Generalized Doubly-Reparameterized Gradient Estimators

Matthias Bauer MSBAUER@GOOGLE.COM
Andriy Mnih AMNIH@GOOGLE.COM
DeepMind, London, UK

1. Introduction

In probabilistic machine learning, we are often interested in optimizing 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 widely used variational autoencoders (VAEs, Kingma and Welling (2014), Rezende et al. (2014)), $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ is the variational posterior and $\boldsymbol{\theta}$ denotes the model parameters.

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 $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$. 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 gradient variance; however, if f depends on $\boldsymbol{\phi}$ only through $\boldsymbol{z}$, we can use reparameterization (Kingma and 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 for the second time, targeting the score function terms. The DReGs estimator has, however, 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 and introduce GDReGs, a generalized doubly-reparameterized gradient estimator that applies to general score functions; we also extend the DReGs estimator to hierarchical models and show that both estimators improve performance on conditional and unconditional image modelling tasks.

2. Background

We are interested in computing gradients of variational objectives $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]$ w.r.t. the variational parameters $\boldsymbol{\phi}$ of the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, and parameters $\boldsymbol{\theta}$ of a second distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$, such as a learnable prior. Here $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ is a general function that can depend on both $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ explicitly.

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

$$\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{z}_1,\ldots,\boldsymbol{z}_K\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K} w_k\right)\right] \qquad w_k = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k)p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \tag{1}$$

Gradient estimation. In practise, we approximate the expectation in $\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}$ by Monte Carlo sampling, so that our estimates of the expectation and of its gradients become random variables. We can distinguish between two types of gradient estimators in this setting: (i) *score function (SF) estimators* and (ii) *pathwise estimators*. Score functions are gradients of a log probability density w.r.t. its parameters, for example $\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$; SF estimators treat the function $f_{\boldsymbol{\phi},\boldsymbol{\theta}}$ 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 $\boldsymbol{z}$ of the function $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ and differentiate the computation path to often achieve lower variance gradients by using the knowledge of $\nabla_{\boldsymbol{z}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$; see Mohamed et al. (2020) for a recent review.

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

$$\nabla^{\text{TD}}_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\underline{\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})} + f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\underline{\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})}\right] \tag{2}$$

$$\nabla^{\text{TD}}_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\underline{\nabla_{\boldsymbol{\theta}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})}\right], \tag{3}$$

and both can give rise to score functions (note $\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) = \nabla_{\log q_{\boldsymbol{\phi}}(\boldsymbol{z})} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \underline{\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})}$).

In the following, we recapitulate how to address the score functions in Eq. (2) using the reparameterization trick and doubly-reparameterized gradients (DReGs, Tucker et al. (2019)), respectively. In Sec. 3 we introduce GDReGs, a generalization of DReGs, that allows us to eliminate the score function in Eq. (3).

Reparameterization. We can use the *reparameterization trick* (Kingma and Welling, 2014; Rezende et al., 2014) to turn the score function, $\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})$, inside the expectation in Eq. (2) into a pathwise derivative of the function $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ as follows: we express the latent variables $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$ through a bijection of new random variables $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$, which are independent of $\boldsymbol{\phi}$, $\boldsymbol{z} = \mathcal{T}_q(\boldsymbol{\epsilon}; \boldsymbol{\phi}) \Leftrightarrow \boldsymbol{\epsilon} = \mathcal{T}_q^{-1}(\boldsymbol{z}; \boldsymbol{\phi})$. This allows us to rewrite expectations w.r.t. $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ as $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] = \mathbb{E}_{q(\boldsymbol{\epsilon})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathcal{T}_q(\boldsymbol{\epsilon}; \boldsymbol{\phi}))\right]$, which moves the parameter dependence into the argument of $f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ and gives rise to a pathwise gradient:

$$\nabla^{\text{TD}}_{\boldsymbol{\phi}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] = \mathbb{E}_{q(\boldsymbol{\epsilon})}\left[\underline{\nabla_{\boldsymbol{\phi}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})} + \underline{\nabla_{\boldsymbol{z}} f_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\phi}} \mathcal{T}_q(\boldsymbol{\epsilon}; \boldsymbol{\phi})}\right]_{\boldsymbol{z}=\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi})}. \tag{4}$$

Double reparameterization. Tucker et al. (2019) reduce gradient variance by replacing the remaining score function in Eq. (4) with its reparameterized counterpart. Double reparameterization is based on the identity (Eq. (5) in Tucker et al. (2019))

$$\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\phi}} \log q_{\boldsymbol{\phi}}(\boldsymbol{z})\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim q(\boldsymbol{\epsilon})}\left[\nabla_{\boldsymbol{z}}^{\mathrm{TD}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\big|_{\boldsymbol{z}=\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi})} \nabla_{\boldsymbol{\phi}} \mathcal{T}_q\left(\boldsymbol{\epsilon};\boldsymbol{\phi}\right)\right] \tag{5}$$

which follows from the fact that both the score function and the reparameterization estimators are unbiased and thus equal in expectation. For the IWAE objective Eq. (1), Tucker et al. (2019) derived the following doubly-reparametererized gradients (DReGs) estimator:

$$\widehat{\nabla}_{\boldsymbol{\phi}}^{\text{DReGs}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\text{IWAE}} = \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log w_k \nabla_{\boldsymbol{\phi}} \mathcal{T}_q(\boldsymbol{\epsilon}_k;\boldsymbol{\phi}); \qquad \boldsymbol{\epsilon}_{1:K} \sim q(\boldsymbol{\epsilon}); \qquad \widetilde{w}_k = \tfrac{w_k}{\sum_{k'=1}^{K} w_{k'}}. \tag{6}$$

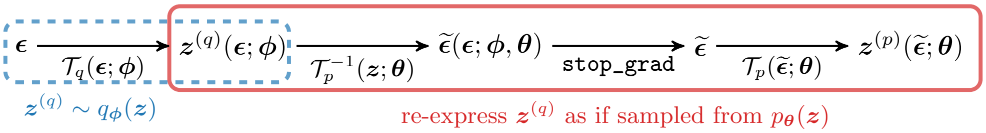
While the DReGs estimator reparameterizes the score function in Eq. (4), the previously proposed STL estimator (Roeder et al., 2017) simply drops it and is usually biased as a result. Crucially, because DReGs relies on reparameterization, it is limited to score functions of the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, making it inapplicable in the more general setting of arbitrary score functions (Eq. (3)). In the following Sec. 3 we introduce the GDReGs estimator that can be applied to these more general score function terms.

In App. C we discuss that the seemingly pathwise gradient in Eq. (4) can actually contain score functions for hierarchical models and explain how to extend DReGs to this case. Additional score functions arise because the distribution parameters (e.g. the mean and covariance) of one stochastic layer depend on the latent variables of previous layers.

3. Generalized DReGs

Here, we generalize DReGs to score function terms that involve distributions $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ different from the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, such as $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$. Such terms appear, for example, when training a VAE with a trainable prior $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ with the ELBO or IWAE objectives. We cannot use DReGs directly here, as it relies on reparameterization of the sampling distribution $q_{\boldsymbol{\phi}}(\boldsymbol{z})$, which means the path would then depend on the parameters $\boldsymbol{\phi}$ and double-reparameterization would only apply to its parameters $\boldsymbol{\phi}$, 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 is to: ⓐ *temporarily* change the path such that it depends on $\boldsymbol{\theta}$; ⓑ perform the reparameterized gradient computation; ⓒ change



$\boldsymbol{\epsilon}$
$\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi})$
$\boldsymbol{z}^{(q)}(\boldsymbol{\epsilon};\boldsymbol{\phi})$
$\mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})$
$\widetilde{\boldsymbol{\epsilon}}(\boldsymbol{\epsilon};\boldsymbol{\phi},\boldsymbol{\theta})$
stop_grad
$\widetilde{\boldsymbol{\epsilon}}$
$\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$
$\boldsymbol{z}^{(p)}(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$
$\boldsymbol{z}^{(q)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$
re-express $\boldsymbol{z}^{(q)}$ as if sampled from $p_{\boldsymbol{\theta}}(\boldsymbol{z})$

Figure 1: 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 $\boldsymbol{\theta}$: $\nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z},\boldsymbol{\theta})}$. Note that $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\mathcal{T}_q(\boldsymbol{\epsilon};\boldsymbol{\phi});\boldsymbol{\theta})$ has a different, usually more complex, distribution than $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$.

