Appendix to Generalized Doubly-Reparameterized Gradient Estimators

Contents of the Appendix

A. Derivation of the GDREGs identity

Here, we detail the derivation of our main result, the GDREGs identity (Eq. (16)), which we restate here.

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

We can derive this identity in two ways: (1) through a reweighting correction as presented in the main paper, see App. A.1; (2) through a flow-like transformation of $\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, see App. A.2.

A.1. Derivation via reweighting

In the main paper (Sec. 4), we explained that we need to make the sampling path of $\boldsymbol{z}$ depend on the parameters $\boldsymbol{\theta}$ to replace the score function $\nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z})$ with a pathwise derivative. At the same time, we evaluate the objective on samples $\boldsymbol{z}$ from the variational posterior $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ during training. The derivation consists of the following three steps:

① *Temporarily* change the path such that it depends on $\boldsymbol{\theta}$. We change the path by first using importance sampling reweighting to temporarily re-write the expectation, $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}[*] = \mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}*\right]$ (step (1a)), and then by employing reparameterization on $p_{\boldsymbol{\theta}}(\boldsymbol{z})$: $\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})$ with $\widetilde{\boldsymbol{\epsilon}}\sim p(\widetilde{\boldsymbol{\epsilon}})=\mathcal{N}(0,\mathbb{I})$ (step (1b)).

② Perform the gradient computation and collect all the terms.

③ Change the path back by un-doing the reparameterization and the reweighting so we can use samples $\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ to estimate the expectation.

$$\nabla_{\boldsymbol{\theta}}^{\text{TD}}\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] \overset{(1a)}{=} \nabla_{\boldsymbol{\theta}}^{\text{TD}}\mathbb{E}_{p_{\boldsymbol{\theta}}(\boldsymbol{z})}\left[\tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})}g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] \overset{(1b)}{=} \nabla_{\boldsymbol{\theta}}^{\text{TD}}\mathbb{E}_{p(\widetilde{\boldsymbol{\epsilon}})}\left[\tfrac{q_{\boldsymbol{\phi}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))}{p_{\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))}g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}))\right] \tag{A.1}$$

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

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

In the derivation we have used the identity $x\nabla_* \log x = \nabla_* x$ repeatedly. By noting that $\nabla^{\text{TD}}_{\boldsymbol{\theta}} \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right] = \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]$, we can cancel these terms on the left hand side of Eq. (A.1) and right hand side of Eq. (A.3). By moving $-\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]$ to the other side we obtain the desired result.

A.2. Derivation via flow-like transformation

Here we provide a second derivation of the GDREGs identity (Eq. (16)) that does not explicitly use a re-weighting of the expectation as in App. A.1, but uses flow-like transformations instead (Rezende & Mohamed, 2015). We already noted in the main text that we can interpret the change of paths as a flow $\boldsymbol{z} \to \widetilde{\boldsymbol{\epsilon}} \to \boldsymbol{z}$, where $\widetilde{\boldsymbol{\epsilon}}$ follows a more complicated distribution than the original $\boldsymbol{\epsilon}$ that might have been used to reparameterize samples $\boldsymbol{z}$ from $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$. Here, we make this connection to normalizing flows more explicit.

We make use of the usual change of probability density formula for flows (see, e.g., Rezende & Mohamed (2015)):

$$y = f(x); x \sim p_x(x) \quad \Rightarrow \quad p_y(y) = p_x(x)\left|\nabla_x f(x)\right|^{-1}. \tag{A.4}$$

Our main insight is that we can reparameterize $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ in many different ways. Usually we sample an independent noise variable from a simple base distribution, for example, $\boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon}) = \mathcal{N}(0, \mathbb{I})$, and use it to reparameterize $\boldsymbol{z}$ as $\boldsymbol{z} = \mathcal{T}_q\left(\boldsymbol{\epsilon}; \boldsymbol{\phi}\right)$. However, we can equally use a different base distribution $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$ and reparameterize $\boldsymbol{z}$ as $\boldsymbol{z} = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta}), \widetilde{\boldsymbol{\epsilon}} \sim \widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$. Note that we reparameterize using $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ in this case. In order for $\boldsymbol{z}$ still to be distributed according to $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, we have to choose $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$ to be itself given by the normalizing flow $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta}), \boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z})$, such that $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}}) = \widetilde{q}(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\phi}, \boldsymbol{\theta})$. It may appear counter-intuitive to transform $\boldsymbol{z}$ as $\boldsymbol{z} \to \widetilde{\boldsymbol{\epsilon}} \to \boldsymbol{z}$ because we are doing and then un-doing a transformation; however, it allows us to make the computation path depend on $\boldsymbol{\theta}$ when computing the gradients. We then separate the forward flow $\boldsymbol{z} = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})$ and the backward flow $\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})$ by moving the forward flow into the path through reparameterization and the backward flow into the integration measure, see Eq. (A.7) below. To move the backward flow into the integration measure, we express $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$ as a change of density:

$$\widetilde{\boldsymbol{\epsilon}} = \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta}); \boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \quad \Rightarrow \quad \widetilde{q}(\widetilde{\boldsymbol{\epsilon}}) = q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1} \tag{A.5}$$

Note that $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$ is different from $q(\boldsymbol{\epsilon}) = \mathcal{N}(0, \mathbb{I})$ typically used to reparameterize samples of the approximate posterior $\boldsymbol{z} = \mathcal{T}_q\left(\boldsymbol{\epsilon}; \boldsymbol{\phi}\right); \boldsymbol{\epsilon} \sim q(\boldsymbol{\epsilon})$! Using the change of density in Eq. (A.5) together with reparameterization as described above, we can re-write the following expectation over $\boldsymbol{z}$ as an integral over $\widetilde{\boldsymbol{\epsilon}}$:

$$\int q_{\boldsymbol{\phi}}(\boldsymbol{z}) g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int \widetilde{q}(\widetilde{\boldsymbol{\epsilon}}) g_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})\right) \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \qquad \text{reparameterization} \tag{A.6}$$

$$= \int \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1}\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} g_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})\right) \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \qquad \text{change of density} \tag{A.7}$$

In the these integrals, $\widetilde{\boldsymbol{\epsilon}}$ is an *independent* variable; its dependence on $\boldsymbol{\theta}$ has been moved into the path $\left(g_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})\right)\right)$ as well as the change of density (Eq. (A.5)). This is one way to understand where the `stop_gradient` in Figs. 2 and 3 comes from.

To derive the GDREGs identity, we take the total derivative of Eq. (A.7) w.r.t. $\boldsymbol{\theta}$ and apply the chain- and product rule

$$\nabla^{\text{TD}}_{\boldsymbol{\theta}} \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} \overset{A.7}{=} \nabla^{\text{TD}}_{\boldsymbol{\theta}} \int \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1}\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} g_{\boldsymbol{\phi},\boldsymbol{\theta}}\left(\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta})\right) \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \tag{A.8}$$

$$= \int \nabla^{\text{TD}}_{\boldsymbol{\theta}} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\Big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})}\right] \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \tag{A.9}$$

$$= \int \nabla^{\text{TD}}_{\boldsymbol{z}} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta}) + \\ + \nabla_{\boldsymbol{\theta}} \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right]_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \tag{A.10}$$

$$= \int \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1}\right] \left(g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla^{\text{TD}}_{\boldsymbol{z}} \log\left(q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1}\right) + \nabla^{\text{TD}}_{\boldsymbol{z}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right)\Big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}; \boldsymbol{\theta}) + \\ + \left[q_{\boldsymbol{\phi}}(\boldsymbol{z})\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1}\right] \left(g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z}; \boldsymbol{\theta})\right|^{-1} + \nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\right)\Big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \mathrm{d}\widetilde{\boldsymbol{\epsilon}} \tag{A.11}$$

where we have separated out the derivatives and used $x\nabla_* \log x = \nabla_* x$. We can now further separate terms and undo the change of density to replace $q_{\boldsymbol{\phi}}(\boldsymbol{z}) \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right|^{-1} = \widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$ (Eq. (A.5)) after taking the derivatives. We obtain

$$\nabla_{\boldsymbol{\theta}}^{\text{TD}} \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \tag{A.12}$$

$$\begin{aligned} &= \int \widetilde{q}(\widetilde{\boldsymbol{\epsilon}}) \Big\{ \nabla_{\boldsymbol{z}}^{\text{TD}} \left(\log q_{\boldsymbol{\phi}}(\boldsymbol{z}) + \log \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right|^{-1} \right) g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \Big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}) + \\ &\quad + \nabla_{\boldsymbol{z}}^{\text{TD}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}) + \nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} + \\ &\quad - g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z})\big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \nabla_{\boldsymbol{\theta}} \log \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right|\big|_{\boldsymbol{z}=\mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})} \Big\} \mathrm{d}\widetilde{\boldsymbol{\epsilon}}. \end{aligned} \tag{A.13}$$

