Generalized Doubly-Reparameterized Gradient Estimators

Matthias Bauer [1] **Andriy Mnih** [1]

Abstract

Efficient low-variance gradient estimation enabled by the reparameterization trick (RT) has been essential to the success of variational autoencoders. Doubly-reparameterized gradients (DREGs) improve on the RT for multi-sample variational bounds by applying reparameterization a second time for an additional reduction in variance. Here, we develop two generalizations of the DREGs estimator and show that they can be used to train conditional and hierarchical VAEs on image modelling tasks more effectively. First, we extend the estimator to hierarchical models with several stochastic layers by showing how to treat additional score function terms due to the hierarchical variational posterior. We then generalize DREGs to score functions of arbitrary distributions instead of just those of the sampling distribution, which makes the estimator applicable to the parameters of the prior in addition to those of the posterior.

1. Introduction

In probabilistic machine learning we often optimize expectations of the form $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]$ w.r.t. to their parameters, where $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ is some objective function, and $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ denote the parameters of the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and other (e.g. model) parameters, respectively. In the case of the influential variational autoencoder (VAE, Kingma & Welling (2014), Rezende et al. (2014)), $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is the variational posterior, $\boldsymbol{\theta}$ denotes the model parameters, and $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}$ is typically either the ELBO (Jordan et al., 1999; Blei et al., 2017) or IWAE (Burda et al., 2016) objective.

In most cases of interest, this expectation is intractable, and we estimate it and its gradients, $\nabla_{\boldsymbol{\phi}}\mathcal{L}$ and $\nabla_{\boldsymbol{\theta}}\mathcal{L}$, using Monte Carlo samples $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$. The resulting gradient estimators are characterized by their *bias* and *variance*. We usually prefer unbiased estimators as they tend to be better-behaved and are better understood. Lower variance is also preferable because it enables faster training by allowing using higher learning rates.

In this paper, we address gradient estimation for continuous variables in variational objectives. A naive implementation of $\nabla_{\boldsymbol{\phi}}\mathcal{L}$ results in a *score function*, or REINFORCE, estimator (Williams, 1992), which tends to have high variance; however, if $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ depends on $\boldsymbol{\phi}$ only through $\boldsymbol{z}$, we can use reparameterization (Kingma & Welling, 2014; Rezende et al., 2014) to obtain an estimator with lower variance by replacing the score function estimator of the gradient with a *pathwise estimator*.

In variational inference, $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ typically depends on $\boldsymbol{\phi}$ not only through $\boldsymbol{z}$ but also through the value of the log density $\log q_{\boldsymbol{\phi}}(\boldsymbol{z})$. Then, the gradient estimators still involve the score function $\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$ despite using reparameterization. Roeder et al. (2017) propose the *sticking the landing* (STL) estimator, which simply drops these score function terms to reduce variance. Tucker et al. (2019) show that STL is biased in general, and introduce the *doubly-reparameterized gradient* (DREGs) estimator for IWAE objectives, which again yields unbiased lower-variance gradient estimates. This is achieved by applying reparameterization a second time, targeting the remaining score function terms.

However, the DREGs estimator has two major limitations: 1) it only applies to latent variable models with a single latent layer; 2) it only applies in cases where the score function depends on the same parameters as the sampling distribution. In this work we address both limitations. Moreover, we show that for hierarchical models with several stochastic layers, gradients that look like pathwise gradients can actually contain additional score function gradients that are not doubly reparameterizable. Despite this, we show that we can still obtain a simple estimator with a sizable reduction in gradient variance for hierarchical IWAE objectives.

Our main contributions are:

- We extend DREGs to hierarchical models;
- We introduce GDREGs, a generalization of DREGs to score functions that depend on a different distribution than the sampling distribution;

[1]DeepMind, London, UK. Correspondence to: Matthias Bauer <msbauer@deepmind.com>, Andriy Mnih <andriy@deepmind.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

- We show how to implement all proposed gradient estimators using automatic differentiation frameworks;
- We evaluate the proposed DREGs and GDREGs estimators on several conditional and unconditional unsupervised learning problems and find that they outperform the regular IWAE estimator.

2. Background

In this work we are interested in computing gradients of variational objectives of the form

$$\mathcal{L}_{\phi,\theta} = \mathbb{E}_{z\sim q_\phi(z)}\left[f_{\phi,\theta}(z)\right] \tag{1}$$

w.r.t. the variational parameters ϕ of the sampling distribution $q_\phi(z)$, and parameters θ of a second distribution $p_\theta(z)$, such as a learnable prior. Here $f_{\phi,\theta}(z)$ is a general function of z that can also explicitly depend on both $q_\phi(z)$ and $p_\theta(z)$. More precisely, we wish to compute

$$\nabla^{\text{TD}}_\phi \mathcal{L}_{\phi,\theta} \qquad \text{and} \qquad \nabla^{\text{TD}}_\theta \mathcal{L}_{\phi,\theta}, \tag{2}$$

where ∇^{TD}_* denotes the total derivative, which we explicitly distinguish from the partial derivative ∇_*.

Arguably the simplest objectives of this form are the negative entropy $\mathcal{L}^{\text{ent}}_{\phi,\theta} = \mathbb{E}_{z\sim q_\phi(z)}\left[\log q_\phi(z)\right]$ and negative cross-entropy $\mathcal{L}^{\text{ce}}_{\phi,\theta} = \mathbb{E}_{z\sim q_\phi(z)}\left[\log p_\theta(z)\right]$.

Importance weighted autoencoders. Another such objective is the importance weighted autoencoder (IWAE) bound (Burda et al., 2016). For a VAE with likelihood $p_\lambda(x|z)$, (learnable) prior $p_\theta(z)$, and variational posterior (or proposal) $q_\phi(z|x)$ the IWAE bound with K importance weights $w_k = \frac{p_\theta(z_k)p_\lambda(x|z_k)}{q_\phi(z_k|x)}$ is given by

$$\mathcal{L}^{\text{IWAE}}_{\phi,\theta} = \mathbb{E}_{z_1,\ldots,z_K\sim q_\phi(z|x)}\left[\log\left(\frac{1}{K}\sum_{k=1}^K w_k\right)\right]. \tag{3}$$

Eq. (3) reduces to the regular ELBO objective for $K = 1$ (Rezende et al., 2014; Kingma & Welling, 2014),

$$\mathcal{L}^{\text{ELBO}}_{\phi,\theta} = \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log \frac{p_\theta(z)p(x|z)}{q_\phi(z|x)}\right]. \tag{4}$$

Burda et al. (2016) showed that using multiple importance samples ($K > 1$) provides the model with more flexibility to learn richer representations (fewer inactive units), and results in better log-likelihood estimates compared to VAEs trained with the single sample ELBO. The estimators discussed in this paper build on these results and lead to further improvements. While we focus on the IWAE objective, our proposed GDREGs estimator applies generally.

Gradient estimation. In practice, the expectation in Eq. (1) and its gradients are intractable, so we approximate them using Monte Carlo sampling, which makes the estimates of the objective and its gradients random variables. The resulting gradient estimators will be unbiased but have non-zero variance. We prefer estimators with lower variance, as they enable fast training by allowing higher learning rates.

