Knowles [2014] and also known as VarGrad [Richter et al., 2020], where its theoretical properties are further analyzed. VIMCO [Mnih and Rezende, 2016] has a similar form, but is specific to the importance weighted multi sample bound [Burda et al., 2016]. A different approach is ARSM [Yin et al., 2019], which reparameterizes the gradient with a Dirichlet distribution and uses swaps to obtain multiple correlated samples. However, it adds some amount of variance due to the continuous reparameterization, and can require up to $C(C-1)/2$ function evaluations per step, which can be computationally expensive. More recently, the unordered set estimator (UNORD) [Kool et al., 2020] uses the Gumbel top-k trick to sample without replacement, and then constructs a baseline using a multiplicative term to preserve unbiasedness.

2 Background

Let $\boldsymbol{z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_D)$ denote D independent categorical variables, where $\boldsymbol{z}_d$ is a one hot encoded categorical sample from $\boldsymbol{z}_d \sim q_{\boldsymbol{\phi}_d}(\boldsymbol{z}_d) = \text{Cat}(\sigma(\boldsymbol{\phi}_d))$, with $\sigma(\boldsymbol{x})_i = e^{\boldsymbol{x}_i} / \sum_j e^{\boldsymbol{x}_j}$ being the softmax function. Let also $\hat{\boldsymbol{\imath}}$ be the basis vector with its i^{th} coordinate set to one, and otherwise zero, such that $P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}) = \sigma(\boldsymbol{\phi})_i$, or equivalently $\mathbb{E}[\boldsymbol{z}] = \sigma(\boldsymbol{\phi})$. Unless otherwise stated, superscripts denote the dimension, and subscripts denote different samples, with vectors and matrices being bold lowercase and bold uppercase symbols, respectively. We are interested in optimizing the following objective with respect to the logits $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_D)$:

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}[f(\boldsymbol{z})], \quad q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \prod_{d=1}^{D} q_{\boldsymbol{\phi}_d}(\boldsymbol{z}_d).$$

Although the score function gradient contains two terms:

$$\nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})} \Big[\nabla_{\boldsymbol{\phi}} f(\boldsymbol{z}) + f(\boldsymbol{z}) \nabla_{\boldsymbol{\phi}} \ln q_{\boldsymbol{\phi}}(\boldsymbol{z}) \Big]$$

the first term is easily estimated, so we omit the subscript in f for notational clarity, and we focus on the latter term in this work. Since CARMS generalizes the multisample LOORF estimator beyond independent samples, we review it here. We also review the ARMS estimator, which uses a copula to generate antithetic samples, but is restricted to binary variables.

2.1 LOORF

Leave-one-out-REINFORCE (LOORF) [Salimans and Knowles, 2014, Kool et al., 2019a], also known as VarGrad [Richter et al., 2020], is a general score function based gradient estimator that uses N i.i.d. samples to construct a baseline for variance reduction, and is competitive with state of the art estimators. Its theoretical properties have recently been analyzed in Richter et al. [2020], where the same estimator results from minimizing the variance of the log ratio between the posterior and approximating distribution in variational inference. The only requirements for LOORF are the ability

to sample $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and evaluate $f(\boldsymbol{z})$. Given N samples $\boldsymbol{z}_1, ..., \boldsymbol{z}_N \overset{iid}{\sim} \text{Cat}(\sigma(\boldsymbol{\phi}))$, it has the form:

$$g_{\text{LOORF}} = \frac{1}{N-1} \sum_{n=1}^{N} \Big(f(\boldsymbol{z}_n) - \bar{f}(\boldsymbol{z}) \Big) \nabla_{\boldsymbol{\phi}} \ln q_{\boldsymbol{\phi}}(\boldsymbol{z}_n) = \frac{1}{N-1} \sum_{n=1}^{N} \Big(f(\boldsymbol{z}_n) - \bar{f}(\boldsymbol{z}) \Big) \big(\boldsymbol{z}_n - \sigma(\boldsymbol{\phi}_n) \big), \quad (1)$$

where $\bar{f}(\boldsymbol{z}) = \frac{1}{N} \sum_{n=1}^{N} f(\boldsymbol{z}_n)$. It is commonly used due to its simplicity and strong performance.

2.2 ARMS

The Antithetic-REINFORCE-MultiSample (ARMS) [Dimitriev and Zhou, 2021] estimator is a recent work that uses any number N of mutually negatively correlated samples. However, although unbiased, it is limited to binary variables, which motivated us to extend it to the categorical case. Since any antithetic copula in two dimensions reduces to the pair $(u, 1-u)$, it generalizes DisARM [Dong et al., 2020], independently discovered as U2G [Yin et al., 2020], which use $N = 2$ samples. ARMS achieves this generalization by using a copula, which is any multivariate distribution whose marginals are uniform random variables:

$$\boldsymbol{u} = (u_1, ..., u_N) \sim \mathcal{C}_N, \quad \forall i : \; u_i \sim \text{Unif}(0, 1).$$

More specifically, ARMS uses a Dirichlet or a Gaussian copula, which both have very strong negative dependence between each dimension of $\boldsymbol{u}$. However any copula can be used instead, with the only two requirements being the ability to generate samples easily, as well as being able to evaluate the bivariate CDF $\Phi(\boldsymbol{u}_i, \boldsymbol{u}_j)$ of the copula, so that the debiasing term can be calculated. Given a copula sample $\boldsymbol{u}$, it uses inverse CDF sampling to convert it into N antithetic Bern(p) samples, which are simply $b_i = \mathbb{1}_{u_i < p}, \; \forall i$. For a D-dimensional vector of antithetic Bernoulli variables $\boldsymbol{b}_1, ..., \boldsymbol{b}_N$ with probabilities $\sigma(\boldsymbol{\phi})$, the N-sample unbiased estimator has the following simple form:

$$g_{\text{ARMS}} = \frac{1}{N-1} \sum_{n=1}^{N} \left(f(\boldsymbol{b}_n) - \frac{1}{n} \sum_{m=1}^{N} f(\boldsymbol{b}_m) \right) \frac{\boldsymbol{b}_n - \sigma(\boldsymbol{\phi})}{1 - \boldsymbol{\rho}}, \quad (2)$$

with $\boldsymbol{\rho} = (\rho_1, ..., \rho_D)$, and $\boldsymbol{\rho}_d = \text{corr}(\boldsymbol{b}_i^d, \boldsymbol{b}_j^d)$ being the correlation of the d^{th} Bernoulli variable. It is easy to compute given the bivariate CDF of the copula.