To evaluate the gradients of the log Jacobians, $\nabla_* \log \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right|$, we can combine the log Jacobians with a simple base distribution $q(\widetilde{\boldsymbol{\epsilon}})$ to obtain $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ because

$$\boldsymbol{z} = \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta}); \widetilde{\boldsymbol{\epsilon}} \sim q(\widetilde{\boldsymbol{\epsilon}}) = \mathcal{N}(0,\mathbb{I}) \quad \Rightarrow \quad p_{\boldsymbol{\theta}}(\boldsymbol{z}) = q(\widetilde{\boldsymbol{\epsilon}}) \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right| \tag{A.14}$$

through reparameterization of $p_{\boldsymbol{\theta}}(\boldsymbol{z})$ and by noting that

$$\nabla_* \log(f(x) \cdot c) = \nabla_* \log(f(x)) \qquad \text{if } c \text{ is constant w.r.t. } * \tag{A.15}$$

and that the simple base distribution $q(\widetilde{\boldsymbol{\epsilon}})$ is constant w.r.t. all gradients:

$$\nabla_* \log \left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right| \overset{(A.15)}{=} \nabla_* \log \left(\left|\nabla_{\boldsymbol{z}} \mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})\right| q(\widetilde{\boldsymbol{\epsilon}})\right) \overset{(A.14)}{=} \nabla_* \log p_{\boldsymbol{\theta}}(\boldsymbol{z}). \tag{A.16}$$

This allows us to simplify Eq. (A.13) as

$$\begin{aligned} \nabla_{\boldsymbol{\theta}}^{\text{TD}} \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \mathrm{d}\boldsymbol{z} = \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \Big\{ &\left(g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{z}}^{\text{TD}} \log \tfrac{q_{\boldsymbol{\phi}}(\boldsymbol{z})}{p_{\boldsymbol{\theta}}(\boldsymbol{z})} + \nabla_{\boldsymbol{z}}^{\text{TD}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}};\boldsymbol{\theta})\Big|_{\widetilde{\boldsymbol{\epsilon}}=\mathcal{T}_p^{-1}(\boldsymbol{z};\boldsymbol{\theta})} + \\ &+ \nabla_{\boldsymbol{\theta}} g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) - g_{\boldsymbol{\phi},\boldsymbol{\theta}}(\boldsymbol{z}) \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) \Big\} \mathrm{d}\boldsymbol{z}, \end{aligned} \tag{A.17}$$

which is identical to Eq. (A.3) and yields the GDREGs identity as explained above.

B. 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. (22) in the main paper.

B.1. Preliminaries on the IWAE objective

The importance weighted autoencoder (IWAE) objective is given by

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

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

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

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 $\mathcal{T}_q(\boldsymbol{\epsilon}_k;\boldsymbol{\phi})$. 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,

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

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. (9)) and use it to derive the following doubly-reparameterized gradient estimator (DREGs) w.r.t. the approximate posterior parameters $\boldsymbol{\phi}$ as:

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

B.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. (16)) 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:

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

$$\overset{(16)}{=} \mathbb{E}_{\boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})} \left[\sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\text{TD}} \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_k)} + \nabla_{\boldsymbol{z}_k}^{\text{TD}} \widetilde{w}_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \right] \tag{B.6}$$

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

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

Thus, the GDREGs estimator is given by:

$$\widehat{\nabla}_{\boldsymbol{\theta}}^{\text{GDREGs}} \mathcal{L}_{\boldsymbol{\phi},\boldsymbol{\theta}}^{\text{IWAE}} = \sum_{k=1}^{K} \left(\widetilde{w}_k \nabla_{\boldsymbol{z}_k}^{\text{TD}} \log p(\boldsymbol{x}|\boldsymbol{z}_k) - \widetilde{w}_k^2 \nabla_{\boldsymbol{z}}^{\text{TD}} \log w_k \right) \nabla_{\boldsymbol{\theta}} \mathcal{T}_p(\widetilde{\boldsymbol{\epsilon}}_k; \boldsymbol{\theta})|_{\widetilde{\boldsymbol{\epsilon}}_k = \mathcal{T}_p^{-1}(\boldsymbol{z}_k;\boldsymbol{\theta})} \qquad \boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x}). \tag{22}$$

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

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

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. (22) vanishes and the overall expression simplifies to

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

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

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

C. 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.

C.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. (B.1) but with importance weights w_k given by

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

Here, $\boldsymbol{z}_{kl}$ denotes the kth importance sample ($k \in \{1, \dots, K\}$) for the lth layer ($l \in \{1, \dots, L\}$). 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:

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

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

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 Sec. 3.

C.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 $\boldsymbol{\phi}_l$:

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

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

where we have used the chain-rule to arrive at Eq. (C.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 $\boldsymbol{\phi}_l$,

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

Applying the DREGs identity to this term and using Eq. (B.3) yields:

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

which agrees with the first term in Eq. (C.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

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

where $\boldsymbol{z}_{kl} = \mathcal{T}_{q_l}(\boldsymbol{\epsilon}_{kl};\boldsymbol{\alpha}_l(\text{pa}_{\boldsymbol{\alpha}}(l),\boldsymbol{\phi}_l)), \forall l\in\{1,\dots,L\}, \forall k\in\{1,\dots,K\}$ 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. E.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.

C.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 $(\boldsymbol{z}_{k1},\dots,\boldsymbol{z}_{kL})\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_1,\dots,\boldsymbol{z}_L|\boldsymbol{x})$, we use the inverse flow of $p_{\boldsymbol{\theta}}$ to obtain new noise variables for each layer, $(\widetilde{\boldsymbol{\epsilon}}_{k1},\dots,\widetilde{\boldsymbol{\epsilon}}_{kL})$. We then use the forward flow of $p_{\boldsymbol{\theta}}$ to obtain back $(\boldsymbol{z}_{k1},\dots,\boldsymbol{z}_{kL})$ but with the gradient path now depending on $\boldsymbol{\theta}$ as discussed in App. A and Sec. 4.

More precisely, we find that

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

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

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

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 $\text{pa}_{\boldsymbol{\alpha}}(\cdot)$), while the other transformations depend on the dependency structure of the prior ($\text{pa}_{\boldsymbol{\beta}}(\cdot)$).

