Differentiable Sampling of Categorical Distributions Using the CatLog-Derivative Trick

Lennert De Smet [1] **Emanuele Sansone** [1] **Pedro Zuidberg Dos Martires** [2]

Abstract

Categorical random variables can faithfully represent the discrete and uncertain aspects of data as part of a discrete latent variable model. Learning in such models necessitates taking gradients with respect to the parameters of the categorical probability distributions, which is often intractable due to their combinatorial nature. A popular technique to estimate these otherwise intractable gradients is the Log-Derivative trick. This trick forms the basis of the well-known REINFORCE gradient estimator and its many extensions. While the Log-Derivative trick allows us to differentiate through samples drawn from categorical distributions, it does not take into account the discrete nature of the distribution itself. Our first contribution addresses this shortcoming by introducing the CatLog-Derivative trick– a variation of the Log-Derivative trick tailored towards categorical distributions. Secondly, we use the CatLog-Derivative trick to introduce IndeCateR, a novel and unbiased gradient estimator for the important case of products of independent categorical distributions with provably lower variance than REINFORCE. Thirdly, we empirically show that the estimates of IndeCateR outperform the state of the art for the same number of samples.

1. Introduction

Categorical random variables naturally emerge in many domains in AI, such as language modelling, reinforcement learning and neural-symbolic AI (De Smet et al., 2023). They are compelling because they can faithfully represent the discrete concepts present in data in a sound probabilistic fashion. Unfortunately, inference in probabilistic models with categorical latent variables is usually computationally intractable due to its combinatorial nature. This intractability often leads to the use of sampling-based approximate inference techniques, which in turn poses problems to gradient-based learning as sampling is an inherently non-differentiable process.

In order to bypass this non-differentiability, two main classes of gradient estimators have been developed. On the one hand, there is a range of unbiased estimators based on the Log-Derivative trick and the subsequent REINFORCE gradient estimator (Williams, 1992). On the other hand, we have biased estimators that use continuous relaxations to which the reparametrisation trick (Ruiz et al., 2016) can be applied, such as the Gumbel-Softmax trick (Jang et al., 2017; Maddison et al., 2017).

A clear advantage of the REINFORCE estimator over relaxation-based estimators is its unbiased nature. However, REINFORCE tends to be sample-inefficient and its gradient estimates exhibit high variance in practice. To resolve these issues, methods have been proposed that modify REINFORCE by, for instance, adding control variates (Richter et al., 2020; Titsias & Shi, 2022). These modified estimators have been shown to deliver more robust gradient estimates than standard REINFORCE.

Instead of modifying REINFORCE, we take a different approach and modify the Log-Derivative trick by explicitly taking into account that we are working with multivariate categorical distributions. We call this first contribution the *CatLog-Derivative trick*. Interestingly, we show that the CatLog-Derivative trick leads to Rao-Blackwellised estimators (Casella & Robert, 1996), immediately giving us a guaranteed reduction in variance. A similar result was shown by Tokui & Sato (2017) using the Gumbel-Max reparametrisation for categorical distributions, which we discuss further in the related work (Section 5). As a second contribution, we propose IndeCateR (read as 'indicator'), a gradient estimator for the special case of independent categorical random variables. IndeCateR is a hyperparameter-free estimator that can be implemented efficiently on modern AI accelerators. Thirdly, we empirically show that IndeCateR is competitive with comparable state-of-the-art gradient estimators.

[1]Department of Computer Science, KU Leuven, Belgium [2]Center for Applied Autonomous Systems, Örebro University, Sweden. Correspondence to: Lennert De Smet <lennert.desmet@kuleuven.be>.

Published at the Differentiable Almost Everything Workshop of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. July 2023.

2. Notation and Preliminaries

Throughout this paper, we consider expectations with respect to multivariate categorical probability distributions, which we write as

$$\mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})] = \sum_{\mathbf{x} \in \Omega(\mathbf{X})} p(\mathbf{x}) f(\mathbf{x}); \tag{1}$$

where we assume this expectation to be finite. The symbol $\mathbf{X}$ denotes a random vector $(X_1, \ldots, X_D)$ of D categorical random variables while $p(\mathbf{X})$ denotes a multivariate probability distribution. The expression $\mathbf{X} \sim p(\mathbf{X})$ indicates that the random vector $\mathbf{X}$ is distributed according to $p(\mathbf{X})$. On the right-hand side of Equation (1) we write the expectation as an explicit sum over $\Omega(\mathbf{X})$, the finite sample space of the random vector $\mathbf{X}$, using $\mathbf{x} = (x_1, \ldots, x_D)$ for the specific assignments of the random vector $(X_1, \ldots, X_D)$.