3 CARMS

The CARMS estimator has a similar form to LOORF, but requires a multiplicative term to remain unbiased. It has two requirements: an easy way to sample antithetic categorical variables, and being able to compute the bivariate probability mass function (PMF), which should be identical for any pair. For clarity, we begin by assuming the ability to do this, and derive the univariate version for two samples, which we then extend to N samples. Next, we generalize CARMS to any number of categorical variables. Lastly, in Section 3.2, we show two different ways of sampling that satisfy both conditions: inverse CDF sampling with an easily computable analytical bivariate PMF, and the Gumbel max trick combined with an empirical estimate of the PMF.

The two sample version of CARMS can be obtained by replacing the two i.i.d. samples in 2-LOORF with an arbitrarily correlated pair of categorical variables and an added debiasing term. For $N = 2$ samples $z, z' \sim \text{Cat}(\sigma(\boldsymbol{\phi}))$, LOORF has the following simple form:

$$g_{\text{LOORF}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{2} \Big(f(\boldsymbol{z}) - f(\boldsymbol{z}') \Big) (\boldsymbol{z} - \boldsymbol{z}'), \quad (3)$$

which, importantly, is unbiased [Kool et al., 2019a, Dimitriev and Zhou, 2021]. This means that using a simple importance weight preserves its unbiasedness, for an arbitrary bivariate categorical distribution $(\boldsymbol{z}, \boldsymbol{z}') \sim \text{Cat}\big(\sigma(\boldsymbol{\phi}), \sigma(\boldsymbol{\phi})\big)$:

$$\mathbb{E}\left[g_{\text{LOORF}}(\boldsymbol{z}, \boldsymbol{z}') \frac{P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}) P(\boldsymbol{z}' = \hat{\boldsymbol{\jmath}})}{P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}})} \right] = \sum_{i,j} P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}}) \frac{P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}) P(\boldsymbol{z}' = \hat{\boldsymbol{\jmath}})}{P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}})} g_{\text{LOORF}}(\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}})$$

$$= \sum_{i,j} P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}) P(\boldsymbol{z}' = \hat{\boldsymbol{\jmath}}) g_{\text{LOORF}}(\hat{\boldsymbol{\imath}}, \hat{\boldsymbol{\jmath}}) = \mathbb{E}_{\boldsymbol{z}, \boldsymbol{z}' \overset{iid}{\sim} \text{Cat}(\sigma(\boldsymbol{\phi}))} \Big[g_{\text{LOORF}}(\boldsymbol{z}, \boldsymbol{z}') \Big] = \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}).$$

We summarize the derivation of the two sample version of CARMS, which we denote as the Categorical-Antithetic-REINFORCE-Two-Sample (CARTS) estimator in the following theorem.

Theorem 1 *Let $(\boldsymbol{z}, \boldsymbol{z}')$ be a sample from an arbitrary bivariate Categorical distribution with marginal distributions $\boldsymbol{z}, \boldsymbol{z}' \sim \text{Cat}(\sigma(\boldsymbol{\phi}))$, and a known bivariate PMF. An unbiased estimator of $\nabla_{\boldsymbol{\phi}} \mathbb{E}[f(\boldsymbol{z})]$ is:*

$$g_{\text{CARTS}}(\boldsymbol{z}, \boldsymbol{z}') = \frac{1}{2}\Big(f(\boldsymbol{z}) - f(\boldsymbol{z}')\Big)(\boldsymbol{z} - \boldsymbol{z}')\boldsymbol{z}^T \boldsymbol{\mathcal{R}} \boldsymbol{z}', \quad \boldsymbol{\mathcal{R}}_{ij} = \frac{\sigma(\boldsymbol{\phi})_i \sigma(\boldsymbol{\phi})_j}{P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}})}. \tag{4}$$

The intuition behind using negatively correlated variables is that we want to avoid the case when $\boldsymbol{z} = \boldsymbol{z}'$, because the sample is then "wasted." We formalize this intuition below, and defer the proof to the appendix.

Theorem 2 *Let $(\boldsymbol{z}, \boldsymbol{z}') \sim Cat(\sigma(\boldsymbol{\phi}), \sigma(\boldsymbol{\phi}))$. If the bivariate PMF satisfies:*

$$\forall i \neq j : P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}}) \geq P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}) P(\boldsymbol{z}' = \hat{\boldsymbol{\jmath}}),$$

with strict inequality for at least one pair, then $Var[g_{CARTS}(\boldsymbol{z}, \boldsymbol{z}')] < Var[g_{LOORF}(\boldsymbol{z}, \boldsymbol{z}')]$.