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

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

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

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

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

$$\overset{16}{=} \mathbb{E}_{\boldsymbol{z}_{1:K}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}\Bigg[\sum_{k=1}^{K}\Bigg(\widetilde{w}_k\nabla^{\text{TD}}_{\boldsymbol{z}_{kl}}\log\frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}_{k1},\dots,\boldsymbol{z}_{kL}|\boldsymbol{x})}{p_{\boldsymbol{\theta}}(\boldsymbol{z}_{k1},\dots,\boldsymbol{z}_{kL})} + \\ + \left(\widetilde{w}_k-\widetilde{w}_k^2\right)\nabla^{\text{TD}}_{\boldsymbol{z}_{kl}}\log w_k\Bigg)\nabla_{\boldsymbol{\theta}_l}\mathcal{T}_{p_l}\left(\widetilde{\boldsymbol{\epsilon}}_{kl};\boldsymbol{\theta}_l\right)|_{\widetilde{\boldsymbol{\epsilon}}_{kl}=\mathcal{T}^{-1}_{p_l}(\boldsymbol{z}_{kl};\boldsymbol{\theta}_l)}\Bigg] \tag{C.15}$$

GDREGs estimator for hierarchical IWAE objectives

$$\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\theta}_l}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \tag{C.16}$$
$$= \sum_{k=1}^{K}\left(\widetilde{w}_k\nabla^{\text{TD}}_{\boldsymbol{z}_{kl}}\log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_{k1},\dots,\boldsymbol{z}_{kL}) - \widetilde{w}_k^2\nabla^{\text{TD}}_{\boldsymbol{z}_{kl}}\log w_k\right)\nabla_{\boldsymbol{\theta}_l}\mathcal{T}_{p_l}\left(\widetilde{\boldsymbol{\epsilon}}_{kl};\boldsymbol{\theta}_l\right)|_{\widetilde{\boldsymbol{\epsilon}}_{kl}=\mathcal{T}^{-1}_{p_l}(\boldsymbol{z}_{kl};\boldsymbol{\theta}_l)}\,;\quad \boldsymbol{z}_{1:K}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$$

where we suppressed dependencies on $\text{pa}_*(l)$ where they are not necessary to simplify notation.

The estimator looks very similar to the GDREGs estimator for a single layer IWAE model Eq. (22). 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. E.2 we show how to implement the hierarchical GDREGs estimator Eq. (C.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.

C.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. (B.3):

$$\nabla^{\text{TD}}_{\boldsymbol{z}}\widetilde{w}_k^2 = 2(\widetilde{w}_k^2 - \widetilde{w}_k^3)\nabla^{\text{TD}}_{\boldsymbol{z}}\log w_k. \tag{C.17}$$

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.

D. Worked example for a 2-layer hierarchical VAE

In this section we show an example of using the IWAE objective with a 2-layer VAE model consisting of a prior $p_{\boldsymbol{\theta}_2}(\boldsymbol{z}_2)p_{\boldsymbol{\theta}_1}(\boldsymbol{z}_1|\boldsymbol{z}_2)$ and likelihood $p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_1,\boldsymbol{z}_2)$. The inference network is bottom-up: $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{x})q_{\boldsymbol{\phi}_2}(\boldsymbol{z}_2|\boldsymbol{x},\boldsymbol{z}_1)$. Therefore $\mathrm{pa}_{\boldsymbol{\alpha}}(2)=\boldsymbol{z}_1$ and $\mathrm{pa}_{\boldsymbol{\beta}}(1)=\boldsymbol{z}_2$.

We hierarchically sample $\boldsymbol{z}_{k1}$ and $\boldsymbol{z}_{k2}$ from the approximate posterior and abbreviate:

$$\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1) \equiv \mathcal{T}_{q_1}(\boldsymbol{\epsilon}_{k1};\boldsymbol{\alpha}_1(\boldsymbol{\phi}_1)) \tag{D.1}$$

$$\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2) \equiv \mathcal{T}_{q_{2|1}}(\boldsymbol{\epsilon}_1;\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1),\boldsymbol{\phi}_2)). \tag{D.2}$$

We also explicitly distinguish between the distribution parameters $\boldsymbol{\alpha}_i$ and the network parameters $\boldsymbol{\phi}_i$ for the posterior as well as the distribution parameters $\boldsymbol{\beta}_i$ and the network parameters $\boldsymbol{\theta}_i$ for the prior. A single importance sample w_k with all of its functional dependencies is given by:

$$w_k = \frac{p_{\boldsymbol{\lambda}}\left(\boldsymbol{x}|\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1),\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2)\right)\;\; p_{\boldsymbol{\beta}_2(\boldsymbol{\theta}_2)}\left(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2)\right)\;\; p_{\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2),\boldsymbol{\theta}_1)}\left(\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1)\right)}{q_{\boldsymbol{\alpha}_1(\boldsymbol{x},\boldsymbol{\phi}_1)}(\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1))\;\; q_{\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1),\boldsymbol{\phi}_2)}\left(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2)\right)}. \tag{D.3}$$

The DREGs estimator for the variational parameters of the lower latent layer, $\boldsymbol{\phi}_1$, is then given by:

$$\widehat{\nabla}^{\text{DREGs}}_{\boldsymbol{\phi}_1}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla^{\text{TD}}_{\boldsymbol{z}_{k1}} \log w_k \nabla_{\boldsymbol{\phi}_1}\mathcal{T}_{q_1}\left(\boldsymbol{\epsilon}_{k1};\boldsymbol{\alpha}_1(\boldsymbol{x},\boldsymbol{\phi}_1)\right); \qquad \boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon}) \tag{D.4}$$

$$\begin{aligned}\nabla^{\text{TD}}_{\boldsymbol{z}_{k1}}\log w_k = {}& \nabla_{\boldsymbol{z}_{k1}}\log w_k - \nabla_{\boldsymbol{\alpha}_{2|1}}\log q_{\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_1(\boldsymbol{\phi}_1),\boldsymbol{\phi}_2)}\left(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2)\right)\nabla_{\boldsymbol{z}_{k1}}\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1),\boldsymbol{\phi}_2)\\ &+ \nabla_{\boldsymbol{\beta}_{1|2}}\log p_{\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2),\boldsymbol{\theta}_1)}\left(\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1)\right)\nabla_{\boldsymbol{z}_{k1}}\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2),\boldsymbol{\theta}_1)\end{aligned} \tag{D.5}$$

where we have expanded the total derivative w.r.t. $\boldsymbol{z}_{k1}$ into the (true) pathwise gradients and two indirect score functions.

Similarly, we can compute the DREGs estimator for the variational parameters of the upper level, $\boldsymbol{\phi}_2$:

$$\widehat{\nabla}^{\text{DREGs}}_{\boldsymbol{\phi}_2}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \widetilde{w}_k^2 \nabla^{\text{TD}}_{\boldsymbol{z}_{k2}} \log w_k \nabla_{\boldsymbol{\phi}_2}\mathcal{T}_{q_{2|1}}\left(\boldsymbol{\epsilon}_{k2};\boldsymbol{\alpha}_2(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\phi}_1),\boldsymbol{\phi}_2)\right); \qquad \boldsymbol{\epsilon}_{1:K}\sim q(\boldsymbol{\epsilon}) \tag{D.6}$$

$$\nabla^{\text{TD}}_{\boldsymbol{z}_{k2}}\log w_k = \nabla_{\boldsymbol{z}_{k2}}\log w_k + \nabla_{\boldsymbol{\beta}_{1|2}}\log p_{\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2),\boldsymbol{\theta}_1)}\left[\boldsymbol{z}_1(\boldsymbol{\phi}_1)\right]\nabla_{\boldsymbol{z}_{k2}}\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\phi}_1,\boldsymbol{\phi}_2),\boldsymbol{\theta}_1) \tag{D.7}$$

For this model structure of the prior and variational posterior, there is only one indirect score function for this gradient. Note that the indirect score functions are computed automatically in our surrogate losses that we introduce in the following section, such that we do not need to compute them manually; the hierarchical DREGs estimator can be implemented without having to trace the dependency structure of the model manually.

