Cold Analysis of Rao-Blackwellized Straight-Through Gumbel-Softmax Gradient Estimator

Alexander Shekhovtsov [1]

Abstract

Many problems in machine learning require an estimate of the gradient of an expectation in discrete random variables with respect to the sampling distribution. This work is motivated by the development of the Gumbel-Softmax family of estimators, which use a temperature-controlled relaxation of discrete variables. The state-of-the art in this family, the Gumbel-Rao estimator uses an extra internal sampling to reduce the variance, which may be costly. We analyze this estimator and show that it possesses a zero temperature limit with a surprisingly simple closed form. The limit estimator, called ZGR, has favorable bias and variance properties, it is easy to implement and computationally inexpensive. It decomposes as the average of the straight through (ST) estimator and DARN estimator — two basic but not very well performing on their own estimators. We demonstrate that the simple ST–ZGR family of estimators practically dominates in the bias-variance tradeoffs the whole GR family while also outperforming SOTA unbiased estimators.

1. Introduction

Discrete variables and discrete structures are important in machine learning. For example, variational autoencoders (VAEs) with binary latent states are helpful in semantic hashing (Shen et al., 2018; Dadaneh et al., 2020; Ñanculef et al., 2020). Vector quantized VAEs (van den Oord et al., 2017) are employed as low-level representations in deep vision models (Mao et al., 2022). Another example is neural networks with discrete (binary or quantized) weights and activations. They allow for a low-latency and energy efficient inference, particularly important for edge devices. Recent results indicate that quantized networks can achieve competitive accuracy with a better efficiency in various applications (Nie et al., 2022). VAEs and quantized networks are two diverse examples that motivate our development and the experimental benchmarks. Other potential applications include conditional computation (Bengio et al., 2013a; Yang et al., 2019; Bulat et al., 2021) reinforcement learning (Yin et al., 2019), learning task-specific tree structures for agglomerative neural networks (Choi et al., 2018), neural architecture search (Chang et al., 2019) and more.

The learning problem in the presence of stochastic discrete variables is often formulated as minimization of the expected loss. Its gradient-based optimization requires gradient of the expectation in the probabilities of random variables (or parameters of networks inferring these probabilities). Unbiased gradient estimators have been developed (Williams, 1992; Grathwohl et al., 2018; Tucker et al., 2017; Gu et al., 2016). These estimators work even for non-differentiable losses, but their high variance is a major limitation. More recent advances (Yin et al., 2019; Kool et al., 2020; Dong et al., 2020; 2021; Dimitriev & Zhou, 2021a;b) reduce the variance by using several cleverly coupled samples. However, the hierarchical / deep case has not been addressed satisfactory. On one hand, due to dependent random variables, the variance of gradient estimators is typically much higher. On the other hand, extensions of coupled sampling methods to networks with L layers of discrete variables (*e.g.*, Dong et al. 2020; Yin et al. 2019) apply their base method in every layer, which requires in each layer to evaluate all the remaining layers till the loss. The computation complexity thus grows quadratically with the number of dependency layers, making these methods too costly for *e.g.* hierarchical VAEs or quantized networks and practically infeasible for autoregressive models.

A different family of methods, fitting practical needs of deep models better, exploits continuation arguments. It includes ST variants (Bengio et al., 2013b; Shekhovtsov & Yanush, 2021; Pervez et al., 2020) and Gumbel-Softmax variants. These methods assume the loss function to be differentiable and try to estimate the derivative with respect to parameters of a discrete distribution from the derivative of the loss function at a discrete sample. Such estimators can be easily incorporated into back-propagation by adjusting the forward and backward passes locally for every discrete

[1]Department of Cybernetics, Czech Technical University in Prague, Czech Republic. Correspondence to: Alexander Shekhovtsov <shekhole@fel.cvut.cz>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

variable. They are, in general, biased. The rationale though is that it may be possible to obtain a low variance estimate at a price of small bias, *e.g.* for a sufficiently smooth loss function (Shekhovtsov & Yanush, 2021).

Gumbel Softmax (Jang et al., 2017; Maddison et al., 2017) enable differentiability through discrete random variables by relaxing them to continuous ones with a distribution approximating the original discrete distribution. The tightness of the relaxation is controlled by the temperature parameter $t > 0$. The bias can be reduced by decreasing the temperature, but the variance grows as $O(1/t)$ (Shekhovtsov, 2021). Gumbel-Softmax Straight-Through (GS-ST) heuristic (Jang et al., 2017) uses the relaxed continuous samples only on the backward pass while discretizing them on the forward pass. The Gumbel-Rao (GR) estimator (Paulus et al., 2021) is a recent improvement of GS-ST, which can substantially reduce its variance by local expectations. However, each local expectation results in an intractable integration in multiple variables, and has to be approximated by sampling. The experiments (Paulus et al., 2021) suggest that this estimator performs better at lower temperatures, which requires more MC samples. Therefore the computation cost becomes a significant obstacle.

In this work, inspired by the performance of GR at low temperatures, we analyze its behavior in the cold limit, *i.e.* $t \to 0$. Note that there is no zero temperature limit estimators corresponding to GS or GS-ST because their variance explodes in the limit. It is not obvious therefore that GR has such a limit estimator. We prove that it does and denote it as ZGR. In the case of binary variables we show that ZGR has a simple analytic expression, matching the already known DARN($\frac{1}{2}$) estimator by Gregor et al. (2014) (also re-discovered as importance reweighed ST by Pervez et al. 2020). In the general categorical case we show that ZGR can be expressed in closed form as $\frac{1}{2}$(ST+DARN), for a specific variant of DARN, giving a simple and efficient new estimator. We show that ZGR is unbiased for all quadratic functions of categorical variables and experimentally show that it achieves a useful bias-variance tradeoff. We observe that both GR and ZGR outperform SOTA unbiased estimators in the two considered applications. We further demonstrate that the interpolated ST–ZGR family of estimators practically dominates in the bias-variance tradeoffs the whole GR family while being drastically simpler.

2. Background

Let x be a categorical random variable taking values in $\mathcal{K} = \{0, \dots K-1\}$ with probabilities $p(x;\eta)$ parametrized by η. Let $\phi(x) \in \mathbb{R}^d$ be its *embedding* into a vector space. Categorical variables are usually represented using the 1-hot encoding, in this case $\phi(x)$ represents one-hot embedding while x itself will be used as an index. For binary and quantized variables we will adopt the embedding $\phi(x) = x$.

Let $\mathcal{L}\colon \mathbb{R}^d \to \mathbb{R}$ be a differentiable loss function. For brevity, let us use the shorthand $\mathcal{L}(x) = \mathcal{L}(\phi(x))$. The elementary problem is to estimate the derivative (Jacobian) of the expected loss

$$J_\eta := \tfrac{\mathrm{d}}{\mathrm{d}\eta}\mathbb{E}[\mathcal{L}(x)] = \tfrac{\mathrm{d}}{\mathrm{d}\eta}\textstyle\sum_x \mathcal{L}(x)p(x;\eta). \tag{1}$$

Hereafter the subscript of the Jacobian will denote the free variable of differentiation. When extended to a network with many (dependent) categorical variables, the problem becomes much more complex. We are interested in a stochastic estimate and would like to interchange the derivative and the expectation in (1) in order to use the "derivative at a sample" as an estimate.

REINFORCE (Williams, 1992). The basic method to achieve the interchange is using the log-derivative trick:

$$\begin{aligned}\tfrac{\mathrm{d}}{\mathrm{d}\eta}\textstyle\sum_x \mathcal{L}(x)p(x;\eta) &= \textstyle\sum_x \mathcal{L}(x)\tfrac{\mathrm{d}p(x;\eta)}{\mathrm{d}\eta} \\ &= \textstyle\sum_x \mathcal{L}(x)p(x;\eta)\tfrac{\mathrm{d}\log p(x;\eta)}{\mathrm{d}\eta} = \mathbb{E}\big[\mathcal{L}(x)\tfrac{\mathrm{d}\log p(x;\eta)}{\mathrm{d}\eta}\big].\end{aligned} \tag{2}$$

The respective estimate is $J^{\mathrm{RF}}_\eta = \mathcal{L}(x)\frac{\mathrm{d}\log p(x;\eta)}{\mathrm{d}\eta}$, where $x \sim p(x;\eta)$. This estimator is clearly unbiased as $\mathbb{E}[J^{\mathrm{RF}}_\eta] = J_\eta$, but has a relatively high variance. As discussed above, existing variance reduction techniques are not computationally efficient for deep models.

Alternatively, one can try to use the reparameterization trick (Kingma & Welling, 2013; Rezende et al., 2014). We can often reparameterize $p(x;\eta)$ as a parametric mapping $x = f_\eta(z)$ with injected noises z following some fixed distribution, *i.e.* not depending on parameters. Then, we can attempt the interchange:

$$\tfrac{\mathrm{d}}{\mathrm{d}\eta}\mathbb{E}_z[\mathcal{L}(f_\eta(z))] \stackrel{?}{=} \mathbb{E}_z[\tfrac{\mathrm{d}}{\mathrm{d}\eta}\mathcal{L}(f_\eta(z))]. \tag{3}$$

However, when $f_\eta(z)$ is not continuously differentiable, which is the case when we reparameterize a discrete random variable, this interchange does not need to hold. Nevertheless, it has proven efficient in practice to consider estimators that smooth the reparameterization function f_η or approximate the derivative at a discrete sample in some other way, leading to an *approximate interchange*. Such estimators can be easily extended to deep models of stochastic variables by simply applying the elementary estimator whenever the respective Jacobian is needed in backpropagation.

ST Let $\bar\phi(\eta) = \mathbb{E}[\phi(x)] = \sum_x \phi(x)p(x;\eta)$ – the *mean embedding* in $\mathbb{R}^d$ under the current distribution of x. In particular, for one-hot embedding it is the vector of probabilities $p(x = \cdot\,;\eta)$. The Straight-Through estimator is

$$J^{\mathrm{ST}}_\eta = J_\phi \tfrac{\mathrm{d}\bar\phi(\eta)}{\mathrm{d}\eta}, \tag{4}$$

where $J_\phi = \frac{d}{d\phi}\mathcal{L}(\phi(x))$ at a random $x \sim p(x;\eta)$. Note that there exist many empirical forms of ST estimators in the literature. The present definition is the same as by Gu et al. (2016) and is consistent with Hinton (2012) and Shekhovtsov & Yanush (2021) in the binary case but is different from, *e.g.*, Bengio et al. (2013a).

DARN Gregor et al. (2014) propose the following estimator, motivated by REINFORCE with an approximate linearization of $\mathcal{L}(x)$:

$$J_\eta^{\mathrm{DARN}(\bar\phi)} = J_\phi(\phi(x) - \bar\phi)\frac{\mathrm{d}\log p(x;\eta)}{\mathrm{d}\eta}, \tag{5}$$

where $x \sim p(x;\eta)$, $J_\phi = \frac{d}{d\phi}\mathcal{L}(\phi(x))$ and $\bar\phi \in \mathbb{R}^d$ is a free choice. When x is binary, the choice $\bar\phi = \frac{1}{2}\sum_x \phi(x)$, the mean embedding under the uniform distribution, ensures that the estimator is unbiased for any quadratic function. However, for categorical variables no $\bar\phi$ with such property exist. Gu et al. (2016) experimentally tested several choices for $\bar\phi$ in the categorical case, including the mean embedding $\bar\phi = \bar\phi(\eta)$, and have found that none performed well.

GS The Gumbel-Softmax estimator (Jang et al., 2017) is a relaxation of the Gumbel-argmax reparameterization. Let $\theta_k = \log p(x{=}k;\eta)$. Let $G_k \sim \mathrm{Gumbel}(0,1)$, $k \in \mathcal{K}$, where $\mathrm{Gumbel}(0,1)$ is the Gumbel distribution with cdf $F(u) = e^{-e^{-u}}$. Then

$$x = \arg\max_k(\theta_k + G_k) \tag{6}$$

is a sample from $p(x;\eta)$. This gives a reparameterization in terms of independent injected noises G (albeit not continuously differentiable). A relaxation is obtained by using a temperatured $\mathrm{softmax}$ instead of $\arg\max$. This construction assumes one-hot embedding ϕ and creates relaxed (continuous) samples in the simplex Δ^{K-1}. Formally, introducing temperature t, it can be written as

$$\tilde\phi = \mathrm{softmax}((\theta + G)/t) =: \mathrm{softmax}_t(\theta + G); \tag{7a}$$

$$J_\theta^{\mathrm{GS}} = \frac{\mathrm{d}\mathcal{L}(\tilde\phi)}{\mathrm{d}\theta} = \frac{\mathrm{d}\mathcal{L}(\tilde\phi)}{\mathrm{d}\tilde\phi}\frac{\mathrm{d}\tilde\phi}{\mathrm{d}\eta}. \tag{7b}$$

There are two practical concerns. First, the loss function is evaluated at a relaxed sample, which may be inefficient, *e.g.*, for large expert models (Shazeer et al., 2017). Second, in a large computation graph the use of relaxed samples can offset all expectations needed for all other gradients. The latter effect can be mitigated by using a smaller temperature, causing relaxed samples $\tilde\phi$ to concentrate close to the corners of the simplex. However, the variance of the estimator grows as $O(\frac{1}{t})$ if t is decreased towards zero (Shekhovtsov, 2021).

GS-ST The Straight-Through Gumbel-Softmax estimator (Jang et al., 2017) is an empirical modification of GS. It uses discrete samples in the forward pass and swaps in the Jacobian of the continuous relaxation in the backward pass:

$$G_k \sim \mathrm{Gumbel}(0,1),\ k \in \mathcal{K}; \tag{8a}$$

$$x = \arg\max_k(\theta_k + G_k); \tag{8b}$$

$$\tilde\phi = \mathrm{softmax}_t(\theta + G); \tag{8c}$$

$$J_\theta^{\mathrm{GS\text{-}ST}} = \frac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi}\frac{\mathrm{d}\tilde\phi}{\mathrm{d}\theta}. \tag{8d}$$

Notice that the hard sample x and the relaxed sample $\tilde\phi$ are entangled through G. Although, x has the law of $p(x;\eta)$ as desired, not changing forward expectations, there still remains bias in estimating the gradient in θ. To make the bias smaller the temperature t should be decreased, however, the bias does not vanish even asymptotically while the variance still grows as $O(1/t)$ (Shekhovtsov, 2021). Values of t between 0.1 and 1 are used in practice (Jang et al., 2017).