We now extend CARTS to N samples, using the following identity, the proof of which can be found in Dimitriev and Zhou [2021]. It states that N-sample LOORF is equivalent to averaging 2-sample LOORF over all $\binom{N}{2}$ pairs:

$$\begin{aligned} g_{\text{LOORF}}(\boldsymbol{z}_1, .., \boldsymbol{z}_N) &= \frac{1}{N}\sum_{n=1}^{N}\left(f(\boldsymbol{z}_n) - \frac{1}{N}\sum_{m=1}^{N} f(\boldsymbol{z}_m)\right)(\boldsymbol{z}_n - \sigma(\boldsymbol{\phi})) \\ &= \tfrac{1}{N(N-1)}\sum_{n \neq m}\frac{1}{2}\Big(f(\boldsymbol{z}_n) - f(\boldsymbol{z}_m)\Big)\big(\boldsymbol{z}_n - \boldsymbol{z}_m\big) = \tfrac{1}{N(N-1)}\sum_{n \neq m} g_{\text{LOORF}}(\boldsymbol{z}_n, \boldsymbol{z}_m) \end{aligned}$$

With the above identity it easily follows that given N antithetic Categorical samples $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_N]^T$, applying CARTS to all pairs results in an unbiased estimator, due to linearity of expectations:

$$\begin{aligned} \mathbb{E}\left[g_{\text{CARMS}}(\boldsymbol{Z}, \boldsymbol{\mathcal{R}})\right] &= \mathbb{E}\left[\tfrac{1}{N(N-1)}\sum_{n \neq m} g_{\text{CARTS}}(\boldsymbol{z}_n, \boldsymbol{z}_m, \boldsymbol{\mathcal{R}})\right] = \mathbb{E}\left[\tfrac{1}{N(N-1)}\sum_{n \neq m} g_{\text{LOORF}}(\boldsymbol{z}_n, \boldsymbol{z}_m)\right] \\ &= \mathbb{E}\left[g_{\text{LOORF}}(\boldsymbol{Z})\right] = \nabla_{\boldsymbol{\phi}} \mathcal{L}(\boldsymbol{\phi}). \end{aligned}$$

We summarize CARMS in the next theorem. and we also rewrite it in a simpler matrix form used in our implementation. The matrix form can be obtained after some algebra, which can be found in the appendix.

Theorem 3 *Let $\boldsymbol{Z} = [\boldsymbol{z}_1, ..., \boldsymbol{z}_N]^T$ be a sample from an arbitrary N-variate Categorical distribution with identical marginal and bivariate distributions, such that $\boldsymbol{z}_i \sim Cat(\sigma(\boldsymbol{\phi}))$, and $\boldsymbol{\mathcal{R}}_{ij} = \sigma(\boldsymbol{\phi})_i \sigma(\boldsymbol{\phi})_j / P(\boldsymbol{z}_n = \hat{\boldsymbol{\imath}}, \boldsymbol{z}_m = \hat{\boldsymbol{\jmath}})$. An unbiased estimator of $\nabla_{\boldsymbol{\phi}} \mathbb{E}[f(\boldsymbol{z})]$ is:*

$$g_{CARMS}(\boldsymbol{Z}) = \tfrac{1}{N(N-1)}\sum_{n \neq m}\frac{1}{2}\Big(f(\boldsymbol{z}_n) - f(\boldsymbol{z}_m)\Big)(\boldsymbol{z}_n - \boldsymbol{z}_m)\boldsymbol{z}_n^T \boldsymbol{\mathcal{R}} \boldsymbol{z}'_m$$

Lemma 4 *Let $f(\boldsymbol{Z}) = [f(\boldsymbol{z}_1), ..., f(\boldsymbol{z}_N)]^T$, $\circ$ denote the Hadamard product, $\mathbf{1}_{N \times N}$ a matrix of ones, and $\boldsymbol{I}_N$ the identity matrix. Define $\boldsymbol{\mathcal{O}} = \frac{1}{N-1}(\mathbf{1}_{NxN} - I_N) \circ (\boldsymbol{Z}\boldsymbol{\mathcal{R}}\boldsymbol{Z}^T)$, and $\boldsymbol{\mathcal{D}} = diag(\boldsymbol{\mathcal{O}}\mathbf{1}_N)$, to be a diagonal matrix. The CARMS estimator can equivalently be written in the following form:*

$$g_{CARMS}(\boldsymbol{Z}) = \frac{1}{N} f(\boldsymbol{Z})^T (\boldsymbol{\mathcal{D}} - \boldsymbol{\mathcal{O}}) \left(\boldsymbol{Z} - \mathbf{1}_N \sigma(\boldsymbol{\phi})^T\right). \tag{5}$$

Furthermore, for independent samples, $\boldsymbol{Z}\boldsymbol{\mathcal{R}}\boldsymbol{Z}^T = \mathbf{1}_{N \times N}$ and LOORF has the form:

$$g_{LOORF}(\boldsymbol{Z}) = \frac{1}{N} f(\boldsymbol{Z})^T \Big(\boldsymbol{I}_{N \times N} - \tfrac{1}{N-1}\left(\mathbf{1}_{NxN} - \boldsymbol{I}_N\right)\Big)\left(\boldsymbol{Z} - \mathbf{1}_N \sigma(\boldsymbol{\phi})^T\right) \tag{6}$$

3.1 Multivariate CARMS

In the univariate case, Monte Carlo estimators are not necessary, as the expectation has C terms and can simply be analytically summed, but for many categorical variables, stochastic gradients are required. For D dimensions, we can sample $\boldsymbol{Z}^d \sim \text{Cat}(\sigma(\boldsymbol{\phi}^d))$ independently and combine them into a $D \times N \times C$ tensor $\boldsymbol{Z}$, with the corresponding $D \times C \times C$ importance ratio tensor $\boldsymbol{\mathcal{R}}$. Below, we use superscripts and subscripts to index the first and second dimension of the tensors. Focusing on the d^{th} dimension, we have:

$$\begin{aligned}
&\nabla_{\boldsymbol{\phi}^d}\mathbb{E}\left[f(\boldsymbol{Z})\right] = \mathbb{E}_{\boldsymbol{Z}^{-d}}\left[\nabla_{\boldsymbol{\phi}^d}\mathbb{E}_{\boldsymbol{Z}^d}\left[f(\boldsymbol{Z}^{-d}, \boldsymbol{Z}^d)\right]\right] \\
&\quad= \mathbb{E}_{\boldsymbol{Z}^{-d}}\left[\mathbb{E}_{\boldsymbol{Z}^d}\left[\tfrac{1}{N(N-1)}\sum_{n\neq m}\frac{1}{2}\Big(f(\boldsymbol{Z}_n^{-d}, \boldsymbol{Z}_n^d) - f(\boldsymbol{Z}_m^{-d}, \boldsymbol{Z}_m^d)\Big)(\boldsymbol{Z}_n^d - \boldsymbol{Z}_m^d)\boldsymbol{Z}_n^{d^T}\boldsymbol{\mathcal{R}}^d\boldsymbol{Z}_m^d\right]\right] \\
&\quad= \mathbb{E}\left[\tfrac{1}{N(N-1)}\sum_{n\neq m}\frac{1}{2}\left(f(\boldsymbol{Z}_n) - f(\boldsymbol{Z}_m)\right)\left(\boldsymbol{Z}_n^d - \boldsymbol{Z}_m^d\right)\boldsymbol{Z}_n^{d^T}\boldsymbol{\mathcal{R}}^d\boldsymbol{Z}_m^d\right].
\end{aligned}$$

Just like the univariate case, we only need N evaluations of f regardless of the dimensionality D.

3.2 Antithetic categorical variables

To be able to use CARMS in practice, we describe two ways to generate antithetic categorical variables in this section. Both methods are based on transformations of uniform random variables, which makes them amenable to antithetic copulas, such as the Gaussian or Dirichlet copula [Dimitriev and Zhou, 2021].

3.2.1 Inverse CDF sampling

The inverse transform sampling is a commonly used identity to transform uniform random variables, which are easy to generate, to another distribution:

$$x \sim F_x(x) \iff u \sim \text{Unif}(0,1),\ x = F_x^{-1}(u),$$

where $F_x(x)$ denotes the CDF of the desired distribution, and is simple for categorical variables. Let $\boldsymbol{p}^{(o)} = (p_1^{(o)}, ..., p_C^{(o)})$ be a vector of probabilities, which are reordered according to some ordering o. Define the left and right boundaries:

$$\boldsymbol{l}_i^{(o)} = \sum_{j=1}^{j-1} p_k^{(o)}, \quad \boldsymbol{r}_j^{(o)} = \sum_{j=1}^{i} p_j^{(o)}. \tag{7}$$

Then, we can transform $u \sim \text{Unif}(0,1)$ by setting $\boldsymbol{z} = \hat{\boldsymbol{j}}$, where j is such that $u \in [\boldsymbol{l}_j^{(o)}, \boldsymbol{r}_j^{(o)}]$. To obtain N antithetic categorical variables, we can sample $\boldsymbol{u} \sim \mathcal{C}_N$ for a given copula and set $\boldsymbol{z}_n$ in a vectorized manner. Importantly, this sampling approach allows us to analytically evaluate the bivariate PMF:

$$\begin{aligned}
&P(\boldsymbol{z}^{(o)} = \hat{\boldsymbol{i}}, \boldsymbol{z}'^{(o)} = \hat{\boldsymbol{j}}) = P\left(u \in [\boldsymbol{l}_i^{(o)}, \boldsymbol{r}_i^{(o)}], u' \in [\boldsymbol{l}_j^{(o)}, \boldsymbol{r}_j^{(o)}]\right) = \\
&\Phi_{\mathcal{C}}(\boldsymbol{l}_i^{(o)}, \boldsymbol{l}_j^{(o)}) + \Phi_{\mathcal{C}}(\boldsymbol{r}_i^{(o)}, \boldsymbol{r}_j^{(o)}) - \Phi_{\mathcal{C}}(\boldsymbol{l}_i^{(o)}, \boldsymbol{r}_j^{(o)}) - \Phi_{\mathcal{C}}(\boldsymbol{r}_i^{(o)}, \boldsymbol{l}_j^{(o)}),
\end{aligned} \tag{8}$$

where $\Phi_{\mathcal{C}}$ denotes the bivariate CDF of the copula. Unfortunately, keeping to one specific ordering does not allow for all possible pairs to have non-zero probabilities, so we randomly reorder $\boldsymbol{p}$ at every sampling step in the following manner. An ordering o consists of two indexes: i, j uniformly sampled from $\{1, ..., C\}$. First we translate $\boldsymbol{p}$ i elements to the right, then swap the last element with the j^{th}:

$$\forall k : \boldsymbol{p}_k = \boldsymbol{p}_{(k+i) \bmod C}, \quad \text{and} \quad \boldsymbol{p}_j \leftrightarrow \boldsymbol{p}_C. \tag{9}$$

This guarantees at least one of the $C(C-1)/2$ orderings has i and j as the first and last elements, respectively. And since i, j are uniformly chosen, this allows us to easily compute the needed bivariate

Algorithm 1 Antithetic inverse CDF categorical sampling

Input: Number of samples N, probabilities $\boldsymbol{p} = \sigma(\boldsymbol{\phi})$, copula $\mathcal{C}$.
Sample $\boldsymbol{u} = (u_1, ..., u_N) \sim \mathcal{C}_N$.
Shuffle $\boldsymbol{p}$ according to Eq. 9, and define the left and right boundaries $\boldsymbol{l}, \boldsymbol{r}$ according to Eq. 7.
Set $\forall n$: $\boldsymbol{z}_n = \hat{\boldsymbol{\jmath}}$, where j is such that $u_n \in [\boldsymbol{l}_j, \boldsymbol{r}_j]$.
For $i, j \in \{1, ..., C\}$: compute $\boldsymbol{\mathcal{R}}_{ij} = \boldsymbol{p}_i \boldsymbol{p}_j / P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}})$ according to Eq. 8.
return: $(\boldsymbol{z}_1, ..., \boldsymbol{z}_N)$, $\boldsymbol{\mathcal{R}}$.