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 applying reparameterization to the new sampling distribution $p_{\boldsymbol{\theta}}(\boldsymbol{z})$: $\boldsymbol{z} = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})$ with $\widetilde{\boldsymbol{\epsilon}} \sim q(\widetilde{\boldsymbol{\epsilon}})$. We derive the gradient identity in Eq. (7) for a general $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$; we refer to it as the generalized DReGs (GDReGs) identity.

GDReGs identity

$$\boxed{\mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}[g(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})] = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}}\Big[\Big(g(\boldsymbol{z})\nabla_{\boldsymbol{z}}^{\mathrm{TD}} \log \tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \nabla_{\boldsymbol{z}}^{\mathrm{TD}} g(\boldsymbol{z})\Big)\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\boldsymbol{\epsilon};\boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z},\boldsymbol{\theta})}\Big]} \tag{7}$$

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{TD}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})] \overset{ⓐ}{=} \nabla_{\boldsymbol{\theta}}^{\mathrm{TD}}\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[\tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] \overset{ⓐ}{=} \nabla_{\boldsymbol{\theta}}^{\mathrm{TD}}\mathbb{E}_{q(\widetilde{\boldsymbol{\epsilon}})}\left[\tfrac{q_{\boldsymbol{\phi}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))}{p_{\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))\right] \tag{8}$$

$$\overset{ⓑ}{=} \mathbb{E}_{q(\widetilde{\boldsymbol{\epsilon}})}\left[\nabla_{\boldsymbol{z}}^{\mathrm{TD}}\left(\tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} g(\boldsymbol{z})\right)\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}) + \tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left(\nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) - g(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right)\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \tag{9}$$

$$\overset{ⓒ}{=} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\left(g(\boldsymbol{z})\nabla_{\boldsymbol{z}}^{\mathrm{TD}} \log \tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \nabla_{\boldsymbol{z}}^{\mathrm{TD}} g(\boldsymbol{z})\right)\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})} + \nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) - g(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right] \tag{10}$$

In the derivation we have used the identity $x\nabla_* \log x = \nabla_* x$ repeatedly. By noting that $\nabla_{\boldsymbol{\theta}}^{\mathrm{TD}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[\nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})]$, we can cancel these terms on the left hand side of Eq. (8) and right hand side of Eq. (10). By moving $-\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})]$ to the other side we obtain the desired result.

Similar to DReGs (Eq. (5)), GDReGs allows us to transform score function gradients into pathwise gradients. But, 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 ($\log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}$ in Eq. (7)). Note that the pathwise derivative $\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$ in Eq. (7) looks like a reparameterization of $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ in terms of a noise variable $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})$ with $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$. We can interpret this sequence of transformations as a normalizing flow (Rezende and 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. 1. Note that to derive the GDReGs identity, we only require $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ to be reparameterizable (▭ in Fig. 1). While $q_{\boldsymbol{\phi}}(\boldsymbol{z})$ may be reparameterizable as well (⬚ in Fig. 1), this is not necessary; we only need to be able to evaluate its density in Eq. (7).

We use the GDReGs identity to address general score functions of the form Eq. (3) and derive the GDReGs estimator for the IWAE objective w.r.t. the prior parameters $\boldsymbol{\theta}$:

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{GDReGs}}\mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \sum_{k=1}^{K}\left(\widetilde{w}_k\nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_k) - \widetilde{w}_k^2\nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log w_k\right)\nabla_{\boldsymbol{\theta}}\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k;\boldsymbol{\theta})\Big|_{\widetilde{\boldsymbol{\epsilon}}_k=\mathcal{T}_p^{-1}(\boldsymbol{z}_k,\boldsymbol{\theta})} \tag{11}$$

with $\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$; full derivation is in App. D. The second term in Eq. (11) looks like the DReGs estimator for $\boldsymbol{\phi}$ in Eq. (6) 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 $\widetilde{w}_k$. We do not apply GDReGs to the likelihood parameters $\boldsymbol{\lambda}$ because $p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z})$ is a distribution over $\boldsymbol{x}$ rather than $\boldsymbol{z}$; in the following we therefore drop the subscript $\boldsymbol{\lambda}$.

Similarly to the DReGs estimator, we can also extend the GDReGs estimator to hierarchical objectives as we explain in App. C.

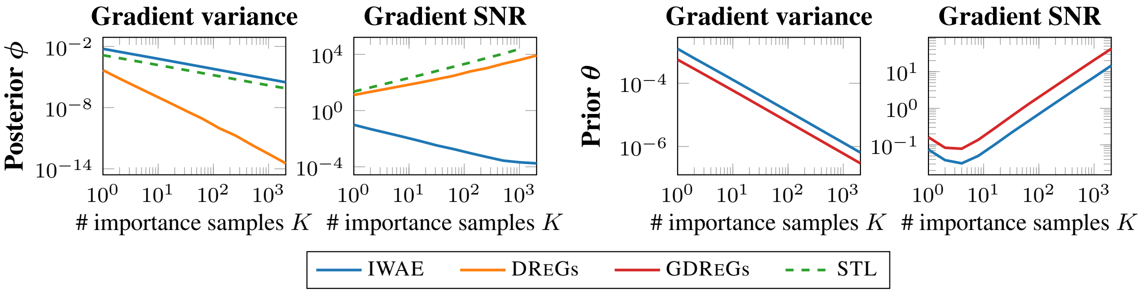


Figure 2: Average gradient variance and gradient signal-to-noise ratio (SNR) for the variational posterior parameters $\boldsymbol{\phi}$ and the prior parameters $\boldsymbol{\theta}$.

We learn all parameters by optimizing the same objective Eq. (1), 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. F for details. While separate objectives might seem computationally expensive, most terms are shared between them, and modern frameworks would avoid such duplicate computation. In practise, we found the runtime increase for training with DREGS and GDREGS estimators to be less than 10% without any optimization of the implementation.

4. Experiments

Here, we empirically evaluate the proposed GDREGS estimator as well as the proposed hierarchical extensions of DREGS and GDREGS, and compare them to the naive IWAE gradient estimator (labelled as IWAE) as well as STL (Roeder et al., 2017).

Illustrative example. We first consider an extended version of the illustrative example introduced 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 and variational posterior and find that (see Fig. 2): (i) for the parameters $\boldsymbol{\phi}$ of the variational posterior, our extended version of the DREGS estimator also resolves the vanishing SNR problem of the naive IWAE estimator (Rainforth et al., 2018) in the hierarchical case by reducing the gradient variance at a faster rate with the number of importance samples; (ii) for the paramters $\boldsymbol{\theta}$ of the prior, the GDREGS estimator has smaller gradient variance and better SNR than the naive IWAE estimator but scales at the same rate with the number of importance samples.

Image modelling with VAEs. We also consider conditional and unconditional image modelling tasks with single layer and hierarchical (multi-layer) VAEs on several standard benchmark datasets: MNIST (LeCun and Cortes, 2010), Omniglot (Lake et al., 2015), and FashionMNIST (Xiao et al., 2017). We use the dynamically binarized versions of the datasets to minimize overfitting. In the hierarchical case, the generative path (prior and likelihood) is top-down whereas the variational posterior is bottom-up; for conditional modelling we predict the bottom half of an image given its top half, as in Tucker et al. (2019); in this

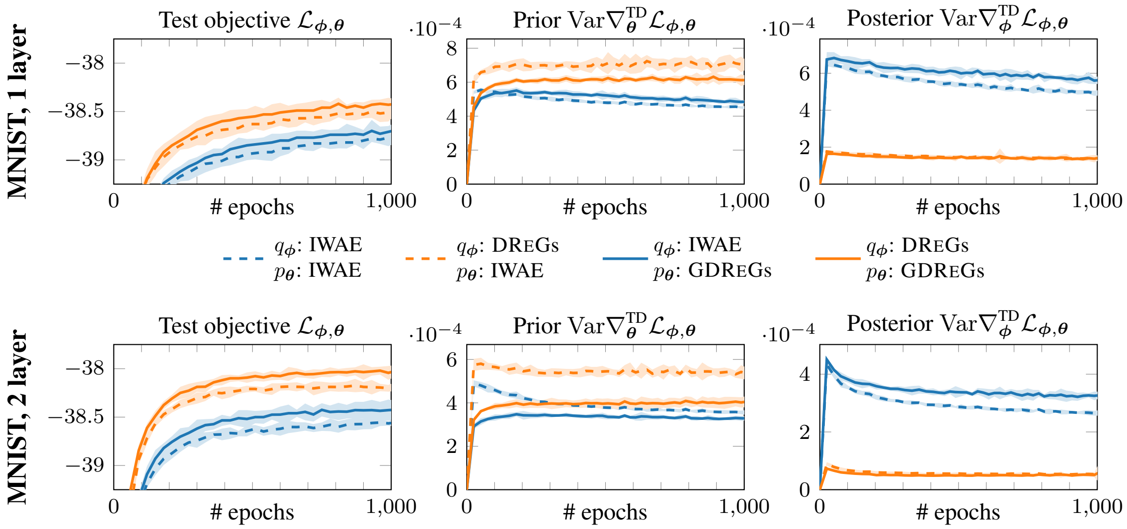


Figure 3: *Conditional* image modelling of MNIST with a VAE with 1 layer (top) and 2 layers (bottom). Shaded areas denote ± 1.96 standard deviations σ over 5 reruns.

case, both the prior and variational posterior additionally depend on a context variable $\boldsymbol{c}$ ($q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c})$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c})$, respectively).

