A. Approximate Posterior Gradients for Latent Gaussian Models

A.1. Model & Variational Objective

Consider a latent variable model, $p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})$, where the prior on $\mathbf{z}$ is a factorized Gaussian density, $p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_p, \operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2)$, and the conditional likelihood, $p_\theta(\mathbf{x}|\mathbf{z})$, depends on the type of data (e.g. Bernoulli for binary observations or Gaussian for continuous observations). We introduce an approximate posterior distribution, $q(\mathbf{z}|\mathbf{x})$, which can be any parametric probability density defined over real values. Here, we assume that q also takes the form of a factorized Gaussian density, $q(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)$. The objective during variational inference is to maximize $\mathcal{L}$ w.r.t. the parameters of $q(\mathbf{z}|\mathbf{x})$, i.e. $\boldsymbol{\mu}_q$ and $\boldsymbol{\sigma}_q^2$:

$$\boldsymbol{\mu}_q^*, \boldsymbol{\sigma}_q^{2*} = \underset{\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2}{\arg\max}\, \mathcal{L}. \tag{1}$$

To solve this optimization problem, we will use the gradients $\nabla_{\boldsymbol{\mu}_q} \mathcal{L}$ and $\nabla_{\boldsymbol{\sigma}_q^2} \mathcal{L}$, which we now derive. The objective can be written as:

$$\begin{aligned} \mathcal{L} &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}) \right] && (2) \\ &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log p_\theta(\mathbf{z}) - \log q(\mathbf{z}|\mathbf{x}) \right]. && (3) \end{aligned}$$

Plugging in $p_\theta(\mathbf{z})$ and $q(\mathbf{z}|\mathbf{x})$:

$$\mathcal{L} = \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[\log p_\theta(\mathbf{x}|\mathbf{z}) + \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_p, \operatorname{diag} \boldsymbol{\sigma}_p^2) - \log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2) \right] \tag{4}$$

Since expectation and differentiation are linear operators, we can take the expectation and derivative of each term individually.

A.2. Gradient of the Log-Prior

We can write the log-prior as:

$$\log \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_p, (\operatorname{diag} \boldsymbol{\sigma}_p^2) = -\frac{1}{2} \log \left((2\pi)^{n_\mathbf{z}} |\operatorname{diag} \boldsymbol{\sigma}_p^2| \right) - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_p)^\intercal (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1} (\mathbf{z} - \boldsymbol{\mu}_p), \tag{5}$$

where $n_\mathbf{z}$ is the dimensionality of $\mathbf{z}$. We want to evaluate the following terms:

$$\nabla_{\boldsymbol{\mu}_q} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[-\frac{1}{2} \log \left((2\pi)^{n_\mathbf{z}} |\operatorname{diag} \boldsymbol{\sigma}_p^2| \right) - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_p)^\intercal (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1} (\mathbf{z} - \boldsymbol{\mu}_p) \right] \tag{6}$$

and

$$\nabla_{\boldsymbol{\sigma}_q^2} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[-\frac{1}{2} \log \left((2\pi)^{n_\mathbf{z}} |\operatorname{diag} \boldsymbol{\sigma}_p^2| \right) - \frac{1}{2} (\mathbf{z} - \boldsymbol{\mu}_p)^\intercal (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1} (\mathbf{z} - \boldsymbol{\mu}_p) \right]. \tag{7}$$

To take these derivatives, we will use the reparameterization trick (Kingma & Welling, 2014; Rezende et al., 2014) to re-express $\mathbf{z} = \boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an auxiliary standard Gaussian variable, and $\odot$ denotes the element-wise product. We can now perform the expectations over $\boldsymbol{\epsilon}$, allowing us to bring the gradient operators inside the expectation brackets. The first term in eqs. 6 and 7 does not depend on $\boldsymbol{\mu}_q$ or $\boldsymbol{\sigma}_q^2$, so we can write:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\nabla_{\boldsymbol{\mu}_q} \left(-\frac{1}{2} (\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon} - \boldsymbol{\mu}_p)^\intercal (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1} (\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon} - \boldsymbol{\mu}_p) \right) \right] \tag{8}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\nabla_{\boldsymbol{\sigma}_q^2} \left(-\frac{1}{2} (\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon} - \boldsymbol{\mu}_p)^\intercal (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1} (\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon} - \boldsymbol{\mu}_p) \right) \right]. \tag{9}$$