Given an order of the random variables in $\mathbf{X}$, we can induce a factorisation of the joint probability distribution as follows

$$p(\mathbf{X}) = \prod_{d=1}^{D} p(X_d \mid \mathbf{X}_{<d}). \tag{2}$$

Here, $\mathbf{X}_{<d}$ denotes the ordered set of random variables $(X_1, \ldots, X_{d-1})$. Similarly, $\mathbf{X}_{>d}$ will denote the ordered set $(X_{d+1}, \ldots, X_D)$ in subsequent sections.

When performing gradient-based learning, we are interested in partial derivatives of the expected value in 1, i.e., $\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})]$. Here, we take the partial derivative of the expectation with respect to the parameter θ and assume that the distribution $p(\mathbf{X})$ and the function $f(\mathbf{X})$ depend on a set of parameters θ with $\theta \in \mathbb{R}^2$. For probability distributions to which the reparametrisation trick does not apply, we can rewrite the partial derivative using the Log-Derivative trick.

Theorem 2.1 (Log-Derivative Trick (Williams, 1992))**.** *Let $p(\mathbf{X})$ be a probability distribution and $f(\mathbf{X})$ such that its expectation is finite, with both functions depending on a set of parameters θ. Then, it holds that*

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})] = \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}\left[\frac{\partial}{\partial \theta} f(\mathbf{X})\right] + \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}\left[f(\mathbf{X}) \frac{\partial}{\partial \theta} \log p(\mathbf{X})\right]. \tag{3}$$

In general, both expectations in Equation (3) are intractable and often estimated with a Monte Carlo scheme. The most immediate such estimation is provided by the REINFORCE gradient estimator (Williams, 1992)

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})] \approx \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial \theta} f(\mathbf{x}^{(n)}) + f(\mathbf{x}^{(n)}) \frac{\partial}{\partial \theta} \log p(\mathbf{x}^{(n)}) \; . \tag{4}$$

The superscript on $\mathbf{x}^{(n)}$ denotes that it is the n^{th} sample vector drawn from $p(\mathbf{X})$.

A well-known problem with the REINFORCE gradient estimator is the high variance stemming from the second term in Equation (4). A growing body of research has been tackling this problem by proposing variance reduction techniques (Grathwohl et al., 2018; Richter et al., 2020; Titsias & Shi, 2022; Tucker et al., 2017). In what follows we will focus on estimating this second term and drop the first term, since it can be assumed to be unproblematic.

3. The CatLog-Derivative Trick

The standard log-derivative trick and its corresponding gradient estimators are applicable to both discrete and continuous probability distributions. However, this generality limits their usefulness when it comes to purely categorical random variables. For example, the REINFORCE gradient estimator suffers from high variance when applied to problems involving high-dimensional multivariate categorical random variables. In such a setting there are exponentially many possible states to be sampled, which makes it increasingly unlikely that a specific state gets sampled. We now introduce the CatLog-Derivative trick that reduces the exponential number of states arising in a multivariate categorical distribution by exploiting the distribution's factorisation. All subsequent statements are proven in the appendix.

Theorem 3.1 (CatLog-Derivative Trick)**.** *Let $p(\mathbf{X})$ be a multivariate categorical probability distribution that depends on a set of parameters θ, then it holds that $\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})]$ is equal to*

$$\sum_{d=1}^{D} \sum_{x \in \Omega(X_d)} \mathbb{E}_{\mathbf{X}_{<d} \sim p(\mathbf{X}_{<d})}\left[\frac{\partial}{\partial \theta} p(x \mid \mathbf{X}_{<d}) \; \mathbb{E}_{\mathbf{X}_{>d} \sim p(\mathbf{X}_{>d} \mid x, \mathbf{X}_{<d})}[f(\mathbf{X}_{<d}, x, \mathbf{X}_{>d})]\right]. \tag{5}$$

Intuitively, the CatLog-Derivative trick decomposes the log-derivative trick into an explicit sum of multiple Log-Derivative tricks, one for each of the categorical random variables present in the multivariate distribution. We show next that this decomposition is effectively equivalent to Rao-Blackwellising (Casella & Robert, 1996) gradient estimators.

Definition 3.2 (The CateR gradient estimator)**.** We define the Categorical REINFORCE (CateR) estimator via the expression

$$\frac{\partial}{\partial \theta} \mathbb{E}_{\mathbf{X} \sim p(\mathbf{X})}[f(\mathbf{X})] \approx \sum_{d=1}^{D} \sum_{x \in \Omega(X_d)} \frac{1}{N} \sum_{n_d=1}^{N} \frac{\partial}{\partial \theta} p(x \mid \mathbf{x}_{<d}^{(n_d)}) f(\mathbf{x}_{<d}^{(n_d)}, x, \mathbf{x}_{>d}^{(n_d)}); \tag{6}$$

where the sample $\mathbf{x}_{>d}^{(n_d)}$ is drawn while conditioning on x and $\mathbf{x}_{<d}^{(n_d)}$. The subscript on n_d indicates that different samples are drawn for every d.

