IMPROVED GRADIENT ESTIMATORS FOR STOCHASTIC DISCRETE VARIABLES

Anonymous authors
Paper under double-blind review

ABSTRACT

In many applications we seek to optimize an expectation with respect to a distribution over discrete variables. Estimating gradients of such objectives with respect to the distribution parameters is a challenging problem. We analyze existing solutions including finite-difference (FD) estimators and continuous relaxation (CR) estimators in terms of bias and variance. We show that the commonly used Gumbel-Softmax estimator is biased and propose a simple method to reduce it. We also derive a simpler piece-wise linear continuous relaxation that also possesses reduced bias. We demonstrate empirically that reduced bias leads to a better performance in variational inference and on binary optimization tasks.

1 INTRODUCTION

Discrete stochastic variables arise naturally in many applications including topic modeling, semi-supervised learning, clustering, variational memory addressing and reinforcement learning. In many cases, the objective is to minimize an expectation of a function of discrete random variables with respect to the distribution parameters:

$$\min_{\boldsymbol{\phi}} \mathcal{L}[\boldsymbol{\phi}], \quad \text{where } \mathcal{L}[\boldsymbol{\phi}] = \mathbb{E}_{q_{\boldsymbol{\phi}}}\big[f(\boldsymbol{z})\big] = \sum_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}) f(\boldsymbol{z}). \tag{1}$$

Here, $\boldsymbol{z}$ is a vector of discrete variables under a $\boldsymbol{\phi}$-parameterized distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$. For example, in variational inference, $f(\boldsymbol{z})$ is the variational lower bound and $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is the approximating posterior.

Eq. (1) is commonly minimized by gradient-based methods, which require estimating the gradient $\partial_{\boldsymbol{\phi}}\mathcal{L}[\boldsymbol{\phi}]$ (see Schulman et al. (2015) for an overview). For continuous random variables whose sampling can be reparameterized as a function of other parameter-independent random variables, the *reparameterization trick* (Kingma & Welling (2013), Rezende et al. (2014)) gives low variance gradient estimates. However, this trick is infeasible for discrete variables due to the discontinuous cumulative distribution function (CDF).

Extensions of the reparameterization trick to discrete distributions can be grouped into finite-difference (FD) estimators, and continuous relaxation (CR) estimators. We present a summary of three recent FD estimators (Tokui & Sato (2017); Yin & Zhou (2018); Lorberbom et al. (2018) to guide practitioners to the salient aspects of their bias-variance tradeoff and computational complexity. Most importantly, we propose a scalable (but still unbiased) variant of Tokui & Sato (2017) that trades decreased computation for slightly increased variance.

We examine CR estimators including the popular Gumbel-Softmax (Jang et al. (2016); Maddison et al. (2016)) in terms of the gradient bias and propose a new method to reduce the bias of all CR estimators for both binary and categorical variables. Based upon this understanding, we develop a piecewise-linear estimator for binary and categorical variables that is simpler and less biased than Gumbel-Softmax.

Lastly, we provide empirical evidence that these improved estimators allow for faster optimization when training variational autoencoders and when optimizing a continuous relaxation of a combinatorial optimization problem.

1.1 RELATED WORK

The most generic approximator of $\partial_{\phi}\mathcal{L}[\phi]$ is the score function (SF) estimator (a.k.a. REINFORCE Williams (1992), Glynn (1990)). SF suffers from high variance and many remedies have been proposed to reduce this variance (Mnih & Gregor (2014); Gregor et al. (2013); Gu et al. (2015); Mnih & Rezende (2016); Tucker et al. (2017); Grathwohl et al. (2017)). Unbiased estimators that require multiple function evaluations have also been proposed Tokui & Sato (2017), Titsias & Lázaro-Gredilla (2015), Lorberbom et al. (2018), Yin & Zhou (2018). These estimators have lower variance but can be computationally demanding. CR estimators often trade bias for variance and previous CR proposals include straight-through estimators (Bengio et al. (2013); Raiko et al. (2014)), the Gumbel-Softmax estimator (Jang et al. (2016); Maddison et al. (2016)), and overlapping smoothing (Vahdat et al. (2018b;a); Rolfe (2016)).

2 THE REPARAMETERIZATION AND MARGINALIZATION ESTIMATOR

We begin a summary of FD estimators with the reparameterization and marginalization (RAM) method of Tokui & Sato (2017). For a single binary variable the expectation in Eq. (1) is enumerated as $\mathcal{L} = \sum_z q_\phi(z)f(z) = q_\phi f(1) + (1-q_\phi)f(0)$, where $q_\phi \equiv q_\phi(z=1)$. The gradient is:

$$\partial_\phi \mathcal{L} = \partial_{\boldsymbol{\phi}} q_\phi(f(1) - f(0)) = \partial_\phi l \; q_\phi(1-q_\phi)(f(1)-f(0)), \tag{2}$$

where $q_\phi = \sigma(l_\phi) = (1+e^{-l_\phi})^{-1}$ and $l_\phi = \text{logit}(q_\phi)$. This derivative involves two function evaluations and contains a finite-difference of $f(z)$. Eq. (2) is an unbiased zero-variance estimate since the summation over z is done explicitly. The generalization to M factorially-distributed random variables $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \prod_{i=1}^M q_{\boldsymbol{\phi},i}(z_i)$ is

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_i \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i} \sum_{\boldsymbol{z}_{\setminus i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\setminus i})(f(z_i=1,\boldsymbol{z}_{\setminus i}) - f(z_i=0,\boldsymbol{z}_{\setminus i})), \tag{3}$$

where again the summation over z_i is performed exactly and $q_{\boldsymbol{\phi},i} \equiv q_{\boldsymbol{\phi},i}(z_i=1)$. The summation over $\boldsymbol{z}_{\setminus i}$ can be estimated with a single sample but the derivative requires $M+1$ function evaluations. This limits the applicability of RAM to moderately-sized models. Note that both $f(z_i=0,\boldsymbol{z}_{\setminus i})$ and $f(z_i=1,\boldsymbol{z}_{\setminus i})$ are evaluated at the same $\boldsymbol{z}_{\setminus i}$ which leads to a lower-variance estimator. For hierarchical $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \prod_{i=1}^M q_{\boldsymbol{\phi},i}(z_i|\boldsymbol{z}_{<i})$ the derivative takes the form:

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_i \sum_{\boldsymbol{z}_{<i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{<i}) \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i} \left[\sum_{\boldsymbol{z}_{>i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{>i}|1,\boldsymbol{z}_{<i}) f(\boldsymbol{z}_{>i},1,\boldsymbol{z}_{<i}) - \sum_{\boldsymbol{z}_{>i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{>i}|0,\boldsymbol{z}_{<i}) f(\boldsymbol{z}_{>i},0,\boldsymbol{z}_{<i}) \right]. \tag{4}$$

where $q_{\boldsymbol{\phi},i} \equiv q_{\boldsymbol{\phi},i}(1|\boldsymbol{z}_{<i})$. The key insight of Tokui & Sato (2017) is to use common random variates $\boldsymbol{z}_{>i}$ when sampling from $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{>i}|1,\boldsymbol{z}_{<i})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z}_{>i}|0,\boldsymbol{z}_{<i})$. This reduces both the variance and the computational cost. Tokui & Sato (2017) show that this estimator is optimal because it exactly sums over the binary variables whose probability distribution is being differentiated.

A RAM estimator can also be constructed for categorical variables. For a single one-hot encoded categorical variable $y = (y^0, ...y^{A-1}), y^a \in \{0,1\}, \sum_a y^a = 1$, the derivative of Eq. (1) is

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \partial_{\boldsymbol{\phi}} \sum_y q_{\boldsymbol{\phi}}(y) f(y) = \sum_a (\partial_{\boldsymbol{\phi}} l^a_{\boldsymbol{\phi}}) \sum_b q^a_{\boldsymbol{\phi}} q^b_{\boldsymbol{\phi}} (f^a - f^b), \tag{5}$$

where $f^a = f(y^a=1)$, and $q^a_{\boldsymbol{\phi}} = e^{l^a_{\boldsymbol{\phi}}} / \sum_b e^{l^b_{\boldsymbol{\phi}}}$. The generalization to many categorical variables proceeds as for binary variables. For example, the derivative of a factorial categorical distribution over $\boldsymbol{y} = \{y^a_i \mid 0 \le a < A, 1 \le i \le M\}$ is

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_i \sum_{\boldsymbol{y}_{\setminus i}} q_{\boldsymbol{\phi}}(\boldsymbol{y}_{\setminus i}) \sum_{a,b} (\partial_{\boldsymbol{\phi}} l^a_{\boldsymbol{\phi},i}) q^a_{\boldsymbol{\phi},i} q^b_{\boldsymbol{\phi},i} \big(f(y^a_i = 1, \boldsymbol{y}_{\setminus i}) - f(y^b_i = 1, \boldsymbol{y}_{\setminus i}) \big). \tag{6}$$

This derivative can again be estimated with a single sample but requires MA function evaluations. Since RAM is unbiased and has the minimal variance (due to the explicit summation over the differentiated variable), we use it as a baseline to evaluate computationally cheaper alternatives.