To compute the GDREGs estimator, we first have to re-express the samples $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ as if they were sampled from $p_{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2}(\boldsymbol{z}_1,\boldsymbol{z}_2)$. We write this reparameterization as

$$\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2) \equiv \mathcal{T}_{p_2}\left(\widetilde{\boldsymbol{\epsilon}}_{k2};\boldsymbol{\beta}_2(\boldsymbol{\theta}_2)\right) \tag{D.8}$$

$$\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2) \equiv \mathcal{T}_{p_{2|1}}\left(\widetilde{\boldsymbol{\epsilon}}_{k1};\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2),\boldsymbol{\theta}_1)\right) \tag{D.9}$$

where $\widetilde{\boldsymbol{\epsilon}}_1$ and $\widetilde{\boldsymbol{\epsilon}}_2$ are noise variables drawn from $\widetilde{q}(\widetilde{\boldsymbol{\epsilon}})$, which is given by the inverse prior flow of the samples drawn from $q_{\boldsymbol{\phi}_1,\boldsymbol{\phi}_2}(\boldsymbol{x}_1,\boldsymbol{x}_2|\boldsymbol{x})$. The full functional dependency of a single importance sample is given by:

$$w_k = \frac{p_{\boldsymbol{\lambda}}\left(\boldsymbol{x}|\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2),\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2)\right)\;\; p_{\boldsymbol{\beta}_2(\boldsymbol{\theta}_2)}\left(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2)\right)\;\; p_{\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2),\boldsymbol{\theta}_1)}\left(\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)\right)}{q_{\boldsymbol{\alpha}_1(\boldsymbol{x},\boldsymbol{\phi}_1)}(\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2))\;\; q_{\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2),\boldsymbol{\phi}_2)}\left(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2)\right)}. \tag{D.10}$$

The GDREGs estimator w.r.t. the prior parameters of the lower stochastic layer, $\boldsymbol{\theta}_1$, is given by:

$$\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\theta}_1}\mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K}\left(\widetilde{w}_k\nabla^{\text{TD}}_{\boldsymbol{z}_{k1}}\log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_{k1},\boldsymbol{z}_{k2}) - \widetilde{w}_k^2\nabla^{\text{TD}}_{\boldsymbol{z}_{k1}}\log w_k\right)\nabla_{\boldsymbol{\theta}_1}\mathcal{T}_{p_1}\left(\widetilde{\boldsymbol{\epsilon}}_{k1};\boldsymbol{\theta}_1\right)\big|_{\widetilde{\boldsymbol{\epsilon}}_{k1}=\mathcal{T}^{-1}_{p_1}(\boldsymbol{z}_{k1};\boldsymbol{\theta}_1)}; \qquad \boldsymbol{z}_{1:K}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$$

$$\nabla^{\text{TD}}_{\boldsymbol{z}_{k1}} \log w_k = \nabla_{\boldsymbol{z}_{k1}} \log w_k - \nabla_{\boldsymbol{\alpha}_{2|1}} \log q_{\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2),\boldsymbol{\phi}_2)} \left(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2)\right) \nabla_{\boldsymbol{z}_{k1}} \boldsymbol{\alpha}_{2|1}(\boldsymbol{x}, \boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2), \boldsymbol{\phi}_2) \tag{D.11}$$

The GDREGs estimator w.r.t. the prior parameters of the upper stochastic layer, $\boldsymbol{\theta}_2$, is given by:

$$\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\theta}_2} \mathcal{L}^{\text{IWAE}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \sum_{k=1}^{K} \left(\widetilde{w}_k \nabla^{\text{TD}}_{\boldsymbol{z}_{k2}} \log p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_{k1},\boldsymbol{z}_{k2}) - \widetilde{w}_k^2 \nabla^{\text{TD}}_{\boldsymbol{z}_{k2}} \log w_k\right) \nabla_{\boldsymbol{\theta}_2} \mathcal{T}_{p_2}\left(\widetilde{\boldsymbol{\epsilon}}_{k2}; \boldsymbol{\theta}_2\right)|_{\widetilde{\boldsymbol{\epsilon}}_{k2}=\mathcal{T}^{-1}_{p_2}(\boldsymbol{z}_{k2};\boldsymbol{\theta}_2)}; \qquad \boldsymbol{z}_{1:K} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}_k|\boldsymbol{x})$$

$$\begin{aligned}\nabla^{\text{TD}}_{\boldsymbol{z}_{k2}} \log w_k = \nabla_{\boldsymbol{z}_{k2}} \log w_k &- \nabla_{\boldsymbol{\alpha}_{2|1}} \log q_{\boldsymbol{\alpha}_{2|1}(\boldsymbol{x},\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2),\boldsymbol{\phi}_2)} \left(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2)\right) \nabla_{\boldsymbol{z}_{k2}} \boldsymbol{\alpha}_{2|1}(\boldsymbol{x}, \boldsymbol{z}_1(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2), \boldsymbol{\phi}_2) \\ &+ \nabla_{\boldsymbol{\beta}_{1|2}} \log p_{\boldsymbol{\beta}_{1|2}(\boldsymbol{z}_2(\boldsymbol{\theta}_2),\boldsymbol{\theta}_1)} \left(\boldsymbol{z}_{k1}(\boldsymbol{\theta}_1,\boldsymbol{\theta}_2)\right) \nabla_{\boldsymbol{z}_{k2}} \boldsymbol{\beta}_{1|2}(\boldsymbol{z}_{k2}(\boldsymbol{\theta}_2), \boldsymbol{\theta}_1)\end{aligned} \tag{D.12}$$

Note how the GDREGs estimator for $\boldsymbol{\theta}_2$ has two indirect score functions for the upper layer where the DREGs estimator for $\boldsymbol{\phi}_2$ only has one. This is due to the opposite factorization (opposite hierarchical dependency structure) of the variational posterior and the prior. This cyclic dependence is also the reason why we cannot replace all indirect score functions with DREGs and GDREGs gradients. Double reparameterization of one of the indirect score functions, leads to another indirect score function, whose double-reparameterization in turn leads back to the first indirect score function but with a different pre-factor.

Again, note that the indirect score functions are computed automatically in our surrogate losses, App. E, and we do not need to manually trace the dependency structure or derive them.

E. Surrogate losses to implement the DREGs and GDREGs estimators for IWAE objectives

As we discussed in Sec. 4.1 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. (3) 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 $\underline{\cdot}$ 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 $\underline{\cdot}$ to indicate that gradients flow. For example, $f(\underline{\boldsymbol{\phi}}, \boldsymbol{\theta})$ means that we backpropagate gradients into $\boldsymbol{\phi}$ but not into $\boldsymbol{\theta}$.

E.1. DREGs for variational posterior parameters $\boldsymbol{\phi}$

E.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:

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

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, $\log q_{\boldsymbol{\alpha}(\underline{\boldsymbol{\phi}})}(\underline{\boldsymbol{z}_k})$. 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. (11). 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.

E.1.2. MULTIPLE STOCHASTIC LAYERS

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

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

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 $\boldsymbol{\alpha}_l(\text{pa}_{\boldsymbol{\alpha}}(l); \boldsymbol{\phi}_l)$ and $\boldsymbol{\beta}_l(\text{pa}_{\boldsymbol{\beta}}(l); \boldsymbol{\theta}_l)$ on the parent latent variables $\text{pa}_{\boldsymbol{\alpha}}(l)$ and $\text{pa}_{\boldsymbol{\beta}}(l)$, 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. (E.2).

E.2. GDREGs for prior parameter $\boldsymbol{\theta}$

E.2.1. SINGLE STOCHASTIC LAYER

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

Taking the derivative of Eq. (E.3) w.r.t. $\boldsymbol{\theta}$ gives rise to the GDREGs estimator for the single stochastic layer IWAE objective. As explained in Sec. 4, 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. 2.

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. (E.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.

E.2.2. MULTIPLE STOCHASTIC LAYERS

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

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

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. 3.

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.

F. Implementation details

In our implementations we use `NumPy` (Harris et al., 2020), `JAX` (Bradbury et al., 2018), `Haiku` (Hennigan et al., 2020), as well as `tensorflow probability` and `tensorflow distributions` (Dillon et al., 2017).

Stopping gradients. All major frameworks allow for gradients to be stopped or interupted. For example, in `TensorFlow` (Abadi et al., 2016) we can use `tf.stop_gradient` and in `JAX` we can use `jax.lax.stop_gradient`. To implement stopped gradients w.r.t. the parameters of a distribution we use `haiku.experimental.custom_getter` contexts, which allow us to manipulate the parameters before they are used to construct the respective networks; in this case we use the context to stop gradients.

Re-expressing samples. To re-express variational posterior samples as if they came from the prior, we directly implement the computation flow as it is described in e.g. Figs. 2 and 3 and detailed in App. E.2.