To simplify notation, we define the following term:

$$\boldsymbol{\xi} \equiv (\operatorname{diag} \boldsymbol{\sigma}_p^2)^{-1/2} (\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon} - \boldsymbol{\mu}_p), \tag{10}$$

allowing us to rewrite eqs. 8 and 9 as:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\nabla_{\boldsymbol{\mu}_q} \left(-\frac{1}{2} \boldsymbol{\xi}^\intercal \boldsymbol{\xi} \right) \right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[-\frac{\partial \boldsymbol{\xi}^\intercal}{\partial \boldsymbol{\mu}_q} \boldsymbol{\xi} \right] \tag{11}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\nabla_{\boldsymbol{\sigma}_q^2}\left(-\frac{1}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi}\right)\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\frac{\partial\boldsymbol{\xi}^\intercal}{\partial\boldsymbol{\sigma}_q^2}\boldsymbol{\xi}\right]. \tag{12}$$

We must now find $\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\mu}_q}$ and $\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\sigma}_q^2}$:

$$\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\mu}_q} = \frac{\partial}{\partial\boldsymbol{\mu}_q}\left((\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p)\right) = (\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2} \tag{13}$$

and

$$\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\sigma}_q^2} = \frac{\partial}{\partial\boldsymbol{\sigma}_q^2}\left((\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p)\right) = (\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}\operatorname{diag}\frac{\boldsymbol{\epsilon}}{2\boldsymbol{\sigma}_q}, \tag{14}$$

where division is performed element-wise. Plugging eqs. 13 and 14 back into eqs. 11 and 12, we get:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\left((\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}\right)^\intercal(\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p)\right] \tag{15}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\left(\operatorname{diag}\frac{\boldsymbol{\epsilon}}{2\boldsymbol{\sigma}_q}\right)^\intercal\left((\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}\right)^\intercal(\operatorname{diag}\boldsymbol{\sigma}_p^2)^{-1/2}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p)\right]. \tag{16}$$

Putting everything together, we can express the gradients as:

$$\nabla_{\boldsymbol{\mu}_q}\mathbb{E}_{\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)}\left[\log\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_p,\operatorname{diag}\boldsymbol{\sigma}_p^2)\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\frac{\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right], \tag{17}$$

and

$$\nabla_{\boldsymbol{\sigma}_q^2}\mathbb{E}_{\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)}\left[\log\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_p,\operatorname{diag}\boldsymbol{\sigma}_p^2)\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\left(\operatorname{diag}\frac{\boldsymbol{\epsilon}}{2\boldsymbol{\sigma}_q}\right)^\intercal\frac{\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right]. \tag{18}$$

A.3. Gradient of the Log-Approximate Posterior

We can write the log-approximate posterior as:

$$\log\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2) = -\frac{1}{2}\log\left((2\pi)^{n_\mathbf{z}}|\operatorname{diag}\boldsymbol{\sigma}_q^2|\right) - \frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_q)^\intercal(\operatorname{diag}\boldsymbol{\sigma}_q^2)^{-1}(\mathbf{z}-\boldsymbol{\mu}_q), \tag{19}$$

where $n_\mathbf{z}$ is the dimensionality of $\mathbf{z}$. Again, we will use the reparameterization trick to re-express the gradients. However, notice what happens when plugging the reparameterized $\mathbf{z} = \boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon}$ into the second term of eq. 19:

$$-\frac{1}{2}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_q)^\intercal(\operatorname{diag}\boldsymbol{\sigma}_q^2)^{-1}(\boldsymbol{\mu}_q + \boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon} - \boldsymbol{\mu}_q) = -\frac{1}{2}\frac{(\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon})^\intercal(\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon})}{\boldsymbol{\sigma}_q^2} = -\frac{1}{2}\boldsymbol{\epsilon}^\intercal\boldsymbol{\epsilon}. \tag{20}$$

This term does not depend on $\boldsymbol{\mu}_q$ or $\boldsymbol{\sigma}_q^2$. Also notice that the first term in eq. 19 depends only on $\boldsymbol{\sigma}_q^2$. Therefore, the gradient of the entire term w.r.t. $\boldsymbol{\mu}_q$ is zero:

$$\nabla_{\boldsymbol{\mu}_q}\mathbb{E}_{\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)}\left[\log\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)\right] = \mathbf{0}. \tag{21}$$

The gradient w.r.t. $\boldsymbol{\sigma}_q^2$ is

$$\nabla_{\boldsymbol{\sigma}_q^2}\left(-\frac{1}{2}\log\left((2\pi)^{n_\mathbf{z}}|\operatorname{diag}\boldsymbol{\sigma}_q^2|\right)\right) = -\frac{1}{2}\nabla_{\boldsymbol{\sigma}_q^2}\left(\log|\operatorname{diag}\boldsymbol{\sigma}_q^2|\right) = -\frac{1}{2}\nabla_{\boldsymbol{\sigma}_q^2}\sum_j\log\sigma_{q,j}^2 = -\frac{\mathbf{1}}{2\boldsymbol{\sigma}_q^2}. \tag{22}$$