We can distinguish between two general types of gradient estimators in this setting: (i) *score function estimators* and (ii) *pathwise estimators*. Score functions are gradients of a log probability density w.r.t. its parameters, such as $\nabla_\phi \log q_\phi(z)$; they treat the function $f_{\phi,\theta}(z)$ as a black box and often yield high variance gradients. In contrast, pathwise estimators move the parameter-dependence from the probability density into the argument z of the function $f_{\phi,\theta}(z)$ and derive the computation path to often achieve lower variance gradients by using the knowledge of $\nabla_z f_{\phi,\theta}(z)$; see Mohamed et al. (2020) for a recent review.

When computing gradients of the objective $\mathcal{L}_{\phi,\theta}$ we have to differentiate both the sampling distribution of the expectation, $q_\phi(z)$, as well as the function $f_{\phi,\theta}(z)$,

$$\begin{aligned}\nabla^{\text{TD}}_\phi \mathbb{E}_{q_\phi(z)}\left[f_{\phi,\theta}(z)\right] &= \\ &= \mathbb{E}_{q_\phi(z)}\left[\nabla_\phi f_{\phi,\theta}(z) + f_{\phi,\theta}(z)\nabla_\phi \log q_\phi(z)\right]\end{aligned} \tag{5}$$

$$\nabla^{\text{TD}}_\theta \mathbb{E}_{q_\phi(z)}\left[f_{\phi,\theta}(z)\right] = \mathbb{E}_{q_\phi(z)}\left[\nabla_\theta f_{\phi,\theta}(z)\right], \tag{6}$$

and both can give rise to score functions. To see that all of the underlined terms indeed contain score functions, note that we can rewrite $\nabla_\phi f_{\phi,\theta}(z)$ as $\nabla_\phi f_{\phi,\theta}(z) = \nabla_{\log q_\phi(z)} f_{\phi,\theta}(z)\nabla_\phi \log q_\phi(z)$ and similarly for $\nabla_\theta f_{\phi,\theta}(z) = \nabla_{\log p_\theta(z)} f_{\phi,\theta}(z)\nabla_\theta \log p_\theta(z)$.

In the following we recapitulate how to address the score functions w.r.t. ϕ in Eq. (5) using the reparameterization trick and doubly-reparameterized gradients (DREGs, Tucker et al. (2019)), respectively. In Sec. 4 we introduce GDREGs, a generalization of DREGs, that eliminates the score function w.r.t. θ in Eq. (6).

Reparameterization. We can use the *reparameterization trick* (Kingma & Welling, 2014; Rezende et al., 2014) to turn the score function $\nabla_\phi \log q_\phi(z)$ inside the expectation in Eq. (5) into a pathwise derivative of the function $f_{\phi,\theta}(z)$ as follows: we express the latent variables $z \sim q_\phi(z)$ through a bijection of new random variables $\epsilon \sim q(\epsilon)$, which are independent of ϕ,

$$z = \mathcal{T}_q(\epsilon;\phi) \Leftrightarrow \epsilon = \mathcal{T}_q^{-1}(z;\phi). \tag{7}$$

This allows us to rewrite expectations w.r.t. $q_\phi(z)$ as $\mathbb{E}_{q_\phi(z)}\left[f_{\phi,\theta}(z)\right] = \mathbb{E}_{q(\epsilon)}\left[f_{\phi,\theta}(\mathcal{T}_q(\epsilon;\phi))\right]$, which moves the parameter dependence into the argument of $f_{\phi,\theta}(z)$ and gives rise to a pathwise gradient:

$$\begin{aligned}\nabla^{\text{TD}}_\phi \mathbb{E}_{q_\phi(z)}\left[f_{\phi,\theta}(z)\right] = \mathbb{E}_{q(\epsilon)}\big[&\nabla_\phi f_{\phi,\theta}(z) + \\ &+ \nabla_z f_{\phi,\theta}(z)\nabla_\phi \mathcal{T}_q(\epsilon;\phi)\big]_{z=\mathcal{T}_q(\epsilon;\phi)}.\end{aligned} \tag{8}$$

In Sec. 3 we discuss that this seemingly pathwise gradient in Eq. (8) can actually contain score functions for more structured or hierarchical models and explain how to extend DREGs to this case. For the remainder of this section we restrict ourselves to simple (single stochastic layer) models.

Double reparameterization. Tucker et al. (2019) further reduce gradient variance by replacing the remaining score function in the reparameterized gradient Eq. (8), $\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) = \nabla_{\log q_{\boldsymbol{\phi}}(\boldsymbol{z})} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$, with its reparameterized counterpart. Double reparameterization is based on the identity Eq. (9) (Eq. 5 in Tucker et al. (2019)),

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\phi}}\log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right] =$$
$$= \mathbb{E}_{\boldsymbol{\epsilon}\sim q(\boldsymbol{\epsilon})}\left[\nabla^{\text{TD}}_{\boldsymbol{z}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\big|_{\boldsymbol{z}=\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi})} \nabla_{\boldsymbol{\phi}}\mathcal{T}_q\left(\boldsymbol{\epsilon};\boldsymbol{\phi}\right)\right] \quad (9)$$
$$= \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\nabla^{\text{TD}}_{\boldsymbol{z}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\phi}}\mathcal{T}_q\left(\boldsymbol{\epsilon};\boldsymbol{\phi}\right)\big|_{\boldsymbol{\epsilon}=\mathcal{T}_q^{-1}(\boldsymbol{z};\boldsymbol{\phi})}\right] \quad (10)$$

which follows from the fact that both the score function and the reparameterization estimators are unbiased and thus equal in expectation. This identity holds for arbitrary $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$; to match the score function in Eq. (8) with the LHS of Eq. (9), we have to choose $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) = \nabla_{\log q_{\boldsymbol{\phi}}(\boldsymbol{z})} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$.

In Eq. (10) we have rewritten the expectation over $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$ in terms of $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$, as this will become useful for our later generalization. Note how, to compute the pathwise gradient, the sample $\boldsymbol{z}$ is mapped back to the noise variable $\boldsymbol{\epsilon} = \mathcal{T}_q^{-1}(\boldsymbol{z};\boldsymbol{\phi})$. $\nabla_{\boldsymbol{\phi}}\mathcal{T}_q(\boldsymbol{\epsilon},\boldsymbol{\phi})$ is also sometimes written as $\nabla_{\boldsymbol{\phi}}\boldsymbol{z}(\boldsymbol{\epsilon};\boldsymbol{\phi})$ (Tucker et al., 2019).

Gradient estimation for the IWAE objective. For the IWAE objective Eq. (3), Tucker et al. (2019) derived the following doubly-reparametererized gradients (DREGs) estimator, which supersedes the previously proposed STL estimator (Roeder et al., 2017):

$$\widehat{\nabla}^{\text{DREGs}}_{\boldsymbol{\phi}}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla^{\text{TD}}_{\boldsymbol{z}_k} \log w_k \nabla_{\boldsymbol{\phi}}\mathcal{T}_q(\boldsymbol{\epsilon}_k;\boldsymbol{\phi}) \quad (11)$$
$$\widehat{\nabla}^{\text{STL}}_{\boldsymbol{\phi}}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \widetilde{w}_k \nabla^{\text{TD}}_{\boldsymbol{z}_k} \log w_k \nabla_{\boldsymbol{\phi}}\mathcal{T}_q(\boldsymbol{\epsilon}_k;\boldsymbol{\phi}) \quad (12)$$

with normalized importance weights $\widetilde{w}_k = \frac{w_k}{\sum_{k'=1}^{K} w_{k'}}$ and $\boldsymbol{\epsilon}_{1:K} \sim q(\boldsymbol{\epsilon})$. While the DREGs estimator doubly-reparameterizes the score function in Eq. (8), the STL estimator simply drops it and is biased as a result. Crucially, because DREGs relies on reparameterization, it is limited to score functions of the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, making it inapplicable in the more general setting of arbitrary score functions, such as $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]$ in Eq. (6).