First, we evaluate the choice of estimator for the parameters of $q_{\boldsymbol{\phi}}(\boldsymbol{z})$. 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 $\boldsymbol{\phi}$ on all tasks (third column in Fig. 3), 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.

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 performance when combined with *any* estimator for the variational posterior, especially for conditional image modelling with deeper models. For unconditional image modelling the improvements are 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 training, when it always has lower variance. Later in training this is only consistently true when also using the DReGs estimator for the variational posterior parameters $\boldsymbol{\phi}$.

5. 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 to 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 lead to better train and test performance on conditional and unconditional image modelling tasks.

Acknowledgments

We thank Chris Maddison for useful comments on the manuscript.

References

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

Tomas Geffner and Justin Domke. Approximation based variance reduction for reparameterization gradients. *arXiv preprint arXiv:2007.14634*, 2020.

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

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

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Andrew Miller, Nick Foti, Alexander D'Amour, and Ryan P Adams. Reducing reparameterization gradient variance. In *Advances in Neural Information Processing Systems*, 2017.

Shakir Mohamed, Mihaela Rosca, Michael Figurnov, and Andriy Mnih. Monte Carlo gradient estimation in machine learning. *Journal of Machine Learning Research*, 2020.

Tom Rainforth, Adam Kosiorek, Tuan Anh Le, Chris Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

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

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

George Tucker, Dieterich Lawson, Shixiang Gu, and Chris J. Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

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

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.

Appendix A. Additional results

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

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

	estimator $\nabla_\phi^{\mathrm{TD}}$	IWAE		STL		DReGs	
	estimator $\nabla_\theta^{\mathrm{TD}}$	**IWAE**	**GDReGs**	**IWAE**	**GDReGs**	**IWAE**	**GDReGs**
MNIST	2 layer	−86.07±0.02	−86.04±0.03	−85.29±0.02	**−85.23±0.03**	**−85.25±0.02**	−85.32±0.02
	3 layer	−85.69±0.02	−85.70±0.02	−85.01±0.03	−84.94±0.05	**−84.87±0.03**	**−84.90±0.04**
Omniglot	2 layer	−105.20±0.02	−105.11±0.02	−104.10±0.05	−104.00±0.05	−104.12±0.05	**−104.05±0.04**
	3 layer	−104.68±0.02	−104.71±0.03	−104.02±0.02	**−103.55±0.03**	−104.71±0.03	**−103.51±0.06**
FMNIST	2 layer	−230.65±0.03	−230.61±0.02	−230.14±0.02	**−229.98±0.02**	−230.04±0.03	**−229.98±0.03**
	3 layer	−230.60±0.03	−230.59±0.03	−230.26±0.04	−229.92±0.03	−229.92±0.02	**−229.87±0.03**

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

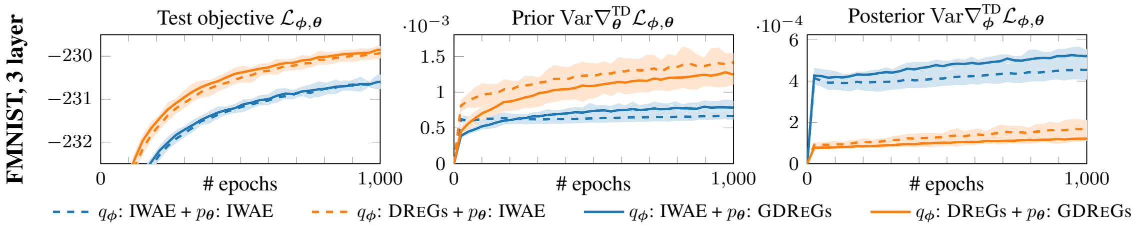


Figure A.1: *Unconditional* image modelling on FashionMNIST; 3 layers.

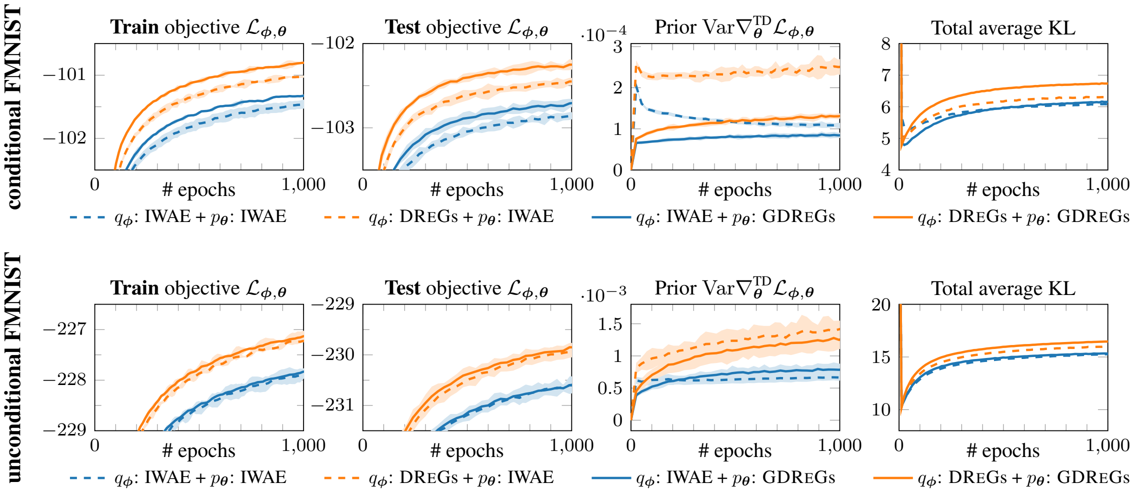


Figure A.2: Train objective *(leftmost column)* and total average KL *(rightmost column)* in addition to the test objective and prior gradient variance for conditional and unconditional models with 3 stochastic layers on FashionMNIST. We find that the KL is lower for conditional models than unconditional models, which indicates that the variational posterior and the prior are closer to each other in this case.

Appendix B. 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 obtained by dropping this score function to reduce the estimator variance. They applied the estimator to hierarchical ELBO models but did 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 bound. 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.

A number of classic techniques from the variance reduction literature have also been applied to variational inference and reparameterization. For example, Miller et al. (2017) and Geffner and Domke (2020) proposed control variates for reparameterization gradients; while 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.

Appendix C. DReGs and GDReGs for hierarchical models

We now consider models with hierarchically structured latent variables and show that even terms that look like pathwise gradients, such as $\nabla_{\boldsymbol{z}}^{\mathrm{TD}} f_{\phi,\theta}(\boldsymbol{z})$ in Eq. (4), can give rise to score functions in this case. *The score functions appear because the distribution parameters of one stochastic layer depend on the latent variables of another layer.*

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 the subsequent level (we omit the true pathwise gradients $(\dots)$):

$$\nabla^{\mathrm{TD}}_{\boldsymbol{z}_2} \log q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2) = \nabla^{\mathrm{TD}}_{\boldsymbol{z}_2} \log q_{\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2,\boldsymbol{\phi}_1)}(\boldsymbol{z}_1) \tag{C.1}$$

$$= \underline{\nabla_{\boldsymbol{\alpha}_{1|2}} \log q_{\boldsymbol{\alpha}_{1|2}}(\boldsymbol{z}_1)} \nabla_{\boldsymbol{z}_2} \boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2, \boldsymbol{\phi}_1) + \dots . \tag{C.2}$$

Similar additional score functions arise for seemingly pathwise gradients of hierarchical and/or autoregressive priors and variational posteriors.