Note that the expectation has been dropped, as the term does not depend on the value of the sampled $\mathbf{z}$. Thus, the gradient of the entire term w.r.t. $\boldsymbol{\sigma}_q^2$ is:

$$\nabla_{\boldsymbol{\sigma}_q^2}\mathbb{E}_{\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)}\left[\log\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_q,\operatorname{diag}\boldsymbol{\sigma}_q^2)\right] = -\frac{\mathbf{1}}{2\boldsymbol{\sigma}_q^2}. \tag{23}$$

A.4. Gradient of the Log-Conditional Likelihood

The form of the conditional likelihood will depend on the data, e.g. binary, discrete, continuous, etc. Here, we derive the gradient for Bernoulli (binary) and Gaussian (continuous) conditional likelihoods.

Bernoulli Output Distribution The log of a Bernoulli output distribution takes the form:

$$\log \mathcal{B}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}) = (\log \boldsymbol{\mu}_\mathbf{x})^\intercal \mathbf{x} + (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal (\mathbf{1} - \mathbf{x}), \tag{24}$$

where $\boldsymbol{\mu}_\mathbf{x} = \boldsymbol{\mu}_\mathbf{x}(\mathbf{z}, \theta)$ is the mean of the output distribution. We drop the explicit dependence on $\mathbf{z}$ and θ to simplify notation. We want to compute the gradients

$$\nabla_{\boldsymbol{\mu}_q} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[(\log \boldsymbol{\mu}_\mathbf{x})^\intercal \mathbf{x} + (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal (\mathbf{1} - \mathbf{x}) \right] \tag{25}$$

and

$$\nabla_{\boldsymbol{\sigma}_q^2} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[(\log \boldsymbol{\mu}_\mathbf{x})^\intercal \mathbf{x} + (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal (\mathbf{1} - \mathbf{x}) \right]. \tag{26}$$

Again, we use the reparameterization trick to re-express the expectations, allowing us to bring the gradient operators inside the brackets. Using $\mathbf{z} = \boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon}$, eqs. 25 and 26 become:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\nabla_{\boldsymbol{\mu}_q} \left((\log \boldsymbol{\mu}_\mathbf{x})^\intercal \mathbf{x} + (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal (\mathbf{1} - \mathbf{x}) \right) \right] \tag{27}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\nabla_{\boldsymbol{\sigma}_q^2} \left((\log \boldsymbol{\mu}_\mathbf{x})^\intercal \mathbf{x} + (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal (\mathbf{1} - \mathbf{x}) \right) \right], \tag{28}$$

where $\boldsymbol{\mu}_\mathbf{x}$ is re-expressed as function of $\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, \boldsymbol{\epsilon}$, and θ. Distributing the gradient operators yields:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial (\log \boldsymbol{\mu}_\mathbf{x})^\intercal}{\partial \boldsymbol{\mu}_q} \mathbf{x} + \frac{\partial (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal}{\partial \boldsymbol{\mu}_q} (\mathbf{1} - \mathbf{x}) \right] \tag{29}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial (\log \boldsymbol{\mu}_\mathbf{x})^\intercal}{\partial \boldsymbol{\sigma}_q^2} \mathbf{x} + \frac{\partial (\log(\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}))^\intercal}{\partial \boldsymbol{\sigma}_q^2} (\mathbf{1} - \mathbf{x}) \right]. \tag{30}$$

Taking the partial derivatives and combining terms gives:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\mu}_q}^\intercal \frac{\mathbf{x}}{\boldsymbol{\mu}_\mathbf{x}} - \frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\mu}_q}^\intercal \frac{\mathbf{1} - \mathbf{x}}{\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}} \right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\mu}_q}^\intercal \frac{\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}}{\boldsymbol{\mu}_\mathbf{x} \odot (\mathbf{1} - \boldsymbol{\mu}_\mathbf{x})} \right] \tag{31}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\sigma}_q^2}^\intercal \frac{\mathbf{x}}{\boldsymbol{\mu}_\mathbf{x}} - \frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\sigma}_q^2}^\intercal \frac{\mathbf{1} - \mathbf{x}}{\mathbf{1} - \boldsymbol{\mu}_\mathbf{x}} \right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon}; \mathbf{0}, \mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_\mathbf{x}}{\partial \boldsymbol{\sigma}_q^2}^\intercal \frac{\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}}{\boldsymbol{\mu}_\mathbf{x} \odot (\mathbf{1} - \boldsymbol{\mu}_\mathbf{x})} \right]. \tag{32}$$