2.1 SAMPLED REPARAMETERIZATION AND MARGINALIZATION

We propose a modification to RAM that allows us to trade decreased computational cost for increased variance. For binary variables, $\partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i} = q_{\boldsymbol{\phi},i}(1 - q_{\boldsymbol{\phi},i})\partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi},i}$ where $q_{\boldsymbol{\phi},i} = \sigma(l_{\boldsymbol{\phi},i})$, so that each term in Eq. (3) or (4) is proportional to $q_{\boldsymbol{\phi},i}(1 - q_{\boldsymbol{\phi},i})$. In many applications, we observe that the distribution of a large number of variables ($q_{\boldsymbol{\phi},i}$) are drawn to 0 or 1 early during optimization. Such variables have negligible contribution to the full derivative. We exploit this observation to reduce the computational cost by including variable z_i in the full gradient with probability $p_i = 4q_{\boldsymbol{\phi},i}(1-q_{\boldsymbol{\phi},i})/\beta$, where β is an adjustable hyperparameter. This means that we replace the derivative in Eq. (3) with

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = E_{\xi\sim p}\Big[\sum_i (\partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi},i})\frac{\beta\xi_i}{4}\sum_{\boldsymbol{z}_{\setminus i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\setminus i})\big(f(z_i = 1, \boldsymbol{z}_{\setminus i}) - f(z_i = 0, \boldsymbol{z}_{\setminus i})\big)\Big], \tag{7}$$

where $\xi_i \in \{0, 1\}$ are Bernoulli variables with probabilities p_i indicating whether z_i is included or not. We evaluate Eq. (7) by sampling ξ and only then evaluating non-zero terms. In Section 4 we show that in the context of variational inference on MNIST the number of function evaluations is reduced by an order of magnitude while still allowing for effective optimization. As always, this computational saving comes at the cost of increased gradient variance which slows training.

In the categorical case, each term in Eq. (6) is accompanied by $q^a_{\boldsymbol{\phi},i} q^b_{\boldsymbol{\phi},i}$ that can be used to assign importance to the (a, b) edge of the simplex for variable y_i. The computational cost MA can be reduced by keeping each term with probability $p_i^{a,b} = 4q^a_{\boldsymbol{\phi},i} q^b_{\boldsymbol{\phi},i}/\beta$.

Appendices A and B summarize two other FD estimators, ARGMAX Lorberbom et al. (2018) and ARM Yin & Zhou (2018), by noting their bias-variance tradeoff and computational complexity.

3 CONTINUOUS RELAXATION ESTIMATORS

Unlike FD estimators that use multiple function evaluations to approximate the gradient, continuous relaxation estimators extend the reparameterization trick to discrete variables by approximating them with continuous variables: $z \to \zeta = g^{\beta}_{\phi}(\rho)$ for $\rho \in \mathcal{U}[0, 1]$, where β is a parameter that controls the approximation. The objective function $\mathcal{L}[\boldsymbol{\phi}] = \sum_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}) f(\boldsymbol{z})$ is replaced with $\tilde{\mathcal{L}}[\boldsymbol{\phi}] = \mathbb{E}_{\boldsymbol{\rho}}\left[f\big(\boldsymbol{\zeta} = g^{\beta}_{\phi}(\rho)\big)\right]$ and the gradients are computed using chain rule

$$\partial_{\boldsymbol{\phi}}\tilde{\mathcal{L}}[\phi] = \mathbb{E}_{\boldsymbol{\rho}}\left[\sum_i \partial_{\zeta_i} f(\boldsymbol{\zeta})\partial_{\boldsymbol{\phi}}\zeta_i\Big|_{\boldsymbol{\zeta}=g(\rho_i; q_{\boldsymbol{\phi},i}(1|\boldsymbol{\zeta}_{<i}))}\right] = \mathbb{E}_{\boldsymbol{\rho}}\left[\sum_i \partial_{\zeta_i} f(\boldsymbol{\zeta})(\partial_{q_i}\zeta_i)(\partial_{\boldsymbol{\phi}} q_i)\Big|_{\substack{\boldsymbol{\zeta}=g(\rho_i; q_i)\\ q_i = q_{\boldsymbol{\phi},i}(1|\boldsymbol{\zeta}_{<i})}}\right] \tag{8}$$

These gradients can be computed efficiently by automatic differentiation libraries. However, because the objective function is changed, CR estimators in Eq. (8) are biased. Nevertheless, in practice the bias is often small enough to allow for effective optimization.

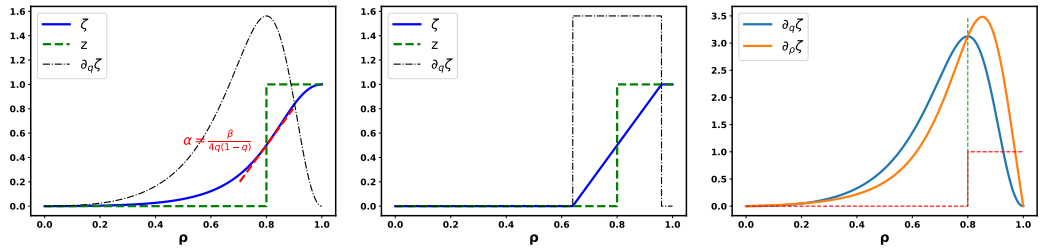
3.1 THE GUMBEL-SOFTMAX ESTIMATOR

The most popular CR estimator is Gumbel-Softmax (GSM) (Jang et al. (2016); Maddison et al. (2016))[1]. For binary variables, this relaxation takes the simple form

$$\zeta_i = \sigma\left(\beta\left[\sigma^{-1}\big(q_i\big) + \sigma^{-1}(\rho_i)\right]\right), \tag{9}$$

where $q_i \equiv q_{\phi,i}(1|\boldsymbol{\zeta}_{<i})$. This relaxation and its derivative $\partial_q\zeta$ are shown in Fig. 1(a). β controls the sharpness of the relaxation and tunes the trade-off between the bias (closeness of ζ to z) and variance of $\partial_q\zeta$. We note that the slope of the relaxed $\zeta(\rho)$ at $\zeta = 1/2$ is $\alpha \equiv \beta/[4q(1 - q)]$ and thus becomes large when q approaches 0 or 1.

[1]More precisely, Maddison et al. (2016) considered problems in variational inference where $f(\boldsymbol{z})$ is relaxed by replacing a generative distribution with its continuous relaxation. In contrast, Jang et al. (2016) directly relaxed discrete variables $z \to \zeta$ without changing the objective, thus replacing $f(\boldsymbol{z}) \to f(\boldsymbol{\zeta})$. This does not produce a consistent objective for a probabilistic model, but as Jang et al. (2016) have shown, works well in practice. We work with generic $f(\boldsymbol{z})$ and thus only consider the approach of Jang et al. (2016).



(a) Gumbel-Softmax (b) Piecewise Linear (c) $\partial_q \zeta$

Figure 1: Continuous relaxations at $\beta = 2$ to a Bernoulli random variable $\text{Ber}(q_\phi = 0.8)$: (a) shows the GSM approximation and (b) shows a piece-wise linear approximation to the discontinuous CDF. (c) shows the effect of a modification to GSM that reduces its bias.