GR Notice that the forward pass in GS-ST is fully determined by x alone and the value of G that generated that x is needed only in the backward pass. Paulus et al. (2021) proposed that the variance of GS-ST can be reduced by computing the conditional expectation in $G|x$, leading to the Gumbel-Rao estimator:

$$J_\theta^{\mathrm{GR}} = \mathbb{E}_{G|x}\left[J_\theta^{\mathrm{GS\text{-}ST}}(G)\right] = \frac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi}\mathbb{E}_{G|x}\left[\frac{\mathrm{d}\tilde\phi}{\mathrm{d}\theta}\right]. \tag{9}$$

Because the value of the loss $\mathcal{L}(x)$ and its gradient do not depend on the specific realization of $G|x$, enabling the equality above, the expectation is localized and can be computed in the backward pass. However, this expectation is in multiple variables and is not analytically tractable. Paulus et al. (2021) use Monte Carlo integration with M samples from $G|x$. In their experiments they report improvement of the mean squared error (MSE) of the estimator when the temperature was decreasing from 1 down to 0.1. The trend suggests that it would improve even further below $t = 0.1$ provided that the conditional expectation is approximated accurately enough. This also suggests that the variance does not grow unbounded with decrease of the temperature in contrast to $O(1/t)$ asymptote for GS or GS-ST.

3. Analysis

Given the experimental evidence about GR estimator, we took the challenge to study its cold asymptotic behavior. The temperatured softmax in (37c) approaches a non-differentiable $\arg\max$ indicator in this limit and we have to handle the limit of the GR estimator with care to obtain correct results. We first analyze the binary case, where derivations are substantially simpler. Proofs of all formal claims can be found in Appendix A.

3.1. Binary Case

In the case with two categories we can simplify the initial GS-ST estimator as follows. We assume $x \in \{0,1\}$ and

$\phi(x) = x$. The argmax trick can be expressed as $x = [\![\theta_1 + G_1 \geq \theta_0 + G_0]\!]$, where $[\![\cdot]\!]$ is the Iverson bracket. Without loss of generality we can assume that the distribution of x is parametrized as $p(x{=}1;\eta) = \sigma(\eta)$, the logistic sigmoid function[1]. Recalling that $\theta_k = \log p(x{=}k;\eta)$, we have $\theta_1 - \theta_0 = \eta$. Next, denoting $Z = G_1 - G_0$, we can write the argmax trick compactly as $x = [\![\eta + Z \geq 0]\!]$. Being the difference of two Gumbel(0,1) variables, Z follows the standard logistic distribution with cdf σ. The GR estimator of derivative in η simplifies as

$$x = [\![\eta + Z \geq 0]\!], \quad Z \sim \text{Logistic}(0,1); \tag{10a}$$

$$\tilde{x} = \sigma_t(\eta + Z); \tag{10b}$$

$$J_\eta^{\text{GR}} = \tfrac{\mathrm{d}\mathcal{L}(x)}{\mathrm{d}x}\mathbb{E}_{Z|x}\big[\tfrac{\mathrm{d}\tilde{x}}{\mathrm{d}\eta}\big], \tag{10c}$$

where $\sigma_t(u) = \sigma(u/t)$ is the temperatured logistic sigmoid function. Although there is no closed form, we can compute (with a careful limit – integral interchange) the series expansion around $t = 0$.

Proposition 1.

$$J_\eta^{\text{GR}} = \tfrac{\mathcal{L}'(x)}{p(x;\eta)} p_Z(\eta)\Big(\tfrac{1}{2} + (2x-1)\tilde{c}_1 t\Big) + O(t^2), \tag{11}$$

where p_Z is the logistic density: $p_Z(\eta) = \sigma(\eta)\sigma(-\eta)$ and $\tilde{c}_1 = (2p_Z(\eta) - 1)\log(2)$.

Corollary 1. In the limit $t \to 0$ the GR estimator becomes the simple binary DARN($\frac{1}{2}$) estimator.

Using the same expansion, we can study the asymptotic bias and variance of GR around $t = 0$, which is detailed in Corollary A.1. This asymptotic expansion allows to make some predictions, In particular, for a linear objective $\mathcal{L}$ the bias is $O(t^2)$ and the squared bias is $O(t^4)$. Therefore the MSE is determined by the variance alone up to $O(t^4)$. The dependence of variance on t for a linear objective is negative in the first order term. Therefore the temperature corresponding to the minimum MSE will be non-zero. We can hope nevertheless that the suboptimality of the limit estimator will be small while there is clearly a gain in simplicity and computation cost.

3.2. General Categorical Case

In the general categorical case, the analysis is more complicated (exchange of the limit and a multivariate integral over $G|x$), but gives novel and rather unexpected results.

Theorem 1 (ZGR)**.** *The Gumbel-Rao estimator for one-hot embedding ϕ in the limit of zero temperature is given by*

$$J_{\theta_i}^{\text{ZGR}} = \begin{cases} \frac{1}{2}(J_{\phi_i} - J_{\phi_x})p(x{=}i;\eta) & \text{if } i \neq x; \\ \frac{1}{2}\sum\limits_{j\neq x}(J_{\phi_x} - J_{\phi_j})p(x{=}j;\eta) & \text{if } i = x. \end{cases} \tag{12}$$

[1]Any other parametrization will result in just an extra deterministic Jacobian in the chain rule.

Proposition 2. ZGR estimator decomposes as

$$\boxed{J_\eta^{\text{ZGR}} = \tfrac{1}{2}(J_\eta^{\text{ST}} + J_\eta^{\text{DARN}(\bar{\phi}(\eta))})}, \tag{13}$$

i.e., with the choice $\bar{\phi} = \bar{\phi}(\eta)$ in DARN.

This expression of ZGR is convenient to implement and generalizes to arbitrary embedding ϕ (since the change of the embedding is just a linear transform). It can be verified that in the binary case with the embedding $\phi(x) = x$, expression (13) is exactly DARN($\frac{1}{2}$) as was claimed in Corollary 1.

To summarize, the complete recipe for ZGR is as follows:

$$\bar{\phi} = \mathbb{E}[\phi(x)] = \textstyle\sum_x \phi(x)p(x;\eta); \tag{14}$$

Sample $x \sim p(x;\eta)$;

$$J_\eta^{\text{ZGR}} = \tfrac{d\mathcal{L}(\phi(x))}{d\phi}\tfrac{1}{2}\Big(\tfrac{d\bar{\phi}}{d\eta} + (\phi(x) - \bar{\phi})\tfrac{d}{d\eta}\log p(x;\eta)\Big).$$

Despite we have derived it from GS family, it needs neither Gumbel noises nor the temperatured softmax and is efficient for discrete distributions beyond categorical.

The decomposition into ST and DARN shows a surprising connection. Neither ST nor DARN perform particularly well in categorical VAEs on their own (Gu et al., 2016; Paulus et al., 2021). However, ZGR, being effectively their average, appears superior. So what is complementary in ST and DARN? We show that ZGR has the following property, truly extending the design principle of binary DARN($\frac{1}{2}$) to the categorical case:

Theorem 2. *ZGR is unbiased for any quadratic loss function $\mathcal{L}$.*

Both ST and DARN($\bar{\phi}(\eta)$) are unbiased for linear functions and the theorem together with representation (13) implies that their biases for quadratic functions are exactly opposite and cancel off in ZGR. This unbiased property extends to multiple variables as follows.

Corollary 2. Let $x_1, \ldots, x_n$ be independent categorical variables and $\mathcal{L}(x_1, \ldots, x_n)$ be such that for all i and all configurations x the restriction $\mathcal{L}(x_i)$ is a quadratic function. Then ZGR is unbiased.

The unbiased property for quadratic functions gives us some intuition about applicability limits of ZGR. Namely, if the loss function is reasonably smooth, such that it can be approximated well by a quadratic function, we expect gradient estimates to be accurate. Compared to ST, which is unbiased for multilinear functions only, we hypothesize that ZGR can capture interactions more accurately.

3.3. Variance Analysis

From the prior work and our results, the following theoretical comparison of variances can be established:

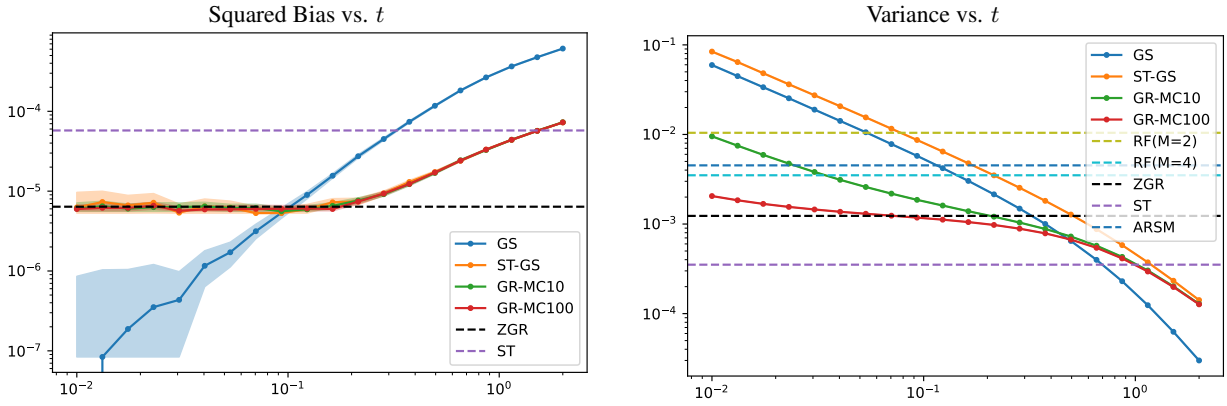

Figure 1: Gradient estimation accuracy in VAE on MNIST-B, 16 variables 16 categories. Average (per parameter) squared bias (*left*) and variance (*right*) of gradient estimators versus temperature at a fixed parameter point (model snapshot after 10 epochs of training with RF(4)). Confidence intervals are 95% empirical intervals of 100 bootstrap samples.

Corollary 3. *For any given number of MC samples M there exists a small enough $t > 0$ such that*

$$\mathbb{V}[J^{\text{ZGR}}] < \mathbb{V}[J^{\text{GR-MC}}] < \mathbb{V}[J^{\text{GS-ST}}]. \tag{15}$$

Proof. We have $\mathbb{V}[J^{\text{ZGR}}] \to \mathbb{V}[J^{\text{GR}}]$ for $t \to 0$ while $\mathbb{V}[J^{\text{GR}}] < \mathbb{V}[J^{\text{GR-MC}}]$ for any t, due to the variance of the MC integration, where the gap grows asymptotically as $1/t$. Therefore, for a small enough t, $\mathbb{V}[J^{\text{ZGR}}] - \mathbb{V}[J^{\text{GR}}]$ is smaller than the MC integration gap. This proves the first inequality in (15). The second inequality follows because GR-MC is a Rao-Blackwellization of GS-ST (an average over $G|x$, see Prop. 2 of Paulus et al. 2021). □

Experimentally, we observe exactly this predicted behavior: while the variance of GR-MC decreases with the number of MC samples, for small enough temperatures it exceeds the variance of ZGR (Fig. 1 (*right*), Fig. B.2). At the same time, the asymptotic expansion of variance of GR in the binary case (Corollary A.1) shows that the variance of GR may have a positive or negative derivative in t around 0. This implies that we cannot strengthen the result to state that there exists $t > 0$ such that for all M the comparison (15) would hold.

Unlike the variances, the biases of GS-ST, GR and GR-MC are exactly the same. In contrast to GS, this common bias has a floor due to the ST heuristic introduced in the GS-ST step. From our analysis it follows that the common floor is achieved by ZGR. This is experimentally clearly visible in Fig. 1 (*left*), although we cannot theoretically guarantee that the limit bias is the smallest amongst all t.

4. Experiments

We compare ZGR with Gumbel-Softmax (**GS**), Straight-through Gumbel-Softmax (**GS-ST**) (Jang et al., 2017), Gumbel-Rao with MC samples (**GR-MC**) (Paulus et al., 2021) and the **ST** estimator (4). We also compare to the REINFORCE with the leave-one-out baseline (Kool et al., 2019) using $M \geq 2$ inner samples, denoted RF(M), which is a strong baseline amongst unbiased estimators. In some tests we include unbiased **ARSM** (Yin et al., 2019), which requires more computation than RF(4) but performs worse. See Appendix B.2 for details of implementations.

4.1. Discrete Variational Autoencoders

We follow established benchmarks for evaluating gradient estimators in discrete VAEs. We use MNIST data with a fixed binarization (Yin et al., 2019) and Omniglot data with dynamic binarization (Burda et al., 2016; Dong et al., 2021). We use the encoder-decoder networks (Yin et al., 2019; Dong et al., 2021), up to the following difference. We embed categorical variables with 2^b states as $\{-1, 1\}^b$ vectors. This allows to vary the number of categorical variables (V) and categories (C) while keeping the total number of latent bits the same, similar to Paulus et al. (2021). Full details can be found in Appendix B.

4.1.1. Zero Temperature Limit

First, we measure the gradient estimation accuracy at a particular point of VAE training, comparing GS family of estimators at different temperatures and aiming to replicate Figure 2b of Paulus et al. (2021). Their plot shows a steady decrease of MSE with the decrease of temperature down to 0.1 and we were expecting ZGR to achieve the lowest MSE.

As an evaluation point we take a VAE model after 10 epochs of training with RF(4). The loss function is the average ELBO of a fixed random mini-batch of size 200. While measuring the variance of a gradient estimator is straightforward, measuring the bias requires a special care. Let $X \in \mathbb{R}^d$ be the estimate by the reference unbiased estima-

Table 1: Nonlinear VAE **training** negative ELBO for static binary MNIST and the down-sampled and dynamically binarized Omniglot. Each value is the mean over 3 random initializations and confidence intervals are $\pm(\max - \min)/2$ of the 3 runs. Bold results are the three best ones per configuration. The test performance can be found in Table B.1.