3. DREGs for hierarchical models

We now show that for models with hierarchically structured latent variables even terms that look like pathwise gradients, such as $\nabla^{\text{TD}}_{\boldsymbol{z}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ in Eq. (8) or $\nabla^{\text{TD}}_{\boldsymbol{z}_k} \log w_k$ in the DREGs or STL estimator for the IWAE objective Eqs. (11) and (12), can give rise to additional score functions. *These additional score functions appear because the distribution parameters of one stochastic layer depend on the latent variables of another layer.* Their appearance is contrary to the intuition that doubly-reparameterized gradient estimators only contain pathwise gradients.

3.1. An illustrative example

To illustrate this, consider a hierarchical model with two layers where we first sample $\boldsymbol{z}_2 \sim q_{\boldsymbol{\phi}_2}(\boldsymbol{z}_2)$ and then $\boldsymbol{z}_1 \sim q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2)$.[1] Note that the conditioning on $\boldsymbol{z}_2$ is through the distribution parameters of $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2)$; to highlight this dependence of $\boldsymbol{z}_1$ on $\boldsymbol{z}_2$, we rewrite $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2) = q_{\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2,\boldsymbol{\phi}_1)}(\boldsymbol{z}_1)$, where we explicitly distinguish between the *distribution* parameters $\boldsymbol{\alpha}_{1|2}$, such as the mean and covariance of a Gaussian, and the *network* parameters $\boldsymbol{\phi}_1$ that parameterize them together with the previously sampled latent $\boldsymbol{z}_2$. A derivative w.r.t. $\boldsymbol{z}_2$ that looks like a pathwise gradient actually gives rise to a score function term at a subsequent level:

$$\nabla^{\text{TD}}_{\boldsymbol{z}_2}\log q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2) = \nabla^{\text{TD}}_{\boldsymbol{z}_2}\log q_{\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2,\boldsymbol{\phi}_1)}(\boldsymbol{z}_1) \quad (13)$$
$$= \nabla_{\boldsymbol{\alpha}_{1|2}}\log q_{\boldsymbol{\alpha}_{1|2}}(\boldsymbol{z}_1)\nabla_{\boldsymbol{z}_2}\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2,\boldsymbol{\phi}_1) + \dots. \quad (14)$$

We omitted (true) pathwise gradients $(\dots)$, as the samples $\boldsymbol{z}_1$ also depend on $\boldsymbol{z}_2$ through reparameterization. Similar additional score functions arise for seemingly pathwise gradients of hierarchical or autoregressive priors and variational posteriors.

3.2. Extending DREGs to hierarchical VAEs

Here we show how to extend DREGs to hierarchical VAEs to effectively reduce gradient variance for the variational posterior despite the results in the previous section. We still consider the IWAE objective (Eq. (3)), but now the latent space $\boldsymbol{z}$ is structured, and both $p_{\boldsymbol{\theta}}$ and $q_{\boldsymbol{\phi}}$ are hierarchically factorized distributions.

Let us consider a 2-layer VAE $\boldsymbol{z}_2 \to \boldsymbol{z}_1 \to \boldsymbol{x}$ and examine the term $\nabla^{\text{TD}}_{\boldsymbol{\phi}_2}\log q_{\boldsymbol{\phi}_1,\boldsymbol{\phi}_2}(\boldsymbol{z}_1,\boldsymbol{z}_2)$, which appears in the total derivative of the IWAE objective, as a concrete example. We have sampled $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ hierarchically using reparameterization: $\boldsymbol{z}_2(\boldsymbol{\phi}_2) \equiv \mathcal{T}_{q_2}(\boldsymbol{\epsilon}_2;\boldsymbol{\alpha}_2(\boldsymbol{\phi}_2))$ and $\boldsymbol{z}_1(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2) \equiv \mathcal{T}_{q_1}(\boldsymbol{\epsilon}_1;\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_2),\boldsymbol{\phi}_1))$:

$$\nabla^{\text{TD}}_{\boldsymbol{\phi}_2}\log\left[q_{\boldsymbol{\alpha}_2(\boldsymbol{\phi}_2)}(\boldsymbol{z}_2(\boldsymbol{\phi}_2))\, q_{\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_2),\boldsymbol{\phi}_1)}(\boldsymbol{z}_1(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2))\right] \quad (15)$$

When computing total derivatives w.r.t. parameters $\boldsymbol{\phi}_2$ of the upper layer, we distinguish between three types of gradients: the (true) pathwise gradients w.r.t. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, a *direct* score function because the distribution parameters $\boldsymbol{\alpha}_2(\boldsymbol{\phi}_2)$

[1] The subscript indices refer to the latent layer indices and not to the importance samples in this case.

directly depend on ϕ_2, and an *indirect* score function because the distribution parameter $\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\phi_2), \phi_1)$ indirectly depends on ϕ_2 through $\boldsymbol{z}_2(\phi_2)$. Indirect score functions do not arise in single stochastic layer models considered by Tucker et al. (2019), and we have three options to estimate them: (1) leave them—this naive estimator is unbiased but potentially has high variance; (2) drop them, similar to STL—this estimator is generally biased; (3) doubly-reparameterize them using DREGs again—this estimator is unbiased, but can generate further indirect score functions.

Total derivatives of other terms in the objective similarly decompose into pathwise gradients as well as direct and indirect score functions. Notably, this includes indirect score functions of the prior $\log p_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2}(\boldsymbol{z}_1, \boldsymbol{z}_2)$, to which DREGs does not apply. In Sec. 4 we introduce the generalized DREGs (GDREGs) estimator that applies in this case.

3.3. DREGs for hierarchical IWAE objectives

For IWAE objectives we find that the indirect score functions come up twice: once when computing pathwise gradients of the initial reparameterization, and a second time (with a different prefactor) when computing pathwise gradients for the double-reparameterization of the direct score functions. The same happens for the (true) pathwise gradients, and it is this double-appearance and the resulting cancellation of prefactors that helps reduce gradient variance for DREGs. Moreover, for general model structures it is impossible to replace all successively arising indirect score functions with pathwise gradients, even by applying GDREGs. For example, when the prior and posterior do not factorize in the same way, double-reparameterization of one continuously creates indirect score functions of the other and vice versa, see Apps. C and C.4 for more details.