Proposition 3.3. *The CateR estimator Rao-Blackwellises REINFORCE.*

Corollary 3.4 (Bias and Variance)**.** *The CateR estimator is unbiased and its variance is upper-bounded by REINFORCE.*

Proof. This follows trivially from Proposition 3.3, the law of total expectation and the law of total variance (Blackwell, 1947; Radhakrishna Rao, 1945). □

4. The IndeCateR Gradient Estimator

Using the CatLog-Derivative trick derived in the previous section we are now going to study a prominent special case of multivariate categorical distributions. That is, we will assume that our probability distribution admits the independent factorisation $p(\mathbf{X}) = \prod_{d=1}^{D} p_d(X_d)$. Note that all D different distributions still depend on the same set of learnable parameters θ. Furthermore, we subscript the individual distributions p_d as they can no longer be distinguished by their conditioning sets. Plugging in this factorisation into Theorem 3.1 gives us the *Independent Categorical REINFORCE* estimator, or IndeCateR for short.

Proposition 4.1 (IndeCateR)**.** *Let $p(\mathbf{X})$ be a multivariate categorical probability distribution that depends on a set of parameters θ and factorises as $p(\mathbf{X}) = \prod_{d=1}^{D} p_d(X_d)$, then the gradient of the expectation $\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X})]$ can be estimated with*

$$\sum_{d=1}^{D} \sum_{x \in \mathcal{X}(X_d)} \nabla_\theta p_d(x) \frac{1}{N} \sum_{n_d=1}^{N} f(\mathbf{x}_{\neq d}^{(n_d)}; x); \qquad (7)$$

where $\mathbf{x}_{\neq d}^{(n_d)}$ are samples drawn from $p(\mathbf{X}_{\neq d})$.

Example 4.2 (Independent Factorisation)**.** Let us consider a multivariate distribution involving three binary, independent categorical random variables. Concretely, this gives us

$$p(X_1; X_2; X_3) = p_1(X_1)p_2(X_2)p_3(X_3); \qquad (8)$$

where X_1, X_2 and X_3 can take values from the set $\mathcal{X}(X) = \{0; 1\}$. Taking this specific distribution and plugging it into Equation (7) for the IndeCateR gradient estimator now gives us

$$\sum_{d=1}^{3} \sum_{x \in \{0;1\}} \nabla_\theta p_d(x) \frac{1}{N} \sum_{n_d=1}^{N} f(\mathbf{x}_{\neq i}^{(n_d)}; x): \qquad (9)$$

In order to understand the difference between the Log-Derivative trick and the CatLog-Derivative trick, we are going to look at the term for $d = 2$ and consider the single-sample estimate

$$\nabla_\theta p_2(0) f(x_1; 0; x_3) + \nabla_\theta p_2(1) f(x_1; 1; x_3); \qquad (10)$$

where x_1 and x_3 are sampled values for the random variables X_1 and X_3. These samples would be different ones for $d \neq 2$. The corresponding complete single sample estimate using REINFORCE instead of IndeCateR would be $\nabla_\theta p_2(x_2) f(x_1; x_2; x_3)$: We see that for REINFORCE we sample all the variables whereas for IndeCateR we perform the explicit sum for each of the random variables in turn and only sample the remaining variables.

Note how, in the case of $D = 1$, Equation (7) reduces to the exact gradient. With this in mind, we can interpret IndeCateR as computing exact gradients for each single random variable X_d with respect to an approximation of the function $\mathbb{E}_{\mathbf{x}_{\neq d} \sim p(\mathbf{x}_{\neq d})}[f(\mathbf{X}_{\neq d}; X_d)]$.

Computational Complexity Consider Equation (7) and observe that none of the random variables has a sample space larger than $K = \max_d(|\mathcal{X}(X_d)|)$. Computing our gradient estimate requires performing three nested sums with lower bound 1 and upper bounds equal to D, K and N, respectively. These summations result in a time complexity of $O(D \cdot K \cdot N)$. Leveraging modern AI accelerators, they can be parallelised to obtain a time complexity of $O(\log D + \log K + \log N)$, which allows for the deployment of IndeCateR in modern deep architectures.

5. Related Work

The work closest related to ours is the RAM estimator introduced by Tokui & Sato (2017). The general idea is to first reparametrise the probability distributions such that they no longer depend on any parameters and to then perform a marginalization. We show in Section 3 that this reparametrisation step is unnecessary. Avoiding reparametrisation has the major advantage that we explicitly retain the conditional dependency structure in the CateR estimator, which allows us to trivially build a special purpose estimator for (conditionally) independent distributions (IndeCateR). Moreover, Tokui & Sato (2017) did not study the setting of a shared parameter space θ between distributions, although this being the most common setting in modern deep-discrete and neural-symbolic architectures. We rectify this omission and show that efficiently implemented Rao-Blackwellised gradient estimators for categorical random variables are a viable option in practice orthogonal to variance reduction schemes based on control variates.