	MNIST-B Number of **C**ategories & Categorical **V**ariables				**Omniglot-28-D** Number of **C**ategories & Categorical **V**ariables			
Method	C2 V192	C4 V96	C16 V48	C64 V32	C2 V192	C4 V96	C16 V48	C64 V32
GS(t=0.1)	91.2±0.1	84.4±0.5	82.8±0.5	86.9±0.9	117.6±0.2	116.0±0.2	117.0±0.1	120.6±0.2
GS-ST(t=0.1)	92.0±0.4	85.4±0.2	84.1±0.5	90.0±0.5	119.4±0.1	116.7±0.2	118.5±0.1	123.2±0.3
GR-MC(t=0.1,M=10)	88.8±0.2	**82.7±0.1**	**81.0±0.1**	**82.4±0.1**	**116.6±0.2**	**114.6±0.1**	**115.7±0.1**	**117.8±0.2**
GR-MC(t=0.1,M=100)	**88.0±0.5**	**82.4±0.2**	**80.5±0.3**	**81.7±0.5**	**116.5±0.1**	**114.4±0.1**	**115.2±0.1**	**116.9±0.3**
ZGR	**88.6±0.2**	83.0±0.1	**80.6±0.4**	**81.9±0.1**	**116.6±0.4**	**114.5±0.1**	**115.4±0.1**	**117.0±0.2**
ST	105.0±0.2	105.4±0.1	106.0±0.1	106.7±0.1	130.2±0.1	130.5±0.1	131.5±0.2	132.0±0.1
RF(M=2)	92.1±0.4	86.3±0.5	88.6±0.5	96.7±0.1	120.5±0.4	118.8±0.1	122.0±0.3	127.5±0.4
RF(M=4)	**88.2±0.3**	**82.7±0.1**	82.2±0.5	87.2±0.7	117.1±0.1	115.8±0.2	117.9±0.1	120.7±0.1

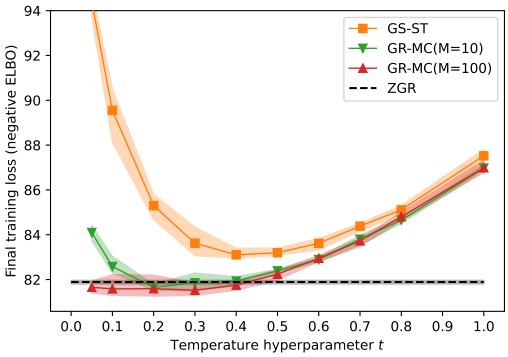

Figure 2: Training ELBO after 500 epochs of VAE training (MNIST-B, C64 V32) with different choices of temperature. The regret of ZGR w.r.t. GR-MC at the optimal temperature is insignificant. The plot shows the mean from 5 random initializations with confidence intervals $\pm(\max - \min)/2$ of the 5 runs.

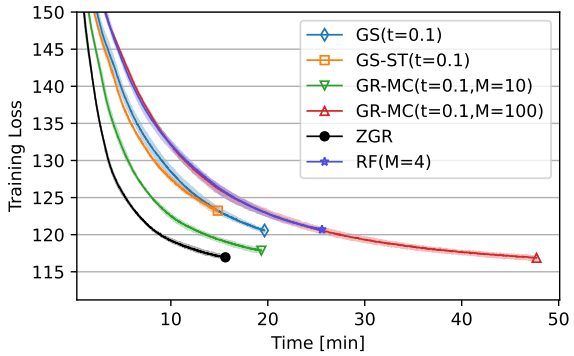

Figure 3: Training loss versus time, VAE on Omniglot-28-D (C64 V32). All methods run for 1000 epochs.

tor and $Y \in \mathbb{R}^d$ be the estimate by the tested estimator. We want to measure the average over parameters (dimensions of the gradient) squared bias, which can be written as $b^2 = \frac{1}{d}\|\mathbb{E}[X] - \mathbb{E}[Y]\|^2$. We obtain $n_1 = 10^4$ independent samples X_i from RF(4) and $n_2 = 10^4$ independent samples Y_i from the tested estimator and compute an unbiased estimate of b^2:

$$\widehat{b^2} = \frac{1}{d}\|\hat{\mu}_1 - \hat{\mu}_2\|^2 - \frac{V_1}{n_1} - \frac{V_2}{n_2}, \tag{16}$$

where $\hat{\mu}_1$ is the sample mean of X and V_1 is the average (over dimensions) sample variance of X and $\hat{\mu}_2$ and V_2 are likewise for Y.

The results are shown in Fig. 1. All of GS-ST and GR estimators share the same bias, consistently with the theory, but differ in variance. GS estimator is asymptotically unbiased but the variance grows as $O(1/t)$. We observe that the variance is by several orders larger than the squared bias. Respectively, MSE, being the sum of the variance and the squared bias is practically indistinguishable from the variance in Fig. 1 and has the opposite trend in comparison to the MSE analysis in (Paulus et al. 2021, Fig 2b.), which we thus deem not reproducible / incorrect.

Note however, that the bias-variance tradeoff in learning is more complex than in the MSE metric. The variance corresponds to uncorrelated errors, evening out with more SGD updates, while the bias is a systematic error which may potentially accumulate. Common optimization methods use exponentially weighted averaging of past gradients (momentum) as an effective way of variance reduction. Some further variance reduction in SGD is possible (*e.g.*, Johnson & Zhang 2013), albeit at an extra memory cost. Therefore, MSE metric is a poor proxy for the performance in the training pipeline and we have to inspect the bias-variance tradeoff of all estimators more carefully (see below).

ZGR still fulfills our expectations of the zero limit estimator: it has the limiting bias, which is the lowest in the GR/GS-ST family, and the limiting variance which is moderate.

4.1.2. Training Performance in VAE

Next we compare the training performance in a setup closely following prior work. In particular we use the same Adam optimizer, batch size, learning rate and training duration as (Dong et al., 2021; Dimitriev & Zhou, 2021a). Full

Table 2: Hierarchical VAE **training** negative ELBO, same notation as in Table 1. The test performance in Table B.2 is consistent with the training performance.

	MNIST-B				Omniglot-28-D			
Method	C2 V192	C4 V96	C16 V48	C64 V32	C2 V192	C4 V96	C16 V48	C64 V32
GS(t=0.1)	**117.9±0.2**	118.7±0.4	126.7±0.1	136.0±0.0	**140.2±0.4**	145.0±0.4	151.9±0.0	162.6±0.2
GS-ST(t=0.1)	120.8±0.9	121.4±0.2	129.7±0.2	139.0±0.1	146.4±0.3	147.8±0.1	154.0±0.1	164.0±0.2
GR(t=0.1,K=10)	**117.0±0.5**	**113.0±0.5**	**117.8±0.4**	**124.9±0.3**	149.8±0.8	**142.9±0.7**	**145.7±0.3**	152.4±0.0
GR(t=0.1,K=100)	118.3±0.8	**111.2±0.5**	**115.0±0.4**	**121.2±0.4**	150.6±0.6	**141.8±0.2**	**143.3±0.4**	**149.6±0.5**
ZGR	**116.5±0.7**	**111.8±0.1**	**115.6±0.2**	**122.0±0.2**	151.2±0.8	**143.5±1.5**	**143.7±0.2**	**149.3±0.3**
ST	125.4±0.2	125.9±0.0	125.2±0.1	125.2±0.1	**146.3±0.1**	146.7±0.1	148.4±0.0	**150.2±0.1**
RF(M=2)	127.1±0.7	129.9±0.3	137.3±0.7	146.1±0.5	147.1±0.6	151.1±0.4	159.1±0.2	169.3±0.2
RF(M=4)	118.6±1.2	121.8±0.5	128.3±0.8	135.0±1.6	**141.9±0.5**	146.1±0.4	154.6±0.3	162.3±0.2

details can be found in Appendix B.3. Results for two datasets and different splittings of latent bits into discrete variables (from binary to 64-way categorical) are presented in Table 1. We observe the following.

1) ST performs poorly — its bias is too high (*c.f.* Fig. 1).

2) ZGR performs no worse than GR-MC variants. In Fig. 2 we additionally verify that at no other temperature GR-MC can achieve significantly better results.

3) ZGR outperforms RF(2) and RF(4), significantly so with more categories. According to published results, recent unbiased methods (Dong et al., 2021; Dimitriev & Zhou, 2021a) appear to improve only marginally over RF with an equal number of samples, *i.e.* the difference is much smaller than between ZGR and RF(2).

4) Finally, we measure the computation time per forward-backward pass in Fig. 5 (*left*) and observe that ZGR is faster than both GR-MC and RF(2). The improvement in speed over GR-MC may appear modest, however GR-MC implementation takes advantage of palatalization and we expect it to hit memory / compute bottlenecks when scaling to larger models with more discrete variables, as already seen in quantization Fig. 5 (*right*). In the training progress versus computation time (Fig. 3) ZGR appears to be the most efficient.

4.1.3. Hierarchical VAEs

Next, we test a hierarchical VAE model with two layers of stochastic binary latent representations, following Grathwohl et al. (2018), with the encoder $q(z|x) = q(z_1|x)q(z_2|z_1)$ and decoder $p(x|z) = p(x|z_1)p(z_1|z_2)$ and the uniform prior $p(z_2)$. Each encoder layer is a linear model that produces $V \times K$ logits with $K = 2^b$, from which V categorical variables are sampled and embedded as vectors in $\{-1, 1\}^b$. The decoder layers are likewise except for the output layer, which is binary.

The results are presented in Table 2. We observed that the differences in the variance between the methods are larger in the beginning of the training and that optimizing this model suffers from getting stuck in suboptimal points (a known issue, *e.g.*, Sønderby et al. (2016)). The evidence supports similar claims as for linear VAEs, except that for Omniglot binary variables (C2), a different subset of methods reaches good results: GS and RF(4) followed by ST.

4.2. Interpolation Between Simple Estimators

We have shown that ZRG performs well in the VAE experiments, generally outperforming GS family of methods with t=0.1 and matching the performance of GR-MC even with a tuned temperature (Fig. 2). A peer reader may nevertheless be not convinced that the GS family can be replaced by single ZGR. For example, GS estimators may employ a temperature schedule and perhaps then outperform ZGR. Furthermore, in other problems it may be beneficial to trade more reduction in the variance for higher bias, which GS family readily provides via tuning the temperature.

To address the above concerns we propose the following. We craft a simple and cheap interpolated estimator using only ST and DARN and show that the set of trade-offs it provides is not worse than the whole GS family. Towards this end, for $p(x;\eta) = \mathrm{softmax}(\eta)$ we define the temperatured ST family $J_\eta^{\mathrm{ST(t)}} = J_\phi \frac{d\bar{\phi}(\eta/t)}{\mathrm{d}\eta}$ and the interpolated ST(t)–ZGR family:

$$(1-\alpha)\mathrm{ST}(t) + \alpha\mathrm{ZGR} \quad \text{for } \alpha \in [0,1]. \tag{17}$$

We notice that ST(t=2) approximately matches GS-ST(t=1).[2] Respectively, it is natural to consider ST–ZGR with $t = 2$ or $t = 1$ and varied α. The optimal design is left to future work. We visualize the bias-variance tradeoffs in Fig. 4 *left* (also in Fig. B.4 at different epochs). Empirically, we observe that there is little to no regret in using the interpolated estimator over GR-MC. If one wishes to use a temperature-scheduled GR-MC, we propose that there exist an equivalent schedule of ST–ZGR with a similar or better performance. A schedule of, *e.g.*, GS-ST involves choosing a monotone function, the starting and the terminal temper-

[2] It can be seen that the GS family smooths the argmax twice: once stochastically and once via $\mathrm{softmax}_t(\eta + G) = \mathbb{E}[\arg\max(\eta + G + G')|G]$ for $G' \sim \mathrm{Gumbel}(0, t)$.

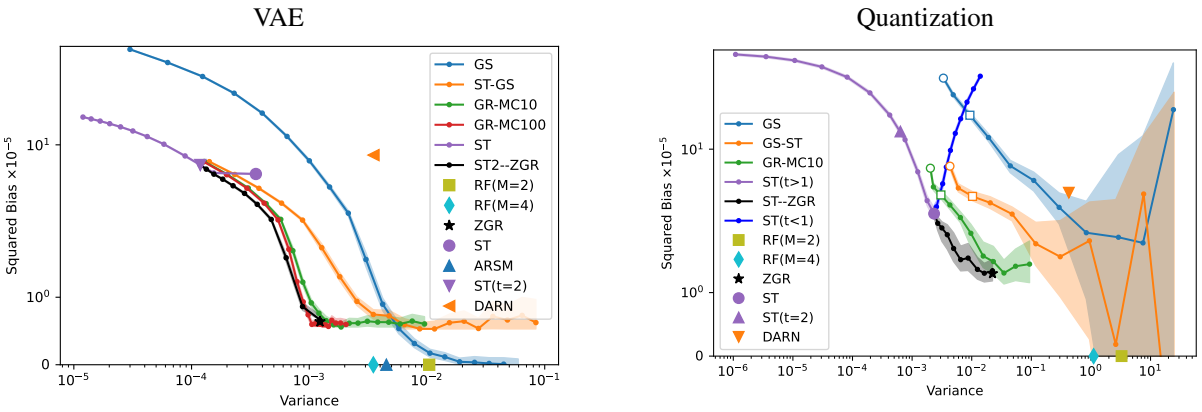

Figure 4: Bias-Variance diagrams. *Left:* non-linear VAE (C16 V16) at epoch 10. *Right:* quantized small network (8C5-MP2-16C5-MP2-128FC-10FC) with ternary weights and activations at epoch 20 (see Appendix B.6). Empty circles and squares mark temperatures $t = 2$ and $t = 1.1$, respectively. The y-axis uses symlog scaling. While different estimators with their hyperparameters can achieve different bias-variance tradeoffs, some choices a dominated by other estimators, having strictly better bias or variance or both. In these examples, the set of non-dominated choices, the Pareto frontier, is represented by families of simple estimators: temperatured ST and interpolated ST–ZGR; and by the unbiased estimator with the lowest variance.

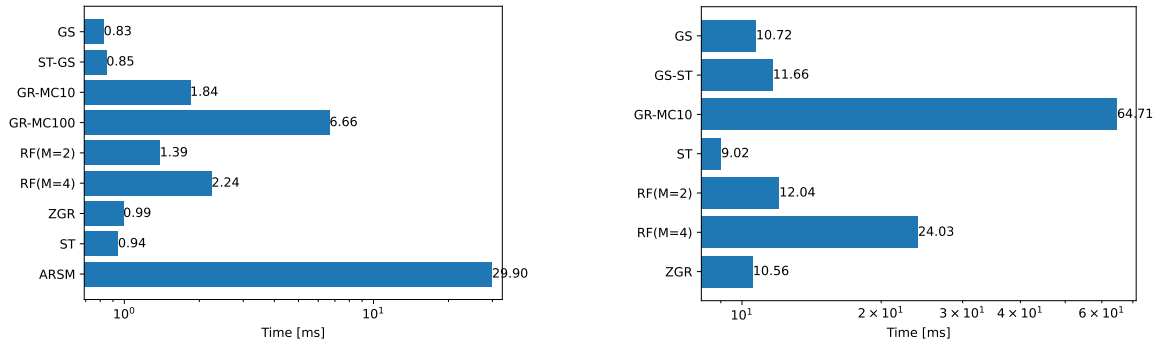

Figure 5: Time [ms] of a forward-backward pass per batch on GPU (Nvidia Tesla P100). *Left:* VAE (C16 V16), batch size 200. *Right:* Quantized network (same as in the main experiment in Table 3) with 2 bits per weigh and activation, batch size 128. The time is measured after optimizing out Python and C++ calling overheads with CUDA Graphs in Pytorch 1.13.

atures (Jang et al., 2017). These hyperparameters have to be found in practice by cross-validation. In particular, the terminal temperature needs to be tuned: in Fig. 4 it is seen that when decreasing the temperature below a certain point, unknown in advance, the bias of GR-MC approaches a floor while the variance continues to grow. We expect that with a similar effort for cross-validation, one can find a schedule for α from 0 to 1 in ST–ZGR that will perform not worse than a scheduled GR-MC.