C.1. 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. (1)), 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 $\boldsymbol{z}_2 \rightarrow \boldsymbol{z}_1 \rightarrow \boldsymbol{x}$ and examine the term $\nabla^{\mathrm{TD}}_{\boldsymbol{\phi}_2} \log q_{\boldsymbol{\phi}_1,\boldsymbol{\phi}_2}(\boldsymbol{z}_1, \boldsymbol{z}_2)$ 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 $\boldsymbol{z}_1(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2) \equiv \mathcal{T}_{q_1}(\boldsymbol{\epsilon}_1; \boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_2), \boldsymbol{\phi}_1))$:

$$\nabla^{\mathrm{TD}}_{\boldsymbol{\phi}_2} \log q_{\underline{\boldsymbol{\alpha}_2(\boldsymbol{\phi}_2)}} \left(\underline{\boldsymbol{z}_2(\boldsymbol{\phi}_2)}\right) q_{\underline{\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_2), \boldsymbol{\phi}_1)}} \left(\underline{\boldsymbol{z}_1(\boldsymbol{\phi}_1, \boldsymbol{\phi}_2)}\right) \tag{C.3}$$

The total derivative w.r.t. parameters of the upper layer, $\boldsymbol{\phi}_2$, gives rise to three types of gradients: (true) pathwise gradients w.r.t. $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, a *direct* score function because the distribution parameters $\boldsymbol{\alpha}_2(\boldsymbol{\phi}_2)$ directly depend on $\boldsymbol{\phi}_2$, and an *indirect* score function because $\boldsymbol{\alpha}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_2), \boldsymbol{\phi}_1)$ indirectly depends on $\boldsymbol{\phi}_2$ through $\boldsymbol{z}_2$. Other terms in the gradient of the objective w.r.t. $\boldsymbol{\phi}$ as well as gradients w.r.t. the $\boldsymbol{\theta}$ parameters decompose similarly. We have three options to estimate each score function individually: (1) leave it—this naive estimator is unbiased but potentially has high variance; (2) drop it, similar to STL—this estimator is generally biased; (3) doubly-reparameterize it using DReGs—this estimator is unbiased, but can generate further score function terms.

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

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

variance for DReGs. Moreover, for most model structures it is impossible to consistently replace all successively arising score functions by doubly-reparameterized gradients. Thus, to extend DReGs to hierarchical models, we leave the indirect score functions unchanged and only doubly reparameterize the direct score functions. We provide detailed derivations and a general DReGs estimator for arbitrary hierarchical structures in App. E, and show how to implement the corresponding surrogate loss functions in App. F.

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 dropping the direct score functions while maintaining the indirect ones, similar to how we extend DReGs to hierarchical models; the STL estimator is biased for IWAE objectives (Tucker et al., 2019).

C.2. Extending GDReGs to hierarchical VAEs

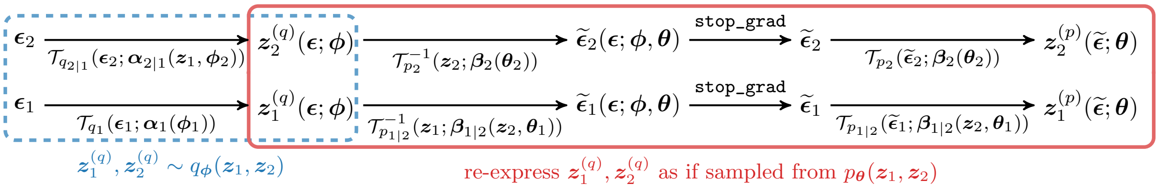

$\boldsymbol{\epsilon}_2$
$\mathcal{T}_{q_{2|1}}(\boldsymbol{\epsilon}_2; \boldsymbol{\alpha}_{2|1}(\boldsymbol{z}_1, \boldsymbol{\phi}_2))$
$\boldsymbol{z}_2^{(q)}(\boldsymbol{\epsilon}; \boldsymbol{\phi})$
$\mathcal{T}_{p_2}^{-1}(\boldsymbol{z}_2; \boldsymbol{\beta}_2(\boldsymbol{\theta}_2))$
$\widetilde{\boldsymbol{\epsilon}}_2(\boldsymbol{\epsilon}; \boldsymbol{\phi}, \boldsymbol{\theta})$
stop_grad
$\widetilde{\boldsymbol{\epsilon}}_2$
$\mathcal{T}_{p_2}(\widetilde{\boldsymbol{\epsilon}}_2; \boldsymbol{\beta}_2(\boldsymbol{\theta}_2))$
$\boldsymbol{z}_2^{(p)}(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})$
$\boldsymbol{\epsilon}_1$
$\mathcal{T}_{q_1}(\boldsymbol{\epsilon}_1; \boldsymbol{\alpha}_1(\boldsymbol{\phi}_1))$
$\boldsymbol{z}_1^{(q)}(\boldsymbol{\epsilon}; \boldsymbol{\phi})$
$\mathcal{T}_{p_{1|2}}^{-1}(\boldsymbol{z}_1; \boldsymbol{\beta}_{1|2}(\boldsymbol{z}_2, \boldsymbol{\theta}_1))$
$\widetilde{\boldsymbol{\epsilon}}_1(\boldsymbol{\epsilon}; \boldsymbol{\phi}, \boldsymbol{\theta})$
stop_grad
$\widetilde{\boldsymbol{\epsilon}}_1$
$\mathcal{T}_{p_{1|2}}(\widetilde{\boldsymbol{\epsilon}}_1; \boldsymbol{\beta}_{1|2}(\boldsymbol{z}_2, \boldsymbol{\theta}_1))$
$\boldsymbol{z}_1^{(p)}(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})$
$\boldsymbol{z}_1^{(q)}, \boldsymbol{z}_2^{(q)} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_1, \boldsymbol{z}_2)$
re-express $\boldsymbol{z}_1^{(q)}, \boldsymbol{z}_2^{(q)}$ as if sampled from $p_{\boldsymbol{\theta}}(\boldsymbol{z}_1, \boldsymbol{z}_2)$

Figure C.1: 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.

When extending GDReGs to hierarchical models, we again encounter direct and indirect score functions (see App. C), and we apply GDReGs to the direct score functions but leave the indirect score functions. See App. E for derivations of a general GDReGs estimator for arbitrary hierarchical structures and App. F for surrogate losses to implement them.

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. C.1 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. E 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.

Appendix D. Derivation of the GDReGs estimator for the IWAE objective

In this section we apply the GDReGs identity derived above to derive the GDReGs estimator for the IWAE objective, Eq. (11) in the main paper.

D.1. Preliminaries on the IWAE objective

The importance weighted autoencoder (IWAE) objective is given by

$$\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{z}_{1:K}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})}\left[\log\left(\frac{1}{K}\sum_{k=1}^{K} w_k\right)\right] \qquad w_k = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z})p(\boldsymbol{x}|\boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \tag{D.1}$$

where w_k are the importance weights (Burda et al., 2016).

Due to the structure of the IWAE objective, any gradient w.r.t. any of its parameters can be written as

$$\nabla^{\text{TD}}_{*}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon})}\left[\sum_{k=1}^{K}\widetilde{w}_k\nabla^{\text{TD}}_{*}\log w_k\right]; \qquad \widetilde{w}_k = \frac{w_k}{\sum_j w_j} \tag{D.2}$$

using the chain rule and $\nabla_* w_k = w_k\nabla_*\log w_k$. $\widetilde{w}_k$ are the normalized importance weights, and we have reparameterized $\boldsymbol{z}_k$ as $\mathcal{T}_q\left(\boldsymbol{\epsilon}_k;\boldsymbol{\phi}\right)$. Typically, the derivatives we are interested in are w.r.t. the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$.

We also note the following identity that we use in the derivation of the doubly reparameterized estimators,

$$\nabla^{\text{TD}}_{\boldsymbol{z}}\widetilde{w}_k = \left(\widetilde{w}_k - \widetilde{w}_k^2\right)\nabla^{\text{TD}}_{\boldsymbol{z}}\log w_k \tag{D.3}$$

which follows from applying the chain-rule and using $\nabla_* w_k = w_k\nabla_*\log w_k$.