3.2 Improved Continuous Relaxation Estimators

In this section, we analyze the bias introduced by Eq. (8) and propose a simple method to reduce it. The bias of $\mathbb{E}_{\boldsymbol{\rho}}\left[\partial_{\zeta_i} f(\boldsymbol{\zeta})\partial_{\boldsymbol{\phi}}\zeta_i\right]$ has two sources: (a) the relaxation of ζ_j for $j \neq i$ and (b) the relaxation of ζ_i. To characterize the latter bias, we start with a single binary variable and write the gradient as the following integral:

$$\partial_\phi \mathcal{L} = \partial_\phi q_\phi \big(f(1) - f(0)\big) = \partial_\phi q_\phi \int_0^1 d\zeta\, \partial_\zeta f(\zeta) = \partial_\phi q_\phi \int_0^1 d\rho\, \frac{\partial \zeta}{\partial \rho} \partial_\zeta f(\zeta). \tag{10}$$

Here, $\zeta(\rho)$ is *any* continuous function satisfying $\zeta(0) = 0$ and $\zeta(1) = 1$. For a non-decreasing function $\zeta(\rho)$, we can view ζ as a random variable with inverse CDF $\zeta(\rho)$ and ρ as a uniform random variable $\rho \in \mathcal{U}[0, 1]$. Given that the corresponding probability density function (PDF) is $q(\zeta) = (\partial\zeta/\partial\rho)^{-1}$, we rewrite the derivative as

$$\partial_\phi \mathcal{L} = \partial_\phi q_\phi\, \mathbb{E}_{\zeta \sim q}\left[\frac{1}{q(\zeta)} \partial_\zeta f(\zeta)\right]. \tag{11}$$

This expectation can be estimated by sampling $\zeta \sim q(\zeta)$. For smooth enough $f(\zeta)$, the variance of this estimate is controlled by $\text{var}\left(q(\zeta)^{-1}\right) = \int_0^1 d\zeta\, q(\zeta)^{-1} - 1$. Thus, the more non-linear $\zeta(\rho)$ is, the higher will be the variance of estimate Eq. (11). This idea can be extended to factorial $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \prod_{i=1}^M q_{\boldsymbol{\phi},i}(z_i)$ as in Eq. (3):

$$\partial_{\boldsymbol{\phi}} \mathcal{L} = \sum_i \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i} \sum_{\boldsymbol{z}_{\setminus i}} q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\setminus i}) \int d\rho_i \frac{\partial \zeta_i}{\partial \rho_i} \partial_{\zeta_i} f(\zeta_i, \boldsymbol{z}_{\setminus i}) = \sum_i \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{\setminus i}), q(\zeta_i)} \left[\frac{\partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i}}{q(\zeta_i)} \partial_{\zeta_i} f(\zeta_i, \boldsymbol{z}_{\setminus i})\right]. \tag{12}$$

At this point there is no relationship between $q(\zeta_i)$ and $q_{\boldsymbol{\phi},i}(z_i)$, and Eq. (12) is just a higher variance version of Eq. (3). However, if we relax $\boldsymbol{z}_{\setminus i} \to \boldsymbol{\zeta}_{\setminus i}$ and choose $\zeta_j = g(\rho_j; q_j)$ we obtain a *biased* estimator

$$\partial_{\boldsymbol{\phi}} \mathcal{L} \approx \sum_i \mathbb{E}_{q(\boldsymbol{\zeta})} \left[\partial_{\zeta_i} f(\boldsymbol{\zeta}) \frac{\partial \zeta_i}{\partial \rho_i} \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i}\right], \tag{13}$$

where the bias comes from the deviation of $\boldsymbol{\zeta}_{\setminus i}$ from $\boldsymbol{z}_{\setminus i}$ in the function evaluations. In other words, Eq. (13) uses an unbiased form for the differentiated variable ζ_i and the only bias comes from relaxing the remaining variables $\boldsymbol{\zeta}_{\setminus i}$.

The variance of each term is controlled by $\text{var}\big(q(\zeta_i)^{-1}\big)$ and is reduced by making $\zeta_i(\rho_i)$ more linear. The bias is reduced by making $\boldsymbol{\zeta}$ closer $\boldsymbol{z}$. Varying $\zeta_i(\rho_i)$ between linear (low variance) and step function (low bias) allows for a controllable trade-off. We call Eq. (13) the improved continuous relaxation (ICR) estimator. The original CR estimator of Eq. (8) for factorial $q_{\boldsymbol{\phi}}$ has the form

$$\partial_{\boldsymbol{\phi}} \tilde{\mathcal{L}}[\boldsymbol{\phi}] = \sum_i \mathbb{E}_{q(\boldsymbol{\zeta})} \left[\partial_{\zeta_i} f(\boldsymbol{\zeta}) \frac{\partial \zeta_i}{\partial q_i} \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},i}\right], \tag{14}$$

and comparing this to the ICR estimate in Eq. (13), we see that CR can be transformed into ICR by the replacing $\partial_{q_i}\zeta_i$ with $\partial_{\rho_i}\zeta_i$. This change can be simply implemented using TensorFlow's $\text{stop_gradient} \equiv \mathbf{sg}$ notation by replacing

$$\zeta_i(\rho_i, q_i) \quad \text{with} \quad \zeta_i\big(\rho_i + q_i - \mathbf{sg}(q_i), \mathbf{sg}(q_i)\big) \tag{15}$$

in Eq. (8). We emphasize that the ICR estimator in Eq. (13) is less biased than the direct CR estimator because, in the case of a single variable, ICR is unbiased while CR is not. Further, this decrease in bias is not accompanied by an increase in variance. We show in Appendix C that similar benefits are obtained for hierarchical $q_{\boldsymbol{\phi}}$.

3.3 PIECE-WISE LINEAR RELAXATION

Inspired by ICR and a better understanding of variance-bias trade-off, we propose a piece-wise linear relaxation (PWL) depicted in Fig. 1(b). The linear part is centered at $\rho = 1 - q$ so that the corresponding binary variable is obtained by $z = \mathrm{round}(\zeta)$. The slope is given by $\alpha = \beta/[4q(1-q)]$ similar to the Gumbel-Softmax slope [2]. The explicit expression for PWL smoothing is

$$\zeta_q(\rho) = \left[0.5 + \alpha(\rho - (1-q))\right]_0^1, \tag{16}$$

where $[x]_0^1 \equiv \min(1, \max[0, x])$ is the hard sigmoid function. This relaxation has several attractive properties. Firstly, we have $\partial_{q_i}\zeta_i = \partial_{\rho_i}\zeta_i$, which means that the CR and ICR estimators coincide for PWL. Secondly, PWL has easily interpretable expressions for bias and variance. In the case of a function of a single variable the variance is given by $\mathrm{var}(\partial_\rho \zeta) = \alpha - 1$ while the bias in computing expectation $\sum_z f(z)$ over the relaxed distribution is $\int d\rho f(\zeta(\rho)) - \sum_z f(z) = \left(\int_0^1 dx f(x) - [f(0) + f(1)]/2\right)/\alpha$. This clearly shows that α trades bias for variance.

Thirdly, we provide a qualitative motivation for the slope to have the form $\alpha \approx 1/[q(1-q)]$. We recall that the gradient in Eq. (13) has the form $\partial_{\boldsymbol{\phi}}\mathcal{L} \approx \mathbb{E}_{q(\boldsymbol{\zeta})}\left[\sum_i \partial_{\zeta_i} f(\boldsymbol{\zeta})(\partial_{\rho_i}\zeta_i) q_i(1-q_i)\partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi},i}\right]$. For smooth enough $f(\boldsymbol{\zeta})$, the variance of this estimate arises mainly from the variance of $\partial_{\rho_i}\zeta_i$ and the relative size of the terms is controlled by $q_i(1-q_i)$. Since the variance of $\partial_{\rho_i}\zeta_i$ is given by $\mathrm{var}(\partial_{\rho_i}\zeta_i) = \alpha_i - 1$, to minimize the variance of the sum, we can choose $\mathrm{var}(\partial_{\rho_i}\zeta_i) \approx 1/[q_i(1-q_i)]$, leading to $\alpha_i \approx 1/[q_i(1-q_i)]$.

Finally, the PWL relaxation defined in Eq. (16) can be considered as the inverse CDF of the PDF:

$$q(\zeta) = (1-\epsilon)\left[(1-q)\delta(\zeta) + q\delta(\zeta - 1)\right] + \epsilon\,\mathcal{U}[0,1], \quad \text{where} \quad \epsilon = [4q(1-q)]/\beta. \tag{17}$$

Eq. (17) is a mixture of two delta distributions centered at zero and one, and a uniform distribution defined in the interval $[0, 1]$. Samples from $q(\zeta)$ are in the continuous interval with probability $[4q(1-q)]/\beta$, and $\zeta = 0/1$ have probability proportional to the probability of the binary states.