Note also that the gain in speed can be traded for an extra variance reduction either by using several samples per SGD step or just by making more SGD steps (with a smaller learning rate and a larger momentum as appropriate).

4.3. Quantized Neural Networks

The mainstream progress in training quantized and binary neural networks, following Hubara et al. (2017), has been achieved so far using empirical variants of ST (with different clamping rules, *etc.*) applied to deterministically quantized models. A sound training approach is to consider a stochastic relaxation, replacing all discrete weights and activations by discrete random variables, leading in the binary case to stochastic binary networks (Peters & Welling, 2018; Roth et al., 2019; Shekhovtsov & Yanush, 2021).

We consider a parameter-efficient stochastic relaxation for quantization (Louizos et al., 2019). In this model the distribution of a quantized weight or activation x is defined by a single real-valued input η via: $x = \lfloor \eta + z \rceil_{\mathcal{K}}$, where $\lfloor \cdot \rceil_{\mathcal{K}}$ rounds to the nearest integer in $\mathcal{K}$ and z is an injected noise. Therefore x is a discrete integer variable with a distribution determined by η. In Appendix B.5 we give a comprehensive evaluation of bias and variance of estimators for a single such quantization unit. In a deep network, the pre-activation input η depends on the weights of the current layer as well

Table 3: FMNIST test error[%] in deterministic mode (no injected noises at test time) for different bit-width per weigh and activation (T denotes ternary). Hyperparameters are selected on the validation set. Best two results in bold. Reference test errors: ReLU 8.9% , Clamp 9.1%.

	Weights [bits] / Activations [bits]			
Method	2/2	T/T	T/1	1/1
GS-ST(t=2)	8.2±0.4	8.2±0.0	8.6±0.3	8.6±0.3
GS-ST(t=1)	**8.0±0.2**	**7.9±0.2**	8.7±0.2	**8.5±0.2**
GS-ST(t=0.5)	8.3±0.1	8.3±0.2	8.8±0.1	9.1±0.1
GR(t=0.5,K=10)	8.1±0.0	8.6±0.1	**8.6±0.2**	8.6±0.3
GR(t=0.1,K=10)	8.4±0.1	8.4±0.1	9.1±0.1	8.8±0.2
ZGR	8.2±0.3	8.3±0.1	8.7±0.3	8.8±0.1
ST	**7.9±0.2**	**8.0±0.3**	**8.1±0.2**	**8.3±0.1**
RF(M=2)	25.3±0.6	28.2±0.7	37.5±1.5	35.2±0.4
RF(M=4)	22.4±0.7	24.5±0.4	31.9±0.1	29.7±0.4

as on the preceding activations, causing a hierarchical dependence.

We train a convolutional network (32C5-MP2-64C5-MP2-512FC-10FC) on FashionMNIST. We do not quantize the input (it has 8 bit resolution in the dataset) and the first and last weight matrices are quantized to 4 bits. All inner layer weights and activations are quantized to 2 bits or below. We evaluate training with *triangular* injected noise with the density $p(z) = \max(0, 1 - |z|)$. For GS-ST variants we enable high temperatures (0.5, 1, 2) as recommended by Louizos et al. (2019). See Appendix B.5 for details of the experimental setup. The results are presented in Table 3. And in Fig. 4 (*right*) we measured the bias-variance tradeoff, using a smaller size network to make it feasible to measure the bias accurately. Details of this experiment are given in Appendix B.6. It is seen that methods differ more substantially in their variance. The best results in Table 3 are obtained by methods with low variance, prominently ST and GS-ST(t=1). Furthermore, the ranking of results in Table 3 is roughly similar to the ranking by the variance alone in Fig. 4 and Fig. B.3. In particular, variance of RF is several orders larger than that of biased estimators and its test accuracy in Table 3 is quite out of the competition. It suggests that the bias of approximate methods is relatively small in this application, not detrimental for optimization. Regarding the performance of ZGR we observe that it still outperforms GR-MC(t=0.1) (consistently with Fig. 4 and Fig. B.5). Both Table 3 and Fig. 4 support the claim that the the simple ST–ZGR family can fully replace temperatured GS, GS-ST and GR-MC families. The gain in speed is more substantial than in VAE, Fig. 5 (*right*).

5. Discussion and Conclusion

We have conducted the following analysis of the GR estimator. We theoretically showed that it has a zero temperature limit and that the limit ZGR estimator has a simple closed form. We studied its properties and connected to ST and DARN estimators. Despite being the limit estimator, ZGR retains nothing of the Gumbel-Softmax design. Indeed, the hard sampling heuristic of GS-ST disposes of relaxed samples on the forward pass. The cold limit disposes of them also on the backward pass. There remains neither Gumbels nor softmax in the design. On the other side, we showed that ZGR generalizes the key design property of DARN($\frac{1}{2}$) to the categorical case: it is unbiased for all quadratic functions. We propose that such rationale can be put forward directly for obtaining improved estimators.

We verified our theoretical findings experimentally by accurately measuring the bias and variance of all estimators, showing that ZGR is indeed the limit estimator with favorable bias and variance. In VAE we provided a refined benchmarking between SOTA biased and unbiased estimators and found out that a low bias is important in this setting and have found ZGR to be highly competitive. We also conducted a new benchmarking of SOTA biased and unbiased estimators for relaxed quantization, which is a deep (hierarchical) model of discrete random variables. We found out that in this application, the bias is less important and methods with the least variance achieve the best performance.

The practical utility of ZGR can be summarized as follows. In VAEs it delivers the same performance in training as GR-MC-100 and outperforms SOTA unbiased estimators and ST. Here the gain is in the simplicity of design, improvement in speed and potentially better scalability. We further demonstrated that the interpolated ST–ZGR family is a decent replacement of the whole GR family, whatever bias-variance tradeoff might be preferred. Another advantage of simplicity is a potentially wider scope of applicability. ZGR (or ST–ZGR) requires to compute only the mean embedding and the probability of a sample. These probabilities can be often computed faster than $O(K)$. In quantization with triangular noise they can be computed in $O(1)$. A more complex thought example is when $p(x; \eta)$ is Markov chain: while the total number of states is exponential, making GS estimators inapplicable, the probability of a sample and the mean embedding (marginals) can be computed in linear time in the chain length.

Our implementation of the described experiments is available for research purposes at `https://gitlab.com/shekhovt/zgr`.

Acknowledgements

I thank my colleagues for support and reviewers for thorough discussion. I gratefully acknowledge Czech OP VVV project "Research Center for Informatics (CZ.02.1.01/0.0/0.0/16019/0000765)" for financial support and computer cluster.

References

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

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

Bulat, A., Martinez, B., and Tzimiropoulos, G. High-capacity expert binary networks. In *ICLR*, 2021.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In *ICLR*, 2016.

Chang, J., zhang, x., Guo, Y., Meng, G., Xiang, S., and Pan, C. DATA: Differentiable architecture approximation. In *NeurIPS*, 2019.

Choi, J., Yoo, K. M., and Lee, S.-g. Learning to compose task-specific tree structures. In *AAAI*, 2018. ISBN 978-1-57735-800-8.

Dadaneh, S. Z., Boluki, S., Yin, M., Zhou, M., and Qian, X. Pairwise supervised hashing with Bernoulli variational auto-encoder and self-control gradient estimator. *ArXiv*, 2020.

Dimitriev, A. and Zhou, M. ARMS: Antithetic-REINFORCE-multi-sample gradient for binary variables. In *ICML*, pp. 2717–2727, 2021a.

Dimitriev, A. and Zhou, M. CARMS: Categorical-antithetic-REINFORCE multi-sample gradient estimator. In *NeurIPS*, 2021b.

Dong, Z., Mnih, A., and Tucker, G. DisARM: An antithetic gradient estimator for binary latent variables. In *NeurIPS*, pp. 18637–18647, 2020.

Dong, Z., Mnih, A., and Tucker, G. Coupled gradient estimators for discrete latent variables. In *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. In *ICLR*, 2018.

Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. Deep autoregressive networks. In *ICML*, 2014.

Gu, S., Levine, S., Sutskever, I., and Mnih, A. Muprop: Unbiased backpropagation for stochastic neural networks. In *ICLR*, May 2016.

Hinton, G. Lecture 15d - Semantic hashing : 3:05 - 3:35, 2012.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Quantized neural networks: Training neural networks with low precision weights and activations. *J. Mach. Learn. Res.*, (1), jan 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pp. 448–456, 2015.

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

Johnson, R. and Zhang, T. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, volume 26, 2013.

Kingma, D. P. and Welling, M. Auto-encoding variational Bayes. *CoRR*, 2013.

Kool, W., van Hoof, H., and Welling, M. Buy 4 REINFORCE samples, get a baseline for free!, 2019.

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

Louizos, C., Reisser, M., Blankevoort, T., Gavves, E., and Welling, M. Relaxed quantization for discretized neural networks. In *ICLR*, 2019.

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

Malik, H. J. and Abraham, B. Multivariate logistic distributions. *The Annals of Statistics*, (3):588–590, 1973.

Mao, C., Jiang, L., Dehghani, M., Vondrick, C., Sukthankar, R., and Essa, I. Discrete representations strengthen vision transformer robustness. In *ICLR*, 2022.

Ñanculef, R., Mena, F. A., Macaluso, A., Lodi, S., and Sartori, C. Self-supervised bernoulli autoencoders for semi-supervised hashing. *CoRR*, 2020.

Nie, G., Xiao, L., Zhu, M., Chu, D., Shen, Y., Li, P., Yang, K., Du, L., and Chen, B. Binary neural networks as a general-propose compute paradigm for on-device computer vision, 2022.

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

Pervez, A., Cohen, T., and Gavves, E. Low bias low variance gradient estimates for boolean stochastic networks. In *ICML*, pp. 7632–7640, 13–18 Jul 2020.

Peters, J. W. and Welling, M. Probabilistic binary neural networks. *arXiv preprint arXiv:1809.03368*, 2018.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pp. 1278–1286, 2014.

Roth, W., Schindler, G., Fröning, H., and Pernkopf, F. Training discrete-valued neural networks with sign activations using weight distributions. In *ECML*, 2019.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

Shekhovtsov, A. Bias-variance tradeoffs in single-sample binary gradient estimators. In *GCPR*, 2021.

Shekhovtsov, A. and Yanush, V. Reintroducing straight-through estimators as principled methods for stochastic binary networks. In *GCPR*, 2021.

Shen, D., Su, Q., Chapfuwa, P., Wang, W., Wang, G., Henao, R., and Carin, L. NASH: Toward end-to-end neural architecture for generative semantic hashing. In *Annual Meeting of the Association for Computational Linguistics*, 2018.

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., and Winther, O. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 29, pp. 3738–3746, 2016.

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

van den Oord, A., Vinyals, O., and kavukcuoglu, k. Neural discrete representation learning. In *NeurIPS*, volume 30, 2017.

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

Wolfram Research, I. Mathematica, Version 13.1, 2021. URL https://www.wolfram.com/mathematica. Champaign, IL, 2022.

Yang, B., Bender, G., Le, Q. V., and Ngiam, J. CondConv: Conditionally parameterized convolutions for efficient inference. In *NeurIPS*, 2019.

Yin, M., Yue, Y., and Zhou, M. ARSM: augment-REINFORCE-swap-merge estimator for gradient backpropagation through categorical variables. In *ICML*, pp. 7095–7104, 2019.

Appendix (Cold Analysis of Rao-Blackwellized Straight-Through Gumbel-Softmax Gradient Estimator)

Contents

A. Proofs

A.1. Binary Case

Proposition 1.

$$J_\eta^{\mathrm{GR}} = \frac{\mathcal{L}'(x)}{p(x;\eta)} p_Z(\eta)\left(\tfrac{1}{2} + (2x-1)\tilde{c}_1 t\right) + O(t^2), \tag{11}$$

where p_Z is the logistic density: $p_Z(\eta) = \sigma(\eta)\sigma(-\eta)$ and $\tilde{c}_1 = (2p_Z(\eta) - 1)\log(2)$.

Proof. The conditional density $p(z|x)$ is

$$p(z|x) = \begin{cases} p_Z(z)[\![\eta + z \geq 0]\!]/p(x{=}1), & \text{if } x = 1; \\ p_Z(z)[\![\eta + z < 0]\!]/p(x{=}0), & \text{if } x = 0. \end{cases} \tag{18}$$

Let us denote $p(x{=}1)$ as just p. The GR estimator expands as

$$J_\eta^{\mathrm{GR}} = \begin{cases} \frac{\mathcal{L}'(1)}{p} \int_{-\infty}^{\eta} \sigma_t'(\eta + z)p(z)\mathrm{d}z, & \text{if } x = 1; \\ \frac{\mathcal{L}'(0)}{1-p} \int_{\eta}^{\infty} \sigma_t'(\eta + z)p(z)\mathrm{d}z, & \text{if } x = 0. \end{cases} \tag{19}$$

Using the change of variables $v = \sigma_t(\eta + z)$, with the inverse $z = t\mathrm{logit}(v) - \eta$, we have

$$\mathrm{d}v = \sigma_t'(\eta + z)\mathrm{d}z \tag{20}$$

and can write the estimator as

$$J_\eta^{\mathrm{GR}} = \frac{\mathcal{L}'(x)}{p(x)}\left(x \int_{\frac{1}{2}}^{1} p_Z(\eta - t\mathrm{logit}v)\mathrm{d}v + (1-x)\int_0^{\frac{1}{2}} p_Z(\eta - t\mathrm{logit}v)\mathrm{d}v\right). \tag{21}$$

Note that $p_Z(\eta - t\text{logit}v)$ is bounded above by a constant $\sup_z p_Z(z) = \frac{1}{4}$. A constant is integrable on $[0, 1]$. By dominated convergence theorem we can take the limits $t \to 0$ under the integral. In particular we can use

$$\lim_{t\to 0} p_Z(\eta - t\text{logit}(v)) = p_Z(\eta) \tag{22}$$

under the integral. In order to get a more detailed view, we make the Taylor series expansion of $p_Z(\eta - t\text{logit}(v))$ and substitute it under the integral. With the help of Mathematica (Wolfram Research, 2021) we obtain:

$$J_\eta^{\text{GR}} = \frac{\mathcal{L}'(x)}{p(x)} p_Z(\eta)\Big(\tfrac{1}{2} + (2x-1)c_1 \log(2)t + c_2 \tfrac{\pi^2}{6} t^2\Big) + O(t^3), \tag{23}$$

where $c_1 = \tanh(\eta/2) = 2p - 1$ and $c_2 = \frac{1}{2}(1 - 3/(\cosh(\eta) + 1))$. □

Corollary 1. In the limit $t \to 0$ the GR estimator becomes the simple binary DARN($\frac{1}{2}$) estimator.