Tucker et al. (2019) derive the DReGs identity (Eq. (5)) and use it to derive the following doubly-reparameterized gradient estimator (DReGs) w.r.t. the approximate posterior parameters $\boldsymbol{\phi}$ as:

$$\widehat{\nabla}^{\text{DReGs}}_{\boldsymbol{\phi}}\mathcal{L}_{\text{IWAE}} = \sum_{k=1}^{K}\widetilde{w}_k^2\nabla^{\text{TD}}_{\boldsymbol{z}_k}\log w_k\nabla^{\text{TD}}_{\boldsymbol{\phi}}\mathcal{T}_q\left(\boldsymbol{\epsilon}_k;\boldsymbol{\phi}\right). \qquad \boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon}) \tag{D.4}$$

D.2. Derivation of the GDReGs estimator

Similarly, we can derive a generalized doubly-reparameterized gradient (GDReGs) estimator w.r.t. the prior parameters $\boldsymbol{\theta}$. We use the GDReGs identity (Eq. (7)) derived above with $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) = \widetilde{w}_k$ and note that the reweighting term $\log\frac{q_{\boldsymbol{\phi}}(z)}{p_{\boldsymbol{\theta}}(z)}$ looks like a log importance weight except for the missing likelihood:

$$\nabla_{\boldsymbol{\theta}}^{\mathrm{TD}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\theta}}^{\mathrm{TD}} \log w_k \right] = \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\theta}}^{\mathrm{TD}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) \right] \tag{D.5}$$

$$\overset{(7)}{=} \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k)} + \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \widetilde{w}_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \right] \tag{D.6}$$

$$\overset{(D.3)}{=} \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k)} + \left(\widetilde{w}_k - \widetilde{w}_k^2 \right) \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log w_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \right] \tag{D.7}$$

$$= \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log p(\boldsymbol{x}|\boldsymbol{z}_k) - \widetilde{w}_k^2 \nabla_{\boldsymbol{z}}^{\mathrm{TD}} \log w_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \right]. \tag{D.8}$$

Thus, the GDReGs estimator is given by:

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{GDReGs}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log p(\boldsymbol{x}|\boldsymbol{z}_k) - \widetilde{w}_k^2 \nabla_{\boldsymbol{z}}^{\mathrm{TD}} \log w_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \quad \boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x}). \tag{11}$$

Note that the $\boldsymbol{z}_k$ are sampled from $q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$ but re-rexpressed as if they came from $p_{\boldsymbol{\theta}}(\boldsymbol{z})$.

We can rewrite the importance weights as

$$w_k = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k) p(\boldsymbol{x}|\boldsymbol{z}_k)}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k|\boldsymbol{x}) p_{\boldsymbol{\theta}}(\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})}. \tag{D.9}$$

Thus, if the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$ is equal to the true posterior $p_{\boldsymbol{\theta}}(\boldsymbol{z}_k|\boldsymbol{x})$, all weights w_k become equal to $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ and thus constant w.r.t. $\boldsymbol{z}_k$. In that case the second term in the GDReGs estimator Eq. (11) vanishes and the overall expression simplifies to

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{GDReGs}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\mathrm{TD}} \log p(\boldsymbol{x}|\boldsymbol{z}_k) \; \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})}. \quad \boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x}) \tag{D.10}$$

In contrast, the usual IWAE gradient involves the score function for $p_{\boldsymbol{\theta}}(\boldsymbol{z}_k)$:

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\mathrm{naive}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \sum_{i=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}_k), \qquad \boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x}). \tag{D.11}$$

Appendix E. Derivation of the DReGs and GDReGs estimator for IWAE objectives of hierarchical VAEs

In this section we derivations of and further details on the extension of DReGs and GDReGs to hierarchical VAEs with the IWAE objective.

E.1. Preliminaries and notation for the hierarchical IWAE objective

For a hierarchically structured model with L stochastic layers the IWAE objective is still given by Eq. (D.1) but with importance weights w_k given by

$$w_k = \frac{p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL})\; p_{\boldsymbol{\theta}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL})}{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}|\boldsymbol{x})}. \tag{E.1}$$

Here, $\boldsymbol{z}_{kl}$ denotes the kth importance sample ($k \in \{1, \ldots, K\}$) for the lth layer ($l \in \{1, \ldots, L\}$). Both the variational posterior and the prior distribution factorize according to their respective hierarchical structure. While the prior factorizes top-down in most cases, the variational posterior can have many different structures. In order for the distributions to be valid in the context of a VAE, we require the individual dependency graphs for the prior (generative path) and the variational posterior (inference path) to be directed acyclic graphs. Cycles would mean that a latent variable conditionally dependent on itself. To keep the dependency structure general, we write the factorization of the variational posterior and prior as follows:

$$q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}|\boldsymbol{x}) = \prod_{l=1}^{L} q_{\boldsymbol{\phi}_l}(\boldsymbol{z}_{kl}|\mathrm{pa}_{\boldsymbol{\alpha}}(l), \boldsymbol{x}) = \prod_{l=1}^{L} q_{\boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}(l);\boldsymbol{\phi}_l)}(\boldsymbol{z}_{kl}) \tag{E.2}$$

$$p_{\boldsymbol{\theta}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}) = \prod_{l=1}^{L} p_{\boldsymbol{\theta}_l}(\boldsymbol{z}_{kl}|\mathrm{pa}_{\boldsymbol{\beta}}(l)) = \prod_{l=1}^{L} p_{\boldsymbol{\beta}_l(\mathrm{pa}_{\boldsymbol{\beta}}(l);\boldsymbol{\theta}_l)}(\boldsymbol{z}_{kl}) \tag{E.3}$$

Here, $\boldsymbol{\alpha}_l(\cdot;\boldsymbol{\phi}_l)$ and $\boldsymbol{\beta}_l(\cdot;\boldsymbol{\theta}_l)$ are the distribution parameters of the variational posterior and prior distribution in the lth layer, respectively, and we have made the dependencies of the conditional distributions explicit; $\mathrm{pa}_{\boldsymbol{\alpha}}(l)$ denotes the "parents" of the latent variable $\boldsymbol{z}_{kl}$ according to the dependency graph of the inference path (the factorization of the posterior); similarly, $\mathrm{pa}_{\boldsymbol{\beta}}(l)$ denotes the latent variables that $\boldsymbol{z}_{kl}$ directly depends on according to the factorization of the prior $p_{\boldsymbol{\theta}}$. Typically, the prior is assumed to factorize top-down, such that $\mathrm{pa}_{\boldsymbol{\beta}}(l) = \boldsymbol{z}_{k(l+1)}$ for all but the top-most layer.

The samples $\boldsymbol{z}_{kl}$ are drawn from the variational posterior and can be expressed through reparameterization as $\boldsymbol{z}_{kl} = \mathcal{T}_{q_l}(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}(l), \boldsymbol{\phi}_l))$, where $\boldsymbol{\epsilon}_{kl}$ is an independent noise variable per importance sample and layer.

We note that it is these dependencies of the distribution parameters $\boldsymbol{\alpha}_l$ and $\boldsymbol{\beta}_l$ on $\mathrm{pa}_{\boldsymbol{\alpha}}(l)$ and $\mathrm{pa}_{\boldsymbol{\beta}}(l)$, respectively, that give rise to the indirect score functions as discussed in App. C.

E.2. Derivation of the hierarchical DReGs estimator for IWAE

With notation fully set up we consider the reparameterized gradients of the IWAE objective w.r.t. the variational parameters in a particular stochastic layer $\boldsymbol{\phi}_l$:

$$\nabla^{\mathrm{TD}}_{\boldsymbol{\phi}_l} \mathcal{L}^{\mathrm{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon})}\left[\sum_{k=1}^{K} \widetilde{w}_k \nabla^{\mathrm{TD}}_{\boldsymbol{\phi}_l} \log w_k\right] \tag{E.4}$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon})}\left[\sum_{k=1}^{K} \widetilde{w}_k \left(\nabla^{\mathrm{TD}}_{\boldsymbol{z}_{kl}} \log w_k \nabla_{\boldsymbol{\phi}_l} \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}\left(l\right), \boldsymbol{\phi}_l)\right) + \nabla_{\boldsymbol{\phi}_l} \log w_k\right)\right] \tag{E.5}$$

where we have used the chain-rule to arrive at Eq. (E.5); the first term contains both the (true) pathwise gradients as well as the indirect score functions; the second term only contains a direct score function as we only take the partial derivative w.r.t. $\boldsymbol{\phi}_l$.