3.4 CATEGORICAL PIECE-WISE LINEAR RELAXATION

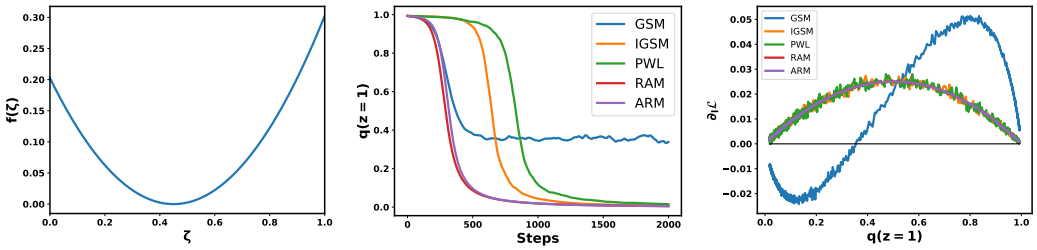
We now extend ICR estimators to categorical variables. For a single categorical variable we apply the integral representation to each edge (a, b) of the simplex in Eq. (5) and relax this pair of variables using PWL: $y \to y^{a,b} = \{y^a = \left[0.5 + \alpha^{a,b}\left(\rho^{a,b} - q^b/(q^a + q^b)\right)\right]_0^1, y^b = 1 - y^a, y^{c\neq a,b} = 0\}$ where $\rho^{a,b} \sim \mathcal{U}[0,1]$ and $\alpha^{a,b}$ is the slope. We replace the summation over the edges of the simplex by sampling one edge at a time with probability $p^{a,b} = (q^a + q^b)/(A-1)$. Details are found in Appendix D and we provide the final result:

$$\mathcal{L} = \mathbb{E}_{(a,b)\sim p^{a,b}}\left[\mathbb{E}_{\rho\in\mathcal{U}[0,1]}\left[f(\tilde{y}^{a,b})\right]\right], \tag{18}$$

where $\tilde{y}^{a,b}$ has the same value as $y^{a,b}$ but has the gradient scaled by $\gamma^{a,b} = (A-1)(q^a + q^b)$.[3] The probabilities of edge selection and the scale factor are chosen to give correct values for the objective and its gradient. Extension of this categorical PWL estimator to multivariate distributions is straightforward; For example, for factorial distributions one relaxes each categorical variable y_i by sampling an edge with probability $p_i^{a,b} = (q_i^a + q_i^b)(A-1)$. The resulting gradient is unbiased for a single variable and introduces a bias in the multivariate case which makes it an ICR estimator.

[2] More generally, the slope of the linear part α can be chosen arbitrarily as long as $\alpha \geq 0.5/\min(q, 1-q)$ so that both $\zeta = 0$ and $\zeta = 1$ have non-zero probability.

[3] In Tensorflow notation $\tilde{y}^{a,b} \equiv \mathbf{sg}(y^{a,b}) + \mathbf{sg}(\gamma^{a,b})(y^{a,b} - \mathbf{sg}(y^{a,b}))$.



(a) Relaxed function (b) Probability $q(z = 1)$ (c) Gradient $\partial_l \mathcal{L}$

Figure 2: Convex, single binary-variable toy example: (a) The relaxed objective $f(\zeta) = (\zeta - 0.45)^2$. (b) The probability $q_\phi(z = 1)$ during optimization. (c) The gradient $\partial_l \mathcal{L}$ of all estimators; the bias of GSM prevents proper minimization.

3.5 Improved Categorical Gumbel-Softmax Estimator

The sampling done in the categorical PWL estimator leads to increased variance of the gradients. As an alternative, we suggest an improved version of the categorical Gumbel-Softmax estimator. Recall that the Gumbel-Softmax estimator for the categorical case has the form:

$$\zeta^a(\rho; \boldsymbol{q}) = \text{softmax}(\beta\,(\log q^a + \log \rho^a)), \;\; \text{where} \;\; \rho^a = \frac{\log u^a}{\sum_b \log u^b}, \; u^b \sim \mathcal{U}[0, 1]. \tag{19}$$

In Eq. (20), we propose a new version of this estimator by applying the same trick as in Eq. (15). This estimator remains biased but we empirically demonstrate that its bias is reduced.

$$\zeta(\rho, \boldsymbol{q}) \to \zeta(\rho + \boldsymbol{q} - \mathbf{sg}(q), \mathbf{sg}(q)). \tag{20}$$

4 Experiments

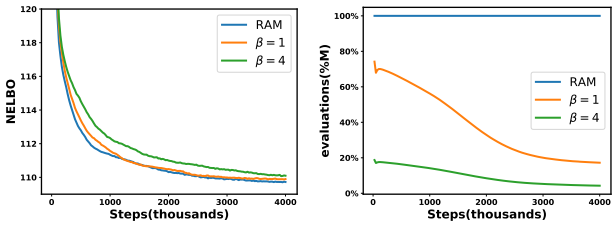
In this section we compare the FD and CR estimators and their improved variants on a number of examples. We start with one-variable toy examples that illustrate the bias of GSM estimator and then move to the training of variational auto-encoders and a combinatorial optimization problem.

4.1 Toy example

We begin with an illustrative single-variable example (Tucker et al. (2017)) with objective $\mathcal{L} = \sum_z q_\phi(z) f(z)$ where $f(z) = (z - 0.45)^2$. The relaxed convex function $f(\zeta)$ depicted in Fig. 2(a) has two local maxima, one of which is the minimum over the discrete domain. We compare five gradient methods: RAM, Eq. (2); ARM (see Appendix B); PWL, Eq. (16); GSM, Eq. (9); and improved Gumbel-Softmax (IGSM), Eq. (15) (see Appendix E for experimental details). Fig. 2(b) shows the evolution of $q_\phi(z = 1)$ during training which demonstrates the bias associated with GSM. To quantify this bias, we plot the value of the gradient of all estimators for different values of $q_\phi(z = 1)$ in Fig. 2(c). We observe that the GSM gradient has the wrong sign for a large interval in $q_\phi(z = 1)$ which prevents GSM from converging to the true minimum.

To understand the nature of the bias that GSM introduces, we plot the derivatives $\partial_q \zeta$ and $\partial_\rho \zeta$ corresponding to GSM and IGSM respectively in Fig. 1(c). We see that $\partial_q \zeta$ is biased towards the value of $z = \text{round}(\zeta)$ that has the highest probability. This means that GSM in Eq. (14) will oversample the derivative $\partial_\zeta f(\zeta)$ from the most probable mode. In example Fig. 2(a) this results in oversampling the derivative from $z = 0$ mode which creates a gradient that pushes optimization away from the true minimum $z = 0$. The bias is reduced as β is increased.

In Appendix H.1 we consider an example for a concave function over a binary variable and observe similar effects. We also consider both concave and convex functions over a categorical variable in Appendix H.2. In all scenarios, the bias of GSM prevents its convergence to the correct minimum.



(a) NELBO (b) Evaluations per sample

Figure 3: NELBO of the RAM and sampled RAM estimators on MNIST trained using the $200\text{H} - 784\text{V}$ architecture having a linear decoder: (a) plots the decrease in training NELBO for RAM and two variants of sampled RAM. (b) shows the computational savings of sampled RAM.

4.2 Discrete Variational Autoencoders

Next, we test the estimators by training variational autoencoders Kingma & Welling (2013) with discrete priors. The objective is the negative expectation lower bound on the log-likelihood (NELBO):

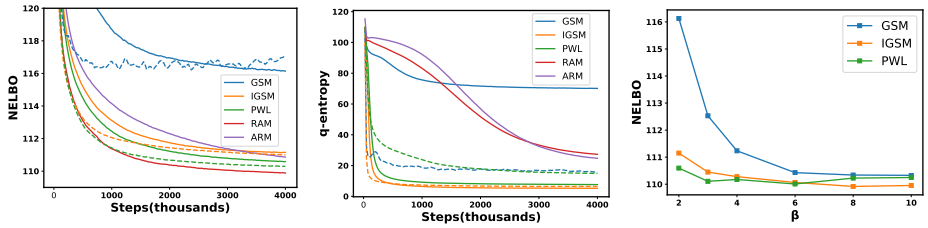
$$\mathcal{L}[\boldsymbol{\phi}, \boldsymbol{\theta}] = \sum_{\boldsymbol{x} \sim \text{data}} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{x}) \right], \text{ where } \quad f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{z}, \boldsymbol{x}) = -\log \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}. \tag{21}$$

Here, $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ is the prior, $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ is the decoder, and $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ is the approximating posterior. $\mathcal{L}$ is minimized with respect to the $\boldsymbol{\theta}$ parameters of the generative model and the $\boldsymbol{\phi}$ parameters of the approximate posterior. The latter minimization corresponds to Eq. (1) and we can apply the various estimators to propagate $\boldsymbol{\phi}$-derivatives through discrete samples $\boldsymbol{z}$. For CR estimators we replace $\boldsymbol{z}$ with $\boldsymbol{\zeta}$ to compute $\partial_{\boldsymbol{\phi}} \mathcal{L}[\boldsymbol{\phi}, \boldsymbol{\theta}] \approx \sum_{\boldsymbol{x} \sim \text{data}} \mathbb{E}_{\rho} \left[\partial_{\boldsymbol{\phi}} f_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{\zeta}, \boldsymbol{x}) \right]$, where the expectation is evaluated with a single relaxed sample per data point $\boldsymbol{x}$. The $\boldsymbol{\theta}$-derivative can be calculated directly from Eq. (21) using the discrete $\boldsymbol{z}$ variables. Thus, the $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ derivatives are evaluated separately requiring two passes through the computation graph with either relaxed or discrete samples. Jang et al. (2016) evaluate both derivatives in one pass (using $\boldsymbol{\zeta}$) thereby introducing bias in the $\boldsymbol{\theta}$ derivatives. We refer to these two possibilities as one/two-pass training and compare their performance.