Proof. From the series expansion, the limit $t \to 0$ is

$$J_\eta^{\text{GR}} = \tfrac{1}{2} \frac{\mathcal{L}'(x)}{p(x)} p_Z(\eta) = \tfrac{1}{2} \frac{\mathcal{L}'(x)}{p(x)} p(1-p), \tag{24}$$

where $p = \sigma(\eta)$. It remains to show that it matches DARN as defined in (5). Note that $\frac{\mathrm{d}\log p(x=1;\eta)}{\mathrm{d}\eta} = \sigma(\eta)(1 - \sigma(\eta)) = p(1-p)$ and $\frac{\mathrm{d}\log p(x=0;\eta)}{\mathrm{d}\eta} = -p(1-p)$. By expanding the cases for $x = 1$ and $x = 0$ we verify that

$$(x - \bar{x}) \frac{\mathrm{d}\log p(x=1;\eta)}{\mathrm{d}\eta} = \tfrac{1}{2} p(1-p), \tag{25}$$

where $\bar{x} = \frac{1}{2}$. □

Corollary A.1. The mean and variance of the GR estimator (10) in the asymptote $t \to 0$ are:

$$\mathbb{E}[J_\eta^{\text{GR}}] = p(1-p)\Big(\tfrac{1}{2}(\mathcal{L}'(1) + \mathcal{L}'(0)) + (\mathcal{L}'(1) - \mathcal{L}'(0))\tilde{c}_1 t\Big) + O(t^2), \tag{26a}$$

$$\mathbb{V}[J_\eta^{\text{GR}}] = (p(1-p))^3\Big(\tfrac{1}{4}(a-b)^2 + \tfrac{1}{2}(a^2 - b^2)\tilde{c}_1 t\Big) + O(t^2), \tag{26b}$$

where $p = \sigma(\eta)$, $a = \frac{\mathcal{L}'(1)}{p}$, $b = \frac{\mathcal{L}'(0)}{1-p}$ and $\tilde{c}_1 = (2p-1)\log(2)$.

Proof. The mean of the estimator is computed from the series expansion up to the first order as

$$\begin{aligned}
&p\tfrac{\mathcal{L}'(1)}{p} p_Z(\eta)\Big(\tfrac{1}{2} + \tilde{c}_1 t\Big) + O(t^2) \\
&+(1-p)\tfrac{\mathcal{L}'(0)}{1-p} p_Z(\eta)\Big(\tfrac{1}{2} + -1\tilde{c}_1 t\Big) + O(t^2) \\
&= p_Z(\eta)\Big(\tfrac{1}{2}(\mathcal{L}'(1) + \mathcal{L}'(1)) + \tilde{c}_1(\mathcal{L}'(1) - \mathcal{L}'(0))t\Big) + O(t^2).
\end{aligned} \tag{27}$$

Since the GR estimator $J_\eta^{\text{GR}}(x)$ is a Bernoulli variable with values $J_\eta^{\text{GR}}(0)$ and $J_\eta^{\text{GR}}(1)$ with probabilities p and $1 - p$, respectively, we can compute is variance simply as

$$\big(J_\eta^{\text{GR}}(1) - J_\eta^{\text{GR}}(0)\big)^2 p(1-p). \tag{28}$$

Using that $p_Z(\eta) = p(1-p)$, the asymptotic expansion of variance up to first order in t is

$$(p(1-p))^3\Big(\big(a(\tfrac{1}{2} + \tilde{c}_1 t) - b(\tfrac{1}{2} - \tilde{c}_1 t)\big)\Big)^2 + O(t^2) \tag{29a}$$

$$= (p(1-p))^3\Big(\tfrac{1}{2}(a-b) + (a+b)\tilde{c}_1 t\Big)^2 + O(t^2), \tag{29b}$$

s where $a = \frac{f'(1)}{p}$, $b = \frac{f'(0)}{1-p}$, $\tilde{c}_1 = \log(2)c_1$. The first order term is

$$\tfrac{1}{2}(p(1-p))^3(a^2 - b^2)\tilde{c}_1 t. \tag{30}$$

It could be positive or negative depending on the values of the derivatives and of p. Let us expand a,b and $c_1 = \tanh(\eta/2) = 2p - 1$. We obtain, up to positive constants,

$$p(1-p)(f'(1)^2(1-p)^2 - f'(0)^2p^2)(2p-1)t. \tag{31}$$

We see that for the corner points, where p approaches either 0 or 1, this linear term is negative. In particular for a linear objective we have $f'(1) = f'(0)$ and the linear term becomes

$$-p(1-p)f'(1)^2(2p-1)^2t, \tag{32}$$

which is non-positive for any p and is zero for $p = \frac{1}{2}$. □

A.2. General Categorical Case

This case is significantly more difficult, as we are dealing with multivariate integration in K Gumbel variables. We will make use of the following statistical relationship.

Lemma A.1. *Let $G_1, \dots G_K$ be independent standard Gumbel random variables. Then Z with components $Z_i = G_i - G_K$ for $i = 1 \dots K-1$ has the multivariate logistic distribution (Malik & Abraham, 1973) with cdf*

$$F_Z(z) = \tfrac{1}{1+\sum_{i=1}^{K-1} e^{-z_i}}. \tag{33}$$

Proof. The cdf and density of Gumbel$(0,1)$ distribution are given respectively by

$$F_G(x) = e^{-e^{-x}}; \quad p_G(x) = e^{-(x+e^{-x})}. \tag{34}$$

The conditional distribution of Z_i given G_K is $F_{Z_i|G_K}(z_i|y) = e^{-e^{-(z_i+y)}}$. The conditional joint distribution of Z given G_K is respectively

$$F_{Z|G_K}(z|y) = \prod_{i=1}^{K-1} e^{-e^{-(z_i+y)}} = \exp(\sum_{i=1}^{K-1} -e^{-(z_i+y)}) = \exp(-e^{-y}\sum_{i=1}^{K-1} e^{-z_i}) = e^{-e^{-y}S}, \tag{35}$$

where $S = \sum_{i=1}^{K-1} e^{-z_i}$. The cdf of Z is obtained by computing the expectation of $F_{Z|G_K}$ in G_K:

$$\begin{aligned} F_Z(z) &= \int_{-\infty}^{\infty} e^{-e^{-y}S} e^{-(y+e^{-y})} \mathrm{d}y = \int_{-\infty}^{\infty} e^{-y-e^{-y}(1+S)} \mathrm{d}y \\ &= \tfrac{1}{1+S} \int_{-\infty}^{\infty} (1+S) e^{-y-(1+S)e^{-y}} \mathrm{d}y = \tfrac{1}{1+S} \int_{-\infty}^{\infty} e^{-v-e^{-v}} \mathrm{d}v = \tfrac{1}{1+S}, \end{aligned} \tag{36}$$

where $v = y - \log(S+1)$ and the last equality is by recognizing the Gumbel density under the integral. □

Theorem 1 (ZGR)**.** *The Gumbel-Rao estimator for one-hot embedding ϕ in the limit of zero temperature is given by*

$$J_{\theta_i}^{\text{ZGR}} = \begin{cases} \frac{1}{2}(J_{\phi_i} - J_{\phi_x})p(x{=}i;\eta) & \text{if } i \neq x; \\ \frac{1}{2}\sum_{j \neq x}(J_{\phi_x} - J_{\phi_j})p(x{=}j;\eta) & \text{if } i = x. \end{cases} \tag{12}$$

Proof. Let us restate the GR estimator:

$$G_k \sim \text{Gumbel}(0,1),\ k \in \mathcal{K}; \tag{37a}$$

$$X = \arg\max_k(\theta_k + G_k); \tag{37b}$$

$$\tilde{\phi} = \text{softmax}_t(\theta + G); \tag{37c}$$

$$J_\theta^{\text{GR}} = \tfrac{\mathrm{d}\mathcal{L}(\phi(X))}{\mathrm{d}\phi} \mathbb{E}_{G|X}\left[\tfrac{\mathrm{d}\tilde{\phi}}{\mathrm{d}\theta}\right]. \tag{37d}$$

The probability space here is determined by G. Let $\mathbb{P}_G$ be the respective probability measure. X is a function of G and is distributed with the pmf $p(x)$ by the sampling procedure and therefore $\mathbb{P}_G(X{=}x) = p(x)$. We can thus rewrite the conditional expectation in (37d) as

$$\int [\![X(u) = x]\!] \tfrac{\mathrm{d}\tilde{\phi}(u)}{\mathrm{d}\theta} \mathrm{d}F_G(u) / \mathbb{P}_G(X{=}x) = \tfrac{1}{p(x)} \int_{\arg\max(\theta+u)=x} \tfrac{\mathrm{d}\tilde{\phi}(u)}{\mathrm{d}\theta} \mathrm{d}F_G(u), \tag{38}$$

where F_G is the joint cdf of G. The condition $\arg\max_k(\theta_k + G_k) = x$ can be expressed as

$$\theta_j + G_j - (\theta_x + G_x) \leq 0, \quad \forall j \neq x. \tag{39}$$

Let us define $\beta_j = \theta_j - \theta_x$ and $Z_j = G_j - G_x$ for $j = 1 \dots K$. Note that $\beta_x = Z_x = 0$ by this definition. Then the constraint can be written as

$$Z \leq -\beta. \tag{40}$$

The integrand $\tilde{\phi}$ expresses in variables β, Z as

$$\tilde{\phi} = \mathrm{softmax}_t(\theta + G) = \mathrm{softmax}_t(\beta + Z). \tag{41}$$

Let us denote $Z_{\neg x} = (Z_j | j \neq x)$. The joint distribution of $Z_{\neg x}$ is the $(K-1)$-variate multivariate logistic distribution (Malik & Abraham, 1973), as detailed in Lemma A.1, with cdf:

$$F_{Z_{\neg x}}(z_{\neg x}) = \tfrac{1}{1+\sum_{i\neq x} e^{-z_i}}. \tag{42}$$

To simplify notation, we let Z_x have the discrete law with mass 1 at a single point $z_x = 0 = -\beta_x$ and extend $F_{Z_{\neg x}}$ to the full joint F_Z accordingly. We then can rewrite the integral as

$$\Big(\int\limits_{z \leq -\beta} \tfrac{\partial}{\partial \beta} \mathrm{softmax}_t(\beta + z) \mathrm{d}F_Z(z) \Big) \tfrac{\partial \beta}{\partial \theta}. \tag{43}$$

The Jacobian $\frac{\partial}{\partial \beta} \mathrm{softmax}_t(\beta + z)$ is a $K \times K$ matrix with indices (k, j) where the column $j = x$ is zero by definition. Let us consider one component of the above integral for $j \neq x$:

$$I_{k,j} = \int\limits_{z \leq -\beta} \tfrac{\partial}{\partial \beta_j} \mathrm{softmax}_t(\beta + z)_k \mathrm{d}F_Z(z). \tag{44}$$

We want to evaluate its limit for $t \to 0$. We cannot push the limit under the integral in this form, we need to transform it first. To shorten the notation, let us denote $a_i = e^{(z_i+\beta_i)/t}$. We change the variable z_j by the mapping

$$T \colon z_j \mapsto v_j = (2 + S_j) \tfrac{a_j}{1+a_j+S_j}. \tag{45}$$

where $S_j = \sum_{i \neq x,j} a_j$. This mapping is monotone increasing and one-to one from $(-\infty, -\beta)$ to $(0, 1)$, therefore the constraint $z_j \leq -\beta_j$ will trivialize.

Let $A_k = \mathrm{softmax}_t(\beta + z)_k$. We can rewrite the integrand $\frac{dA_k}{d\beta_j} \mathrm{d}z_j$ as follows:

$$\begin{aligned} &\tfrac{\mathrm{d}A_k}{\mathrm{d}\beta_j} \mathrm{d}z_j = \tfrac{\mathrm{d}A_k}{\mathrm{d}z_j} \mathrm{d}z_j && (A_k \text{ depends on } \beta_j \text{ in the same way as on } z_j) \\ =&\tfrac{\mathrm{d}A_k}{\mathrm{d}z_j} \tfrac{\mathrm{d}z_j}{\mathrm{d}v} \mathrm{d}v = \tfrac{\mathrm{d}A_k}{\mathrm{d}v_j} \mathrm{d}v. && (46) \end{aligned}$$

We will thus need to evaluate $C_j := \frac{\mathrm{d}A_k}{\mathrm{d}v_j} = \frac{\mathrm{d}A_k}{\mathrm{d}a_j} \frac{\mathrm{d}a_j}{\mathrm{d}v_j} = \frac{\mathrm{d}A_k}{\mathrm{d}a_j} (\frac{\mathrm{d}v_j}{\mathrm{d}a_j})^{-1}$. For $j = k$ we simply have

$$A_k = \tfrac{a_j}{1+a_j+S_j} = \tfrac{1}{2+S_j} v_j; \;\; C_j = \tfrac{1}{2+S_j}. \tag{47}$$

For $j \neq k$ we have

$$\tfrac{\mathrm{d}A_k}{da_j} = \tfrac{\mathrm{d}}{da_j} \tfrac{a_k}{1+a_j+S_j} = \tfrac{-a_k}{(1+a_j+S_j)^2} \tag{48a}$$

$$\tfrac{\mathrm{d}v_j}{da_j} = (2 + S_j) \Big(\tfrac{1}{1+a_j+S_j} - \tfrac{a_j}{(1+a_j+S_j)^2} \Big) = \tfrac{(2+S_j)(1+S_j)}{(1+a_j+S_j)^2} \tag{48b}$$

$$C_j = \tfrac{-a_k}{(2+S_j)(1+S_j)}. \tag{48c}$$

The integral $I_{k,j}$ with the change of variable $z_j \mapsto v_j$ expresses as

$$\int\limits_{z_i \leq -\beta_i \;\; \forall i \neq j} \int_0^1 C_j f_t(v_j | z_{\neg j}) \mathrm{d}v_j \mathrm{d}F_{Z_{\neg j}}(z_{\neg j}), \tag{49}$$

where $z_{\neg j} = (z_i | i \neq j)$ and

$$f_t(v_j|z_{\neg j}) = p_{Z_j|Z_{\neg j}}(T^{-1}(v_j)|z_{\neg j}). \tag{50}$$

The dependance of f_t on t is through T, while C_j depends on t and $z_{\neg j}$. Note that f is a squashed density and is itself not a density.