We can rewrite this direct score function gradient because only one term in the (log-)importance weight directly depends on $\boldsymbol{\phi}_l$,

$$\nabla_{\boldsymbol{\phi}_l} \log w_k = -\nabla_{\boldsymbol{\phi}_l} \log q_{\boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}(l);\boldsymbol{\phi}_l)}(\boldsymbol{z}_{kl}). \tag{E.6}$$

Applying the DReGs identity to this term and using Eq. (D.3) yields:

$$\mathbb{E}_{\boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon})}\left[\sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\phi}_l} \log w_k\right] = -\mathbb{E}_{\boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon})}\left[\sum_{k=1}^{K} (\widetilde{w}_k - \widetilde{w}_k^2) \nabla^{\mathrm{TD}}_{\boldsymbol{z}_{kl}} \log w_k \nabla_{\boldsymbol{\phi}_l} \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}\left(l\right), \boldsymbol{\phi}_l)\right)\right] \tag{E.7}$$

which agrees with the first term in Eq. (E.5) up to the prefactor. Thus, both the true pathwise gradients as well as the indirect score functions appear twice and the prefactors partly cancel to give rise to the DReGs estimator for hierarchical IWAE objectives:

DReGs estimator for hierarchical IWAE objectives

$$\widehat{\nabla}^{\mathrm{DReGs}}_{\boldsymbol{\phi}_l} \mathcal{L}^{\mathrm{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla^{\mathrm{TD}}_{\boldsymbol{z}_{kl}} \log w_k \nabla_{\boldsymbol{\phi}_l} \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}\left(l\right), \boldsymbol{\phi}_l)\right); \qquad \boldsymbol{\epsilon}_{1:K} \sim q(\boldsymbol{\epsilon}) \tag{E.8}$$

where $\boldsymbol{z}_{kl} = \mathcal{T}_{q_l}(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}\left(l\right), \boldsymbol{\phi}_l)), \forall l \in \{1, \ldots, L\}, \forall k \in \{1, \ldots, K\}$ through reparameterization.

We emphasize that the total derivative w.r.t. $\boldsymbol{z}_{kl}$ contains pathwise gradients as well as indirect score functions for both the variational posterior as well as for the prior. The hierarhical DReGs estimator otherwise looks very similar to the DReGs estimator in the single layer case (Tucker et al., 2019).

In App. F.1 we explain how to implement this estimator effectively and in a structure-agnostic way. That is, we do *not* have to derive a new estimator for each new dependency graph of the variational posterior or the prior.

E.3. Derivation of the hierarchical GDReGs estimator for IWAE

Next, we derive the expression for the GDReGs estimator for hierarchical VAEs with IWAE objective.

Applying the GDReGs identity entails re-expressing the samples $\boldsymbol{z}_{kl}$ from the variational posterior as if they were sampled from the prior. Starting form a sample $(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}) \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L|\boldsymbol{x})$, we use the inverse flow of $p_{\boldsymbol{\theta}}$ to obtain new noise variables for each layer, $(\widetilde{\boldsymbol{\epsilon}}_{k1}, \ldots, \widetilde{\boldsymbol{\epsilon}}_{kL})$. We then use the forward flow of $p_{\boldsymbol{\theta}}$ to obtain back $(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL})$ but with the gradient path now depending on $\boldsymbol{\theta}$ as discussed in Sec. 3.

More precisely, we find that

$$\boldsymbol{z}_{kl}^{(q)} = \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\mathrm{pa}_{\boldsymbol{\alpha}}(l), \boldsymbol{\phi}_l)\right) \qquad \text{original sampling of } (\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}) \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L|\boldsymbol{x}) \tag{E.9}$$

$$\widetilde{\boldsymbol{\epsilon}}_{kl} = \mathcal{T}_{p_l}^{-1}\left(\boldsymbol{z}_{kl}^{(q)}; \boldsymbol{\beta}_l(\mathrm{pa}_{\boldsymbol{\beta}}(l), \boldsymbol{\theta}_l)\right) \qquad \text{inverse prior flow to obtain new “noise” variables} \tag{E.10}$$

$$\boldsymbol{z}_{kl} = \mathcal{T}_{p_l}\left(\widetilde{\boldsymbol{\epsilon}}_{kl}; \boldsymbol{\beta}_l(\mathrm{pa}_{\boldsymbol{\beta}}(l), \boldsymbol{\theta}_l)\right) \qquad \text{forward prior flow to re-express the } \boldsymbol{z}_{kl} \tag{E.11}$$

where $\boldsymbol{\epsilon}_{kl} \sim q(\boldsymbol{\epsilon})$ follows a simple distribution that is different from the more complicated distribution of $\widetilde{\boldsymbol{\epsilon}}_{kl}$. Note how the initial reparameterization of a sample $\boldsymbol{z}_{kl}$ depends on the dependency structure of the variational posterior (through $\mathrm{pa}_{\boldsymbol{\alpha}}(\cdot)$), while the other transformations depend on the dependency structure of the prior ($\mathrm{pa}_{\boldsymbol{\beta}}(\cdot)$).

As for DReGs, we note that only one term in the log importance weight directly depends on the variable $\boldsymbol{\theta}_l$,

$$\nabla_{\boldsymbol{\theta}_l} \log w_k = \nabla_{\boldsymbol{\theta}_l} \log p_{\boldsymbol{\beta}_l(\mathrm{pa}_{\boldsymbol{\beta}}(l); \boldsymbol{\theta}_l)}(\boldsymbol{z}_{kl}). \tag{E.12}$$

With these prerequesits, we can compute the GDReGs estimator for parameters $\boldsymbol{\theta}_l$ of the lth stochastic layer.

$$\nabla_{\boldsymbol{\theta}_l}^{\mathrm{TD}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\theta}_l} \log w_k\right] \tag{E.13}$$

$$\overset{E.12}{=} \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\left[\sum_{k=1}^{K} \widetilde{w}_k \nabla_{\boldsymbol{\theta}_l} \log p_{\boldsymbol{\beta}_l(\mathrm{pa}_{\boldsymbol{\beta}}(l); \boldsymbol{\theta}_l)}(\boldsymbol{z}_{kl})\right] \tag{E.14}$$

$$\begin{aligned} \overset{7}{=} \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Bigg[\sum_{k=1}^{K} \Bigg(&\widetilde{w}_k \nabla_{\boldsymbol{z}_{kl}}^{\mathrm{TD}} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL})} + \\ &+ \left(\widetilde{w}_k - \widetilde{w}_k^2\right) \nabla_{\boldsymbol{z}_{kl}}^{\mathrm{TD}} \log w_k\Bigg) \nabla_{\boldsymbol{\theta}_l} \mathcal{T}_{p_l}\left(\widetilde{\boldsymbol{\epsilon}}_{kl}; \boldsymbol{\theta}_l\right)\Big|_{\widetilde{\boldsymbol{\epsilon}}_{kl} = \mathcal{T}_{p_l}^{-1}(\boldsymbol{z}_{kl}; \boldsymbol{\theta}_l)}\Bigg] \end{aligned} \tag{E.15}$$

GDReGs estimator for hierarchical IWAE objectives

$$\widehat{\nabla}_{\boldsymbol{\theta}_l}^{\mathrm{GDReGs}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\mathrm{IWAE}} = \tag{E.16}$$

$$= \sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_{kl}}^{\mathrm{TD}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_{k1}, \ldots, \boldsymbol{z}_{kL}) - \widetilde{w}_k^2 \nabla_{\boldsymbol{z}_{kl}}^{\mathrm{TD}} \log w_k\right) \nabla_{\boldsymbol{\theta}_l} \mathcal{T}_{p_l}\left(\widetilde{\boldsymbol{\epsilon}}_{kl}; \boldsymbol{\theta}_l\right)\Big|_{\widetilde{\boldsymbol{\epsilon}}_{kl} = \mathcal{T}_{p_l}^{-1}(\boldsymbol{z}_{kl}; \boldsymbol{\theta}_l)}$$

with $\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$, and where we suppressed dependencies on $\mathrm{pa}_*(l)$ where they are not necessary to simplify notation.