Following Maddison et al. (2016); Jang et al. (2016); Tucker et al. (2017), we consider four architectures with binary variables denoted by $200\text{H} - 784\text{V}$, $200\text{H} - 200\text{H} - 784\text{V}$, $200\text{H} \sim 784\text{V}$, and $200\text{H} \sim 200\text{H} \sim 784\text{V}$ (see Appendix E for details). First, we compare the sampled RAM estimator of Eq. (7) on $200\text{H} - 784\text{V}$ model for different values of β, where probability of updating variable z_i is $p_i = 4q_i(1 - q_i)/\beta$. Fig. 3(a) shows the NELBO on the training set. As expected, increased β leads to increased gradient variance which slows training. Fig. 3(b) shows the average number of function evaluations performed in Eq. (7). Many units become deterministic early in training leading to significant computational savings with the sampled RAM method.

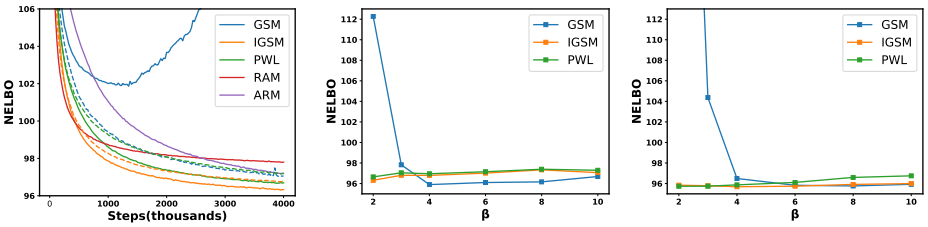
In Fig. 4(a), we include several CR estimators on the same architecture and observe that the GSM estimator performs significantly worse than other estimators. With linear decoders, the objective function Eq. (21) is convex in z_i. Thus, similar to the example in Fig. 2, GSM learns a distribution with higher entropy which leads to poorer performance. The entropy of all estimators during training is shown in Fig. 4(b). We also note that two-pass training performs better then one-pass training for all estimators. Although two-pass training requires twice the computation in the worst case, this overhead is negligible for these models due to GPU parallelization. We observe that ARM performs poorly confirming that its high variance impedes training. The PWL estimator with two-pass training performs on par with RAM, while being more computationally efficient. We used $\beta = 2$ in the above experiments, similar to Jang et al. (2016); Maddison et al. (2016). To understand the dependence on β, we plot final train NELBO in Fig. 4(c). We find that improved estimators are less sensitive to the choice of β.

In Fig. 5, we repeat the experiments for the non-linear architecture $200\text{H} \sim 784\text{V}$. The one-pass GSM estimator exhibits instability which is remedied by two-pass training. However, two-pass GSM still performs worse than IGSM and PWL due to its bias. Interestingly, one-pass training works better for IGSM/PWL. We observe this repeatedly in the nonlinear models. Unlike the linear case, the RAM estimator converges faster initially but later in training is outperformed by the higher-variance IGSM, PWL and ARM. It is likely that additional noise prevents the latent units from being turned off early in training which is known to cause poor performance in VAEs. As in the linear case we plot the dependence of the final train NELBO on β in Fig. 5(b). The GSM estimator outperforms



(a) train NELBO (b) Entropy of $q_\phi(z)$ (c) Final train NELBO

Figure 4: MNIST training on the linear architecture 200H − 784V: (a) compares training NELBO of CR estimators (all estimators use $\beta = 2$). Solid/dashed lines correspond to one/two-pass training. (b) shows the entropy of the learned posterior approximations; the GSM bias induces much higher entropy. (c) dependence of the final trained NELBO on β; higher β corresponds to lower bias.



(a) train NELBO (b) Final train NELBO (c) Final train NELBO, KL anneal 10% of training

Figure 5: MNIST training on the non-linear architecture 200H $\sim$ 784V: (a) compares training NELBO of CR estimators (all estimators use $\beta = 2$). Solid/dashed lines correspond to one/two-pass training. (b) Final training NELBO for different β, (c) Final training NELBO using KL annealing; explicit annealing erases the gains of GSM.

IGSM and PWL for $\beta \geq 4$ because its bias favors higher entropy approximating posteriors which inhibit latent units from turning off early in training. A well known resolution for inactive latent units is KL-annealing Bowman et al. (2016). Fig. 5(c) shows that KL annealing indeed improves IGSM and PWL by inhibiting over-pruning of latent units[4]. As with the linear case, the IGSM and PWL estimators are more stable with respect to the choice of β.

Additional results for other VAE models and for categorical variables on both MNIST and OMNIGLOT are presented in Appendix H.3 with similar conclusions. Interested reader is referred to Appendix F for experimental results on training encoder part of a VAE with pretrained generative model. Appendix G compares estimators for solving the discrete maximum clique optimization problem analyzed in Patish & Ullman (2018).

5 Conclusion

We have reviewed several finite-difference (FD) and continuous relaxation (CR) estimators of the gradients of the objective Eq. (1). FD estimators like RAM Tokui & Sato (2017) can give unbiased low variance estimates but often require multiple function evaluations. We proposed a less expensive version of RAM estimator that requires order of magnitude fewer function evaluations with a controllable decrease in performance. On the other hand, CR estimators, like Gumbel-Softmax (GSM), require a single pass through the objective function and can be computed efficiently giving low variance but biased gradients. We analyzed the nature of the bias introduced by CR estimators and proposed a way to reduce it. This gives rise to improved CR (ICR) estimators, like improved GSM and piece-wise linear. These ICR estimators are unbiased for a single variable and less biased for many variables. Experiments on VAE training and discrete optimization confirm the theoretical predictions and illustrate the advantage of less-biased estimators.

[4]In other experiments, not reported here, we observe that the entropy regularizing benefits of the GSM bias can be achieved with explicit entropy regularization of the objective.

REFERENCES

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pp. 10–21, 2016.

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

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoff Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*, 2017.

Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. Deep autoregressive networks. *arXiv preprint arXiv:1310.8499*, 2013.

Shixiang Gu, Sergey Levine, Ilya Sutskever, and Andriy Mnih. Muprop: Unbiased backpropagation for stochastic neural networks. *arXiv preprint arXiv:1511.05176*, 2015.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. *arXiv preprint arXiv:1611.01144*, 2016.

David S Johnson and Michael A Trick. *Cliques, coloring, and satisfiability: second DIMACS implementation challenge, October 11-13, 1993*, volume 26. American Mathematical Soc., 1996.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Guy Lorberbom, Andreea Gane, Tommi Jaakkola, and Tamir Hazan. Direct optimization through $\arg\max$ for discrete variational auto-encoder. *arXiv preprint arXiv:1806.02867*, 2018.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Andriy Mnih and Karol Gregor. Neural variational inference and learning in belief networks. *arXiv preprint arXiv:1402.0030*, 2014.

Andriy Mnih and Danilo Rezende. Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, pp. 2188–2196, 2016.

Uri Patish and Shimon Ullman. Cakewalk sampling. *arXiv preprint arXiv:1802.09030*, 2018.

Tapani Raiko, Mathias Berglund, Guillaume Alain, and Laurent Dinh. Techniques for learning binary stochastic feedforward neural networks. *arXiv preprint arXiv:1406.2989*, 2014.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pp. 1278–1286, 2014.

Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, pp. 3528–3536, 2015.

Michalis Titsias and Miguel Lázaro-Gredilla. Local expectation gradients for black box variational inference. In *Advances in neural information processing systems*, pp. 2638–2646, 2015.

Seiya Tokui and Issei Sato. Evaluating the variance of likelihood-ratio gradient estimators. In *International Conference on Machine Learning*, pp. 3414–3423, 2017.

George Tucker, Andriy Mnih, Chris J Maddison, John Lawson, and Jascha Sohl-Dickstein. Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pp. 2624–2633, 2017.

Arash Vahdat, Evgeny Andriyash, and William G Macready. DVAE#: Discrete variational autoencoders with relaxed Boltzmann priors. In *Neural Information Processing Systems (NIPS)*, 2018a.

Arash Vahdat, William G. Macready, Zhengbing Bian, Amir Khoshaman, and Evgeny Andriyash. DVAE++: Discrete variational autoencoders with overlapping transformations. In *International Conference on Machine Learning (ICML)*, 2018b.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pp. 5–32. Springer, 1992.

Mingzhang Yin and Mingyuan Zhou. Arm: Augment-reinforce-merge gradient for discrete latent variable models. *arXiv preprint arXiv:1807.11143*, 2018.