In the code listings Listings 1 and 2 we provide (pseudo-)code for a simple implementation of the surrogate objectives. Listing 1 contains the import statements as well as the function and class definitions to create parameterized distributions that allow for

1. stopping the gradients into their parameters, and
2. re-expressing samples using the bijector interface in `tensorflow probability`.

In Listing 2 we implement surrogate objectives for the naive and the GDREGs estimators of the cross-entropy. More specifically, we wish to estimate:

$$
\nabla_{\boldsymbol{\theta}}^{\text{TD}} \mathcal{L}^{ce} = \nabla_{\boldsymbol{\theta}}^{\text{TD}} \mathbb{E}_{\boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right]. \tag{F.1}
$$

The naive estimator (using a single Monte Carlo sample) is given by

$$
\widehat{\nabla}_{\boldsymbol{\theta}}^{\text{naive}} \mathcal{L}^{ce} = \nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\boldsymbol{z}) \qquad \boldsymbol{z} \sim q_{\boldsymbol{\phi}}(\boldsymbol{z}). \tag{F.2}
$$

The corresponding GDREGs estimator (again using a single MC sample) is given by Eq. (21):

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

Listing 1: Function and class definitions necessary to define the surrogate objectives for the DREGs and GDREGs estimators.

```
from typing import List
import haiku as hk
import jax
import jax.lax as lax
import jax.numpy as jnp
from tensorflow_probability.substrates import jax as jtfp
jtfd = jtfp.distributions


def stop_grad_getter(next_getter, value, _):
  """A custom getter that stops gradients of parameters."""
  return lax.stop_gradient(next_getter(value))


def reparameterize_as_if_from(p: jtfd.TransformedDistribution,
                              z_q: jnp.ndarray) -> jnp.ndarray:
  """Reparameterize samples z_q as if they were sampled from p.

  Transforms: z_q -> stop_gradient(noise) -> z_q_as_if_from_p
  This transformation leaves the numerical value unchanged but alters the
    gradients.

  Args:
    p: a TransformedDistribution object
    z_q: Samples to be reparameterized.

  Returns:
    Reparameterized samples.
  """

  eps = p.bijector.inverse(z_q)
  eps = jax.lax.stop_gradient(eps)
  return p.bijector.forward(eps)


class ConditionalNormal(hk.Module):
  """A Normal distribution that is conditioned through an MLP."""

  def __init__(
      self,
      output_size: int,
      hidden_layer_sizes: List[int],
      name: str = "conditional_normal"):
    """Creates a conditional Normal distribution.

    Args:
      output_size: The dimension of the random variable.
      hidden_layer_sizes: The sizes of the hidden layers of the fully connected
        network used to condition the distribution on the inputs.
      name: The name of this distribution.
    """
    super(ConditionalNormal, self).__init__(name=name)
    self.name = name
    self.fcnet = hk.nets.MLP(
        output_sizes=hidden_layer_sizes + [2 * output_size],
        activation=jnp.tanh,
```

```
        activate_final=False,
        with_bias=True,
        name=name + "_fcnet")

  def condition(self, inputs):
    """Computes the parameters of a normal distribution based on the inputs."""
    outs = self.fcnet(inputs)
    mu, sigma = jnp.split(outs, 2, axis=-1)
    sigma = jax.nn.softplus(sigma)
    return mu, sigma

  def __call__(self, inputs, **kwargs):
    """Creates a normal distribution conditioned on the inputs."""
    # Optional `stop_gradient_params` argument stops the parameters
    # of the distribution.
    if kwargs.get("stop_gradient_params", False):
      with hk.experimental.custom_getter(stop_grad_getter):
        mu, sigma = self.condition(inputs)
    else:
      mu, sigma = self.condition(inputs)

    # Optional `as_flow` argument parameterizes the distribution as a flow
    # to have access to `Bijector.inverse` and `Bijector.forward`
    # to use with the function `reparameterize_as_if_from`
    if kwargs.get("as_flow", False):
      bijector = jtfp.bijectors.Chain(
          [jtfp.bijectors.Shift(shift=mu), jtfp.bijectors.Scale(scale=sigma)])
      base = jtfd.Normal(loc=jnp.zeros_like(mu), scale=jnp.ones_like(sigma))
      return jtfd.TransformedDistribution(
          distribution=base, bijector=bijector, name=self.name + "_flow")

    return jtfd.Normal(loc=mu, scale=sigma)
```

Listing 2: Code to implement the surrogate objective for the naive estimator of the cross-entropy as well as for the GDREGs estimator Eq. (21). Computing derivatives w.r.t. the parameters of the prior `p` using automatic differentiation gives rise to the correct expressions for the estimators.

```
# Create distributions
# Inputs `x` and context `c`
q = ConditionalNormal(x)
p = ConditionalNormal(c)
q_stop = ConditionalNormal(x, stop_gradient_params=True)
p_stop = ConditionalNormal(c, stop_gradient_params=True)
p_flow = ConditionalNormal(c, as_flow=True)

# Sample from the variational posterior
z_q = q.sample(sample_shape=[num_samples], seed=hk.next_rng_key())  # [k, bs, z]

# Reparameterize the samples from q as if they were sampled from p
z_q_as_p = reparameterize_as_if_from(p_flow, z_q)

# Cross-entropy surrogate losses
cross_entropy_naive = p.log_prob(z_q)
cross_entropy_gdregs = q_stop.log_prob(z_q_as_p) - p_stop.log_prob(z_q_as_p)
```

G. Experimental details and additional results

In this section we provide additional experimental details as well as additional results.

G.1. Illustrative example

As discussed in the main text, we use a 2-layer linear VAE inspired by the single layer example of Rainforth et al. (2018); Tucker et al. (2019). We use a top-down generative model $\boldsymbol{z}_2 \rightarrow \boldsymbol{z}_1 \rightarrow \boldsymbol{x}$, with $\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{x} \in \mathbb{R}^D$ and $D = 5$. The hierarchical prior is given by $\boldsymbol{z}_2 \sim \mathcal{N}(0, \mathbb{I})$, $\boldsymbol{z}_1|\boldsymbol{z}_2 \sim \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_2), \boldsymbol{\sigma}^2_{\boldsymbol{\theta}}(\boldsymbol{z}_2))$, and the likelihood is given by $\boldsymbol{x}|\boldsymbol{z}_1 \sim \mathcal{N}(\boldsymbol{z}_1, \mathbb{I})$. We choose a bottom-up variational posterior that factorizes as: $q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{\mu}_{\boldsymbol{\phi}_1}(\boldsymbol{x}), \boldsymbol{\sigma}^2_{\boldsymbol{\phi}_1}(\boldsymbol{x}))$ 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 functions $\boldsymbol{\mu}_*$ and $\boldsymbol{\sigma}_*$ are given by linear functions with weights and biases; the likelihood and the upper layer of the prior do not have any learnable parameters.

To generate data, we sample 512 datapoints from the model where we have set $\boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{z}_2) = \boldsymbol{z}_2$ and $\boldsymbol{\sigma}_{\boldsymbol{\theta}}(\boldsymbol{z}_2) = 1$.

We then train the parameters $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ in all linear layers using SGD on the IWAE objective til convergence. We then evaluate the gradient variance and gradient 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.

In Fig. 4 in the main paper we show the average gradient variance and gradient signal-to-noise ratio (SNR). The average is taken over all parameters of either the variational posterior or the prior. The gradient variance and gradient SNR for individual parameters exhibit the same qualitative behaviour.