Next we show dominated convergence of $h_t(v_j, z_{\neg j}) = C_j f_t(v_j|z_{\neg j})$ in $t \to 0$. If $h_t(v_j, z_{\neg j})$ converges point-wise and bounded above by an integrable function, then the limit $t \to 0$ can be taken under the integral.

We show a constant bound on $h_t(v_j, z_{\neg j})$ as follows. Note that $|C_j| \leq 1$. We then have

$$|h_t(v_j, z_{\neg j})| \leq \sup_{v \in (0,1), z_{\neg j}} f(v_j|z_{\neg j}) = \sup_z p_{Z_j|Z_{\neg j}}(z_j|z_{\neg j}), \tag{51}$$

which is the supremum of the conditional density of the standard multivariate distribution and is equal to some constant c independent of t. The integral of a constant function c over $(0,1) \times \mathbb{R}^{K-1}$ with respect to the measure $\mathrm{d}v_j \mathrm{d}F_{Z_{\neg j}}(z_{\neg j})$ is c.

The point-wise limit is as follows. For z satisfying the constraints $z_i + \beta_i \leq 0$ strictly for $i \neq j, x$, we have

$$\lim_{t\to 0} a_i = \lim_{t \to 0} e^{(\beta_i + z_i)/t} = 0 \text{ and } \lim_{t \to 0} S_j = 0. \tag{52}$$

Therefore we have

$$\lim_{t \to 0} C_j = \begin{cases} \frac{1}{2} & \text{if } j = k; \\ 0 & \text{if } j \neq k \wedge k \neq x; \\ -\frac{1}{2} & \text{if } j \neq k \wedge k = x. \end{cases} \tag{53}$$

The inverse of mapping T is given by the relations

$$a_j = \tfrac{v_j(1+S_j)}{2+S_j - v_j}; \quad z_j = -\beta_j + t \log(a_j). \tag{54}$$

It is seen that the limit of $\log(a_j)$ is finite and therefore

$$\lim_{t \to 0} T^{-1}(v_j) = -\beta_j. \tag{55}$$

and

$$\begin{aligned} \lim_{t\to 0} f_t(v_j|z_{\neg j}) = \lim_{t \to 0} p_{Z_j|Z_{\neg j}}(T^{-1}(v_j)|z_{\neg j}) &= p_{Z_j|Z_{\neg j}}(\lim_{t\to 0} T^{-1}(v_j)|z_{\neg j}) \\ &= p_{Z_j|Z_{\neg j}}(-\beta_j|z_{\neg j}). \end{aligned} \tag{56}$$

By dominated convergence theorem, we can now claim

$$\lim_{t\to 0} I_{k,j} = 0 \text{ if } j \neq k \text{ and } k \neq x. \tag{57}$$

And elsewise, if $j = k$ or $k = x$,

$$\lim_{t\to 0} I_{k,j} = \int_{z_i \leq -\beta_i \ \forall i \neq j} \pm\tfrac{1}{2} p_{Z_j|Z_{\neg j}}(-\beta_j|z_{\neg j}) \mathrm{d}F_{Z_{\neg j}}(z_{\neg j}) = \pm\tfrac{1}{2} \tfrac{\partial}{\partial z_j} F_Z(z)\Big|_{z=-\beta} \tag{58}$$

$$= \mp\tfrac{1}{2} \tfrac{\partial}{\partial \beta_j} \tfrac{1}{1+\sum_{i\neq x} e^{\beta_i}} = \mp\tfrac{1}{2} \tfrac{\partial}{\partial \beta_j} \tfrac{e^{\beta_x}}{\sum_i e^{\beta_i}} = \mp\tfrac{1}{2} \tfrac{\partial}{\partial \beta_j} \operatorname{softmax}(\beta)_x \tag{59}$$

$$= \pm\tfrac{1}{2} p(x) p(j), \tag{60}$$

where the upper sign corresponds to the case $j = k$ and the lower to $k = x$.

Let us denote $\hat{I} = \lim_{t\to 0} I$. Multiplying it with the incoming derivative J_ϕ on the left, we obtain:

$$(J_\phi \hat{I})_j = \tfrac{1}{2}(J_{\phi_j} - J_{\phi_x}) p(x) p(j). \tag{61}$$

And finally, multiplying (61) with the Jacobian $\frac{\partial \beta}{\partial \theta}$ on the right per (43) and with the factor $\frac{1}{p(x)}$ per (38), we obtain

$$J^{\mathrm{ZGR}}_{\theta_i} = \begin{cases} \frac{1}{2}(J_{\phi_i} - J_{\phi_x}) p(i) & \text{if } i \neq x; \\ -\frac{1}{2} \sum_{j \neq x} (J_{\phi_j} - J_{\phi_x}) p(j) & \text{if } i = x. \end{cases} \tag{62}$$

□

Proposition 2. ZGR estimator decomposes as

$$\boxed{J_\eta^{\text{ZGR}} = \tfrac{1}{2}(J_\eta^{\text{ST}} + J_\eta^{\text{DARN}(\bar\phi(\eta))})}, \tag{13}$$

i.e., with the choice $\bar\phi = \bar\phi(\eta)$ in DARN.

Proof. Let p denote the vector of probabilities $(p(x{=}k;\eta)|k = 0,\dots,K-1)$. Recall that we have derived ZGR under the assumption of on-hot embedding ϕ, inherited from GS. In this case $J_\phi\phi(i) = J_{\phi_i}$ and $\bar\phi_k = \sum_i \phi(i)_k p_i = p_k$.

Note that ZGR (62) defines the gradient in the parametrization θ used in Gumbel Rao and initially in Gumbel-Softmax, while ST and DARN estimators are given by us with respect to η. We need to bring these two to a common basis. We chose to reconstruct J_p^{ZGR} because both J_p^{ST} and J_p^{DARN} are particularly simple:

$$J_{p_i}^{\text{ST}} = J_\phi\phi(i) = J_{\phi_i}, \tag{63}$$

$$J_{p_i}^{\text{DARN}} = J_\phi(\phi_i - \bar\phi)[\![x{=}i]\!]/p(x) = (J_{\phi_i} - J_\phi p)[\![x{=}i]\!]/p(x). \tag{64}$$

Note, because p lies in the simplex, gradients in p are defined up to an additive constant to all coordinates. In other words any such additive constant is irrelevant and will not affect the gradient in η.

In order to reconstruct J_p^{ZGR} we represent $J_\theta^{\text{ZGR}} = J_p^{\text{ZGR}} P$, where P is the Jacobian of softmax, given by

$$P = \operatorname{diag}(p) - pp^\mathsf{T} = \operatorname{diag}(p)(I - \mathbf{1}p^\mathsf{T}). \tag{65}$$

We first note that J_θ^{ZGR} satisfies $\sum_i J_{\theta_i}^{\text{ZGR}} = 0$ (as any gradient should, but not necessarily a stochastic estimator) and therefore

$$J_\theta^{\text{ZGR}} = J_\theta^{\text{ZGR}}(I - \mathbf{1}p^\mathsf{T}) = J_\theta^{\text{ZGR}} \operatorname{diag}(p)^{-1} P. \tag{66}$$

We obtained:

$$J_{p_i}^{\text{ZGR}} = \begin{cases} \frac{1}{2}(J_{\phi_i} - J_{\phi_x}) & \text{if } i \neq x; \\ -\frac{1}{2}\sum_{j\neq x}(J_{\phi_j} - J_{\phi_x})p(j)/p(x) & \text{if } i = x, \end{cases} \tag{67}$$

up to a constant, *i.e.* adding the same number c to all components. We further add the constant $\frac{1}{2}J_{\phi_x}$ and obtain

$$J_{p_i}^{\text{ZGR}} = \begin{cases} \frac{1}{2}J_{\phi_i} & \text{if } i \neq x; \\ \frac{1}{2}J_{\phi_x} - \frac{1}{2}\sum_{j\neq x}(J_{\phi_j} - J_{\phi_x})p(j)/p(x) & \text{if } i = x, \end{cases} \tag{68}$$

Subtracting $\frac{1}{2}J_p^{\text{ST}}$, the reminder is $\frac{1}{2}J_p^{\text{RE}}$ with

$$J_{p_i}^{\text{RE}} = [\![i{=}x]\!]\tfrac{1}{p(x)}\textstyle\sum_{j\neq x}(J_{\phi_x} - J_{\phi_j})p(j). \tag{69}$$

Simplifying

$$\textstyle\sum_{j\neq x}(J_{\phi_x} - J_{\phi_j})p(j) = \sum_j (J_{\phi_x} - J_{\phi_j})p(j) = J_{\phi_x} - \sum_j J_{\phi_j}p(j) \tag{70}$$

we obtain

$$J_{p_i}^{\text{RE}} = [\![i{=}x]\!]\tfrac{1}{p(x)}\Big(J_{\phi_x} - \textstyle\sum_j J_{\phi_j}p(j)\Big). \tag{71}$$

and we see that $J_p^{\text{RE}} = J_p^{\text{DARN}}$ with $\bar\phi = p = \bar\phi(\eta)$. □

Theorem 2. *ZGR is unbiased for any quadratic loss function $\mathcal{L}$.*

Proof. Since ZGR estimator is linear in $\mathcal{L}$ (estimate for a linear combination of two loss functions is the linear combination of estimates), it is sufficient to prove the claim for one-hot embedding ϕ and some elementary functions forming a basis for all quadratic functions. With one-hot embedding we have $\bar\phi(\eta)_i = p(x{=}i;\eta) = p_i$.

Let us start with a linear monomial $\mathcal{L}(x) = \phi(x)_i$. The expected loss is $\mathbb{E}[\mathcal{L}(x)] = p(x{=}i;\eta)$. The true gradient is

$$J_\eta = \tfrac{\mathrm{d}}{\mathrm{d}\eta} p(x{=}i;\eta). \tag{72}$$

Substituting $J_{\phi_k} = [\![k{=}i]\!]$ in ST we have

$$J^{\mathrm{ST}}_\eta = \tfrac{\mathrm{d}\mathcal{L}(\phi(x))}{\mathrm{d}\phi} \tfrac{\mathrm{d}\bar{\phi}(\eta)}{\mathrm{d}\eta} = \tfrac{\mathrm{d}\bar{\phi}(\eta)_i}{\mathrm{d}\eta} = J_\eta. \tag{73}$$

This may come as a surprise for someone, but ST for a single categorical variable is exact (zero bias and zero variance). The expectation of J^{DARN} simplifies as follows for any $\bar{\phi}$ and a linear loss function, ensuring that J_ϕ is constant in x:

$$\begin{aligned}\mathbb{E}[J^{\mathrm{DARN}}_\eta] &= \textstyle\sum_x p(x) J_\phi (\phi(x) - \bar{\phi}) \frac{1}{p(x)} \frac{\mathrm{d}p(x;\eta)}{\mathrm{d}\eta} = J_\phi \sum_x (\phi(x) - \bar{\phi}) \frac{\mathrm{d}p(x;\eta)}{\mathrm{d}\eta} \\ &= J_\phi \textstyle\sum_x \phi(x) \frac{\mathrm{d}p(x;\eta)}{\mathrm{d}\eta} - J_\phi \bar{\phi} \frac{\mathrm{d}}{\mathrm{d}\eta} \sum_x p(x;\eta) = J_\phi \frac{\mathrm{d}\bar{\phi}(\eta)}{\mathrm{d}\eta}.\end{aligned} \tag{74}$$

Substituting $J_{\phi_k} = [\![k{=}i]\!]$ and $\bar{\phi}(\eta) = p$ we obtain

$$\mathbb{E}[J^{\mathrm{DARN}}_\eta] = \tfrac{\mathrm{d}p(x{=}i;\eta)}{\mathrm{d}\eta} = J_\eta, \tag{75}$$

reconfirming that DARN is unbiased for linear function of categorical variables as expected. It follows that $\frac{1}{2}(J^{\mathrm{ST}}_\eta + J^{\mathrm{DARN}}_\eta)$ is also unbiased.

Let us now consider the elementary quadratic function $\mathcal{L}(\phi(x)) = \phi(x)_i^2 - \phi(x)_i$. For all discrete assignments it is zero, therefore the true gradient of its expected value is zero. We have

$$J_{\phi_k}(x) = \begin{cases} 2\phi(x)_i - 1 & k = i \\ 0 & k \neq i. \end{cases} \tag{76}$$

Therefore $J^{\mathrm{ST}}_{p_k} = 0$ for $k \neq i$ and

$$\mathbb{E}[J^{\mathrm{ST}}_{p_i}] = \mathbb{E}[2\phi(x)_i - 1] = 2p_i - 1. \tag{77}$$

For $J^{\mathrm{DARN}}_{p_i}$ we have

$$J^{\mathrm{DARN}}_{p_i} = (J_{\phi_i}(x) - J_\phi(x)p) \tfrac{1}{p(x)} [\![x{=}i]\!] \tag{78}$$

$$= (1 - p_i)(2\phi(x)_i - 1) \tfrac{1}{p(x)} [\![x{=}i]\!]. \tag{79}$$

Its expectation is

$$(1 - p_i)(2\phi_i(i) - 1) = 1 - p_i. \tag{80}$$

For $J^{\mathrm{DARN}}_{p_k} = 0$ for $k \neq i$ we have

$$J^{\mathrm{DARN}}_{p_k} = (J_{\phi_k}(x) - J_\phi(x)p) \tfrac{1}{p(x)} [\![x{=}k]\!] \tag{81a}$$

$$= -J_\phi(x)p \tfrac{1}{p(x)} [\![x{=}k]\!] \tag{81b}$$

$$= -p_i(2\phi(x)_i - 1) \tfrac{1}{p(x)} [\![x{=}k]\!]. \tag{81c}$$

Its expectation is

$$-p_i(2\phi_i(k) - 1) = p_i. \tag{82}$$

By subtracting p_i from all ordinates of J^{DARN}_p, we obtain an equivalent (having identical derivative in η) form where $\mathbb{E}[J^{\mathrm{DARN}}_{p_k}] = 0$ for all $k \neq i$ and $\mathbb{E}[J^{\mathrm{DARN}}_{p_i}] = 1 - 2p_i$, which cancels with $\mathbb{E}[J^{\mathrm{ST}}_{p_i}]$.