Gaussian Output Density The log of a Gaussian output density takes the form:

$$\log \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_\mathbf{x}, \operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2) = -\frac{1}{2} \log \left((2\pi)^{n_\mathbf{x}} |\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2| \right) - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\intercal (\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2)^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}), \tag{33}$$

where $\boldsymbol{\mu}_\mathbf{x} = \boldsymbol{\mu}_\mathbf{x}(\mathbf{z}, \theta)$ is the mean of the output distribution and $\boldsymbol{\sigma}_\mathbf{x}^2 = \boldsymbol{\sigma}_\mathbf{x}^2(\theta)$ is the variance. We assume $\boldsymbol{\sigma}_\mathbf{x}^2$ is not a function of $\mathbf{z}$ to simplify the derivation, however, using $\boldsymbol{\sigma}_\mathbf{x}^2 = \boldsymbol{\sigma}_\mathbf{x}^2(\mathbf{z}, \theta)$ is possible and would simply result in additional gradient terms in $\nabla_{\boldsymbol{\mu}_q} \mathcal{L}$ and $\nabla_{\boldsymbol{\sigma}_q^2} \mathcal{L}$. We want to compute the gradients

$$\nabla_{\boldsymbol{\mu}_q} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[-\frac{1}{2} \log \left((2\pi)^{n_\mathbf{x}} |\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2| \right) - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\intercal (\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2)^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) \right] \tag{34}$$

and

$$\nabla_{\boldsymbol{\sigma}_q^2} \mathbb{E}_{\mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \operatorname{diag} \boldsymbol{\sigma}_q^2)} \left[-\frac{1}{2} \log \left((2\pi)^{n_\mathbf{x}} |\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2| \right) - \frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x})^\intercal (\operatorname{diag} \boldsymbol{\sigma}_\mathbf{x}^2)^{-1} (\mathbf{x} - \boldsymbol{\mu}_\mathbf{x}) \right]. \tag{35}$$

The first term in eqs. 34 and 35 is zero, since $\boldsymbol{\sigma}_{\mathbf{x}}^2$ does not depend on $\boldsymbol{\mu}_q$ or $\boldsymbol{\sigma}_q^2$. To take the gradients, we will again use the reparameterization trick to re-express $\mathbf{z} = \boldsymbol{\mu}_q + \boldsymbol{\sigma}_q \odot \boldsymbol{\epsilon}$. We now implicitly express $\boldsymbol{\mu}_{\mathbf{x}}$ as $\boldsymbol{\mu}_{\mathbf{x}}(\boldsymbol{\mu}_q, \boldsymbol{\sigma}_q^2, \theta)$. We can then write:

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\nabla_{\boldsymbol{\mu}_q}\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})^\intercal(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right)\right] \tag{36}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\nabla_{\boldsymbol{\sigma}_q^2}\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})^\intercal(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right)\right]. \tag{37}$$

To simplify notation, we define the following term:

$$\boldsymbol{\xi} \equiv (\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}), \tag{38}$$

allowing us to rewrite eqs. 36 and 37 as

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\nabla_{\boldsymbol{\mu}_q}\left(-\frac{1}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi}\right)\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\frac{\partial\boldsymbol{\xi}^\intercal}{\partial\boldsymbol{\mu}_q}\boldsymbol{\xi}\right] \tag{39}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\nabla_{\boldsymbol{\sigma}_q^2}\left(-\frac{1}{2}\boldsymbol{\xi}^\intercal\boldsymbol{\xi}\right)\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[-\frac{\partial\boldsymbol{\xi}^\intercal}{\partial\boldsymbol{\sigma}_q^2}\boldsymbol{\xi}\right]. \tag{40}$$

We must now find $\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\mu}_q}$ and $\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\sigma}_q^2}$:

$$\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\mu}_q} = \frac{\partial}{\partial\boldsymbol{\mu}_q}\left((\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right) = -(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\mu}_q} \tag{41}$$

and

$$\frac{\partial\boldsymbol{\xi}}{\partial\boldsymbol{\sigma}_q^2} = \frac{\partial}{\partial\boldsymbol{\sigma}_q^2}\left((\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right) = -(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\sigma}_q^2}. \tag{42}$$

Plugging these expressions back into eqs. 39 and 40 gives

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\mu}_q}^\intercal((\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2})^\intercal(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\mu}_q}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\sigma}_{\mathbf{x}}^2}\right] \tag{43}$$

and

$$\mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\sigma}_q^2}^\intercal((\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2})^\intercal(\operatorname{diag}\boldsymbol{\sigma}_{\mathbf{x}}^2)^{-1/2}(\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}})\right] = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\sigma}_q^2}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\sigma}_{\mathbf{x}}^2}\right]. \tag{44}$$

Despite having different distribution forms, Bernoulli and Gaussian output distributions result in approximate posterior gradients of a similar form: the Jacobian of the output model multiplied by a weighted error term.