Such variance reduction methods for REINFORCE aim to reduce the variance by subtracting a mean-zero term, called the baseline, from the estimate (Bengio et al., 2013; Paisley et al., 2012). Progress in this area is mainly driven by multi-sample, i.e., sample-dependent baselines (Grathwohl et al., 2018; Titsias & Shi, 2022; Tucker et al., 2017), and leave-one-out baselines (Kool et al., 2019; 2020; Mnih & Rezende, 2016; Richter et al., 2020). Variance can further be reduced by coupling multiple samples and exploiting their dependencies (Dimitriev & Zhou, 2021; Dong et al., 2021; Yin et al., 2020). A general drawback of baseline variance reduction methods is that they often involve a certain

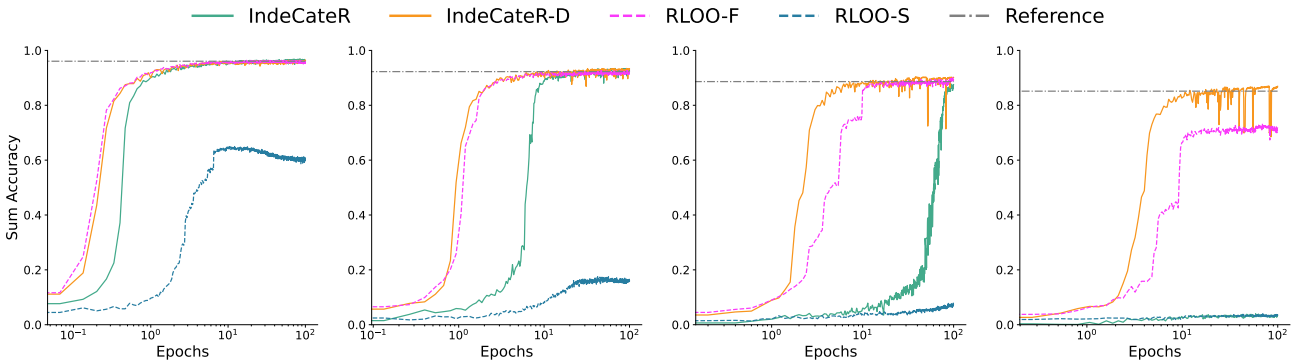


Figure 1: Test set accuracy of predicting the correct sum value versus number of epochs for the MNIST addition. From left to right, the plots show curves for 4, 8, 12, and 16 MNIST digits. The dashed, grey line at the top of each plot represents a hypothetical sum-classifier that has access to a 99% accurate MNIST digit classifier. The x-axis is in log-scale.

amount of computational overhead, usually in the form of learning optimal parameters or computing statistics. This computational overhead can be justified by assuming that the function in the expectation is costly to evaluate. In such cases, it might pay off to perform extra computations if that means the expensive function is evaluated fewer times.

Another popular, yet diametrically opposite, approach to low variance gradient estimates for categorical random variables is the concrete distribution (Maddison et al., 2017). The concrete distribution is a continuous relaxation of the categorical distribution using the Gumbel-Softmax trick (Jang et al., 2017). Its main drawback is that it results in biased gradient estimates. Even though this bias can be controlled by a temperature parameter, the tuning of this parameter is a highly non-trivial matter in practice.

Both approaches can additionally benefit from Rao-Blackwellisation, as illustrated by Liu et al. (2019) for REINFORCE and Paulus et al. (2021) for relaxed methods. However, these works still only focus on the univariate case. In contrast, we perform Rao-Blackwellisation by exploiting multivariate dependencies. In general, reducing variance by exploiting dependencies is orthogonal to much of the existing literature. Hence, it might prove interesting to examine whether it is possible to combine existing techniques with our work for a further reduction in variance.

6. Experiments

We study the behaviour of IndeCateR on a problem from the neural-symbolic literature, the addition of MNIST digits (Manhaeve et al., 2018). Given a set of D MNIST digits, the task is to predict the sum of the digits. The only provided supervision is the correct sum and no direct label of any digit. The difficulty of the problem scales exponentially, as there are 10^D states in the sample space. There are only $10D + 1$ possible labels, resulting in very sparse supervision. In the field of neural-symbolic AI such problems are either solved exactly (Manhaeve et al., 2018) or by simplifying the underlying combinatorial structure (Huang et al., 2021; Manhaeve et al., 2021). Exact methods scale very poorly while simplifying the combinatorics introduces problematic biases. In contrast, we will study the MNIST addition problem using sampling and unbiased gradient estimators. The general architecture is as follows. Each of the D different MNIST images is passed through a neural classifier, which gives probabilities for each class. These probabilities are used to sample a number between 0 and 9 for each image. The numbers are summed up and compared to the label using a binary cross-entropy loss.

Using IndeCateR in a neural-symbolic setting we achieve two things. On the one hand, we use the sampling in IndeCateR as an unbiased search, replacing the usual symbolic search. On the other, we render this stochastic search differentiable by estimating gradients instead of performing the costly exact computation.