Next we consider a bilinear monomial in ϕ: $\mathcal{L}(\phi(x)) = \phi(x)_1\phi(x)_2$, where we have taken indices 1 and 2, without loss of generality. Its is zero for all discrete assignments and therefore the gradient of its expectation is zero. We have

$$J_{phi_1} = \phi_2(x) = [\![x{=}2]\!] \tag{83a}$$
$$J_{phi_2} = \phi_1(x) = [\![x{=}1]\!]. \tag{83b}$$

For ST we have $J_p^{\mathrm{ST}} = J_\phi$ and

$$\mathbb{E}[J_{p_1}^{\mathrm{ST}}] = p_2, \tag{84a}$$
$$\mathbb{E}[J_{p_2}^{\mathrm{ST}}] = p_1, \tag{84b}$$
$$\mathbb{E}[J_{p_k}^{\mathrm{ST}}] = 0, \quad k \neq 1, 2. \tag{84c}$$

For DARN part we have:

$$J_{p_1}^{\mathrm{DARN}} = [\![x{=}1]\!]\tfrac{1}{p_1}(J_{\phi_1} - J_{\phi_1}p_1 - J_{\phi_2}p_2), \tag{85a}$$
$$J_{p_2}^{\mathrm{DARN}} = [\![x{=}2]\!]\tfrac{1}{p_2}(J_{\phi_2} - J_{\phi_1}p_1 - J_{\phi_2}p_2), \tag{85b}$$
$$J_{p_k}^{\mathrm{DARN}} = [\![x{=}k]\!]\tfrac{1}{p_k}(-J_{\phi_1}p_1 - J_{\phi_2}p_2), \quad k \neq 1, 2. \tag{85c}$$

In the expectation, substituting J_ϕ:

$$\mathbb{E}[J_{p_1}^{\mathrm{DARN}}] = [\![x{=}1]\!]\tfrac{1}{p_1}(\phi_2(1) - \phi_2(1)p_1 - \phi_1(1)p_2) = -p_2, \tag{86a}$$
$$\mathbb{E}[J_{p_2}^{\mathrm{DARN}}] = [\![x{=}2]\!]\tfrac{1}{p_2}(\phi_1(2) - \phi_2(2)p_1 - \phi_1(2)p_2) = -p_1, \tag{86b}$$
$$\mathbb{E}[J_{p_k}^{\mathrm{DARN}}] = [\![x{=}k]\!]\tfrac{1}{p_k}(-\phi_2(k)p_1 - \phi_1(k)p_2) = 0, \quad k \neq 1, 2. \tag{86c}$$

This exactly cancels with ST.

The elementary functions we have considered form a basis in the space of all quadratic functions. By linearity argument, $J^{ZGR} = \frac{1}{2}(J^{ST} + J^{DARN})$ is unbiased for all quadratic functions. □

B. Details of Experiments

Here we give detailed specifications of our experiments. The implementation of all experiments will be made publicly available upon publication. During the review period, we will be happy to answer questions and share the code with reviewers confidentially through the OpenReview platform.

B.1. Dataset

In quantized training we use **MNIST**[1] and **FashionMNIST**[2] datasets. Each contains 60000 training and 10000 test images. We used 54000 images for training and 6000 for validation.

In VAE training, following the prior work, we use a decoder with Bernoulli output layer, which requires binary datasets. **MNIST-B** is a binarized MNIST with a fixed threshold of 0.5, same as in (Yin et al., 2019). The original Omniglot dataset is of the size 105×105 and contains binary images. However the established benchmarks use its down-sampled version (to size 28×28), which is then dynamically sampled: binary pixel values are generated with probabilities proportional to the original pixel values (Burda et al., 2016; Dong et al., 2021), which we denote as **Omniglot-28-D**. The down-scaled dataset published by (Burda et al., 2016)[3] was used, same as in the public implementation of (Dong et al., 2021). It contains about 24000 training images, which were split into training (90%) and validation (10%) parts and currently we are not using the validation part.

B.2. Methods

ZGR for categorical variables can be implemented as shown in Fig. B.1. It is a plug-in estimator, meaning that it is sufficient to use it for every tensor of categorical variables in a hierarchical model and the gradient in all parameters will be computed by back-propagation automatically.

[1] http://yann.lecun.com/exdb/mnist/
[2] https://github.com/zalandoresearch/fashion-mnist
[3] https://github.com/yburda/iwae/raw/master/datasets/OMNIGLOT/chardata.mat

```
def ZGR(p:Tensor)->Tensor:
    """Returns a categorical sample from p [*,C] (over axis=-1) as one-hot vector, with
        ZGR gradient.
    """
    index = distributions.categorical.Categorical(probs=p).sample()
    x = functional.one_hot(index, num_classes=p.shape[-1]).to(p)
    logpx = p.log().gather(-1, index.unsqueeze(-1))# log p(x)
    dx_ST = p
    dx_RE = (x - p.detach()) * logpx
    dx = (dx_ST + dx_RE) / 2
    return x + (dx - dx.detach()) # value of x with backprop through dx
```

Figure B.1: ZGR implementation in Pytorch for a general categorical variable.

Gumbel-Softmax (**GS**) and Straight-through Gumbel-Softmax (**GS-ST**) (Jang et al., 2017) are shipped with pytorch[4].

For Gumbel-Rao with MC samples (**GR-MC**) we adopted the public reimplementation by nshepperd [5], which is parallel over MC-samples.

RF(M) we implemented according to (Kool et al., 2019), Eq. 8. The part of the computation relevant to the encoder is propagated forward and backward only once. In the decoder we perform as many backward passes as forward, as this reduces variance of the gradient in decoder parameters. In quantization our implementation performs a backward pass for each forward pass.

For **ARSM** (Yin et al., 2019) we made own reimplementation, cross-checked with the authors tensor-flow implementation [6]. As with RF(M), we also performed a backward pass for each forward pass.

B.3. VAE

Model In our model each categorical variable is encoded as a vector of ± 1, corresponding to the bit representation of x, similar to (Paulus et al., 2021). There is a fixed number of total hidden bits (192), which are split into several categorical variables. For example 192 1b variables or 32 6-bit variables. This way the number of weights in the network stays constant. The network architecture is adopted from (Yin et al., 2019):

$$\text{Linear(784,512)} \to \text{LReLU} \to \text{Linear(512,256)} \to \text{LReLU} \to \text{Linear(256, D*K)},$$

where in the last layer we have D of K-way categorical units and LReLU has a leaky coefficient of 0.2 (same as in (Dong et al., 2021), default in tensorflow). The output of the encoder defines logits of the encoder Bernoulli model $q(z_i{=}1|x)$, where x is the input binary image and z is the latent discrete state.

The decoder has exactly the reverse Linear-LReLU architecture and outputs logits of conditionally independent Bernoulli generative model $p(x_i{=}1|z)$. We optimize the standard evidence lower bound (ELBO) (Kingma & Welling, 2013) with prior distribution $p(z)$ uniform and not learned.

We do not perform any special data-based initializations like subtracting data mean in the encoder in (Dong et al., 2021).

Optimization In the forward pass all methods produce a sample, from which a stochastic estimate of the gradient with respect to the decoder parameters is readily computed by backpropagation through decoder. We compute the KL term in ELBO analytically for a mini-batch and use its exact gradient. The estimation problem (1) occurs for the gradient of the data term with respect to the encoder parameters, where the estimators through discrete variables are applied.

All methods, including GS that optimizes ELBO with relaxed samples, are evaluated by the correct ELBO with discrete samples.

In the VAE experiments we measure the gradient accuracy and the training performance and do not make use of validation or test sets. First, this is reasonable when comparing quality of gradient estimators, regardless generalization. Second, the prior work (Dong et al., 2021) has verified that improvement in the training ELBO translates into improvement of the test

[4] Pytorch function `torch.functional.gumbel_softmax`

[5] https://github.com/nshepperd/gumbel-rao-pytorch

[6] https://github.com/ARM-gradient/ARSM

ELBO and IWAE bounds.

Following (Dong et al., 2021) we train with Adam with learning rate 10^{-4} using batch size 50. Furthermore we tried to match the training time that of (Dong et al., 2021). For MNIST we perform 500 epochs, and for Omniglot-28-D we perform 1000 epochs, roughly equivalent in booth cases to their $500K$ iterations with batch size 50.

B.3.1. HIERARCHICAL VAE

Model The model extends the 2-layer linear hierarchical VAE model of Grathwohl et al. (2018) to categorical variables. As above, each categorical variable is encoded as a vector of ± 1, corresponding to the bit representation of x. There is a fixed number of total hidden bits in each layer (192), which are split into V K-way categorical variables. The encoder $q(z_1|x)$ is a linear mapping Linear(784,V*K). The encoder $q(z_2|z_1)$ is a linear mapping Linear(192,V*K). The decoders are likewise in the reverse directions. The prior distribution $p(z_2)$ is set to uniform and not learned.

Optimization We optimize the ELBO:

$$\log p(x) \geq \sum_{z_1,z_2} q(z_1|x)q(z_2|z_1)(\log p(x|z_1) + \log p(z_1|z_2) + \log p(z_2) - \log q(z_1|x) - \log q(z_2|z_1)). \tag{87}$$

The expectations such as $\sum_{z_2} q(z_2|z_1) \log q(z_2|z_1)$ are computed in closed form. The optimization parameters are kept the same with the VAE setup above.

B.4. Bias-Variance Analysis

We conducted the bias-variance analysis for VAE at different training stages. Namely, we trained the model using RF(4) for 1, 10, 100, and 200 epochs and at each stage evaluated bias and variance of all gradient estimators. The model used 16 categorical variables of 16 categories (64 total latent bits). The estimates of the squared bias average and variance, both on average over all network parameters, are obtained using 10000 samples of the reference RF(4) estimator and 10000 samples of the evaluated estimator. The temperature range for GS family is $[0.01, 2]$, uniformly in the log space. The temperature range for ST(t) family is $[1, 8]$, uniformly in the log space. The range of α in ST–ZGR(α) family is $[0, 1]$.

The results are displayed in Fig. B.2. At the very beginning of training, the picture looks substantially different in that there is some bias reversal in GS-ST and derived estimators. However from epoch 10 and on the trends and relative ordering of methods stabilizes, with only RF(4) slightly overtaking ZGR in variance. The different picture after 1 epoch suggests that it would be beneficial in practice to warm-up the training with a few epochs using an estimator with lower variance.

B.5. Quantized Neural Networks

Experimental Setup In this experiment we train a convolutional network 32C5-MP2-64C5-MP2-512FC-10FC, closely replicating the model evaluated by Louizos et al. (2019) for MNIST. Each activation quantization is preceded by batch normalization (Ioffe & Szegedy, 2015).

All gradient estimators are working with the same network, parametrization and initialization. In the case of logistic noise the noise standard deviation is learnable and is initialized to $1/3$. All methods are applied with Adam optimizer for 200 epochs. For every method we select the bast validation performance with the grid search for the learning rate from $\{10^{-3}, 3.3 \cdot 10^{-4}, 10^{-4}\}$. We used the step-wise learning rate schedule decreasing the learning rate 10 times at epochs 100 and 150. The whole procedure is repeated for 3 different initialization seeds and we report the mean test error over seeds and $\pm(\max - \min)/2$ over seeds.

For validation and testing, we evaluate the network in the 'deterministic' mode, turning off all injected noises. This corresponds to a simple deterministic quantized model to be deployed.

Single Unit Quantization We include the following toy experiment that well illustrates properties of different estimators. We evaluate bias and variance of all estimators on a simple function of a single quantized variable. Let η be a real-valued parameter. Let $p(x; \eta)$ be given by the stochastic quantization model with $K = 4$ states and a particular noise type. Given a test function $\mathcal{L}(x)$ we can compute the true gradient of $\mathbb{E}[\mathcal{L}(x)]$. For each estimator we draw 10^4 samples to compute its mean and standard deviation for each value of η. The results are presented in Fig. B.5. In this plot we show several combinations of loss functions and noises. The test functions are: *linear* $\mathcal{L}(x) = x$; *quadratic* $\mathcal{L}(x) = \frac{1}{2}(x - c)^2$ and *sigmoid* $\mathcal{L}(x) = \sigma(2(x - c))$, where $c = (K - 1)/2$ is chosen for centering. The noises shown refer to the *logistic* noise with std = 1/3 as used at initialization by Louizos et al. 2019 and the *triangular* noise with the density $p(z) = \max(0, 1 - |z|)$.

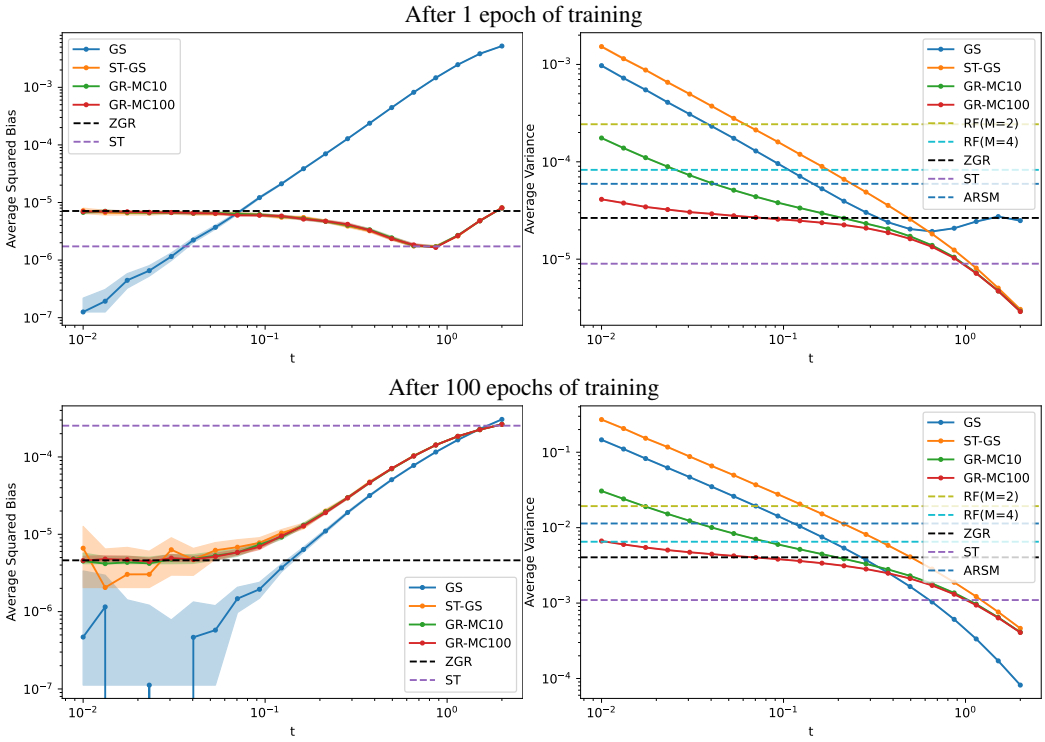

Figure B.2: Gradient estimation accuracy in VAE on MNIST-B. Average (per parameter) squared bias (*left*) and variance (*right*) of gradient estimators versus temperature for a model snapshot at a particular iteration of training with RF(4). VAE network with 16 categorical variables with 16 categories.