A.5. Summary

Putting the gradient terms from $\log p_\theta(\mathbf{x}|\mathbf{z})$, $\log p_\theta(\mathbf{z})$, and $\log q(\mathbf{z}|\mathbf{x})$ together, we arrive at

Bernoulli Output Distribution:

$$\nabla_{\boldsymbol{\mu}_q}\mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\mu}_q}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\mu}_{\mathbf{x}}\odot(\mathbf{1}-\boldsymbol{\mu}_{\mathbf{x}})} - \frac{\boldsymbol{\mu}_q+\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon}-\boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right] \tag{45}$$

$$\nabla_{\boldsymbol{\sigma}_q^2}\mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\sigma}_q^2}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\mu}_{\mathbf{x}}\odot(\mathbf{1}-\boldsymbol{\mu}_{\mathbf{x}})} - \left(\operatorname{diag}\frac{\boldsymbol{\epsilon}}{2\boldsymbol{\sigma}_q}\right)^\intercal\frac{\boldsymbol{\mu}_q+\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon}-\boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right] - \frac{\mathbf{1}}{2\boldsymbol{\sigma}_q^2} \tag{46}$$

Gaussian Output Distribution:

$$\nabla_{\boldsymbol{\mu}_q}\mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\mu}_q}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\sigma}_{\mathbf{x}}^2} - \frac{\boldsymbol{\mu}_q+\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon}-\boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right] \tag{47}$$

$$\nabla_{\boldsymbol{\sigma}_q^2}\mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})}\left[\frac{\partial\boldsymbol{\mu}_{\mathbf{x}}}{\partial\boldsymbol{\sigma}_q^2}^\intercal\frac{\mathbf{x}-\boldsymbol{\mu}_{\mathbf{x}}}{\boldsymbol{\sigma}_{\mathbf{x}}^2} - \left(\operatorname{diag}\frac{\boldsymbol{\epsilon}}{2\boldsymbol{\sigma}_q}\right)^\intercal\frac{\boldsymbol{\mu}_q+\boldsymbol{\sigma}_q\odot\boldsymbol{\epsilon}-\boldsymbol{\mu}_p}{\boldsymbol{\sigma}_p^2}\right] - \frac{\mathbf{1}}{2\boldsymbol{\sigma}_q^2} \tag{48}$$

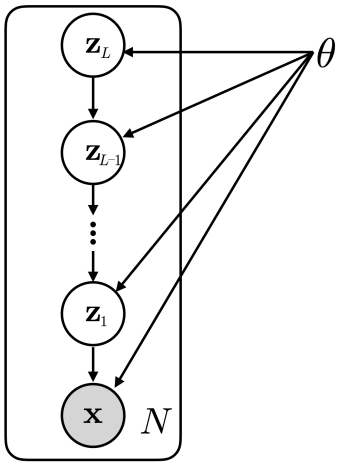

Figure 1. Plate notation for a hierarchical latent variable model consisting of L levels of latent variables. Variables at higher levels provide empirical priors on variables at lower levels. With data-dependent priors, the model has more flexibility.

A.6. Approximate Posterior Gradients in Hierarchical Latent Variable Models

Hierarchical latent variable models factorize the latent variables over multiple levels, $\mathbf{z} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_L\}$. Latent variables at higher levels provide *empirical priors* on latent variables at lower levels. Here, we assume a first-order Markov graphical structure, as shown in Figure 1, though more general structures are possible. For an intermediate latent level, we use the notation $q(\mathbf{z}_\ell|\cdot) = \mathcal{N}(\mathbf{z}_\ell; \boldsymbol{\mu}_{\ell,q}, \operatorname{diag} \boldsymbol{\sigma}^2_{\ell,q})$ and $p(\mathbf{z}_\ell|\mathbf{z}_{\ell+1}) = \mathcal{N}(\mathbf{z}_\ell; \boldsymbol{\mu}_{\ell,p}, \operatorname{diag} \boldsymbol{\sigma}^2_{\ell,p})$ to denote the approximate posterior and prior respectively. Analogously to the case of a Gaussian output density in a one-level model, the approximate posterior gradients at an intermediate level ℓ are:

$$\nabla_{\boldsymbol{\mu}_{q,\ell}} \mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_{\ell-1,p}}{\partial \boldsymbol{\mu}_{\ell,q}}^\intercal \frac{\boldsymbol{\mu}_{\ell-1,q} + \boldsymbol{\sigma}_{\ell-1,q} \odot \boldsymbol{\epsilon}_{\ell-1} - \boldsymbol{\mu}_{\ell-1,p}}{\boldsymbol{\sigma}^2_{\ell-1,p}} - \frac{\boldsymbol{\mu}_{\ell,q} + \boldsymbol{\sigma}_{\ell,q} \odot \boldsymbol{\epsilon}_\ell - \boldsymbol{\mu}_{\ell,p}}{\boldsymbol{\sigma}^2_{\ell,p}} \right], \tag{49}$$

$$\nabla_{\boldsymbol{\sigma}^2_q} \mathcal{L} = \mathbb{E}_{\mathcal{N}(\boldsymbol{\epsilon};\mathbf{0},\mathbf{I})} \left[\frac{\partial \boldsymbol{\mu}_{\ell-1,p}}{\partial \boldsymbol{\sigma}^2_{\ell,q}}^\intercal \frac{\boldsymbol{\mu}_{\ell-1,q} + \boldsymbol{\sigma}_{\ell-1,q} \odot \boldsymbol{\epsilon}_{\ell-1} - \boldsymbol{\mu}_{\ell-1,p}}{\boldsymbol{\sigma}^2_{\ell-1,p}} - \left(\operatorname{diag} \frac{\boldsymbol{\epsilon}_\ell}{2\boldsymbol{\sigma}_{\ell,q}} \right)^\intercal \frac{\boldsymbol{\mu}_{\ell,q} + \boldsymbol{\sigma}_{\ell,q} \odot \boldsymbol{\epsilon}_\ell - \boldsymbol{\mu}_{\ell,p}}{\boldsymbol{\sigma}^2_{\ell,p}} \right] - \frac{\mathbf{1}}{2\boldsymbol{\sigma}^2_{\ell,q}}. \tag{50}$$

The first terms inside each expectation are "bottom-up" gradients coming from reconstruction errors at the level below. The second terms inside the expectations are "top-down" gradients coming from priors generated by the level above. The last term in the variance gradient acts to reduce the entropy of the approximate posterior.

B. Implementing Iterative Inference Models

Here, we provide specific implementation details for these models. Code for reproducing the experiments will be released online.

B.1. Input Form

Approximate posterior gradients and errors experience distribution shift during inference and training. Using these terms as inputs to a neural network can slow down and prevent training. For experiments on MNIST, we found the $\log$ transformation method proposed by (Andrychowicz et al., 2016) to work reasonably well: replacing $\nabla_{\boldsymbol{\lambda}} \mathcal{L}$ with the concatenation of $[\alpha \log(|\nabla_{\boldsymbol{\lambda}} \mathcal{L}| + \epsilon), \operatorname{sign}(\nabla_{\boldsymbol{\lambda}} \mathcal{L})]$, where α is a scaling constant and ϵ is a small constant for numerical stability. We also encode the current estimates of $\boldsymbol{\mu}_q$ and $\log \boldsymbol{\sigma}^2_q$. For experiments on CIFAR-10, we instead used layer normalization (Ba et al., 2016) to normalize each input to the iterative inference model. This normalizes each input over the non-batch dimension.

Algorithm 1 Iterative Amortized Inference

Input: data $\mathbf{x}$, generative model $p_\theta(\mathbf{x}, \mathbf{z})$, inference model f
Initialize $t = 0$
Initialize $\nabla_\phi = 0$
Initialize $q(\mathbf{z}|\mathbf{x})$ with $\boldsymbol{\lambda}_0$
repeat
  Sample $\mathbf{z} \sim q(\mathbf{z}|\mathbf{x})$
  Evaluate $\mathcal{L}_t = \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}_t; \theta)$
  Calculate $\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t$ and $\nabla_\phi \mathcal{L}_t$
  Update $\boldsymbol{\lambda}_{t+1} = f_t(\nabla_{\boldsymbol{\lambda}} \mathcal{L}_t, \boldsymbol{\lambda}_t; \phi)$
  $t = t + 1$
  $\nabla_\phi = \nabla_\phi + \nabla_\phi \mathcal{L}_t$
until $\mathcal{L}$ converges
$\theta = \theta + \alpha_\theta \nabla_\theta \mathcal{L}$
$\phi = \phi + \alpha_\phi \nabla_\phi$