Thus, to extend DREGs to hierarchical models, we leave the indirect score functions unchanged and only doubly reparameterize the direct score functions. The extended DREGs estimator for IWAE models with arbitrary hierarchical structures is given by Eq. (20)

DREGs estimator for hierarchical IWAE

$$\widehat{\nabla}^{\text{DREGs}}_{\phi_l} \mathcal{L}^{\text{IWAE}}_{\phi,\boldsymbol{\theta}} = \qquad \boldsymbol{\epsilon}_{\{1:K\}l} \sim q(\boldsymbol{\epsilon}) \quad (20)$$
$$= \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla^{\text{TD}}_{\boldsymbol{z}_{kl}} \log w_k \nabla_{\phi_l} \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\text{pa}_{\boldsymbol{\alpha}}(l), \phi_l)\right)$$

where l denotes the stochastic layer, $\text{pa}_{\boldsymbol{\alpha}}(l)$ is the set of latent variables that $\boldsymbol{z}_{kl}$ depends on, and $\boldsymbol{z}_{kl} = \mathcal{T}_{q_l}(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\text{pa}_{\boldsymbol{\alpha}}(l), \phi_l))$ through reparameterization. We provide a detailed derivation in App. C and a worked example for a VAE with two stochastic layers in App. D. In Apps. E and E.1 we show how to implement this estimator using automatic differentiation by using a *surrogate loss function*, whose forward computation we discard, but whose backward computation exactly corresponds to the estimator in Eq. (20). Alternatively, one could implement a custom gradient for the objective that directly implements Eq. (20); however, we found our approach using a surrogate loss function to be simpler both conceptually and implementation-wise.

Roeder et al. (2017) apply the STL estimator to hierarchical ELBO objectives but do not discuss indirect score functions. Their experimental results are consistent with maintaining the indirect score functions, similar to how we extend DREGs to hierarchical models; the STL estimator is biased for IWAE objectives (Tucker et al., 2019).

4. Generalized DREGs

Here, we generalize DREGs to score functions that involve distributions $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ different from the sampling distribution $q_{\phi}(\boldsymbol{z})$, as in Eq. (6). In other words, we would like to replace score function terms of the form $\mathbb{E}_{q_{\phi}(\boldsymbol{z})}\left[g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$ with doubly-reparameterized pathwise gradients. Such terms appear, for example, when training a VAE with a trainable prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ with the ELBO or IWAE objectives. DREGs cannot be used directly in this case as it relies on reparameterization of the sampling distribution $q_{\phi}(\boldsymbol{z})$, so that the path depends on the parameters ϕ,

GDREGs identity

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z})}[g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z})}\left[\left(g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z}) \nabla^{\text{TD}}_{\boldsymbol{z}} \log \tfrac{q_{\phi}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \nabla^{\text{TD}}_{\boldsymbol{z}} g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z})\right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z},\boldsymbol{\theta})}\right] \quad (16)$$

$$\nabla^{\text{TD}}_{\boldsymbol{\theta}} \mathbb{E}_{q_{\phi}(\boldsymbol{z})}\left[g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z})\right] \overset{①}{=} \nabla^{\text{TD}}_{\boldsymbol{\theta}} \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[\tfrac{q_{\phi}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z})\right] \overset{①}{=} \nabla^{\text{TD}}_{\boldsymbol{\theta}} \mathbb{E}_{q(\widetilde{\boldsymbol{\epsilon}})}\left[\tfrac{q_{\phi}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))}{p_{\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))} g_{\phi,\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))\right]; \quad q(\widetilde{\boldsymbol{\epsilon}}) = \mathcal{N}(0, \mathbb{I}) \quad (17)$$

$$\overset{②}{=} \mathbb{E}_{q(\widetilde{\boldsymbol{\epsilon}})}\left[\nabla^{\text{TD}}_{\boldsymbol{z}}\left(\tfrac{q_{\phi}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} g(\boldsymbol{z})\right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta}) + \tfrac{q_{\phi}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left(\nabla_{\boldsymbol{\theta}} g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z}) - g(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right)\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \quad (18)$$

$$\overset{③}{=} \mathbb{E}_{q_{\phi}(\boldsymbol{z})}\left[\left(g(\boldsymbol{z}) \nabla^{\text{TD}}_{\boldsymbol{z}} \log \tfrac{q_{\phi}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \nabla^{\text{TD}}_{\boldsymbol{z}} g(\boldsymbol{z})\right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})} + \nabla_{\boldsymbol{\theta}} g_{\phi,\boldsymbol{\theta}}(\boldsymbol{z}) - g(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right] \quad (19)$$

Figure 1: The GDREGs identity and a brief derivation in three steps: ① *temporarily* change the path so that it depends on $\boldsymbol{\theta}$; ② perform the reparameterized gradient computation; ③ change the path back so we can use samples $\boldsymbol{z} \sim q_{\phi}(\boldsymbol{z})$ to estimate the expectation. See App. A for details and an alternative derivation using the change of density formula.

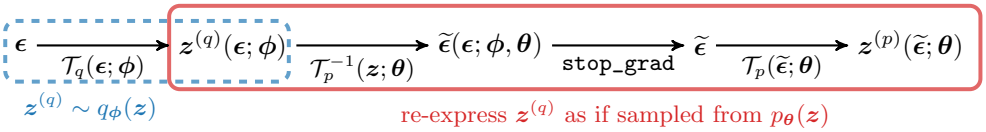

Figure 2: Computational flow to re-express a sample $\boldsymbol{z}$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ as if it were sampled from $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. Its numerical value and distribution remain unchanged but the pathwise gradient through it now depends on $\boldsymbol{\theta}$: $\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z},\boldsymbol{\theta})}$. $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi});\boldsymbol{\theta})$ follows a different, usually more complex, distribution from $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$.

whereas the score function is with w.r.t. parameters $\boldsymbol{\theta}$ of a different distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$.

To make progress *we need to make the path depend on the parameters* $\boldsymbol{\theta}$ while still sampling $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ during training. Our solution consists of three steps (also see Fig. 1):

① *temporarily* change the path so that it depends on $\boldsymbol{\theta}$;
② perform the reparameterized gradient computation;
③ change the path back so we can use samples $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ to estimate the expectation.

We change the path by first using an importance sampling reweighting to temporarily re-write the expectation, $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[*] = \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}*\right]$, and then employing reparameterization on the new sampling distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$: $\boldsymbol{z} = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$ with $\widetilde{\boldsymbol{\epsilon}} \sim q(\widetilde{\boldsymbol{\epsilon}})$. Following this recipe, we derive the gradient identity in Eq. (16) for general $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ that we refer to as generalized DREGs (or GDREGs for short) identity.

Like DREGs (Eq. (10)), GDREGs allows us to transform score functions into pathwise gradients. Yet, unlike DREGs, GDREGs applies to general score functions and contains a correction term that vanishes when $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ and $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ are identical ($\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}$ term in Eq. (16)).

Note that the pathwise derivative $\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$ in Eq. (16) looks like we reparameterized an independent noise variable $\widetilde{\boldsymbol{\epsilon}}$ using $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, where the numerical value of the noise variable is given by $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})$ and $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$. We can interpret this sequence of transformations as a normalizing flow (Rezende & Mohamed, 2015) $\boldsymbol{z} \to \widetilde{\boldsymbol{\epsilon}} \to \boldsymbol{z}$, such that $\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}) = \mathcal{T}_p(\mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta});\boldsymbol{\theta}) = \boldsymbol{z}$. We can think of this procedure as *re-expressing the sample* $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ *as if it came from* $p_{\boldsymbol{\theta}}(\boldsymbol{z})$: Its numerical value $\boldsymbol{z}$ remains unchanged and it is still distributed according to $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, yet its pathwise gradient $\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$ depends on $\boldsymbol{\theta}$. We illustrate the corresponding computational flow in Fig. 2 and provide an example implementation with code in App. F

Note that to derive the GDREGs identity, we only require $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ to be reparameterizable (in Fig. 2). While $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ may be reparameterizable as well (in Fig. 2), this is not necessary; we only need to be able to evaluate its density in Eq. (16).