We compare IndeCateR to REINFORCE leave-one-out (RLOO) (Kool et al., 2019; Salimans & Knowles, 2014) as it is a strong representative of methods reducing variance for REINFORCE (Figure 1). IndeCateR is given 10 samples leading to D 10 10 function evaluations. Two instances of RLOO are used, RLOO-F and RLOO-S. The former uses D 10 10 samples and function evaluations while the latter only uses 10. IndeCateR-D is an augmented version of IndeCateR that takes 10 new samples for each of the D terms in Equation (7), which would not be possible following the work of Tokui & Sato (2017). We see that IndeCateR-D is the only method capable of scaling and solving the MNIST addition for 16 digits.

Acknowledgements

This research is funded by TAILOR, a project from the EU Horizon 2020 research. It was also supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. We also have to acknowledge support from Flanders AI, FWO and the KU Leuven Research Fund. Emanuele Sansone is partially funded by the KU Leuven Research Fund (C14/18/062) and the Flemish Government (AI Research Program).

References

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

Blackwell, D. Conditional expectation and unbiased sequential estimation. *The Annals of Mathematical Statistics*, 1947.

Casella, G. and Robert, C. P. Rao-blackwellisation of sampling schemes. *Biometrika*, 1996.

De Smet, L., Martires, P. Z. D., Manhaeve, R., Marra, G., Kimmig, A., and De Readt, L. Neural probabilistic logic programming in discrete-continuous domains. *UAI*, 2023.

Dimitriev, A. and Zhou, M. Carms: Categorical-antithetic-reinforce multi-sample gradient estimator. *NeurIPS*, 2021.

Dong, Z., Mnih, A., and Tucker, G. Coupled gradient estimators for discrete latent variables. *NeurIPS*, 2021.

Grathwohl, W., Choi, D., Wu, Y., Roeder, G., and Duvenaud, D. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *ICLR*, 2018.

Huang, J., Li, Z., Chen, B., Samel, K., Naik, M., Song, L., and Si, X. Scallop: From probabilistic deductive databases to scalable differentiable reasoning. *NeurIPS*, 2021.

Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *ICLR*, 2017.

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

Kool, W., van Hoof, H., and Welling, M. Estimating gradients for discrete random variables by sampling without replacement. *ICLR*, 2020.

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

LeCun, Y. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Liu, R., Regier, J., Tripuraneni, N., Jordan, M., and Mcauliffe, J. Rao-blackwellized stochastic gradients for discrete distributions. *ICML*, 2019.

Maddison, C. J., Mnih, A., and Teh, Y. W. The concrete distribution: A continuous relaxation of discrete random variables. *ICLR*, 2017.

Manhaeve, R., Dumancic, S., Kimmig, A., Demeester, T., and De Raedt, L. Deepproblog: Neural probabilistic logic programming. *advances in neural information processing systems*, 31, 2018.

Manhaeve, R., Marra, G., and De Raedt, L. Approximate inference for neural probabilistic logic programming. *KR*, 2021.

Mnih, A. and Rezende, D. Variational inference for monte carlo objectives. *International Conference on Machine Learning*, 2016.

Niepert, M., Minervini, P., and Franceschi, L. Implicit mle: backpropagating through discrete exponential family distributions. *NeurIPS*, 2021.

Paisley, J., Blei, D. M., and Jordan, M. I. Variational bayesian inference with stochastic search. *ICML*, 2012.

Paulus, M. B., Maddison, C. J., and Krause, A. Rao-blackwellizing the straight-through gumbel-softmax gradient estimator. *ICLR*, 2021.

Radhakrishna Rao, C. Information and accuracy attainable in the estimation of statistical parameters. *Bulletin of the Calcutta Mathematical Society*, 1945.

Richter, L., Boustati, A., Nüsken, N., Ruiz, F., and Akyildiz, O. D. Vargrad: a low-variance gradient estimator for variational inference. *NeurIPS*, 2020.

Rolfe, J. T. Discrete variational autoencoders. *ICLR*, 2017.

Ruiz, F. J. R., Titsias, M. K., and Blei, D. M. The generalized reparameterization gradient. *NeurIPS*, 2016.

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

Titsias, M. and Shi, J. Double control variates for gradient estimation in discrete latent variable models. *AISTATS*, 2022.

Tokui, S. and Sato, I. Evaluating the variance of likelihood-ratio gradient estimators. *ICML*, 2017.

Tucker, G., Mnih, A., Maddison, C. J., Lawson, J., and Sohl-Dickstein, J. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. *NeurIPS*, 2017.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Reinforcement learning*, 1992.

Wu, J., Hu, W., Xiong, H., Huan, J., Braverman, V., and Zhu, Z. On the noisy gradient descent that generalizes as sgd. In *ICML*. PMLR, 2020.