G.2. Conditional and unconditional image modelling

For conditional and unconditional image modelling, we use VAEs with one or multiple stochastic layers, where the generative path is top-down and the inference path is bottom-up, as specified in Eq. (23), which we reproduce here for convenience:

$$
\begin{aligned}
q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{c}) &= q_{\boldsymbol{\phi}_1}(\boldsymbol{z}_1|\boldsymbol{x}, \boldsymbol{c}) \prod_{l=2}^{L} q_{\boldsymbol{\phi}_l}(\boldsymbol{z}_l|\boldsymbol{z}_{l-1}, \boldsymbol{x}, \boldsymbol{c}) \\
p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{c}) &= p_{\boldsymbol{\theta}_L}(\boldsymbol{z}_L|\boldsymbol{c}) \prod_{l=1}^{L-1} p_{\boldsymbol{\theta}_l}(\boldsymbol{z}_l|\boldsymbol{z}_{l+1}, \boldsymbol{c}) \\
p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}) &= p_{\boldsymbol{\lambda}}(\boldsymbol{x}|\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L).
\end{aligned}
\tag{23}
$$

For conditional image modelling, we predict the bottom half of an image given its top half as in Tucker et al. (2019), providing the top half as an additional context input $\boldsymbol{c}$ to the prior and variational posterior. Given the above model structure, $\mathrm{pa}_{\boldsymbol{\alpha}}(l) = \boldsymbol{z}_{k(l-1)}$ and $\mathrm{pa}_{\boldsymbol{\beta}}(l) = \boldsymbol{z}_{k(l+1)}$.

Each conditional distribution in Eq. (23) is given by an MLP of 2 hidden layers of 300 `tanh` units each. If a distribution has multiple inputs, we concatenate them along the feature dimension. The prior and variational posterior are all given by diagonal Gaussian distributions, whereas the likelihood is given by a Bernoulli distribution. The unconditional prior distribution in the uppermost layer $p_{\boldsymbol{\theta}_L}(\boldsymbol{z}_L)$ is given by a standard Normal distribution, $p_{\boldsymbol{\theta}_L}(\boldsymbol{z}_L) = \mathcal{N}(0, \mathbb{I})$. All latents $\boldsymbol{z}_l$ are 50 dimensional.

To avoid overfitting, we use dynamically binarized versions of the datasets. We use the Adam optimizer with default learning rate $3 \cdot 10^{-4}$ and default parameters $b_1 = 0.9$, $b_2 = 0.999$, and $\epsilon = 10^{-8}$. We use a batch size of 64 and $K = 64$ importance samples for training and evaluation. Note that for testing we report test objective values rather than an estimate of $\log p(\boldsymbol{x})$ by using a large number of importance samples. We do this as we are interested in the relative behaviour of the estimators.

In Fig. G.1 we provide further plots that show the evolution of the test objective and the gradient variance of the prior, the variational posterior, as well as the likelihood throughout training on *conditional modelling* of MNIST and FashionMNIST (predict bottom half given top half). As in the main paper we find that using GDREGs for the prior instead of the naive estimator (denoted as IWAE) always improves performance on the test objective regardless of the estimator for the variational posterior. We also note that GDREGs always reduces gradient variance for the prior early in training and also typically throughout training, especially for deeper models and when combined with the DREGs estimator for the variational posterior.

Interestingly, using the GDREGs estimator for the prior leads to an increase in the gradient variance for the variational posterior when we use the naive estimator but similar or even lower posterior gradient variance when combined with the DREGs estimator (third column in Fig. G.1). It always leads to a slight improvement in gradient variance for the likelihood parameters (fourth column in Fig. G.1). We hypothesize that this is the case because lower gradient variance in for one set

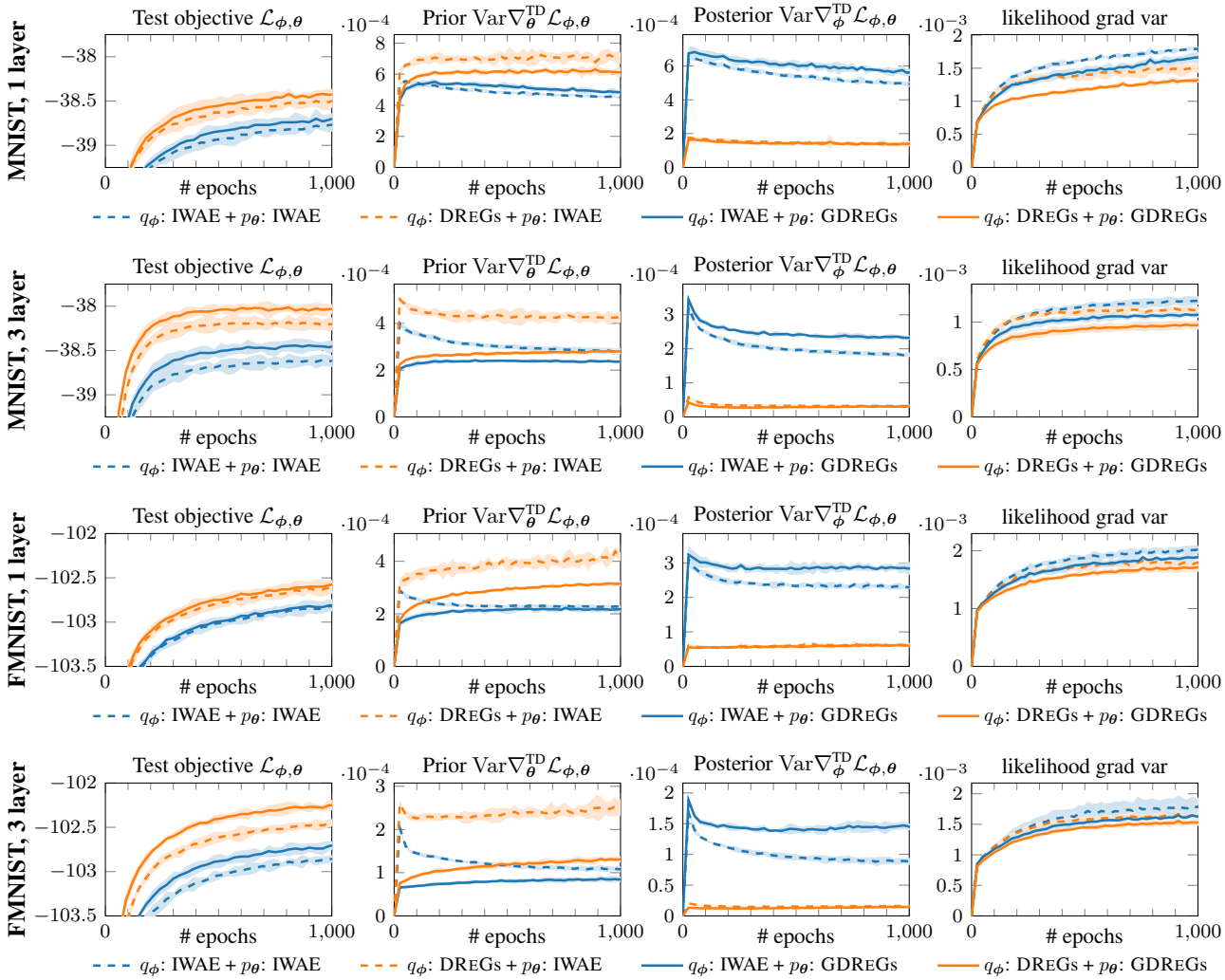

Figure G.1: Additional results for *conditional* image modelling with VAEs on MNIST and FashionMNIST (FMNIST) with 1 stochastic layer and 3 stochastic layers. For each task we show the test objective, the prior gradient variance, the variational posterior gradient variance, and the likelihood gradient variance. Means over 5 reruns; shaded areas denote $\pm$ 1.96 standard deviations σ.

of parameters makes it easier to estimate the gradients for another set as a secondary effect.

In Fig. G.2 we show that the training objective behaves qualitatively similar to the test objective in that the applying the DREGs or GDREGs estimator results in improved objective values in the same way, regardless of whether we consider the training or test objective. That is, our hierarchical extension of DREGs results in a better training objective values for both conditional and unconditional tasks. GDREGs is particularly helpful for conditional tasks.

Moreover, in the main text we had hypothesized that GDREGs performs better on conditional image modelling tasks than unconditional tasks because access to the context makes the variational posterior and the prior more similar. In the case of analytically computed cross-entropy in App. H, we derive that the GDREGs estimator outperforms a naive estimator of the score function in terms of gradient variance when the posterior and prior are similar. We hypothesize that this also holds more generally for the IWAE objective, where we cannot compute the gradient variance in closed form but only estimate it empirically. To investigate this, we computed the total average KL (rightmost column in Fig. G.2) over all latent variables and found that it is indeed lower for conditional than unconditional modelling, which indicates that the distributions are closer together in this case.