Table B.1: Non-linear VAE **test** negative ELBO.

	MNIST-B				**Omniglot-28-D**			
Method	C2 V192	C4 V96	C16 V48	C64 V32	C2 V192	C4 V96	C16 V48	C64 V32
GS(t=0.1)	94.7±0.1	86.8±0.4	84.2±0.6	87.9±0.9	120.0±0.2	118.1±0.2	118.9±0.1	122.2±0.2
GS-ST(t=0.1)	94.0±0.2	87.0±0.3	85.0±0.4	90.8±0.5	121.3±0.2	118.6±0.1	120.3±0.0	124.5±0.4
GR(t=0.1,K=10)	**92.5±0.3**	**85.2±0.1**	**82.5±0.1**	**83.7±0.2**	**118.7±0.4**	**116.8±0.3**	**117.7±0.2**	**119.8±0.3**
GR(t=0.1,K=100)	**92.5±0.7**	**85.2±0.1**	**82.2±0.4**	**83.2±0.6**	**118.8±0.1**	**116.4±0.1**	**117.1±0.1**	**118.8±0.2**
ZGR	**94.0±0.1**	**86.2±0.2**	**82.5±0.3**	**83.4±0.0**	**119.0±0.3**	**116.6±0.2**	**117.3±0.2**	**119.0±0.2**
ST	105.3±0.4	105.8±0.4	106.2±0.3	107.0±0.3	131.1±0.1	131.4±0.1	132.2±0.0	132.7±0.2
RF(M=2)	97.5±0.3	88.9±0.3	89.9±0.5	97.5±0.1	123.3±0.4	121.1±0.1	123.8±0.3	128.9±0.3
RF(M=4)	99.1±0.1	87.6±0.1	84.3±0.5	89.0±0.5	120.6±0.1	118.4±0.3	120.1±0.0	122.8±0.1

Table B.2: Hierarchical VAE **test** negative ELBO.

	MNIST-B				**Omniglot-28-D**			
Method	C2 V192	C4 V96	C16 V48	C64 V32	C2 V192	C4 V96	C16 V48	C64 V32
GS(t=0.1)	123.7±0.7	120.8±0.4	128.8±0.3	138.1±0.3	**141.2±0.2**	146.1±0.5	152.5±0.1	162.2±0.2
GS-ST(t=0.1)	124.0±0.6	121.1±0.2	129.2±0.2	138.7±0.1	145.8±0.4	147.3±0.1	153.4±0.0	162.7±0.2
GR(t=0.1,M=10)	**119.5±1.1**	**112.8±0.6**	**117.8±0.5**	**124.8±0.4**	149.0±0.8	**142.6±0.5**	**145.3±0.3**	**151.8±0.1**
GR(t=0.1,M=100)	**119.1±1.5**	**111.2±0.5**	**115.0±0.2**	**121.3±0.3**	149.8±0.6	**141.5±0.4**	**143.2±0.4**	**149.2±0.3**
ZGR	**118.2±2.2**	**111.8±0.0**	**115.6±0.2**	**121.9±0.3**	150.3±0.8	**143.2±1.5**	**143.6±0.3**	**149.1±0.4**
ST	124.7±0.1	125.2±0.1	124.7±0.1	125.1±0.1	**145.8±0.3**	146.3±0.3	148.1±0.1	149.8±0.0
RF(M=2)	129.3±0.2	129.3±0.2	136.8±0.9	145.7±0.3	146.5±0.7	150.2±0.2	158.0±0.3	168.0±0.4
RF(M=4)	120.4±1.0	121.7±0.6	127.8±0.7	134.7±1.6	**141.6±0.7**	145.6±0.5	153.8±0.4	161.3±0.2

The bias of the GS family quickly decreases with the temperature. ZRG estimator achieves the same expected value as GS-ST in the limit of small temperature illustrated by GS-ST(t=0.1) and the variance comparable to that of GR(t=0.1, M=100). We also verify that ZGR has zero bias for quadratic objectives as we have shown theoretically.

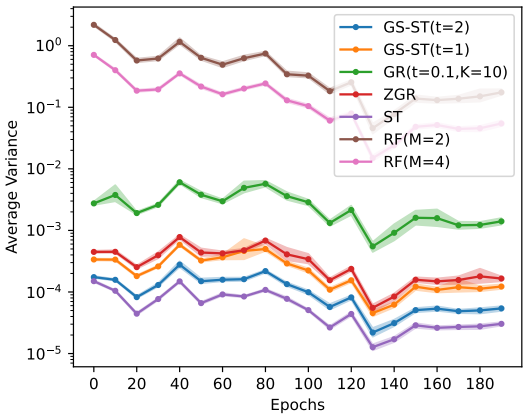

Figure B.3: Quantization: average (per parameter) variance in the first layer of the network of different estimators along the training trajectory of ZGR (ternary weights/activations, triangular noise).

B.6. Bias-Variance Analysis

The bias-variance analysis for quantization uses the following setup. In order to be able to measure bias with a sufficient accuracy, we trained a small model 8C5-MP2-16C5-MP2-128FC-10FC with ternary weights and activations. The model is trained using ZGR for 20 epochs. We estimate the squared bias and variance, both on average over parameters in layer 0 of the network. Because of high variance of all estimators, the experiment required $4 \cdot 10^5$ samples of the reference method RF(4) and $4 \cdot 10^4$ of each of the evaluated estimators per hyperparameter point. The temperature range for GS family is $[\mathbf{0.1}, 2]$, uniformly in the log space. The temperature range for ST(t) family is $[1, 8]$, uniformly in the log space. The range of α in ST–ZGR(α) family is $[0, 1]$. In the plot Fig. 4 we show ST–ZGR interpolation between ST(t=1) and ZGR. It is seen that ZGR achieves a smaller bias compared to ST but at a price of a substantial increase in the variance. To contrast it to ST trend we also show $ST < 1$ for temperatures ranging in $[1, 0.5]$, uniformly in the log space.

Additionally, we evaluate the variance of all methods in the full size model (which is easier to estimate accurately), during the training in Fig. B.3.

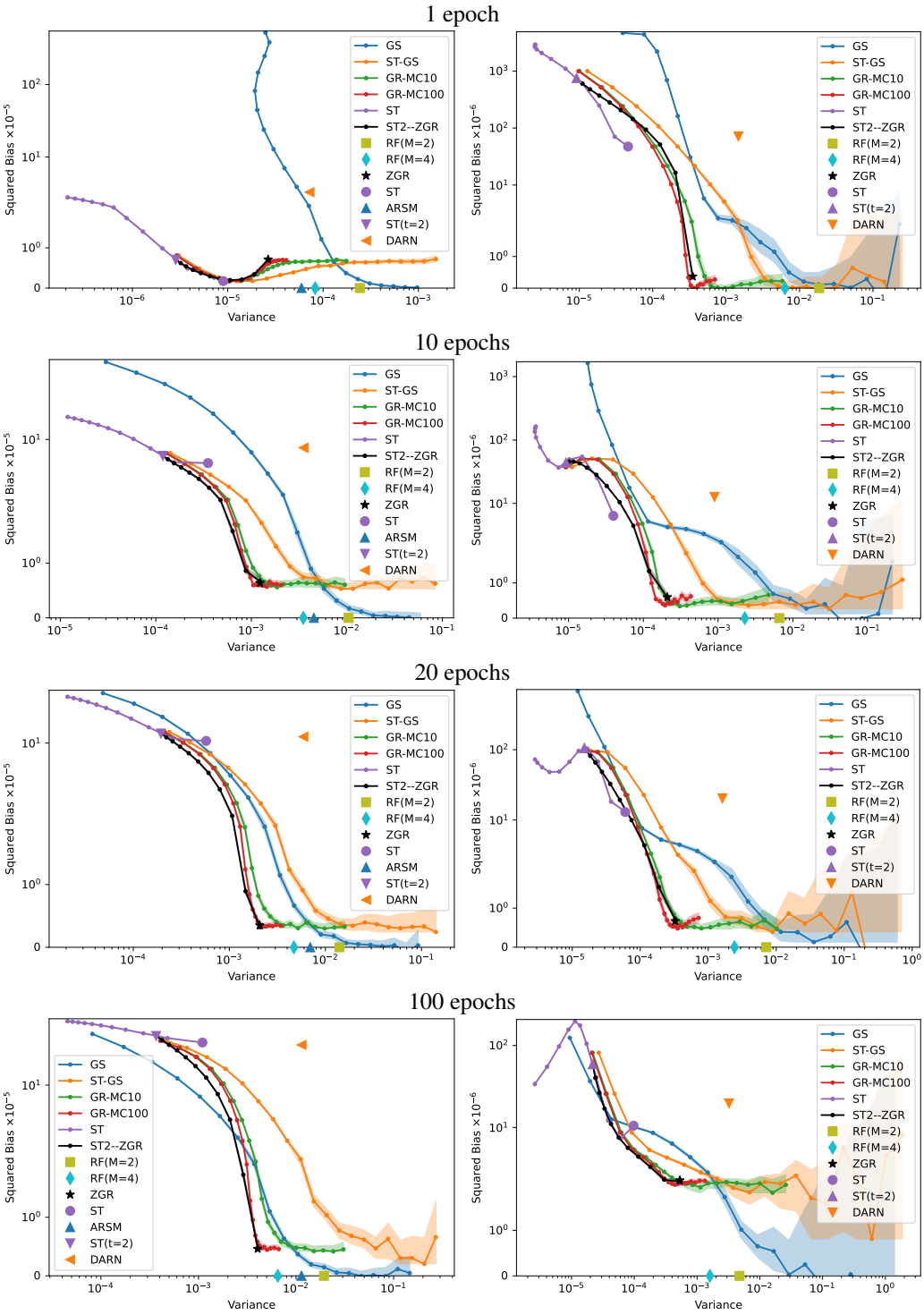

Figure B.4: Bias-variance tradeoff diagrams as in Fig. 4 during different training stages. *Left*: VAE, *right*: HVAE.

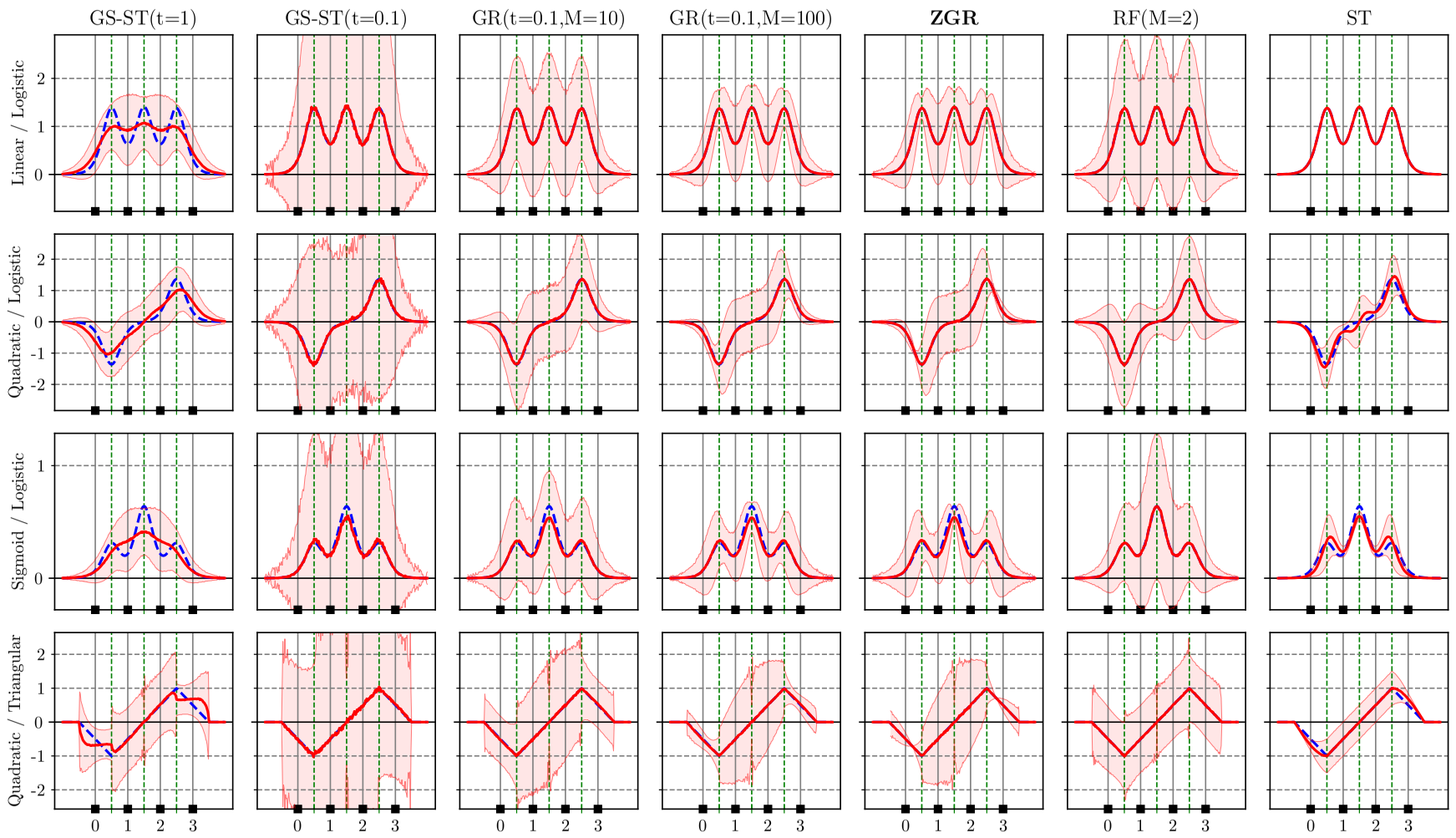

Figure B.5: Single unit quantization: performance of estimators in stochastic quantization with selected combinations of test functions and injected noise for varied input η of the stochastic quantizer. The dashed blue line is the exact gradient. Red line is the mean of the estimator and the red shaded area shows ± 1 std. GR with 100 is able to reduce the variance of GS-ST substantially. ZGR reduces the variance by an edge further while keeping the bias equal to the theoretical bias of GR with zero temperature. The variance of RF(M=2) is substantially higher. Plain ST estimator has a yet smaller variance but a larger bias. See Appendix B.5 for details of the experiment.