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

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

A. Proofs of Theorems

A.1. Proof of Theorem 3.1

Theorem 3.1 (CatLog-Derivative Trick)**.** *Let $p(\mathbf{X})$ be a multivariate categorical probability distribution that depends on a set of parameters θ, then it holds that $\nabla_\theta \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X})]$ is equal to*

$$\sum_{d=1}^{D} \sum_{x \in \mathcal{X}(X_d)} \mathbb{E}_{\mathbf{x}_{<d} \sim p(\mathbf{x}_{<d})}[\nabla_\theta p(x \mid \mathbf{x}_{<d}) \, \mathbb{E}_{\mathbf{x}_{>d} \sim p(\mathbf{x}_{>d} \mid x, \mathbf{x}_{<d})}[f(\mathbf{x}_{\neq d}, x)]]. \tag{5}$$

Proof. We start by applying the standard log-derivative trick and fill in the product form of the categorical distribution followed by pulling this product out of the logarithm and the expectation

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X}) \nabla_\theta \log p(\mathbf{X})] = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}\left[f(\mathbf{X}) \nabla_\theta \log \textstyle\prod_{d=1}^{D} p(X_d \mid \mathbf{X}_{<d})\right]; \tag{11}$$

$$= \sum_{d=1}^{D} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X}) \nabla_\theta \log p(X_d \mid \mathbf{X}_{<d})]. \tag{12}$$

To continue, we write out the expectation explicitly and write the sum for the random variable X_d separately, resulting in

$$\sum_{d=1}^{D} \sum_{\mathbf{x} \in \mathcal{X}(\mathbf{X}_{\neq d})} \sum_{x \in \mathcal{X}(X_d)} p(\mathbf{x}_{\neq d}, x) f(\mathbf{x}_{\neq d}, x) \nabla_\theta \log p(x \mid \mathbf{x}_{<d}). \tag{13}$$

Next, we factorize the joint probability $p(\mathbf{x}_{\neq d}, x)$ as $p(\mathbf{x}_{>d} \mid x, \mathbf{x}_{<d}) p(x \mid \mathbf{x}_{<d}) p(\mathbf{x}_{<d})$. Multiplying the second of these factors with $\nabla_\theta \log p(x \mid \mathbf{x}_{<d})$ gives us $\nabla_\theta p(x \mid \mathbf{x}_{<d})$. Finally, plugging $\nabla_\theta p(x \mid \mathbf{x}_{<d})$ into Equation (13) gives the statement for the theorem. □

A.2. Proof of Proposition 3.3

Proposition A.1. *The CateR estimator Rao-Blackwellises REINFORCE.*

Proof. We start from Equation (4) for the REINFORCE estimator, where we ignore the first term and factorize the probability distribution similar to Equation (12)

$$\nabla_\theta \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X})] \quad \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}^{(n)}) \nabla_\theta \log p(\mathbf{x}^{(n)}) \quad \sum_{d=1}^{D} \frac{1}{N} \sum_{n=1}^{N} f(\mathbf{x}^{(n)}) \nabla_\theta \log p(x_d^{(n)}). \tag{14}$$

For notational conciseness, we dropped the subscript on n_d and simply use n to identify single samples. Now we compare Equation (6) and Equation (14) term-wise for $N \to 1$

$$\sum_{n=1}^{N} \mathbb{E}_{X_d \sim p(X_d \mid \mathbf{x}_{<d}^{(n)})}\left[f(\mathbf{x}_{<d}^{(n)}, X_d, \mathbf{x}_{>d}^{(n)}) \nabla_\theta \log p(X_d \mid \mathbf{x}_{<d}^{(n)})\right] = \sum_{n=1}^{N} f(\mathbf{x}_{<d}^{(n)}, x_d^{(n)}, \mathbf{x}_{>d}^{(n)}) \nabla_\theta \log p(x_d^{(n)}).$$

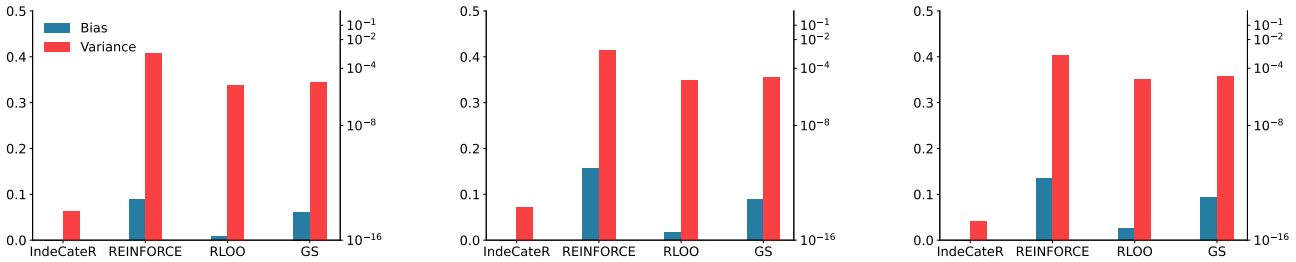
In the equation above, we see that for CateR we take the expected value for X_d (left-hand side) and compute it exactly using an explicit sum over the space $\mathcal{X}(X_d)$, whereas for REINFORCE (right-hand side) we use sampled values. This means, in turn, that on the left we have the Rao-Blackwellised version of the right-hand side. Doing this for every d gives us a Rao-Blackwellised version for REINFORCE, i.e., the CateR estimator. □