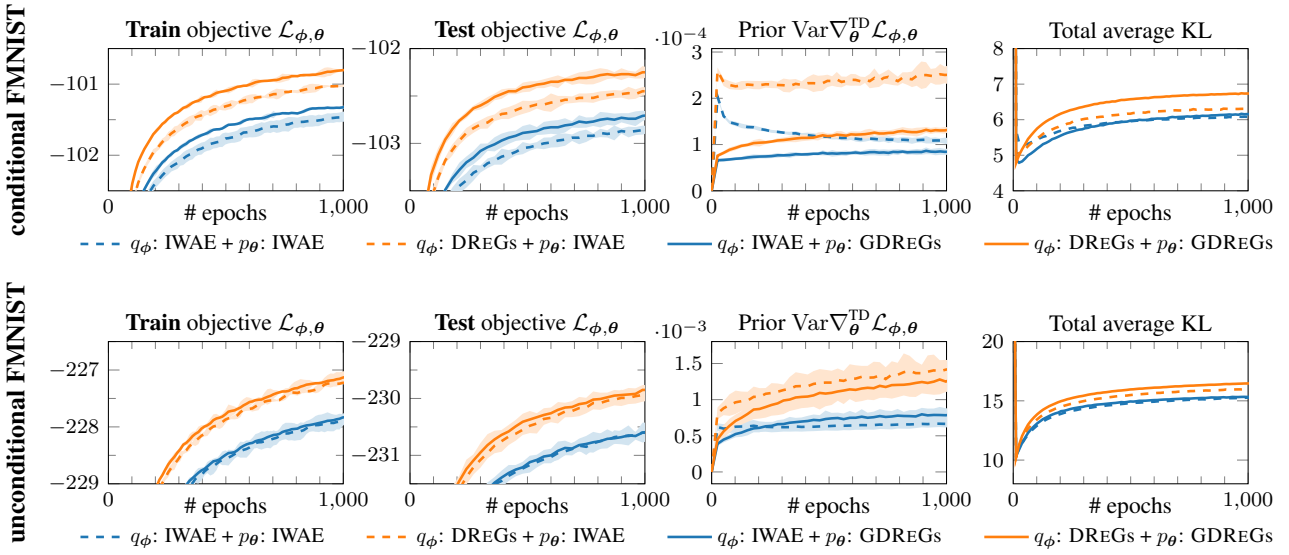

Figure G.2: Train objective *(leftmost column)* and total average KL *(rightmost column)* in addition to the test objective and prior gradient variance for conditional and unconditional FashionMNIST on a model with 3 stochastic layers.

G.3. Offline evaluation of the DREGs and GDREGs estimator

In the image modelling experiments in Sec. 5.2 we discussed the gradient variance of the different estimators for the posterior and prior parameters. However, we only analyzed estimators *online* on their respective runs; that is, in Fig. 6 the prior and posterior gradient variance shown corresponds to the variance of the estimator also used during training.

Here, we provide an ablation study for conditional image modelling on MNIST with a 2-layer VAE where we evaluate the different estimators offline; that is, for each combination of estimators used for training we also show the gradient variance of the other estimators.

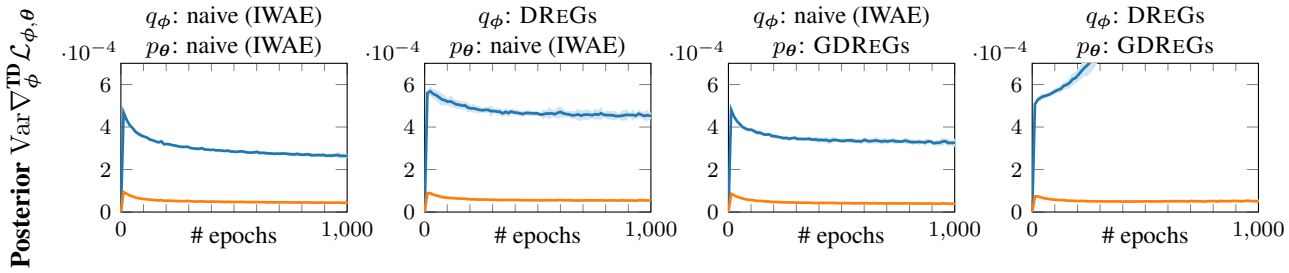

Figure G.3: Gradient variance of the **posterior** parameters ϕ as estimated with the naive (IWAE) estimator (—) and the DREGs estimator (—) for different combinations of estimators used during training as indicated by the sub-figure title. For example, in the leftmost plot we compare the posterior gradient variance as estimated by the naive (IWAE) estimator to that of the DREGs estimator on an experiment where we trained both the posterior parameters ϕ as well as the prior parameters θ with the naive (IWAE) estimator.

In Fig. G.3 we consider the gradient variance w.r.t. the **variational posterior** parameters ϕ and compare the naive (IWAE) estimator to the DREGs estimator. We find that, regardless which combination of estimators has been used during training, the DREGs estimator *always* results in a better (lower) gradient variance than the naive estimator.

Similarly, in Fig. G.4 we consider the gradient variance w.r.t. the **prior** parameters θ and compare the naive (IWAE) estimator to the GDREGs estimator. We find that generally the GDREGs gradient estimates have lower variance than the naive (IWAE) estimates. However, when we use the naive estimator for the prior parameters during training, this reduction is smaller and may only be present in the beginning of training. However, consistently, the GDREGs gradient estimates

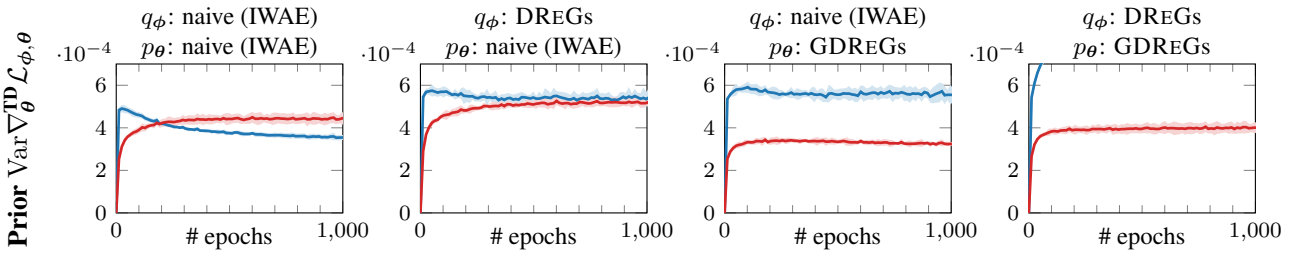

Figure G.4: Gradient variance of the **prior** parameters $\boldsymbol{\theta}$ as estimated with the naive (IWAE) estimator (——) and the GDREGs estimator (——) for different combinations of estimators used during training as indicated by the sub-figure title. For example, in the leftmost plot we compare the prior gradient variance as estimated by the naive (IWAE) estimator to that of the GDREGs estimator on an experiment where we trained both the posterior parameters $\boldsymbol{\phi}$ as well as the prior parameters $\boldsymbol{\theta}$ with the naive (IWAE) estimator.

have lower variance when we use the DREGs estimator during training to estimate the variational posterior parameters.

H. The cross-entropy for Gaussian distributions

Here, we investigate the properties of the GDREGs estimator compared to the naive estimator in a setting where all quantities of interest can be computed in closed form, the cross-entropy of two Gaussian distributions, $q_{\boldsymbol{\phi}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_q, \boldsymbol{\sigma}_q)$ and $p_{\boldsymbol{\theta}}(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_p, \boldsymbol{\sigma}_p)$.

The negative cross-entropy is given by

$$\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \mathbb{E}_{\boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\log p_{\boldsymbol{\theta}}(\boldsymbol{z})\right] = \tfrac{1}{2}\log(2\pi) + \log\boldsymbol{\sigma}_p + \frac{\boldsymbol{\sigma}_q^2 + (\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^2}{2\boldsymbol{\sigma}_p^2} \tag{H.1}$$

Note that in this analytic case, we can compute the gradients without having to sample. However, in the following we want to compare the naive (score function) estimator to the GDREGs estimator.