In the simplest case of the cross-entropy objective $\mathcal{L}^{\text{ce}} = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[\log p_{\boldsymbol{\theta}}(\boldsymbol{z})]$ (as in the ELBO with a sample-based KL estimate), $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) = 1$, and the GDREGs identity Eq. (16) gives rise to the following GDREGs estimator:

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\text{GDREGs}}\mathcal{L}^{\text{ce}} = \nabla_{\boldsymbol{z}} \log \tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} \nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})\Big|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z},\boldsymbol{\theta})} \tag{21}$$

with $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$. For Gaussian distributions $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, the cross-entropy and its gradients can be computed in closed form, which we can think of as a perfect estimator with zero bias and zero variance. Moreover, the expectation and variance of both the naive score function as well as the GDREGs estimator in Eq. (21) can be computed in closed form. We provide full derivations and a discussion of this special case in App. H as well as an example implementation in terms of (pseudo-)code in App. F. The main results are: (i) GDREGs has lower variance gradients than the score function when $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ overlap substantially, which is typically the case at the beginning of training; (ii) we can derive a closed-form control variate that depends on a ratio of the means and variances of the two distributions and that is strictly superior to the naive score function estimator and the GDREGs estimator in terms of gradient variance. However, the analytic expression for the cross-entropy has even lower (zero) gradient variance in this case.

4.1. GDREGs for VAE objectives

We can now use the GDREGs identity Eq. (16) to derive generalized doubly-reparameterized estimators for expectations of general score functions of the form $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})]$, also see Eq. (6). In App. B we derive the following GDREGs estimator of the IWAE objective w.r.t. the prior parameters $\boldsymbol{\theta}$:

GDREGs estimator for IWAE

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\text{GDREGs}}\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\text{IWAE}} = \textstyle\sum_{k=1}^{K}\big(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\text{TD}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_k) - \\ - \widetilde{w}_k^2 \nabla_{\boldsymbol{z}_k}^{\text{TD}} \log w_k\big)\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k;\boldsymbol{\theta})\Big|_{\widetilde{\boldsymbol{\epsilon}}_k=\mathcal{T}_p^{-1}(\boldsymbol{z}_k,\boldsymbol{\theta})} \tag{22}$$

with $\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. The second term in Eq. (22) looks like the DREGs estimator for $\boldsymbol{\phi}$ in Eq. (11) except that the samples $\boldsymbol{z}_k$ are now re-expressed as if they came from $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. In addition we obtain a term that involves the likelihood $p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z})$ and is linear in $\widetilde{w}_k$. Note that we do not apply GDREGs to the likelihood parameters $\boldsymbol{\lambda}$ because $p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z})$

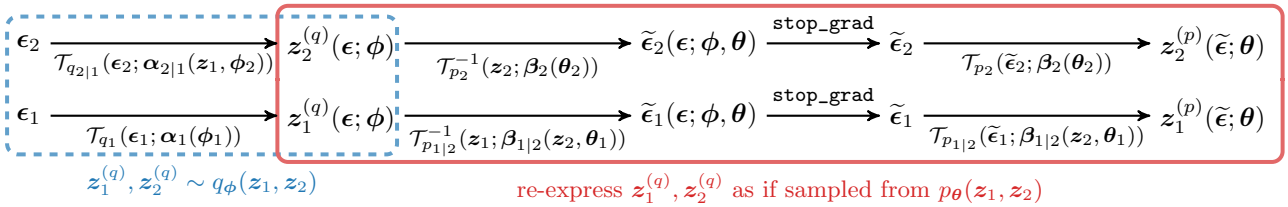

Figure 3: Computational flow to re-express samples $\boldsymbol{z}_1, \boldsymbol{z}_2$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z}_1, \boldsymbol{z}_2) = q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1) q_{\boldsymbol{\phi}_2}(\boldsymbol{z}_2|\boldsymbol{z}_1)$ as if they were sampled from $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{z}_2) = p_{\boldsymbol{\theta}_2}(\boldsymbol{z}_2) p_{\boldsymbol{\theta}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2)$. Their numerical values and distribution remain unchanged but the gradient flow through them changes. Note that $\widetilde{\boldsymbol{\epsilon}}_i$ follows a different, usually more complex, distribution from $\boldsymbol{\epsilon}_i$. $\boldsymbol{\alpha}_i$ and $\boldsymbol{\beta}_i$ denote the distribution parameters of the variatonal posterior and the prior, respectively.

is a distribution over $\boldsymbol{x}$ rather than $\boldsymbol{z}$; in the following we therefore drop the subscript $\boldsymbol{\lambda}$.

We learn all parameters by optimizing the same objective function Eq. (3), but employ different gradient estimators for different subsets of parameters. In practice, we implement these estimators using different *surrogate objectives* for the likelihood, proposal, and prior parameters, see App. E for details. While separate objectives seem computationally expensive, most terms are shared between them, and modern frameworks avoid such duplicate computation. In practice, we found the runtime increase for training with DREGs and GDREGs estimators to be smaller than 10% without any optimization of the implementation.

4.2. Extending GDREGs to hierarchical VAEs

When extending GDREGs to hierarchical models, we again encounter direct and indirect score functions (see Sec. 3). Like for the posterior parameter $\boldsymbol{\phi}$ we apply GDREGs to the direct score functions but leave the indirect score functions unchanged. The full GDREGs estimator for IWAE objectives with arbitrary hierarchial structure is given in App. C Eq. (C.16), see App. C for a derivation. In App. D we provide a worked example and in App. E we again show how to use surrogate losses to implement the estimator in practice.

To apply GDREGs we need to re-express samples from $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ as if they came from $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. We do this for the entire hierarchy jointly. In Fig. 3 we illustrate the necessary computational flow for the example of a 2-layer VAE with the variational posterior factorized in the opposite direction from the generative process; see App. C for the general case. We draw samples $\boldsymbol{z}_1, \boldsymbol{z}_2 \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_1, \boldsymbol{z}_2) = q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1) q_{\boldsymbol{\phi}_2}(\boldsymbol{z}_2|\boldsymbol{z}_1)$ (by transforming independent variables $\boldsymbol{\epsilon}_i$) and then re-express them as if they were sampled from the prior $p_{\boldsymbol{\theta}_2}(\boldsymbol{z}_2) p_{\boldsymbol{\theta}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2)$, which factorizes in the opposite direction. While the numerical values of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ remain unchanged, $\boldsymbol{z}_1$ is now dependent on $\boldsymbol{z}_2$ and both depend on the respective $\boldsymbol{\theta}$ parameters when computing gradients; we can view $(\boldsymbol{z}_1, \boldsymbol{z}_2)$ as samples that were obtained by transforming independent variables $(\widetilde{\boldsymbol{\epsilon}}_1, \widetilde{\boldsymbol{\epsilon}}_2)$ that follow a more complicated distribution than $(\boldsymbol{\epsilon}_1, \boldsymbol{\epsilon}_2)$. As in the single-layer case, only $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ needs to be reparameterizable.