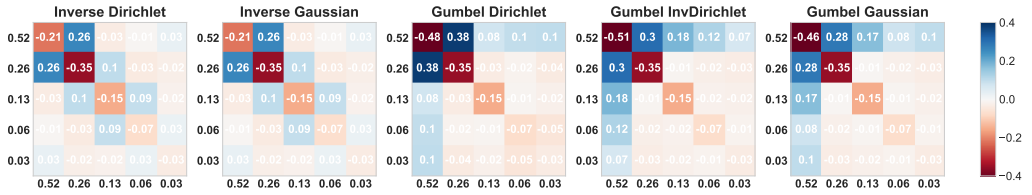

Figure 1: Correlation matrix for each pair of variables $\rho_{ij} = \text{corr}(z_i, z_j)$ using both types of categorical sampling and different copulas.

PMF for $\boldsymbol{\mathcal{R}}$, given all of the orderings:

$$P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}}) = \frac{2}{C(C-1)} \sum_{o \in O} P(\boldsymbol{z}^{(o)} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}'^{(o)} = \hat{\boldsymbol{\jmath}}) \tag{10}$$

Although it would be simpler to permute the elements randomly, a sum over $C!$ orderings would quickly become prohibitive as C increases. We summarize the inverse CDF sampling method in Algorithm 1.

3.2.2 Gumbel max sampling

It is not strictly necessary to analytically calculate the bivariate PMF. If the variance is not too large, a Monte Carlo estimation suffices. In such a case, we can use a simpler sampling approach using the well known Gumbel max trick [Gumbel, 1954, Jang et al., 2017, Maddison et al., 2017]. Since a Gumbel distribution $g \sim \text{Gumbel}(0, 1) \iff g = -\ln(-\ln(u)), u \sim \text{Unif}(0, 1)$ we can again use copulas to encode negative correlations between samples. Given $\boldsymbol{u} \sim \mathcal{C}_N$, an antithetic Gumbel categorical sample is $\boldsymbol{z} = \hat{\boldsymbol{\jmath}}$ such that $j = \text{argmax}_i \boldsymbol{\phi}_i - \ln(-\ln(u_i))$, where $\boldsymbol{\phi}$ are the logits of the desired categorical distribution. This is also equivalent to the following exponential racing [Yin et al., 2019, Zhang and Zhou, 2018] sampling: $\boldsymbol{z} = \hat{\boldsymbol{\jmath}}$, such that $j = \text{argmin}_i \epsilon_i$, where $\epsilon_i \sim \text{Exp}(e^{\boldsymbol{\phi}_i})$.

If we let $\boldsymbol{Z} = [\boldsymbol{z}_1, .., \boldsymbol{z}_N]^T$ as before, an simple empirical estimate of the bivariate PMF computed using all pairs and only matrix operations is:

$$P(\boldsymbol{z} = \hat{\boldsymbol{\imath}}, \boldsymbol{z}' = \hat{\boldsymbol{\jmath}}) \approx \frac{1}{N(N-1)} \boldsymbol{Z}^T \left(\mathbf{1}_{N \text{x} N} - I_N\right) \boldsymbol{Z}.$$

In Fig. 1 we show what bivariate PMF both approaches produce. The results are similar for both a(n inverted) Dirichlet or Gaussian copula, with larger differences between the categorical sampling method.

4 Experimental results

In this section, we first illustrate the variance reduction that CARMS offers on a toy example. We also optimize a categorical variational autoencoder (VAE) [Kingma and Welling, 2014, Rezende et al., 2014], and a stochastic network for structured output prediction, which are standard tasks [Jang et al., 2017] for categorical variables, done on three different benchmark datasets. Each experiment uses both antithetic categorical approaches for CARMS: inverse CDF sampling and the Gumbel max trick, denoted as CARMS-I and CARMS-G, respectively. For CARMS-G we clip the empirical ratio values to avoid numerical instabilities, and we use the Dirichlet copula for both. We compare our approach to three state-of-the-art unbiased estimators: LOORF/VarGrad [Kool et al., 2019a, Richter

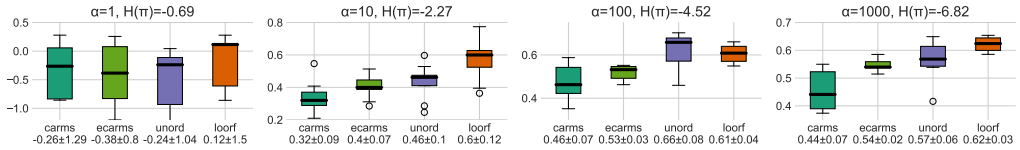

Figure 2: Log variance of the gradient of different estimators with respect to the logits on a toy problem. Columns correspond to different entropy levels of the logits.

et al., 2020], the unordered set estimator (UNORD) [Kool et al., 2020] and ARSM [Yin et al., 2019]. The code for all experiments is available in the supplementary material.

4.1 Toy example

We first showcase the variance reduction over other methods in a simple toy example, where we take the gradient with respect to the logits $\boldsymbol{\phi}$ of:

$$\mathcal{L}(\boldsymbol{\phi}) = \mathbb{E}[f(\boldsymbol{z}_1, \dots, \boldsymbol{z}_D)], \quad \boldsymbol{z}_d \sim \text{Cat}(\sigma(\boldsymbol{\phi_d})), \quad f(\boldsymbol{z}_1, \dots, \boldsymbol{z}_D) = \sum_{d=1}^{D} \sum_{c=1}^{C} d \cdot c \cdot \boldsymbol{z}_{dc}.$$

For simplicity, let the number of categories, dimensions, and samples be $C = D = N = 3$. The probabilities are randomly sampled from a Dirichlet distribution: $\sigma(\boldsymbol{\phi}) \sim \text{Dir}(\mathbf{1}_C \cdot \alpha)$, but are identical for all methods for a given α. We vary the entropy of the probabilities from high ($\alpha = 1$) to low ($\alpha = 1000$). In a high entropy setting, there is little difference between the estimators, as the variance itself is very high, but differences emerge as we increase α and lower the entropy. The combined log variance of the gradient of each logit, for different methods and different α, is shown in Fig 2.