A.3. Proof of Proposition 4.1

Proposition A.2 (IndeCateR)**.** *Let $p(\mathbf{X})$ be a multivariate categorical probability distribution that depends on a set of parameters θ and factorises as $p(\mathbf{X}) = \prod_{d=1}^{D} p_d(X_d)$, then the gradient of the expectation $\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{X})]$ can be estimated with*

$$\sum_{d=1}^{D} \sum_{x \in \mathcal{X}(X_d)} \nabla_\theta p_d(x) \frac{1}{N} \sum_{n_d=1}^{N} f(\mathbf{x}_{\neq d}^{(n_d)}, x); \tag{7}$$

where $\mathbf{x}_{\neq d}^{(n_d)}$ are samples drawn from $p(\mathbf{X}_{\neq d})$.



(a) $D = 12$ and $|\Omega(X_d)| = 3$ (b) $D = 6$ and $|\Omega(X_d)| = 10$ (c) $D = 3$ and $|\Omega(X_d)| = 100$

Figure 2: We report the (empirical) bias and variance for the different estimators and distributions in comparison to the exact gradient. Bias and variance were computed using sample means and averaging them over 1000 runs.

Proof. We start by looking at the expression in Equation (5). Using the fact that we have a set of independent random variables, we can simplify $p(x \mid x_{<d})$ to $p_d(x)$. As a result, the gradient of the expected value can be rewritten as

$$\nabla \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}[f(\mathbf{x})] = \sum_{d=1}^{D} \sum_{x=1}^{K} \mathbb{E}_{\mathbf{x}_{<d} \sim p(\mathbf{x}_{<d})} \nabla p_d(x) \mathbb{E}_{\mathbf{x}_{>d} \sim p(\mathbf{x}_{>d})}[f(\mathbf{x}_{<d}, x, \mathbf{x}_{>d})] \tag{15}$$

$$= \sum_{d=1}^{D} \sum_{x=1}^{K} \nabla p_d(x) \mathbb{E}_{\mathbf{x}_{\neq d} \sim p(\mathbf{x}_{\neq d})}[f(\mathbf{x}_{<d}, x, \mathbf{x}_{>d})] \tag{16}$$

Drawing N samples for the $D - 1$ independent random variables $\mathbf{x}_{\neq d}$ and for each term in the sum over d then gives us the estimate stated in the proposition. □

B. Additional Experiments

Apart from the provided neural-symbolic experiment, we performed two more traditional experiments from the gradient estimation literature. A synthetic study of the quality of the IndeCateR gradients and the optimisation of a discrete variational auto-encoder (DVAE).

B.1. Synthetic: Exact Gradient Comparison

For small enough problems the gradient for multivariate categorical random variables can be computed exactly via explicit enumeration. Inspired by Niepert et al. (2021), we compare the estimates of

$$\nabla \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}\left[\sum_{d=1}^{D} \log p(x_d = b_d)\right] \tag{17}$$

to its exact value. Here, θ are the logits that directly parametrise a categorical distribution $p(\mathbf{x}) = \prod_{d=1}^{D} p(x_d)$ and b_d denotes an arbitrarily chosen element of $\Omega(x_d)$. We compare the gradient estimates from IndeCateR, REINFORCE, RLOO, and Gumbel-Softmax (GS) by varying the number of distributions D and the cardinality of the distributions.

In Figure 2 we show the (empirical) bias and variance for the different estimators. Each estimator was given 1000 samples, while IndeCateR was only given a single one. Hence, IndeCateR has the fewest function evaluations as $D \cdot K$ is smaller than 1000 for each configuration. IndeCateR offers gradient estimates close to the exact ones with orders of magnitude lower variance for all three settings. RLOO exhibits the smallest difference in bias, yet it can not compete in terms of variance. Furthermore, the computation times were of the same order of magnitude for all methods. This is in stark contrast to the estimator presented by Tokui & Sato (2017), where a two-fold increase in computation time of RAM with respect to REINFORCE is reported.