The naive estimator of this score function is given by:

$$\widehat{\nabla}^{\mathrm{naive}}_{\boldsymbol{\theta}}\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}}\log p_{\boldsymbol{\theta}}(\boldsymbol{z}); \qquad \boldsymbol{z}\sim q_{\boldsymbol{\phi}}(\boldsymbol{z}) \tag{H.2}$$

while the GDREGs estimator is given by (see Eq. (21)):

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

H.1. Gradient variance of the estimators

In the case under consideration, the parameters $\boldsymbol{\theta}$ are given by the mean and variance of the prior, $\boldsymbol{\theta} = \{\boldsymbol{\mu}_p, \boldsymbol{\sigma}_p\}$, and we can compute both the expectation as well as the variance of these gradient estimators in closed form.

Naive estimator

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\mathrm{naive}}_{\boldsymbol{\mu}_p}\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\mu}_q - \boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2} \tag{H.3}$$

$$\mathrm{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\mathrm{naive}}_{\boldsymbol{\mu}_p}\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\sigma}_q^2}{\boldsymbol{\sigma}_p^4} \tag{H.4}$$

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\mathrm{naive}}_{\boldsymbol{\sigma}_p}\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\sigma}_q^2 - \boldsymbol{\sigma}_p^2}{\boldsymbol{\sigma}_p^3} + \frac{(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^2}{\boldsymbol{\sigma}_p^3} \tag{H.5}$$

$$\mathrm{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\mathrm{naive}}_{\boldsymbol{\sigma}_p}\mathcal{L}^{\mathrm{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = 2\frac{\boldsymbol{\sigma}_q^4}{\boldsymbol{\sigma}_p^6} + 4\boldsymbol{\sigma}_q^2\frac{(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^2}{\boldsymbol{\sigma}_p^6} \tag{H.6}$$

Note that all operations are element-wise.

The proposed GDREGs estimator

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\mu}_p}\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\mu}_q - \boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2} \tag{H.7}$$

$$\text{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\mu}_p}\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\sigma}_q^2}{\boldsymbol{\sigma}_p^4}\frac{\left(\boldsymbol{\sigma}_p^2 - \boldsymbol{\sigma}_q^2\right)^2}{\boldsymbol{\sigma}_q^4} \tag{H.8}$$

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\sigma}_p}\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{\boldsymbol{\sigma}_q^2 - \boldsymbol{\sigma}_p^2}{\boldsymbol{\sigma}_p^3} + \frac{\left(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\right)^2}{\boldsymbol{\sigma}_p^3} \tag{H.9}$$

$$\text{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\widehat{\nabla}^{\text{GDREGs}}_{\boldsymbol{\sigma}_p}\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = 2\frac{\left(\boldsymbol{\sigma}_q^2 - \boldsymbol{\sigma}_p^2\right)^2}{\boldsymbol{\sigma}_p^6} + \frac{\left(\boldsymbol{\sigma}_p^2 - 2\boldsymbol{\sigma}_q^2\right)^2}{\boldsymbol{\sigma}_q^2}\frac{\left(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p\right)^2}{\boldsymbol{\sigma}_p^6} \tag{H.10}$$

Note that all operations are element-wise.

Both estimators have equal expectation; this is because GDREGs is an unbiased estimator.

Comparing the variances in Eq. (H.4) and Eq. (H.8) we note that the GDREGs estimator has a lower gradient variance than the naive estimator for the mean parameters $\boldsymbol{\mu}_p$ if

$$\boldsymbol{\sigma}_p^2 \leq 2\boldsymbol{\sigma}_q^2. \tag{H.11}$$

Similarly, comparing Eq. (H.6) and Eq. (H.10), we find that the GDREGs estimator has lower variance than the naive estimator for the variance parameters $\boldsymbol{\sigma}_p$ if

$$\boldsymbol{\sigma}_p^2 \leq 4\boldsymbol{\sigma}_q^2\left(1 - \frac{\boldsymbol{\sigma}_q^2}{(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^2 + 2\boldsymbol{\sigma}_q^2}\right) \tag{H.12}$$

In the case of $\boldsymbol{\mu}_p = \boldsymbol{\mu}_q$, this also reduces to $\boldsymbol{\sigma}_p^2 \leq 2\boldsymbol{\sigma}_q^2$.

Thus, we expect the GDREGs estimator to perform better than the naive estimator when $p_{\boldsymbol{\theta}}(\boldsymbol{z})$and $q_{\boldsymbol{\phi}}(\boldsymbol{z})$are close together.

H.2. Constructing the optimal control variate

Because the GDReGs estimators and the naive estimators have the same expectation, we can build a control variate out of their difference:

$$\left[\hat{\nabla}^{\text{naive}}_{\boldsymbol{\theta}} + \alpha(\hat{\nabla}^{\text{GDReG}}_{\boldsymbol{\theta}} - \hat{\nabla}^{\text{naive}}_{\boldsymbol{\theta}})\right]\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}. \tag{H.13}$$

We can then compute its optimal strength α^*, by minimizing its variance,

$$\alpha^* = \arg\min_{\alpha}\text{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\left[\hat{\nabla}^{\text{naive}}_{\boldsymbol{\theta}} + \alpha(\hat{\nabla}^{\text{GDReG}}_{\boldsymbol{\theta}} - \hat{\nabla}^{\text{naive}}_{\boldsymbol{\theta}})\right]\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right]. \tag{H.14}$$

We find that:

$$\alpha^*_{\boldsymbol{\mu}} = \frac{\boldsymbol{\sigma}_q^2}{\boldsymbol{\sigma}_p^2} \tag{H.15}$$

$$\alpha^*_{\boldsymbol{\sigma}} = \frac{2\boldsymbol{\sigma}_q^2}{\boldsymbol{\sigma}_p^2}\frac{(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^2 + \boldsymbol{\sigma}_q^2}{(\boldsymbol{\mu}_p - \boldsymbol{\mu}_q)^2 + 2\boldsymbol{\sigma}_q^2} \tag{H.16}$$

$$\text{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\left[\hat{\nabla}^{\text{naive}}_{\boldsymbol{\mu}_p} + \alpha^*_{\boldsymbol{\mu}}(\hat{\nabla}^{\text{GDReG}}_{\boldsymbol{\mu}_p} - \hat{\nabla}^{\text{naive}}_{\boldsymbol{\mu}_p})\right]\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = 0 \tag{H.17}$$

$$\text{Var}_{q_{\boldsymbol{\phi}}(\boldsymbol{z})}\left[\left[\hat{\nabla}^{\text{naive}}_{\boldsymbol{\sigma}_p} + \alpha^*_{\boldsymbol{\sigma}}(\hat{\nabla}^{\text{GDReG}}_{\boldsymbol{\sigma}_p} - \hat{\nabla}^{\text{naive}}_{\boldsymbol{\sigma}_p})\right]\mathcal{L}^{\text{ce}}_{\boldsymbol{\phi},\boldsymbol{\theta}}\right] = \frac{2\boldsymbol{\sigma}_q^4}{\boldsymbol{\sigma}_p^6}\frac{(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^2}{(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^2 + 2\boldsymbol{\sigma}_q^2} \tag{H.18}$$

Note that the expression for the optimal strength has different form for the mean $\boldsymbol{\mu}_p$ and variance $\boldsymbol{\sigma}_p$ parameters. Moreover, note that the analytic estimator has zero gradient variance whereas our estimator with control variate still has non-zero gradient variance for the variance parameters $\boldsymbol{\sigma}_p$.

This optimal estimator holds for a single layer VAE, where both the variational posterior as well as the prior are Gaussian. In a hierarchical model where both the prior and the posterior are factorized top-down, the same derivation holds for the lowest stochastic layer (similar to how semi-analytic approximations for the conditional KL can be derived in this case). Unfortunately, in this case the expectations cannot be computed in closed form anymore.