4.2 Categorical variational autoencoder

For this task, we follow the experimental setting from ARMS [Dimitriev and Zhou, 2021], except we use categorical instead of binary latent variables, and maximize the ELBO:

$$\text{ELBO}(\boldsymbol{\phi}) = \mathbb{E}\Big[\ln \frac{p(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Big] \approx \sum_{n=1}^{N} \ln \frac{p(\boldsymbol{x}|\boldsymbol{z}_n)p(\boldsymbol{z}_n)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_n|\boldsymbol{x})}, \quad z_1, \dots, z_N \sim \text{Cat}_N(\sigma(\boldsymbol{\phi})),$$

where $\text{Cat}_N(\sigma(\boldsymbol{\phi}))$ denotes an N-variate categorical distribution with identical marginals. The number of categories is $C \in \{3, 5, 10\}$ with $D = \lfloor 200/C \rfloor$ latent variables, respectively, to make the total computational effort similar. In the binary case $C = 2$, CARMS reduces to ARMS, for which a thorough comparison has already been produced. The task is training a categorical VAE using either a linear or nonlinear encoder/decoder pair on three different datasets: Dynamic(ally binarized) MNIST [LeCun et al., 2010], Fashion MNIST [Xiao et al., 2017], and Omniglot [Lake et al., 2015]. All datasets are freely available under the MIT license, and do not contain any personally identifiable information or offensive content. For a fair comparison, all methods use the same learning rate, optimizer, model architecture, and number of samples. Since ARSM uses a variable number of function evaluations per step, we use one sample per step, for which ARSM uses around twice as many evaluations as the other methods. The results are combined from five independent runs for each experimental configuration.

The VAE consists of a stochastic layer with $\lfloor 200/C \rfloor$ units, each of which is a C-way categorical variable. For the nonlinear case, there are additionally two layers of 200 units with LeakyReLU [Maas et al., 2013] activations. The prior logits are optimized using SGD with a learning rate of 10^{-2}, whereas the encoder and decoder are optimized using Adam [Kingma and Ba, 2015] with a learning rate of 10^{-4}, following Yin et al. [2019]. The optimization is run for 10^6 steps, with a batch size of 50, from which the global dataset mean is subtracted. The models are trained on an Intel Xeon Platinum 8280 2.7GHz CPU, and an individual run took 5-8 hours on one core of the machine, with a total carbon emissions estimated to be 28.19 kg of CO_2 [Lacoste et al., 2019]. An exception is UNORD for 10 samples, which was significantly more computation heavy. Although the paper states that a step can be performed in $O(2^C)$, the provided code requires $O(C!)$ evaluations per step.

Table 1: Final training 100 sample log likelihood of VAEs using different estimators, where the stochastic layer contains C=3, 5, or 10 categories, with $\lfloor 200/C \rfloor$ latent variables and C samples per gradient step, respectively. Results are reported on three datasets: Dynamic MNIST, Fashion MNIST, and Omniglot over 5 runs, with the best performing methods in bold.

Categories			CARMS-I	CARMS-G	LOORF	UNORD	ARSM
Dynamic MNIST	Linear	3	**-105.34 ± 0.25**	**-105.36 ± 0.24**	-105.64 ± 0.23	**-105.23 ± 0.23**	-107.35 ± 0.56
		5	**-103.53** ± 0.13	**-103.35 ± 0.18**	-103.54 ± 0.15	**-103.50 ± 0.13**	-106.13 ± 0.53
		10	-103.22 ± 0.05	**-103.12 ± 0.06**	-103.48 ± 0.06	-103.56 ± 0.06	-106.71 ± 0.58
	Nonlinr	3	**-94.85 ± 0.28**	**-94.60 ± 0.28**	-95.12 ± 0.21	-95.21 ± 0.22	-99.62 ± 0.50
		5	-93.05 ± 0.14	**-92.60 ± 0.12**	-92.91 ± 0.16	-92.98 ± 0.12	-98.89 ± 0.43
		10	**-92.13 ± 0.05**	-92.42 ± 0.10	-92.44 ± 0.04	-93.05 ± 0.14	-97.76 ± 0.41
Fashion MNIST	Linear	3	**-245.44 ± 0.22**	-245.69 ± 0.19	-245.8 ± 0.19	-245.90 ± 0.21	-247.51 ± 0.45
		5	-242.06 ± 0.13	**-241.90 ± 0.10**	-242.17 ± 0.09	-242.43 ± 0.12	-244.63 ± 0.47
		10	**-240.44 ± 0.03**	-240.52 ± 0.04	-240.91 ± 0.04	-241.08 ± 0.06	-243.29 ± 0.36
	Nonlinr	3	**-233.13 ± 0.16**	**-233.20 ± 0.16**	-233.75 ± 0.10	-233.34 ± 0.12	-237.93 ± 0.20
		5	**-231.72 ± 0.09**	**-231.67 ± 0.13**	-232.01 ± 0.05	-232.19 ± 0.09	-237.29 ± 0.34
		10	**-230.77 ± 0.05**	-231.16 ± 0.05	-231.35 ± 0.03	-231.74 ± 0.02	-235.88 ± 0.17
Omniglot	Linear	3	**-114.53 ± 0.12**	-114.73 ± 0.15	-114.90 ± 0.13	-114.77 ± 0.15	-116.34 ± 0.42
		5	**-114.01 ± 0.10**	**-113.93 ± 0.11**	**-114.01 ± 0.10**	**-114.00 ± 0.09**	-115.72 ± 0.35
		10	**-114.97 ± 0.06**	**-114.99 ± 0.06**	-115.17 ± 0.05	-115.55 ± 0.08	-117.48 ± 0.35
	Nonlinr	3	**-110.19 ± 0.23**	**-110.16 ± 0.21**	**-110.33 ± 0.27**	**-110.31 ± 0.21**	-115.13 ± 0.51
		5	**-109.14 ± 0.09**	-109.35 ± 0.13	-109.39 ± 0.16	-109.55 ± 0.14	-115.07 ± 0.37
		10	**-108.66 ± 0.08**	**-108.62 ± 0.07**	-108.98 ± 0.06	-109.54 ± 0.15	-115.11 ± 0.34