A ARGMAX

Lorberbom et al. (2018) proposed a FD estimator that we refer to as ARGMAX. For a single variable, ARGMAX relies on an identity that approximates $\partial_{\boldsymbol{\phi}}\mathcal{L} = \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi}}[f(1) - f(0)] = (\partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi}}) q_{\boldsymbol{\phi}}[1 - q_{\boldsymbol{\phi}}][f(1) - f(0)]$. With $q_{\boldsymbol{\phi}} = \sigma(l_{\boldsymbol{\phi}})$, Lorberbom et al. (2018) show that

$$\sigma(l_{\boldsymbol{\phi}})[1 - \sigma(l_{\boldsymbol{\phi}})][f(1) - f(0)] = \lim_{\epsilon \to 0} \mathbb{E}_{\rho}\left[\frac{\Theta\big(\epsilon f(1) + l_{\boldsymbol{\phi}} + \sigma^{-1}(\rho)\big) - \Theta\big(\epsilon f(0) + l_{\boldsymbol{\phi}} + \sigma^{-1}(\rho)\big)}{\epsilon}\right] \tag{22}$$

where Θ is the Heaviside step function and $\rho \sim \mathcal{U}[0, 1]$.[5] Lorberbom et al. (2018) approximate the right side of Eq. (22) by sampling ρ and evaluating at finite ϵ which introduces bias and variance. Decreasing ϵ decreases bias but increases variance.

In the categorical case the gradient of Eq. (5) can correspondingly be written as

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_a \partial_{\boldsymbol{\phi}} l^a_{\boldsymbol{\phi}} \lim_{\epsilon \to 0} \frac{1}{\epsilon} \mathbb{E}_{\rho}\left[\big[a = \arg\max_b(\epsilon f^b + l^b_{\boldsymbol{\phi}} + \gamma^b)\big] - \big[a = \arg\max_b(l^b_{\boldsymbol{\phi}} + \gamma^b)\big]\right], \tag{23}$$

where $\gamma_b = -\log(-\log(\rho^b))$ are Gumbel variables and $[pred]$ is the indicator function (1 if $pred$ is true and 0 otherwise). This derivative requires A function evaluations, similar to RAM. Lorberbom et al. (2018) extend the single variable result to multivariate distributions similar to Eqs. (3) and (4). ARGMAX has the same computational complexity as RAM, but is biased and has higher variance than RAM. Thus, it is suboptimal to Eqs. (3) and (4) and for this reason we do not perform experiments with ARGMAX.

B THE AUGMENT-REINFORCE-MERGE ESTIMATOR

FD estimators are computationally expensive and require multiple function evaluations per gradient. A notable exception is the Augment-REINFORCE-Merge (ARM) estimator introduced in Yin & Zhou (2018). ARM provides an unbiased estimate using only two function evaluations for the factorized multivariate distribution, regardless of the number of variables. For a single binary variable, the ARM derivative is given by

$$\partial_\phi \mathcal{L} = \partial_\phi l \; q_\phi(1 - q_\phi)(f(1) - f(0)) = \partial_\phi l \; \mathbb{E}_{\rho \sim \mathcal{U}[0,1]}[(f(z^{(2)}) - f(z^{(1)}))(\rho - 0.5)],$$
$$z^{(1)} = \Theta(q_\phi - \rho) \quad z^{(2)} = \Theta(\rho - 1 + q_\phi). \tag{24}$$

The expectation is approximated with a single sample ρ. This estimator has a significantly lower variance than REINFORCE since the expectation contains the difference $f(z^{(2)}) - f(z^{(1)})$ rather then the function itself. For multivariate f and factorial $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ the derivative is

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_i \partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi},i} \; \mathbb{E}_{\boldsymbol{\rho} \sim \mathcal{U}[0,1]}[(f(z^{(2)}_i, \boldsymbol{z}^{(1)}_{/i}) - f(z^{(1)}_i, \boldsymbol{z}^{(1)}_{/i}))(\rho_i - 0.5)], \tag{25}$$

where $z^{(1)}_i = \Theta(q_{\boldsymbol{\phi},i} - \rho_i)$ and $z^{(2)}_i = \Theta(\rho_i - 1 + q_{\boldsymbol{\phi},i})$. Yin & Zhou (2018) observed that one can replace $f(z^{(2)}_i, \boldsymbol{z}^{(1)}_{/i}) \to f(z^{(2)}_i, \boldsymbol{z}^{(2)}_{/i})$ without changing the expectation which allows evaluation by a single sample and two function evaluations $f(\boldsymbol{z}^{(1)}), f(\boldsymbol{z}^{(2)})$ regardless of M:

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \sum_i \partial_{\boldsymbol{\phi}} l_{\boldsymbol{\phi},i} \; \mathbb{E}_{\boldsymbol{\rho} \sim \mathcal{U}[0,1]}[(f(\boldsymbol{z}^{(2)}) - f(\boldsymbol{z}^{(1)}))(\rho_i - 0.5)]. \tag{26}$$

However, this change comes at the cost of higher variance of the expectation[6]. Thus, while ARM estimator provides a low-variance gradient estimate for a single variable, it introduces high variance

[5] We motivate this identity by noting that the non-zero contribution comes from the region $\sigma(-\epsilon f(1) - l_{\boldsymbol{\phi}}) \leq \rho \leq \sigma(-\epsilon f(0) - l_{\boldsymbol{\phi}})$ assuming $f(1) > f(0)$. For small ϵ, samples land within this region with probability $p \sim \sigma(l_{\boldsymbol{\phi}})(1 - \sigma(l_{\boldsymbol{\phi}}))\epsilon[f(1) - f(0)]$ giving rise to the identity. The variance of Eq. (22) is $\mathbb{V}\mathrm{ar}\big(\mathrm{Ber}(p)\big)/\epsilon^2 \propto \sigma(l_{\boldsymbol{\phi}})(1 - \sigma(l_{\boldsymbol{\phi}}))[f(1) - f(0)]/\epsilon$ to leading order in $1/\epsilon$.

[6] Denoting $g^{(a)}(\boldsymbol{z}_{/i}) = f(z^{(a)}_i, \boldsymbol{z}_{/i}), a = 1, 2$, we can write

$$\mathrm{VAR}_{\boldsymbol{z}_{/i}}[g^{(2)}(\boldsymbol{z}_{/i}) - g^{(1)}(\boldsymbol{z}_{/i})] = \mathrm{VAR}_{\boldsymbol{z}_{/i}}[g^{(2)}(\boldsymbol{z}_{/i})] + \mathrm{VAR}_{\boldsymbol{z}_{/i}}[g^{(1)}(\boldsymbol{z}_{/i})] - 2\mathrm{COV}_{\boldsymbol{z}_{/i}}[g^{(1)}(\boldsymbol{z}_{/i}), g^{(2)}(\boldsymbol{z}_{/i})],$$
$$\mathrm{VAR}_{\boldsymbol{z}_{/i}, \boldsymbol{z}'_{/i}}[g^{(2)}(\boldsymbol{z}_{/i}) - g^{(1)}(\boldsymbol{z}'_{/i})] = \mathrm{VAR}_{\boldsymbol{z}_{/i}}[g^{(2)}(\boldsymbol{z}_{/i})] + \mathrm{VAR}_{\boldsymbol{z}_{/i}}[g^{(1)}(\boldsymbol{z}_{/i})]. \tag{27}$$

If the functions $g^{(a)}(\boldsymbol{z}_{/i})$ are highly correlated the ARM estimator will have a much higher variance than RAM.

for multivariate functions compared to RAM. The ARM estimator has straightforward extensions to hierarchical $q(\boldsymbol{z})$ (similar to Eq. (4)) and to categorical variables Yin & Zhou (2018).

C IMPROVED CR ESTIMATOR FOR BAYESIAN NETWORKS

In this appendix we derive the improved CR estimator for a hierarchical $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \prod_i q_{\boldsymbol{\phi},i}(z_i|\boldsymbol{z}_{<i})$. For simplicity, we do this for two variables $q_{\boldsymbol{\phi}}(z_1, z_2) = q_{\boldsymbol{\phi},2}(z_2|z_1)q_{\boldsymbol{\phi},1}(z_1)$ with the gradient given in Eq. (4):