B.2. Output Form

For the output of these models, we use a gated updating scheme, where approximate posterior parameters are updated according to

$$\boldsymbol{\lambda}_{t+1} = \mathbf{g}_t \odot \boldsymbol{\lambda}_t + (\mathbf{1} - \mathbf{g}_t) \odot f_t(\nabla_{\boldsymbol{\lambda}} \mathcal{L}, \boldsymbol{\lambda}_t; \phi). \tag{51}$$

Here, $\odot$ represents element-wise multiplication and $\mathbf{g}_t = g_t(\nabla_{\boldsymbol{\lambda}} \mathcal{L}, \boldsymbol{\lambda}_t; \phi) \in [0, 1]$ is the gating function for $\boldsymbol{\lambda}$ at time t, which we combine with the iterative inference model f_t. We found that this yielded improved performance and stability over the additive updating scheme used in (Andrychowicz et al., 2016).

B.3. Training

To train iterative inference models for latent Gaussian models, we use stochastic estimates of $\nabla_\phi \mathcal{L}$ from the reparameterization trick. We accumulate these gradient estimates during inference, then update both ϕ and θ jointly. We train using a fixed number of inference iterations.

C. Experiment Details

Inference model and generative model parameters (ϕ and θ) were trained jointly using the adam optimizer (Kingma & Ba, 2014). The learning rate was set to 0.0002 for both sets of parameters and all other optimizer parameters were set to their default values. Learning rates were decayed exponentially by a factor of 0.999 each epoch. All models utilized exponential linear unit (ELU) activation functions (Clevert et al., 2015), although we found other non-linearities to work as well. Unless otherwise stated, all inference models were symmetric to their corresponding generative models. Iterative inference models for all experiments were implemented as feed-forward networks to make comparison with standard inference models easier.

C.1. Two-Dimensional Latent Gaussian Models

We trained models with 2 latent dimensions and a point estimate approximate posterior. That is, $q(\mathbf{z}|\mathbf{x}) = \delta(\mathbf{z} = \boldsymbol{\mu}_q)$ is a Dirac delta function at the point $\boldsymbol{\mu}_q = (\mu_1, \mu_2)$. We trained these models on binarized MNIST. The generative models consisted of a neural network with 2 hidden layers, each with 512 units. The output of the generative model was the mean of a Bernoulli distribution. The optimization surface of each model was evaluated on a grid of range [-5, 5] in increments of 0.05 for each latent variable. The iterative inference model shown in Figure 3 encodes $\mathbf{x}$, $\boldsymbol{\varepsilon}_\mathbf{x}$, and $\boldsymbol{\varepsilon}_\mathbf{z}$.

C.2. $\mathcal{L}$ During Inference

We trained one-level models on MNIST using iterative inference models that encode gradients ($\nabla_{\boldsymbol{\lambda}} \mathcal{L}$) for 16 iterations. We compared against stochastic gradient descent (SGD), SGD with momentum, RMSProp, and Adam, using learning rates in $\{0.5, 0.4, 0.3, 0.2, 0.1, 0.01, 0.001\}$ and taking the best result. In addition to performance over iterations, we also compared the optimization techniques on the basis of wall clock time. Despite requiring more time per inference iteration, we observed

that the iterative inference model still outperformed the conventional optimization techniques.

C.3. Reconstructions Over Inference Iterations

We trained iterative inference models on MNIST, Omniglot, and SVHN by encoding approximate posterior gradients ($\nabla_{\boldsymbol{\lambda}}\mathcal{L}$) for 16 iterations. For CIFAR-10, we trained an iterative inference model by encoding errors for 10 inference iterations. For MNIST and Omniglot, we used a generative model architecture with 2 hidden layers, each with 512 units, a latent space of size 64, and a symmetric iterative inference model. For SVHN and CIFAR-10, we used 3 hidden layers in the iterative inference and 1 in the generative model, with 2,048 units at each hidden layer and a latent space of size 1,024.

C.4. Gradient Magnitudes

While training iterative inference models, we recorded approximate posterior gradient magnitudes at each inference iteration. We observed that, on average, the magnitudes decreased during inference optimization. This decrease was more prevalent for the approximate posterior mean gradients. For Figure 6, we trained an iterative inference model on RCV1 by encoding gradients ($\nabla_{\boldsymbol{\lambda}}\mathcal{L}$) for 16 inference iterations. The generative model contained a latent variable of size 512 and 2 fully-connected layers of 512 units each. The inference model was symmetric.