5. Experiments

In this section we empirically evaluate the hierarchical extension of DREGs and its generalization to GDREGs, and compare them to the naive IWAE gradient estimator (labelled as IWAE) as well as STL (Roeder et al., 2017). First, we illustrate that DREGs and GDREGs increase the gradient signal-to-noise ratio (SNR) and reduce gradient variance compared to the naive estimator on a simple hierarchical example (Sec. 5.1); second, we show that they also reduce gradient variance in practice and improve test performance on several generative modelling tasks with VAEs with one or more stochastic layers (Sec. 5.2). We highlight that both the extension of DREGs to more than one stochastic layer as well as training the prior with GDREGs are novel contributions of this work.

5.1. Illustrative example: linear VAE

We first consider an extended version of the illustrative example by Rainforth et al. (2018) and Tucker et al. (2019) to show that hierarchical DREGs and GDREGs increase the gradient signal-to-noise ratio (SNR) and reduce gradient variance compared to the naive IWAE gradient estimator.

We consider a 2-layer linear VAE with hierarchical prior $\boldsymbol{z}_2 \sim \mathcal{N}(0, \mathbb{I})$, $\boldsymbol{z}_1|\boldsymbol{z}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_2), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\boldsymbol{z}_2))$, Gaussian noise likelihood $\boldsymbol{x}|\boldsymbol{z}_1 \sim \mathcal{N}(\boldsymbol{z}_1, \mathbb{I})$, and bottom up variational posterior $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}_1}(\boldsymbol{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}_1}(\boldsymbol{x}))$ and $q_{\boldsymbol{\phi}_2}(\boldsymbol{z}_2|\boldsymbol{z}_1) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}_2}(\boldsymbol{z}_1), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}_2}(\boldsymbol{z}_1))$. All $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i$ are linear functions, and $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{x} \in \mathbb{R}^D$. We sample 512 datapoints in $D = 5$ dimensions from a model with $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_2) = \boldsymbol{z}_2$ and $\boldsymbol{\sigma}_{\boldsymbol{\theta}}(\boldsymbol{z}_2) = 1$. We then train the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ using SGD and the IWAE objective til convergence and evaluate the gradient variance and signal-to-noise ratio for each estimator. For the proposal parameters $\boldsymbol{\phi}$ we compare DREGs to the naive score function (labelled as IWAE) and to STL; for the prior parameters $\boldsymbol{\theta}$ we compare GDREGs to IWAE.

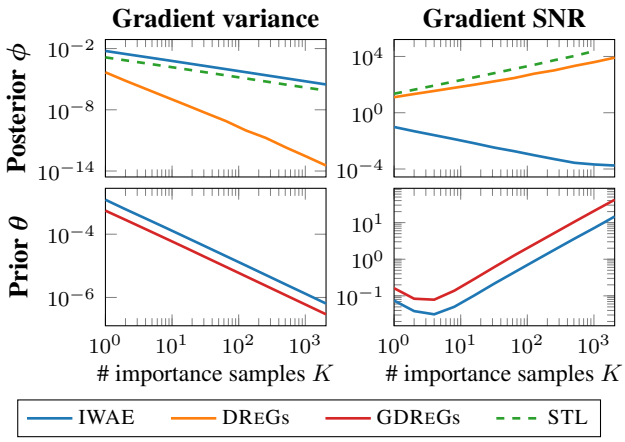

Figure 4: Average gradient variance *(left)* and signal-to-noise ratio (SNR) *(right)* for the proposal parameters ϕ *(top)* and the prior parameters θ *(bottom)*.

We find that our extension of DREGs to hierarchical models behaves qualitatively the same as in the single layer case considered by Tucker et al. (2019), see Fig. 4 *(top)*: While the SNR for the naive estimator vanishes with increasing number of importance samples, the SNR increases for DREGs. This can be explained by the faster rate with which the gradient variance decreases for DREGs compared to IWAE and STL. While STL has an even better SNR, its gradients are biased.

When considering gradients w.r.t. the prior parameters we find that the SNR is higher and the gradient variance is lower for GDREGs compared to the naive estimator (IWAE), see Fig. 4 *(bottom)*. However, they grow and shrink at the same rate for both estimators as the number of importance samples is increased.

5.2. Image modelling with VAEs

In the remainder of this paper we consider image modelling tasks with VAEs on several standard benchmark datasets: MNIST (LeCun & Cortes, 2010), Omniglot (Lake et al., 2015), and FashionMNIST (Xiao et al., 2017). We use dynamically binarized versions of all datasets to minimize overfitting.

We consider both single layer and hierarchical (multi-layer) VAEs and evaluate them on unconditional and conditional modelling tasks using the IWAE objective, Eq. (3). In the hierarchical case, the generative path (prior and likelihood) is top-down whereas the variational posterior is bottom-up, see Fig. 5 and Eq. (23) for a full description of the model and a 2-layer unconditional example. For conditional modelling we predict the bottom half of an image given its top half, as in Tucker et al. (2019); in this case, both the prior and variational posterior also depend on a context variable

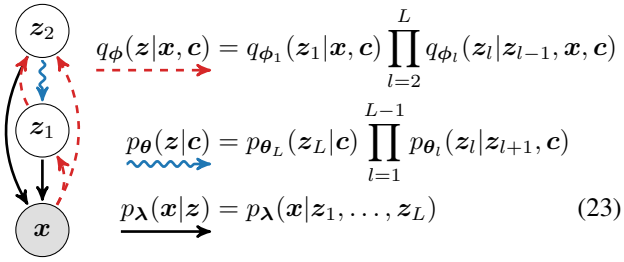

Figure 5: Model specification and 2-layer example for conditional and unconditional image modelling.

$\boldsymbol{c}$, $q_{\phi}(\boldsymbol{z}|\boldsymbol{x},\boldsymbol{c})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c})$, respectively. We use a factorized Bernoulli likelihood along with factorized Gaussians for the variational posterior and prior. Every conditional distribution in Eq. (23) is parameterized by an MLP with two hidden layers of 300 `tanh` units each, and all latent spaces have 50 dimensions. Unless stated otherwise, we train all models for 1000 epochs using the Adam optimizer (Kingma & Ba, 2015) with default learning rate of $3 \cdot 10^{-4}$, a batch size of 64, and $K = 64$ importance samples; see App. G for details.

As mentioned in Sec. 4.1, we use separate surrogate objectives to compute the gradient estimators for the likelihood, posterior, and prior parameters. While we always train the likelihood parameters $\boldsymbol{\lambda}$ on the naive IWAE objective, we consider the naive IWAE estimator (labelled as IWAE), STL, and DREGs for the variational posterior parameters ϕ, and IWAE and GDREGs for the prior parameters $\boldsymbol{\theta}$. See App. E for details on the implementation of the estimators. We present the results for conditional modelling in Tab. 1 and Fig. 6, and for unconditional modelling in Tab. 2 and Fig. 7; see App. G for more experimental results.

Estimators for the variational parameters ϕ. First, we evaluate the choice of estimator for the parameters of $q_{\phi}(\boldsymbol{z})$. Like Tucker et al. (2019) for the single layer case, we find that our extension of DREGs to hierarchical models leads to a dramatic reduction in gradient variance for the variational posterior parameters ϕ on all tasks (third column in Figs. 6 and 7), which translates to an improved test objective in all cases considered. DREGs is unbiased and typically outperforms the (biased) STL estimator. We also observed similar improvements on the training objective.