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},1}\left[\sum_{z_2} q_{\boldsymbol{\phi},2}(z_2|1)f(1,z_2) - \sum_{z_2} q_{\boldsymbol{\phi},2}(z_2|0)f(0,z_2)\right] + \sum_{z_1} q_{\boldsymbol{\phi},1}(z_1)\partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},2}(1|z_1)\left[f(z_1,1) - f(z_1,0)\right]$$

where $q_{\boldsymbol{\phi},1} = q_{\boldsymbol{\phi},1}(z_1 = 1)$. We denote the two contributions to $\partial_{\boldsymbol{\phi}}\mathcal{L}$ by $\partial_{\boldsymbol{\phi}}\mathcal{L}^{(1,2)}$ and determine them separately. Using the integral trick (10) the second contribution can be written as

$$\partial_{\boldsymbol{\phi}}\mathcal{L}^{(2)} = \sum_{z_1} q_{\boldsymbol{\phi},1}(z_1)\partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},2}(1|z_1)\int d\rho_1 \frac{\partial\zeta_2}{\partial\rho_2}\partial_{\zeta_2} f(z_1,\zeta_2) \approx \mathbb{E}_{\boldsymbol{\rho}}\left[\partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},2}(1|\zeta_1)\frac{\partial\zeta_2}{\partial\rho_2}\partial_{\zeta_2} f(\zeta_1,\zeta_2)\right] \tag{28}$$

where $\zeta_2 = g(\rho_2; q_2(1|\zeta_1))$ and $\zeta_1 = g(\rho_1; q_1)$. The first term can be handled similarly:

$$\partial_{\boldsymbol{\phi}}\mathcal{L}^{(1)} = \partial_{\boldsymbol{\phi}} q_{\boldsymbol{\phi},1}\int d\rho_1 \frac{\partial\zeta_1}{\partial\rho_1}\partial_{\zeta_1}\left(\sum_{z_2} q_{\boldsymbol{\phi},2}(z_2|\zeta_1)f(\zeta_1,z_2)\right) \approx \mathbb{E}_{\boldsymbol{\rho}}\left[\partial_{\phi} q_{\boldsymbol{\phi},1}\frac{\partial\zeta_1}{\partial\rho_1}\partial_{\zeta_1} f(\zeta_1,\zeta_2)\right] \tag{29}$$

Combining these contributions we arrive at an expression very similar to the reparameterization trick with the replacement $\partial_{q_i}\zeta_i \to \partial_{\rho_i}\zeta_i$:

$$\partial_{\boldsymbol{\phi}}\mathcal{L} \approx \partial_{\boldsymbol{\phi}}\mathbb{E}_{\boldsymbol{\rho}}\left[f(\zeta_1,\zeta_2)\right]_{\partial_{q_i}\zeta_i \to \partial_{\rho_i}\zeta_i} \tag{30}$$

D CATEGORICAL PWL ESTIMATOR

Here, we derive the PWL estimator for categorical variables. The derivative for a single categorical variable $y = (y^0, ...y^{A-1})$ with $y^a \in \{0,1\}$ and $\sum_a y^a = 1$ is given in Eq. (5) as

$$\partial_{\boldsymbol{\phi}}\mathcal{L} = \partial_{\boldsymbol{\phi}}\sum_y q_{\boldsymbol{\phi}}(y)f(y) = \sum_a \partial_{\boldsymbol{\phi}} l^a_{\boldsymbol{\phi}} \sum_b q^a q^b (f^a - f^b), \tag{31}$$

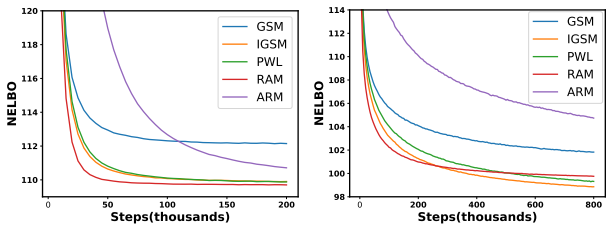
We again apply the integral trick (10) to represent the difference $f^a - f^b$. To do that we relax the variable y so that it interpolates between $(y^a, y^b) = (1,0)$ and $(y^a, y^b) = (0,1)$ as $y \to y^{a,b} = \left\{y^a = \left[0.5 + \alpha^{a,b}(\rho^{a,b} - q^b/(q^a+q^b))\right]_0^1, y^b = 1 - y^a, y^{c\neq a,b} = 0\right\}$ where $\rho^{a,b} \in \mathcal{U}[0,1]$ and $\alpha^{a,b}$ is the slope. The relaxed objective then takes the form

$$\mathcal{L} = \int_0^1 \prod_{a<b} d\rho^{a,b} \sum_{a<b} w^{a,b} f(y^{a,b}), \tag{32}$$

where $w^{a,b}$ are weights to be determined. The gradient of this relaxed objective with respect to the logit l^a is

$$\partial_{l^a}\mathcal{L} = \mathbb{E}_\rho\left[\sum_{b\neq a} w^{a,b}\partial_{\rho^{a,b}} f(y^{a,b})\frac{q^a q^b}{(q^a+q^b)^2}\right] = \sum_{b\neq a} w^{a,b}\frac{q^a q^b}{(q^a+q^b)^2}(f^a - f^b). \tag{33}$$

Comparison with Eq. (31) gives $w^{a,b} = (q^a + q^b)^2$. However, computing the sum over the edges of the simplex is prohibitively expensive, so we choose to replace it with sampling from the set of edges with probability $p^{a,b} = (q_a + q_b)/(\sum_{a<b} q^a + q^b) = (q^a + q^b)/(A-1)$. The reason for choosing this distribution is that Eq. (32) must give correct value of the objective (not just the derivative) as



(a) 200H − 784V (b) 200H ∼ 784V

Figure 6: Encoder training the with fixed pre-trained decoder on MNIST: (a) the linear architecture. (b) the non-linear architecture.

the relaxation parameter $\beta \to \infty$, which requires the probability of each state $y^a = 1$ to be equal to q^a. For relaxed edge (a, b) the probability of $y^a = 1$ is equal to $q^a/(q^a + q^b)$, and thus the total probability of $y^a = 1$ is:

$$\sum_{b \neq a} p^{a,b} \frac{q^a}{q^a + q^b} = \sum_{b \neq a} \frac{q^a + q^b}{A - 1} \frac{q^a}{q^a + q^b} = q^a. \tag{34}$$

Finally, to get the correct weights $w^{a,b}$ we must rescale each term by a factor $\gamma^{a,b} = (A-1)(q^a+q^b)$ so that $w^{a,b} = \gamma^{a,b} p^{a,b}$. In summary, the relaxed objective has the following form:

$$\mathcal{L} = \mathbb{E}_{(a,b)\sim p^{a,b}} \left[\mathbb{E}_{\rho \in \mathcal{U}[0,1]} \left[f(\tilde{y}^{a,b}) \right] \right], \tag{35}$$

where $\tilde{y}^{a,b}$ has the same value as $y^{a,b}$ but has the gradient scaled by $\gamma^{a,b} = (A - 1)(q^a + q^b)$.

E Experimental details

For the toy example of Section 4.1 we optimize $\mathcal{L}$ with respect to $q_\phi(z = 1)$ using Adam (Kingma & Ba (2014)) for 2000 iterations with learning rate $r = 0.01$ using minibatches of size 100 to reduce the variance of gradients. We initialize $q_\phi(z = 1)$ to the wrong maximum of the relaxed function by setting $q_\phi(z = 1) = \sigma(5)$.

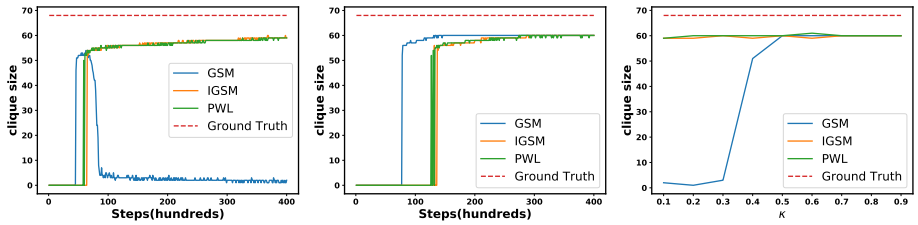
For VAE training experiments, following Maddison et al. (2016); Jang et al. (2016); Tucker et al. (2017), we consider four architectures with binary variables denoted by $200\text{H} - 784\text{V}$, $200\text{H} - 200\text{H} - 784\text{V}$, $200\text{H} \sim 784\text{V}$, and $200\text{H} \sim 200\text{H} \sim 784\text{V}$. Here $-$ denotes a linear layer, while $\sim$ denotes two layers of 200 hidden units with $\tanh$ nonlinearity and batch normalization. In our experiments we use the Adam optimizer with default parameters and a fixed learning rate 0.0003, run for $4 \cdot 10^6$ steps with minibatches of size 100. We repeat each experiment 5 times with random seed and plot the mean.

F VAE encoder training

During training of VAEs, the encoder and decoder models interact in complex ways. To eliminate these effects to more directly identify the impact of better gradients, we use a fixed pre-trained decoder, and minimize Eq. (21) with respect to encoder parameters $\boldsymbol{\phi}$ only. As shown in Fig. 6, the conclusions from joint training remain unchanged: GSM underperforms due to its bias. Interestingly however, RAM still overfits early in training for the $200\text{H} \sim 784\text{V}$ model, showing that better gradients can negatively affect optimization.