The estimator looks very similar to the GDReGs estimator for a single layer IWAE model Eq. (11). Note that just like above for hierarchical DReGs, the total gradients w.r.t. $\boldsymbol{z}_{kl}$ give rise to both (true) pathwise gradients as well as indirect score functions through the hierarchical dependencies of the variational posterior and prior.

In App. F.2 we show how to implement the hierarchical GDReGs estimator Eq. (E.16) effectively and in a way that is agnostic to the structure of the model. That is, we do not have to derive a separate estimator for every dependency graph of the variational posterior and prior.

E.4. Double reparameterization and indirect score functions

In principle, we could apply double-reparameterization to the indirect score functions as well. However, as we explain now, we often cannot doubly-reparameterize *all* indirect score functions; moreover, even in cases where this is possible, it is still impractical, as the corresponding estimator depends on the exact model structure and would require adaptation to each dependency graph of the prior and variational posterior.

Double reparameterization of indirect score functions works in the same way as for the direct score functions except that $g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})$ is given by $\widetilde{w}_k^2$ instead of $\widetilde{w}_k$ in this case. The derivatives of $\widetilde{w}_k^2$ have a similar reproducing property as we observed in Eq. (D.3):

$$\nabla_{\boldsymbol{z}}^{\mathrm{TD}} \widetilde{w}_k^2 = 2(\widetilde{w}_k^2 - \widetilde{w}_k^3) \nabla_{\boldsymbol{z}}^{\mathrm{TD}} \log w_k. \tag{E.17}$$

Thus, double reparameterization of the indirect score functions similarly gives rise to further indirect score functions. We note that these indirect score functions only appear for the "children" of the current stochastic layer, that is, stochastic variables in those layers that depend on the current layer. In this context, "children" refers to *all* children w.r.t. the dependency structure of both, the variational posterior *and* the prior. For a particular layer l we obtain indirect score functions from double reparameterization of all of its (direct or indirect) parent nodes. Following the dependency structure, we could collect all of these terms and reparameterize them to obtain pathwise gradients only.

However, a problem arises, because we need to account for dependencies of both the variational posterior *and* the prior. Reparameterization of a score function gives rise to indirect score functions in all its "children" layers for both the variational posterior and the prior. For general hierarchical structures, this leads to cycles, in that some of the children of one dependency tree (the variational posterior) are the parents in the other (the prior) and/or vice versa. In this case we are never able to collect all the terms and fully reparameterize all the score functions.

Moreover, even if the joint dependency graph of the variational posterior and the prior were acyclic, this derivation would be structure-specific and would need to be repeated for each hierarchical structure. We therefore do not doubly reparameterize the indirect score functions.

Appendix F. Surrogate losses to implement the DReGs and GDReGs estimators for IWAE objectives

As we discussed in Sec. 3 and similar to Tucker et al. (2019), we use surrogate loss functions to compute the gradients w.r.t. the likelihood, proposal, and prior parameters. That is, we use different losses, such that backpropagation results in the respective gradient estimator.

While Tucker et al. (2019) use a single surrogate loss to compute the gradient estimators for all parts of the objective, we choose to use separate surrogate losses for each of the three parameter groups (likelihood, variational posterior, prior). In principle, we could combine them into a single loss, but in order to keep presentation simple we keep them separate. Computationally this does not make a difference as modern deep learning frameworks avoid duplicate computation.

For the likelihood parameters, we use the regular (negative) IWAE objective Eq. (1) as a loss. That is, the gradient estimator for the likelihood parameters is given by the gradient of the negative IWAE objective.

To construct the other surrogate losses we need to stop the gradients at various points in the computation graph. In the following, we use the shorthand notation $\underset{\sim\times}{}$ to indicate that we stop gradients into the underlined parts of an expression. Where it might be ambiguous, or to highlight where we do *not* stop gradients, we use the shorthand $\underset{\checkmark}{\underline{}}$ to indicate that gradients flow. For example, $f(\underset{\checkmark}{\underline{\boldsymbol{\phi}}}, \underset{\sim\times}{\boldsymbol{\theta}})$ means that we backpropagate gradients into $\boldsymbol{\phi}$ but not into $\boldsymbol{\theta}$.

F.1. DReGs for variational posterior parameters $\boldsymbol{\phi}$

F.1.1. Single stochastic layer

Here we reproduce part of the surrogate loss for the variational parameters $\boldsymbol{\phi}$ by Tucker et al. (2019) for the single stochastic layer case:

$$\begin{aligned} L_{\mathrm{DReGs}}(\boldsymbol{\phi}) &= \sum_{k=1}^{K} \underset{\sim\times}{\widetilde{w}_k^2} \Big(\log p_{\boldsymbol{\lambda}}(\boldsymbol{x} | \underset{\checkmark}{\underline{\boldsymbol{z}_k}}) + \log p_{\boldsymbol{\beta}(\boldsymbol{\theta})}(\underset{\checkmark}{\underline{\boldsymbol{z}_k}}) - \log q_{\boldsymbol{\alpha}(\underset{\sim\times}{\boldsymbol{\phi}})}(\underset{\checkmark}{\underline{\boldsymbol{z}_k}}) \Big) \\ \underset{\checkmark}{\underline{\boldsymbol{z}_k}} &= \mathcal{T}_q\left(\boldsymbol{\epsilon}_k; \underset{\checkmark}{\underline{\boldsymbol{\phi}}}\right) \qquad\qquad \boldsymbol{\epsilon}_k \sim q(\boldsymbol{\epsilon}_k) \end{aligned} \tag{F.1}$$

That is, we sample $\boldsymbol{z}_k \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k | \boldsymbol{x})$ as usual (by reparameterizing independent noise variables $\boldsymbol{\epsilon}_k$) but stop the gradients of the parameters that parameterize the distributions when evaluating their densities, $\log q_{\boldsymbol{\alpha}(\underset{\sim\times}{\boldsymbol{\phi}})}(\underset{\checkmark}{\underline{\boldsymbol{z}_k}})$. In addition we stop the gradients around the normalized importance weights $\widetilde{w}_k$. Differentiating L_{DReGs} w.r.t. the proposal parameters $\boldsymbol{\phi}$ yields the DReGs estimator Eq. (6). Note that we do not explicitly stop gradients into $\boldsymbol{\lambda}$ or $\boldsymbol{\theta}$ because we use separate surrogate losses for those parameter groups. If we were to use a combined loss, we would potentially have to stop gradients into these parameters as well, depending on the estimator used.

To practically implement this surrogate loss, we use two copies of the variational posterior distribution. An unaltered one (no stopped gradients) to sample $\boldsymbol{z}$ and one with gradients into the proposal parameters stopped to evaluate the log densities. The stopped gradient makes sure that we do not obtain a direct score function as we have doubly-reparameterized it.

Note that for single-stochastic-layer models we could also stop the gradients of the distribution parameters $\boldsymbol{\alpha}$ instead as they only depend on $\boldsymbol{\phi}$. We emphasize that this is not possible for hierarchical models as this would eliminate the indirect score functions and thus produce potentially biased gradients.

F.1.2. Multiple stochastic layers

For multiple layers, the surrogate loss for the DReGs estimator Eq. (E.8) is given by:

$$
\begin{aligned}
L_{\text{DReGs}}(\boldsymbol{\phi}) &= \sum_{k=1}^{K} \underset{\times}{\underline{\widetilde{w}_k^2}} \log w_k \\
\log w_k &= \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\underset{\checkmark}{\underline{\boldsymbol{z}_{k1}}}, \underset{\checkmark}{\underline{\ldots}}, \underset{\checkmark}{\underline{\boldsymbol{z}_{kL}}}) + \sum_{l=1}^{L} \log p_{\boldsymbol{\beta}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\beta}}(l)}};\boldsymbol{\theta}_l)}(\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}}) - \sum_{l=1}^{L} \log q_{\boldsymbol{\alpha}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\alpha}}(l)}};\underset{\times}{\underline{\boldsymbol{\phi}_l}})}(\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}}) \\
\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}} &= \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\alpha}}(l)}}, \underset{\checkmark}{\underline{\boldsymbol{\phi}_l}})\right) \qquad\qquad \boldsymbol{\epsilon}_{kl} \sim q(\boldsymbol{\epsilon}_{kl})
\end{aligned}
\tag{F.2}
$$

Again, we do not explicitly stop gradients into $\boldsymbol{\lambda}$ or $\boldsymbol{\theta}_l$ as we only take gradients w.r.t. $\boldsymbol{\phi}_l$.