Estimators for the prior parameters $\boldsymbol{\theta}$. Second, we consider the estimators for the $\boldsymbol{\theta}$ parameters of the prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$. Using the GDREGs estimator instead of the naive IWAE estimator consistently improves the train and test objective when combined with *any* estimator for the variational posterior, especially for conditional image modelling with deeper models. For unconditional image modelling the improvements are only marginal, though using GDREGs never hurts. In terms of gradient variance for the prior parameters $\boldsymbol{\theta}$, GDREGs consistently performs better in the beginning of

		estimator ∇_ϕ^{TD}	IWAE		STL		DREGs	
		estimator $\nabla_\theta^{\text{TD}}$	IWAE	GDREGs	IWAE	IWAE	IWAE	GDREGs
MNIST	1 layer		$-38.77_{\pm 0.01}$	$-38.71_{\pm 0.02}$	$-38.76_{\pm 0.03}$	$-38.68_{\pm 0.03}$	$-38.50_{\pm 0.01}$	$\mathbf{-38.44_{\pm 0.01}}$
	2 layer		$-38.55_{\pm 0.02}$	$-38.42_{\pm 0.03}$	$-38.24_{\pm 0.02}$	$-38.14_{\pm 0.02}$	$-38.20_{\pm 0.01}$	$\mathbf{-38.02_{\pm 0.02}}$
	3 layer		$-38.63_{\pm 0.01}$	$-38.44_{\pm 0.02}$	$-38.20_{\pm 0.01}$	$-38.10_{\pm 0.02}$	$-38.20_{\pm 0.01}$	$\mathbf{-38.04_{\pm 0.01}}$
Omniglot	1 layer		$-55.84_{\pm 0.02}$	$-55.66_{\pm 0.03}$	$-55.80_{\pm 0.05}$	$-55.62_{\pm 0.05}$	$-55.34_{\pm 0.02}$	$\mathbf{-55.24_{\pm 0.02}}$
	2 layer		$-55.27_{\pm 0.03}$	$-54.98_{\pm 0.02}$	$-54.66_{\pm 0.03}$	$\mathbf{-54.28_{\pm 0.02}}$	$-54.73_{\pm 0.02}$	$-54.36_{\pm 0.03}$
	3 layer		$-55.35_{\pm 0.02}$	$-54.93_{\pm 0.02}$	$-54.64_{\pm 0.03}$	$\mathbf{-54.21_{\pm 0.03}}$	$-54.72_{\pm 0.02}$	$-54.28_{\pm 0.02}$
FMNIST	1 layer		$-102.84_{\pm 0.02}$	$-102.80_{\pm 0.02}$	$-102.99_{\pm 0.02}$	$-102.88_{\pm 0.02}$	$-102.61_{\pm 0.01}$	$\mathbf{-102.58_{\pm 0.01}}$
	2 layer		$-102.74_{\pm 0.02}$	$-102.68_{\pm 0.01}$	$-102.65_{\pm 0.02}$	$-102.48_{\pm 0.03}$	$-102.40_{\pm 0.01}$	$\mathbf{-102.30_{\pm 0.02}}$
	3 layer		$-102.86_{\pm 0.01}$	$-102.71_{\pm 0.01}$	$-102.68_{\pm 0.01}$	$-102.42_{\pm 0.02}$	$-102.46_{\pm 0.01}$	$\mathbf{-102.26_{\pm 0.01}}$

Table 1: Test objective values (higher is better) on *conditional* image modelling with a VAE model trained with IWAE. Higher is better; errorbars denote $\pm$ 1.96 standard errors ($\sigma/\sqrt{5}$) over 5 reruns.

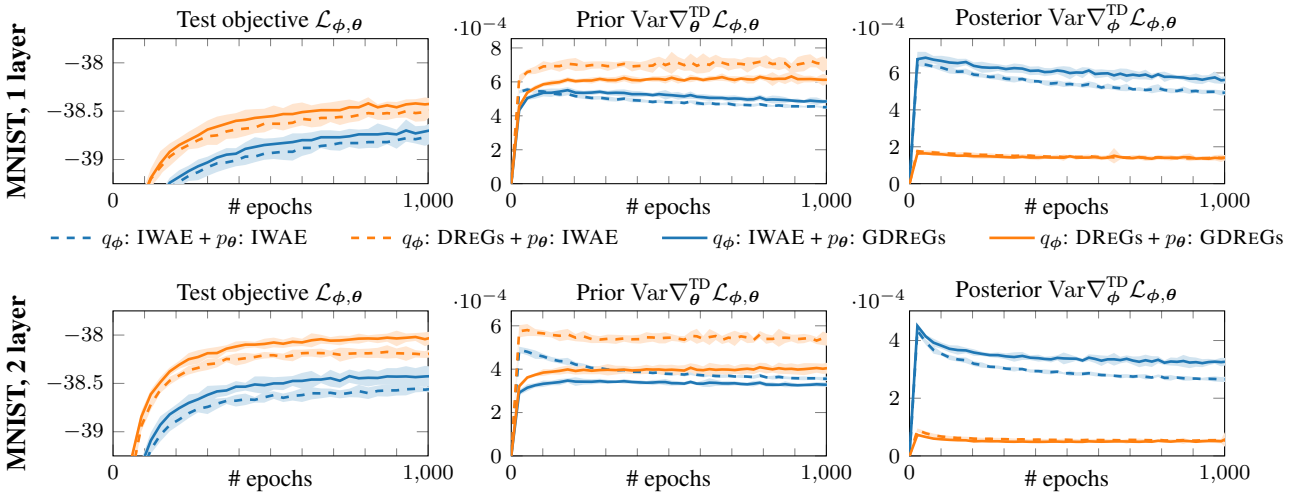

Figure 6: *Conditional* image modelling of MNIST with a VAE with 1 layer (top) and 2 layers (bottom). Shaded areas denote $\pm$ 1.96 standard deviations σ over 5 reruns.

training, when it always has lower variance. However, later in training this is only consistently true when also using the DREGs estimator for the variational posterior parameters ϕ. We hypothesize that the GDREGs estimator yields larger improvements for conditional modelling because the prior and posterior distribution are closer to each other due to the conditioning, and we saw that GDREGs works particularly well in this case for Gaussian distributions, also see App. H. To quantify this "closeness" we compared the KL of the variational posterior to the prior on the same dataset and found it to be about twice as large for unconditional modelling than for conditional modelling, see App. G.

We also note that the gradient variance for the prior parameters θ is higher when using the DREGs estimator for the variational posterior parameters ϕ, compared to the naive IWAE estimator (compare orange and blue lines in the middle column of Figs. 6 and 7). This is an indirect effect of altered learning dynamics. We suspect that better posterior gradient estimates with DREGs lead to generative models that fit the data better, which in turn results in larger gradient variance for the prior. This effect is absent in the illustrative example in Fig. 4 because we evaluate the gradient variance on the same fixed model for all estimators. In App. G.3 we compare the estimators *offline* for different combinations of estimators during training. The results are in line with our online results in this section: for the gradients of the variational posterior the DREGs estimator *always* has lower variance than the naive (IWAE) estimator; for the gradients of the prior the GDREGs estimator typically has lower variance, though in some cases only in the beginning of training.