B.2. Synthetic: Optimisation

We now study an optimization setting (Titsias & Shi, 2022), where the goal is to maximise the expected value

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})}\left[\frac{1}{D} \sum_{i=1}^{D} (x_i - 0.499)^2\right], \tag{18}$$

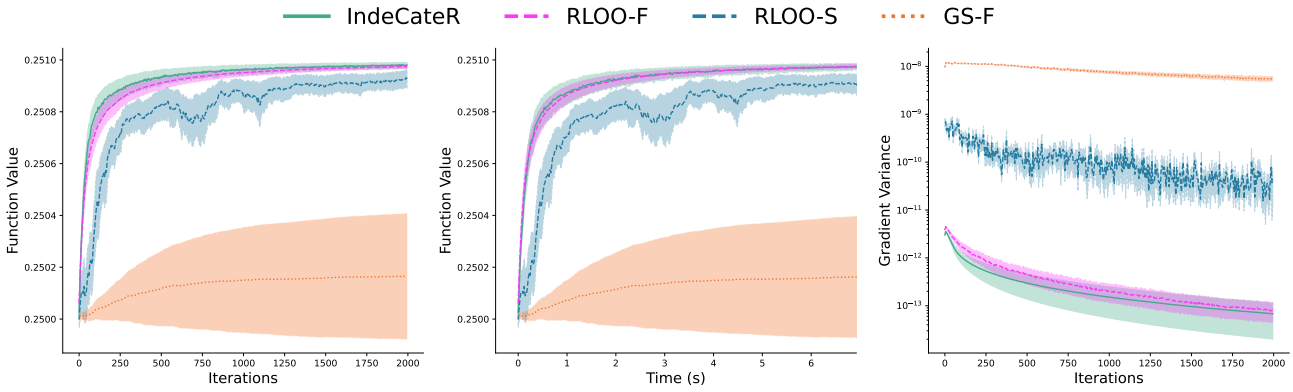


Figure 3: We plot the function value for different estimators against iterations (left) and time (middle). On the right, we plot the variance of the gradients against iterations. Statistics were obtained by taking the average and standard error over 10 runs. IndeCateR and RLOO-S both use 2 samples, while RLOO-F and Gumbel-Softmax (GS) use 800 samples. The number of function evaluations is equal for IndeCateR, RLOO-F, and GS-F. We performed a hyperparameter search for the learning rate and the temperature of GS-F. Parameters were optimised using RMSProp.

where $p(\mathbf{X})$ factorizes into D independent binary random variables. The true maximum is given by $p(X_d = 1) = 1$ for all d. This task is challenging because of the small impact of the individual values for each X_d on the expected value for higher values of D. We set D to 200 and report the results in Figure 3, where we compare IndeCateR to RLOO and Gumbel-SoftMax.

In Figure 3 and subsequent figures we use the notation RLOO-F and RLOO-S, which we define as follows. If IndeCateR takes N samples, then it performs $D \quad K \quad N$ functional evaluations with $K = \max_d j \quad (X_d)j$. As such, we define RLOO-S as drawing the same number of samples as IndeCateR, which translates to N function evaluations. For RLOO-F we match the number of function evaluations, which means that it takes $D \quad K \quad N$ samples. We give an analogous meaning to GS-S and GS-F for the Gumbel-SoftMax gradient estimator.

IndeCateR distinguishes itself by having both the lowest variance and quickest convergence across all methods, even compared to RLOO-F. Additionally, the time to compute all gradient estimates does not differ significantly for the different methods and leads to the same conclusions. It is striking to see that the Gumbel-Softmax estimator struggles in this task, which is likely due to its bias in combination with the sensitive loss function.

B.3. Discrete Variational Auto-Encoder

As a third experiment we analyse the ELBO optimisation behaviour of a discrete variational auto-encoder (DVAE) (Rolfe, 2017). We optimise the DVAE on the three main datasets from the literature, being MNIST (LeCun, 1998), F-MNIST (Xiao et al., 2017) and Omniglot (Lake et al., 2015). The encoder component of the network has three dense hidden layers of sizes 384 and 256 ending in a latent 200-dimensional Bernoulli variable. The decoder takes samples from this variable followed by hidden layers of size 256, 384 and 784. IndeCateR again uses two samples, hence we can compare to the same configurations of RLOO and Gumbel-Softmax as in Section B.2, i.e., equal samples (GS-S and RLOO-S) and equal function evaluations (GS-F and RLOO-F).

As evaluation metrics, we show the negated training and test set ELBO in combination with the variance of the gradients throughout training. We opted to report all metrics in terms of computation time (Figure 4), but similar results in terms of iterations are given in the appendix.

A first observation is that IndeCateR performs remarkably well in terms of convergence speed. It beats all other methods on both MNIST and F-MNIST in terms of training ELBO, only having to yield to the Gumbel-Softmax trick with equal function evaluations on Omniglot. However, we can observe a disadvantage of the quick convergence in terms of generalisation performance when looking at the test set ELBO. RLOO-F and IndeCateR exhibit overfitting on the training data for MNIST, resulting in an overall higher negative test set ELBO. We speculate that the relaxation for the Gumbel-Softmax or the higher variance (Wu et al., 2020) of other methods act as a regulariser for the network. As one would ideally like to separate