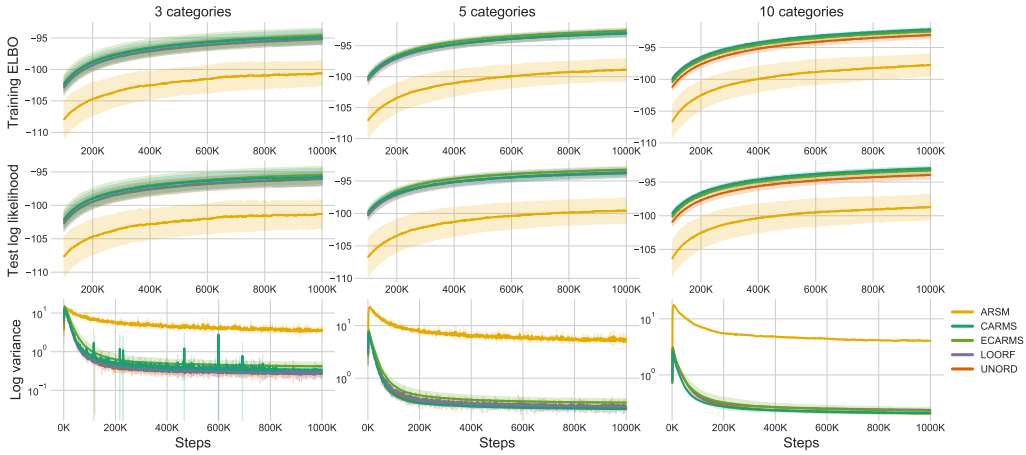

Figure 3: Training a nonlinear categorical VAE with different estimators on Dynamic MNIST using ELBO. Columns correspond to $C \in \{3, 5, 10\}$ categories with C samples per gradient step, respectively. Rows correspond to the 100 sample training and test log likelihood, and the variance of the gradient with respect to the logits of the encoder network. Results for different datasets and other networks can be found in the Appendix.

In Fig 3, we plot the training and test log likelihood using 100 samples, and gradient variance w.r.t the logits over time, for a nonlinear network on dynamic MNIST. Similar plots for other datasets and network types can be found in the appendix. Also shown in Table 1 is the final training log likelihood using 100 samples for all three datasets, network types, categories, and gradient estimators. The corresponding table with the final test log likelihood can be found in the appendix. In general, both versions of CARMS perform comparably, and result in slightly higher log likelihood than other methods. Because Gumbel CARMS uses an empirical estimate of the debiasing ratio, it has higher variance and slightly worse performance. When limiting the number of function evaluations, there is a gap between ARSM (which used on average $2C$ evaluations per step, out of a maximum of $C(C-1)/2$ per step) compared to the other methods, which used C evaluations. This is possibly due to the continuous reparameterization that ARSM uses, which adds variance.

Table 2: Final training log likelihood of a categorical network for conditional estimation using different gradient estimators, where the stochastic layer contains $C = 3$, 5, or 10 categories, with $\lfloor 200/C \rfloor$ latent variables and C samples per gradient step, respectively. Results are reported on three datasets: Dynamic MNIST, Fashion MNIST, and Omniglot over 5 runs, with the best performing methods in bold.

Categories		CARMS-I	CARMS-G	LOORF	UNORD	ARSM
Dynamic MNIST	3	**57.98 ± 0.10**	58.35 ± 0.14	58.19 ± 0.14	**58.06 ± 0.04**	60.22 ± 0.19
	5	**57.57 ± 0.05**	57.85 ± 0.12	57.78 ± 0.08	**57.6 ± 0.05**	59.38 ± 0.13
	10	**58.17 ± 0.26**	**58.33 ± 0.13**	**58.20 ± 0.14**	**58.18 ± 0.17**	59.32 ± 0.11
Fashion MNIST	3	**132.83 ± 0.08**	**132.90 ± 0.08**	133.10 ± 0.05	133.06 ± 0.06	134.56 ± 0.35
	5	**132.68 ± 0.05**	132.81 ± 0.12	132.91 ± 0.07	132.94 ± 0.14	134.09 ± 0.12
	10	**133.32 ± 0.15**	**133.43 ± 0.23**	133.54 ± 0.17	**133.38 ± 0.10**	134.02 ± 0.21
Omniglot	3	**65.57 ± 0.10**	66.05 ± 0.14	65.92 ± 0.26	65.81 ± 0.09	68.00 ± 0.09
	5	**65.65 ± 0.18**	66.16 ± 0.32	65.92 ± 0.23	**65.78 ± 0.06**	67.99 ± 0.32
	10	**66.76 ± 0.31**	66.94 ± 0.24	**66.87 ± 0.07**	**66.66 ± 0.24**	68.35 ± 0.16