G Optimization

We apply gradient estimators for discrete optimization. Following Patish & Ullman (2018) we study the NP-hard task of finding a maximal clique in a graph. To model this problem, the binary variable $z_i = 1/0$ indicates the presence/absence of vertex i in a maximal clique. If $A_{i,j}$ is the adjacency



(a) $\kappa = 0.1$ (b) $\kappa = 0.9$ (c) Final max clique size

Figure 7: Finding maximal cliques in the $C1000.9$ graph: (a) and (b) show the course of optimization for two settings of κ and different gradient estimators. (c) plots the final maximal clique size across κ for the estimators.

matrix of the graph and $d = \sum_i z_i$ is the size of the clique, then the objective function considered in Patish & Ullman (2018) is

$$f(z) = -\frac{\sum_{ij} z_i A_{ij} z_j}{d(d-1+\kappa)}, \tag{36}$$

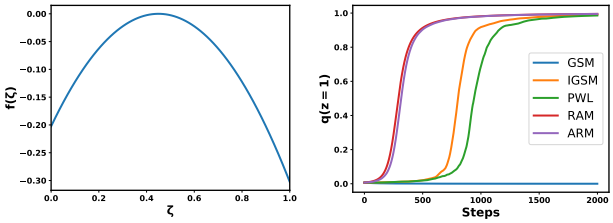
where $\kappa \in [0, 1]$ is a hyperparameter. Patish & Ullman (2018) use factorial $q_{\boldsymbol{\phi}}(z)$ to minimize $\mathcal{L}[\boldsymbol{\phi}] = \sum_{\boldsymbol{z}} q_{\boldsymbol{\phi}}(\boldsymbol{z}) f(\boldsymbol{z})$. At the end of optimization $q_{\boldsymbol{\phi}}(z)$ typically collapses to a single state corresponding to a clique. Most often this clique is a local minimum and not the maximal clique.

We illustrate the performance of CR estimators on a particular graph (1000 nodes, 450000 edges), labeled $C1000.9$ from the DIMACS data set Johnson & Trick (1996). We minimize $f(\boldsymbol{z})$ using Adam with default settings, learning rate 0.01 for 40000 iterations. We perform 1000 minimizations in parallel and choose the best clique found at each iteration. Fig. 7(a),(b) show the size of clique found by each of the estimators for two values of the hyperparameter $\kappa = 0.1, 0.9$. At $\kappa = 0.1$ the bias of GSM causes trapping in the wrong minima but ICR converges to a good local minimum. In contrast, at $\kappa = 0.9$ the GSM bias accelerates convergence to a good local minimum. Fig. 7(c) shows the dependence of the final clique size on κ; unbiased estimators provide more stable results.

H Additional Experiments

H.1 Binary Concave Toy Example

We evaluate performance on the concave function $f(\zeta) = -(\zeta - 0.45)^2$ shown in Fig. 8(a). The training setup is identical to Section 4.1 but with $q_\phi(z = 1)$ initialized to $\sigma(-5)$. Fig. 8(b) demonstrates that the GSM optimization gets trapped in the wrong minimum due to its bias towards the dominant mode $\zeta = 0$.

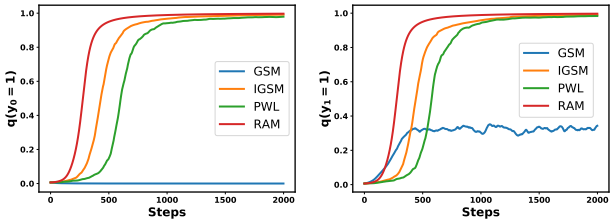


(a) Concave relaxed function (b) Probability $q(z = 1)$

Figure 8: Single binary variable concave toy example. (a) Relaxed function $f(\zeta) = -(\zeta - 0.45)^2$ (b) Probability $q_\phi(z = 1)$ during optimization

H.2 CATEGORICAL TOY EXAMPLE

We consider a categorical example with convex and concave functions of a single categorical variable y having 10 values. We take $f(y) = \pm\sum_a (g^a - y^a)^2$ such that $g^0 = 0.9, g^1 = 1.1$, and $g^{i>1} = 1$. The convex function has a minimum at $y^1 = 1$, while the concave function is minimized at $y^0 = 1$. We compare 4 estimators for minimizing this function: RAM of Eq. (5), PWL of Eq. (18), GSM of Eq. (19) and IGSM of Eq. (20). The probability of the true minimum is shown in Fig. 9. In both the convex and concave cases the GSM estimator exhibits a bias preventing it from finding the minimum. IGSM is less biased than GSM which allows it to find the true minimum. The PWL estimator is unbiased but has higher variance then IGSM which slows down its optimization.

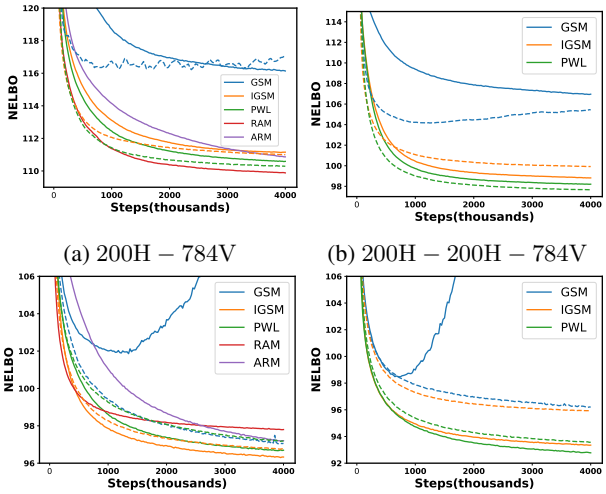


(a) Concave function (b) Convex function

Figure 9: The probability of the true minimum for a single categorical-variable toy example with (a) concave function $f(y) = -\sum_a (g^a - y^a)^2$ and (b) convex function $f(y) = \sum_a (g^a - y^a)^2$

H.3 DISCRETE VARIATIONAL AUTOENCODERS

Fig. 10 shows the results for all 4 architectures $200\text{H} - 784\text{V}$, $200\text{H} - 200\text{H} - 784\text{V}$, $200\text{H} \sim 784\text{V}$, and $200\text{H} \sim 200\text{H} \sim 784\text{V}$ at $\beta = 2$. Hierarchical models with two layers of latent units in Fig. 10(b),(d) exhibit similar trends to the factorial case considered in the main text. The GSM estimator converges to a higher NELBO in the linear case and becomes unstable in the non-linear case.



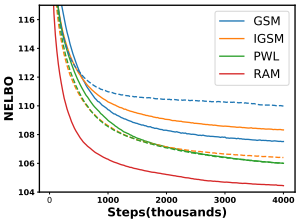
(a) $200\text{H} - 784\text{V}$ (b) $200\text{H} - 200\text{H} - 784\text{V}$

(c) $200\text{H} \sim 784\text{V}$ (d) $200\text{H} \sim 200\text{H} \sim 784\text{V}$

Figure 10: MNIST training (CR estimators use $\beta = 2$). Solid/dashed lines correspond to one/two-pass training.

We also compare the performance of GSM and IGSM estimators in the categorical case $20 \times 10\text{H} - 784\text{V}$ with latent space being 20 categorical variables with 10 classes Fig. 11. We see that IGSM

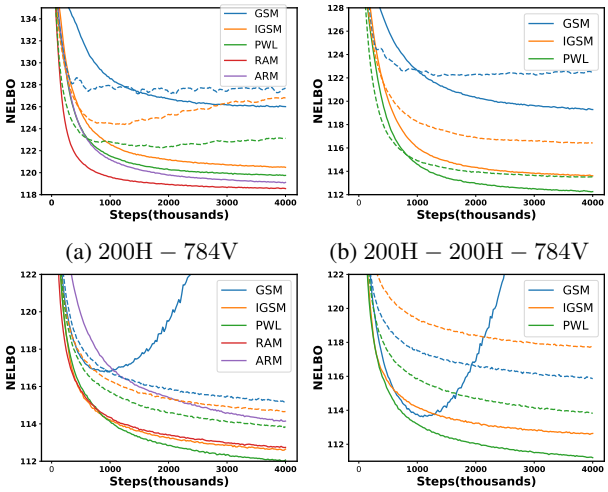
outperforms GSM estimator when using two-pass training. This indirectly confirms that IGSM is less biased. The PWL estimator performs best indicating the advantages ICR estimators.



(a)

Figure 11: MNIST training on categorical linear architecture $20 \times 10\text{H} - 784\text{V}$ (CR estimators use $\beta = 2$). Solid/dashed lines correspond to one/two-pass training.

For completeness we also show the training curves for OMNIGLOT dataset at $\beta = 2$ in Fig 12. Interestingly, one-pass training performs better in all cases. PWL estimator performs best among considered CR estimators.



(a) $200\text{H} - 784\text{V}$ (b) $200\text{H} - 200\text{H} - 784\text{V}$

(c) $200\text{H} \sim 784\text{V}$ (d) $200\text{H} \sim 200\text{H} \sim 784\text{V}$

Figure 12: OMNIGLOT training (CR estimators use $\beta = 2$). Solid/dashed lines correspond to one/two-pass training.