The indirect score functions arise due to the indirect dependence of the distribution parameters $\boldsymbol{\alpha}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\alpha}}(l)}}; \underset{\times}{\underline{\boldsymbol{\phi}_l}})$ and $\boldsymbol{\beta}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\beta}}(l)}}; \boldsymbol{\theta}_l)$ on the parent latent variables $\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\alpha}}(l)}}$ and $\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\beta}}(l)}}$, respectively. Note how the former depends on the hierarchical structure of the variational posterior, whereas the latter depends on the hierarchical structure of the prior.

To implement this surrogate loss effectively, we again use two copies of the variational posterior distribution. One un-altered one (without stopped gradiends) from which we sample the individual reparameterized $\boldsymbol{z}_{kl}$ and through which gradients can flow; we use these samples to evaluate densities at and to parameterize the distribution parameters at subsequent layers. Derivatives w.r.t. $\boldsymbol{\phi}_l$ will then give rise to pathwise gradients and indirect score functions. We use the second copy of the variational posterior, where we have stopped the parameters $\boldsymbol{\phi}_l$, to evaluate the density at for the log importance weights in the last summand of Eq. (F.2).

F.2. GDReGs for prior parameter $\boldsymbol{\theta}$

F.2.1. Single stochastic layer

$$
\begin{aligned}
L_{\text{GDReGs}}(\boldsymbol{\theta}) &= \sum_{k=1}^{K} \underset{\times}{\underline{\widetilde{w}_k}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\underset{\checkmark}{\underline{\boldsymbol{z}_k}}) - \underset{\times}{\underline{\widetilde{w}_k^2}}\Big(\log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\underset{\checkmark}{\underline{\boldsymbol{z}_k}}) + \log p_{\boldsymbol{\beta}\underset{\times}{\underline{(\boldsymbol{\theta})}}}(\underset{\checkmark}{\underline{\boldsymbol{z}_k}}) - \log q_{\boldsymbol{\alpha}(\boldsymbol{\phi})}(\underset{\checkmark}{\underline{\boldsymbol{z}_k}})\Big) \\
\underset{\checkmark}{\underline{\boldsymbol{z}_k}} &= \mathcal{T}_p\left(\underset{\times}{\underline{\widetilde{\boldsymbol{\epsilon}}_k}}; \underset{\checkmark}{\underline{\boldsymbol{\theta}}}\right) \\
\underset{\times}{\underline{\widetilde{\boldsymbol{\epsilon}}_k}} &= \underset{\times}{\underline{\mathcal{T}_p^{-1}\left(\mathcal{T}_q\left(\boldsymbol{\epsilon}_k; \boldsymbol{\phi}\right); \boldsymbol{\theta}\right)}} \qquad\qquad \boldsymbol{\epsilon}_k \sim q(\boldsymbol{\epsilon}_k)
\end{aligned}
\tag{F.3}
$$

Taking the derivative of Eq. (F.3) w.r.t. $\boldsymbol{\theta}$ gives rise to the GDReGs estimator for the single stochastic layer IWAE objective. As explained in Sec. 3, we need to re-express $\boldsymbol{z}_k$ such that its path depends on $\boldsymbol{\theta}$. In effect, we first sample $\boldsymbol{z}_k = \mathcal{T}_q\left(\boldsymbol{\epsilon}_k; \boldsymbol{\phi}\right)$, then compute the new noise variable $\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})$, and re-compute $\boldsymbol{z}_k = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})$. Note that we have to stop gradients into the noise variables $\widetilde{\boldsymbol{\epsilon}}_k$ to obtain the correct gradient estimator. This explains the `stop_grad` in Fig. 1.

As above, we do not explicitly stop gradients into $\boldsymbol{\lambda}$ and $\boldsymbol{\phi}$ as we use separate losses for these parameter groups and only compute gradients of Eq. (F.3) w.r.t. $\boldsymbol{\theta}$.

To effectively implement this loss, we use two copies of the prior distribution. One that we implement as a normalizing flow and a second one with stopped gradients into the parameters. We then proceed as follows:

- Compute the new noise variables $\widetilde{\boldsymbol{\epsilon}}_k$ by using the inverse flow $\mathcal{T}_p^{-1}$ on the samples $\boldsymbol{z}_k$ from the variational posterior.
- Stop the gradients into $\widetilde{\boldsymbol{\epsilon}}_k$.
- Use the forward flow $\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})$ to re-compute $\boldsymbol{z}_k$ but with path dependent on $\boldsymbol{\theta}$. These samples when derived w.r.t. $\boldsymbol{\theta}$ will give rise to the pathwise gradients.
- Use the second copy of the prior (with stopped gradients into its parameters) to evaluate the log density at the samples $\boldsymbol{z}_k$. The stopped gradients make sure that we do not obtain the direct score function.

F.2.2. Multiple stochastic layers

For multiple stochastic layers the surrogate loss that gives rise to the GDReGs estimator Eq. (E.16) is given by:

$$
\begin{aligned}
L_{\text{GDReGs}}(\boldsymbol{\theta}) &= \sum_{k=1}^{K} \underset{\times}{\uwave{\widetilde{w}_k}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\underset{\checkmark}{\underline{\boldsymbol{z}_{k1}}}, \underset{\checkmark}{\underline{\ldots}}, \underset{\checkmark}{\underline{\boldsymbol{z}_{kL}}}) - \underset{\times}{\uwave{\widetilde{w}_k^2}} \log w_k \\
\log w_k &= \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\underset{\checkmark}{\underline{\boldsymbol{z}_{k1}}}, \underset{\checkmark}{\underline{\ldots}}, \underset{\checkmark}{\underline{\boldsymbol{z}_{kL}}}) + \sum_{l=1}^{L} \log p_{\boldsymbol{\beta}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\beta}}(l)}}; \underset{\times}{\uwave{\boldsymbol{\theta}_l}})}(\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}}) - \sum_{l=1}^{L} \log q_{\boldsymbol{\alpha}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\alpha}}(l)}}; \boldsymbol{\phi}_l)}(\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}}) \\
\underset{\checkmark}{\underline{\boldsymbol{z}_{kl}}} &= \mathcal{T}_{p_l}\left(\underset{\times}{\uwave{\widetilde{\boldsymbol{\epsilon}}_{kl}}}; \boldsymbol{\beta}_l(\underset{\checkmark}{\underline{\mathrm{pa}_{\boldsymbol{\beta}}(l)}}, \underset{\checkmark}{\underline{\boldsymbol{\theta}_l}})\right) \\
\underset{\times}{\uwave{\widetilde{\boldsymbol{\epsilon}}_{kl}}} &= \underset{\times}{\uwave{\mathcal{T}_{p_l}^{-1}\left(\boldsymbol{z}_{kl}^{(q)}; \boldsymbol{\beta}_l\left(\mathrm{pa}_{\boldsymbol{\beta}}(l), \boldsymbol{\theta}_l\right)\right)}} \\
\boldsymbol{z}_{kl}^{(q)} &= \mathcal{T}_{q_l}\left(\boldsymbol{\epsilon}_{kl}; \boldsymbol{\alpha}_l\left(\mathrm{pa}_{\boldsymbol{\alpha}}(l), \boldsymbol{\phi}_l\right)\right) \qquad\qquad \boldsymbol{\epsilon}_{kl} \sim q(\boldsymbol{\epsilon}_{kl})
\end{aligned}
\tag{F.4}
$$

As for the single layer case, we need to re-express variational posterior samples $\boldsymbol{z}_{kl}$ as if they were sampled from the prior. To obtain the correct gradients, we again have to stop gradients into the new noise variables $\widetilde{\boldsymbol{\epsilon}}_{kl}$, also see Fig. C.1.

As for hierarchical DReGs, the indirect score functions stem from the second and third term of $\log w_k$ and arise because the distribution parameters $\boldsymbol{\alpha}_l$ and $\boldsymbol{\beta}_l$ depend on the "parent" stochastic layers.

As before we use two copies of the prior distribution, one with regular gradients that is set up as a flow, and a second with stopped gradients into the parameters. This allows us to implement the GDReGs estimator regardless of the model structure.