4.3 Structured prediction with stochastic categorical networks

We also compare all methods on the standard benchmark task of predicting the lower half of an image from the upper half, i.e. the conditional distribution $p(\boldsymbol{x}_l|\boldsymbol{x}_u)$, where $\boldsymbol{x}_u$ and $\boldsymbol{x}_l$ denote the upper and lower half of an image, respectively. We use a stochastic categorical network to estimate this distribution, with the objective: $\mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}_m|\boldsymbol{x}_u)}\left[\frac{1}{M}\sum_{m=1}^{M}\ln p(\boldsymbol{x}_l|\boldsymbol{z}_m)\right]$, where $\boldsymbol{z}$ denotes a stochastic categorical layer. The encoder/decoder pair each contain one hidden layer with $\lfloor 200/C \rfloor$ latent variables and a LeakyReLU activation, with the optimization performed for $C \in \{3, 5, 10\}$ categories, on all three datasets, with identical settings for each gradient estimator for a fair comparison. We use $M = 1$ for training, and $M = 1000$ for evaluation on both the train and test set. In Table 2, we show the final training set log likelihood, with the corresponding test log likelihood table in the appendix, though the results are qualitatively similar. The results are similar to the VAE experiment, with the inverse CDF CARMS having a slightly higher log likelihood. However, the differences between estimators are less pronounced, with the unordered set estimator is being mostly on par with CARMS, and ARSM only slightly trailing the other methods, with a much smaller gap.

5 Discussion

We have presented a novel approach for training categorical variables, which extends the ARMS estimator for the binary case. It goes beyond i.i.d. samples by using a copula to generate antithetic categorical samples, but preserves unbiasedness by including a multiplicative term. For i.i.d. samples, the form of the estimator reduces to LOORF. We showcase its usefulness on several datasets, a different number of categories and types of deep neural networks. In variational inference tasks, and conditional estimation, CARMS outperforms other state of the art estimators. The main limitation of this work is its specificity to categorical (including binary) variables. We hope to extend this, e.g. to Plackett-Luce models for top-k sampling [Kool et al., 2019b, Grover et al., 2019], which has important applications in ranking [Dadaneh et al., 2020]. There is also a general limitation that CARMS shares with other state-of-the-art estimators, which is higher complexity than LOORF, a very simple but strong baseline. Future work includes investigating theoretical properties and large scale applications, as well as possible general antithetic gradient estimators, and we plan to investigate adaptive correlations that take into account the properties of f to further reduce the variance.

Potential societal impact This work is focused on better optimization for categorical variables, which includes any network containing a stochastic categorical layer. In particular, generative models such as VAEs are widely used, and are sometimes trained on more human centered datasets. These models can sometimes be used to impersonate a person's face or voice. We only used non-human datasets such as MNIST, but with enough knowledge and using the available code, anyone, including bad actors, can train the same model on human content for malicious purposes. However, we strongly believe that wide knowledge dissemination and open source code is crucial for reproducibility in all of science.

References

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

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

S. Z. Dadaneh, S. Boluki, M. Zhou, and X. Qian. Arsm gradient estimator for supervised learning to rank. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3157–3161. IEEE, 2020.

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

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

Z. Dong, A. Mnih, and G. Tucker. Coupled gradient estimators for discrete latent variables. *Third Symposium on Advances in Approximate Bayesian Inference*, 2021.

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

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

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

A. Grover, E. Wang, A. Zweig, and S. Ermon. Stochastic optimization of sorting networks via continuous relaxations. *arXiv preprint arXiv:1903.08850*, 2019.

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

E. J. Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

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

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

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

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

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

W. Kool, H. Van Hoof, and M. Welling. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *International Conference on Machine Learning*, pages 3499–3508. PMLR, 2019b.

W. Kool, H. van Hoof, and M. Welling. Estimating gradients for discrete random variables by sampling without replacement. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rklEj2EFvB`.

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

A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

G. Lorberbom, T. S. Jaakkola, A. Gane, and T. Hazan. Direct optimization through arg max for discrete variational auto-encoder. In *Advances in Neural Information Processing Systems 32*, pages 6200–6211, 2019.

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

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

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

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

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

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

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

A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

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

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

L. Richter, A. Boustati, N. Nüsken, F. J. Ruiz, and Ö. D. Akyildiz. Vargrad: A low-variance gradient estimator for variational inference. In *Advances in Neural Information Processing Systems*, volume 33, pages 13481–13492, 2020.

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

F. J. R. Ruiz, M. K. Titsias, and D. M. Blei. The generalized reparameterization gradient. In *Advances in Neural Information Processing Systems 29*, pages 460–468, 2016.

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

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

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

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

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arxiv.org/abs/1708.07747*, 2017.

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

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

Q. Zhang and M. Zhou. Nonparametric bayesian lomax delegate racing for survival analysis with competing risks. *arXiv preprint arXiv:1810.08564*, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   [Yes] Our work focuses specifically on improving training that includes stochastic categorical variables, where we offer improvement over the state of the art.
   (b) Did you describe the limitations of your work?
   [Yes] We describe the limitations in the discussion section.
   (c) Did you discuss any potential negative societal impacts of your work?
   [Yes] We discuss societal impacts in the discussion section.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them?
   [Yes] We read the guidelines and ensured the paper conforms to them.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?
   [Yes] The theorems/lemmas contain all assumptions.
   (b) Did you include complete proofs of all theoretical results?
   [Yes] Proofs are given either in the main text or the supplemental material (appendix).
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   [Yes] The code, data, and instructions are available in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
   [Yes] All of information is available in the supplementary material.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   [Yes] We provide the mean and variance over 5 runs for all experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?
   [Yes] As stated in the results section, in total it took 5-8 hours on one CPU machine for a full run of all methods on all datasets, categories and network types, and an estimated 28kg of CO_2.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators?

[Yes] We included the proper dataset citations. The code was written by the authors.

(b) Did you mention the license of the assets?

[Yes] All datasets are available under the MIT license, mentioned in the results section.

(c) Did you include any new assets either in the supplemental material or as a URL?

[N/A] There are no new assets used. The existing datasets and code are either part of the supplemental material, or are downloaded when training starts.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?

[N/A] Under the MIT license, permission is granted to any person to use, copy, modify, merge, publish etc.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?

[Yes] Yes, it is stated in the results section that no personally identifiable information or offensive content is present in any of the datasets.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]