C.5. Additional Inference Iterations

We used an architecture of 2 hidden layers, each with 512 units, for the output model and inference models. The latent variable contained 64 dimensions. We trained all models for 1,500 epochs. We were unable to run multiple trials for each experimental set-up, but on a subset of runs for standard and iterative inference models, we observed that final performance had a standard deviation less than 0.1 nats, below the difference in performance between models trained with different numbers of inference iterations.

C.6. Additional Latent Samples

We used an architecture of 2 hidden layers, each with 512 units, for the output model and inference models. The latent variable contained 64 dimensions. Each model was trained by drawing the corresponding number of samples from the approximate posterior distribution to obtain ELBO estimates and gradients. Iterative inference models were trained by encoding the data ($\mathbf{x}$) and the approximate posterior gradients ($\nabla_{\boldsymbol{\lambda}}\mathcal{L}$) for 5 inference iterations. All models were trained for 1,500 epochs.

C.7. Comparison with Standard Inference Models

C.7.1. MNIST

For MNIST, one-level models consisted of a latent variable of size 64, and the inference and generative networks both consisted of 2 hidden layers, each with 512 units. Hierarchical models consisted of 2 levels with latent variables of size 64 and 32 in hierarchically ascending order. At each level, the inference and generative networks consisted of 2 hidden layers, with 512 units at the first level and 256 units at the second level. At the first level of latent variables, we also used a set of deterministic units, also of size 64, in both the inference and generative networks. Hierarchical models included batch normalization layers at each hidden layer of the inference and generative networks; we found this beneficial for training both standard and iterative inference models. Both encoder and decoder networks in the hierarchical model utilized highway skip connections at each layer at both levels. Iterative models were trained by encoding data and errors for 5 inference iterations.

C.7.2. CIFAR-10

For CIFAR-10, one-level models consisted of a latent variable of size 1,024, an encoder network with 3 hidden layers of 2,048 units, and a decoder network with 1 hidden layer with 2,048 units. We found this set-up performed better than a symmetric encoder and decoder for both standard and iterative inference models. Hierarchical models were the same as the one-level model, adding another latent variable of size 512, with another 3 layer encoder of with 1,024 units and a 1 layer decoder with 1,024 units. Both encoder and decoder networks in the hierarchical model utilized highway skip connections at each layer at both levels. Models were all trained for 150 epochs. We annealed the KL-divergence term during the first 50 epochs when training hierarchical models. Iterative inference models were trained by encoding the data and gradients for 5

inference iterations.

C.7.3. RCV1

We followed the same processing procedure as (Krishnan et al., 2017), encoding data using normalized TF-IDF features. For encoder and decoder, we use 2-layer networks, each with 2,048 units and ELU non-linearities. We use a latent variable of size 1,024. The iterative inference model was trained by encoding gradients for 10 steps. Both models were trained using 5 approximate posterior samples at each iteration. We evaluate the models by reporting perplexity on the test set (Table 2). Perplexity, P, is defined as

$$P \equiv \exp(-\frac{1}{N}\sum_i \frac{1}{N_i}\log p(\mathbf{x}^{(i)})), \tag{52}$$

where N is the number of examples and N_i is the total number of word counts in example i. We evaluate perplexity by estimating each $\log p(\mathbf{x}^{(i)})$ with 5,000 importance weighted samples. We also report an upper bound on perplexity using $\mathcal{L}$.

References

Andrychowicz, Marcin, Denil, Misha, Gomez, Sergio, Hoffman, Matthew W, Pfau, David, Schaul, Tom, and de Freitas, Nando. Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems*, pp. 3981–3989, 2016.

Ba, Jimmy Lei, Kiros, Jamie Ryan, and Hinton, Geoffrey E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Clevert, Djork-Arné, Unterthiner, Thomas, and Hochreiter, Sepp. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

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

Kingma, Diederik P and Welling, Max. Stochastic gradient vb and the variational auto-encoder. In *Second International Conference on Learning Representations, ICLR*, 2014.

Krishnan, Rahul G, Liang, Dawen, and Hoffman, Matthew. On the challenges of learning with inference networks on sparse, high-dimensional data. *arXiv preprint arXiv:1710.06085*, 2017.

Rezende, Danilo Jimenez, Mohamed, Shakir, and Wierstra, Daan. Stochastic backpropagation and approximate inference in deep generative models. *Proceedings of the 31st International Conference on Machine Learning*, pp. 1278–1286, 2014.

Figure 2. Reconstructions over inference iterations (left to right) for examples from (top to bottom) MNIST, Omniglot, SVHN, and CIFAR-10. Corresponding data examples are shown on the right of each panel.