6. Related work

Roeder et al. (2017) observed that the reparameterization gradient estimator for the ELBO contains a score function term and proposed the STL estimator that simply drops this term to reduce the estimator variance. They considered

	estimator ∇_ϕ^{TD}	IWAE		STL		DREGs	
	estimator $\nabla_\theta^{\text{TD}}$	IWAE	GDREGs	IWAE	GDREGs	IWAE	GDREGs
MNIST	2 layer	$-86.07_{\pm 0.02}$	$-86.04_{\pm 0.03}$	$-85.29_{\pm 0.02}$	$\mathbf{-85.23_{\pm 0.03}}$	$\mathbf{-85.25_{\pm 0.02}}$	$-85.32_{\pm 0.02}$
	3 layer	$-85.69_{\pm 0.02}$	$-85.70_{\pm 0.02}$	$-85.01_{\pm 0.03}$	$-84.94_{\pm 0.05}$	$\mathbf{-84.87_{\pm 0.03}}$	$\mathbf{-84.90_{\pm 0.04}}$
Omniglot	2 layer	$-105.20_{\pm 0.02}$	$-105.11_{\pm 0.02}$	$-104.10_{\pm 0.05}$	$\mathbf{-104.00_{\pm 0.05}}$	$-104.12_{\pm 0.05}$	$\mathbf{-104.05_{\pm 0.04}}$
	3 layer	$-104.68_{\pm 0.02}$	$-104.71_{\pm 0.03}$	$-104.02_{\pm 0.02}$	$\mathbf{-103.55_{\pm 0.03}}$	$-104.71_{\pm 0.03}$	$\mathbf{-103.51_{\pm 0.06}}$
FMNIST	2 layer	$-230.65_{\pm 0.03}$	$-230.61_{\pm 0.02}$	$-230.14_{\pm 0.02}$	$\mathbf{-229.98_{\pm 0.02}}$	$-230.04_{\pm 0.03}$	$\mathbf{-229.98_{\pm 0.03}}$
	3 layer	$-230.60_{\pm 0.03}$	$-230.59_{\pm 0.03}$	$-230.26_{\pm 0.04}$	$-229.92_{\pm 0.03}$	$-229.92_{\pm 0.02}$	$\mathbf{-229.87_{\pm 0.03}}$

Table 2: Test objective values on *unconditional* image modelling with a VAE model trained with IWAE.

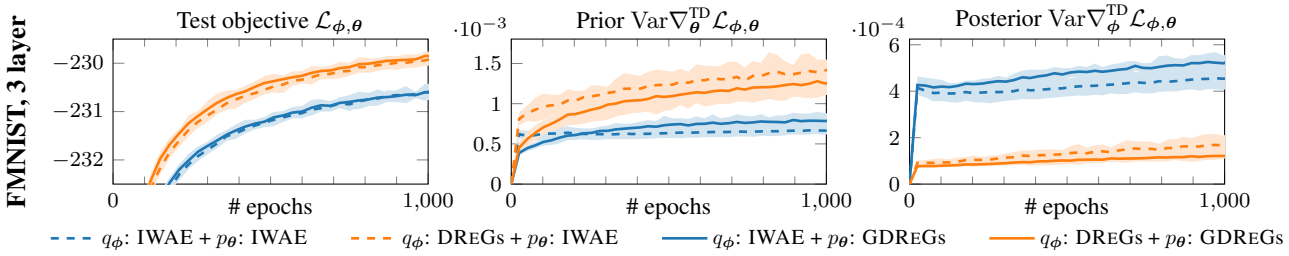

Figure 7: *Unconditional* image modelling on FashionMNIST; 3 layers.

hierarchical ELBO models but do not discuss how to treat indirect score functions. While the STL estimator is unbiased for the ELBO objective, Tucker et al. (2019) showed that it is biased for more general objectives such as the IWAE. They proposed the DREGs estimator that yields unbiased and low variance gradients for IWAE and resolves the diminishing signal-to-noise issue of the naive IWAE gradients first discussed by Rainforth et al. (2018). We extend DREGs to hierarchical models, discuss how to treat the indirect score functions, and generalize it to general score functions by introducing GDREGs.

Several classic techniques from the variance reduction literature have been applied to variational inference and reparameterization. For example, Miller et al. (2017) and Geffner & Domke (2020) proposed control variates for reparameterization gradients; Ruiz et al. (2016) used importance sampling with a proposal optimized to reduce variance. Such approaches are orthogonal to methods such as (G)DREGs and STL, and can be combined with them for greater variance reduction (Agrawal et al., 2020).

7. Conclusion

In this paper we generalized the recently proposed doubly-reparameterized gradients (DREGs, Tucker et al. (2019)) estimator for variational objectives in two ways. First, we showed that for hierarchical models such as VAEs seemingly pathwise gradients can actually contain score functions, and how to consistently and effectively extend DREGs in this case. Second, we introduced GDREGs, a doubly-reparameterized gradient estimator that applies to general score functions, while DREGs is limited to score functions of the variational distribution. Finally, we demonstrated that both generalizations can improve performance on conditional and unconditional image modelling tasks.

While we present and discuss the GDREGs estimator in the context of deep probabilistic models, it applies generally to score function gradients of the form $\mathbb{E}_{q_\phi(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$. Applying it to other problem settings of this type such as normalizing flows is an exciting area of future research.

Acknowledgements

We thank Chris Maddison as well as the anonymous reviewers for feedback on the manuscript.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, 2016.

Agrawal, A., Sheldon, D. R., and Domke, J. Advances in black-box VI: normalizing flows, importance weighting, and optimization. In *Advances in Neural Information Processing Systems 33*, 2020.

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 2017.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary,

C., Maclaurin, D., and Wanderman-Milne, S. JAX: composable transformations of Python+NumPy programs, 2018.

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

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M. D., and Saurous, R. A. Tensorflow distributions. *arXiv preprint arXiv:1711.10604*, 2017.

Geffner, T. and Domke, J. Approximation based variance reduction for reparameterization gradients. In *Advances in Neural Information Processing Systems 33*, 2020.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Rio, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. Array programming with numpy. *Nature*, Sep 2020.

Hennigan, T., Cai, T., Norman, T., and Babuschkin, I. Haiku: Sonnet for JAX, 2020.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. *An Introduction to Variational Methods for Graphical Models*. MIT Press, Cambridge, MA, USA, 1999.

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

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

Lake, B. M., Salakhutdinov, R., and Tenenbaum, J. B. Human-level concept learning through probabilistic program induction. *Science*, 2015.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Miller, A., Foti, N., D'Amour, A., and Adams, R. P. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*, 2017.

Mohamed, S., Rosca, M., Figurnov, M., and Mnih, A. Monte carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 2020.

Rainforth, T., Kosiorek, A., Le, T. A., Maddison, C., Igl, M., Wood, F., and Teh, Y. W. Tighter variational bounds are not necessarily better. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Rezende, D. J. and Mohamed, S. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, 2014.

Roeder, G., Wu, Y., and Duvenaud, D. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In *Advances in Neural Information Processing Systems 30*, 2017.

Ruiz, F. J. R., Titsias, M. K., and Blei, D. M. Overdispersed black-box variational inference. In *Uncertainty in Artificial Intelligence*, 2016.

Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

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

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.