SUPPLEMENTARY FILE:
The Usual Suspects?
Reassessing Blame for VAE Posterior Collaps

This document contains companion technical material regarding our ICML 2020 submission. Note that herein all equation numbers referencing back to the main submission document will be be prefixed with an 'M' to avoid confusion, i.e, (M.#) will refer to equation (#) from the main text. Similar notation differentiates sections, tables, and figures, e.g., Section M.#, etc.

1. Network Structure, Experimental Settings, and Additional Results

Three different kinds of network structures were used in the experiments: fully connected networks, convolution networks, and residual networks. For all these structures, we set the dimension of the latent variable $\boldsymbol{z}$ to 64. We now describe the network details accordingly.

Fully Connected Network: This experiment is only applied on the simple Fashion-MNIST dataset, which contains 60000 28×28 black-and-while images. These images are first flattened to a 784 dimensional vector. Both the encoder and decoder have multiple number of 512-dimensional hidden layers, each followed by ReLU activations.

Convolution Network: The original images are either $32 \times 32 \times 3$ (Cifar10, Cifar100 and SVHN) or $64 \times 64 \times 3$ (CelebA and ImageNet). In the encoder, we use a multiple number (denoted as t) of 3×3 convolution layers for each spatial scale. Each convolution layer is followed by a ReLU activation. Then we use a 2×2 max pooling to downsample the feature map to a smaller spatial scale. The number of channels is doubled when the spatial scale is halved. We use 64 channels when the spatial scale is 32×32. When the spatial scale reaches 4×4 (there should be 512 channels in this feature map), we use an average pooling to transform the feature map to a vector, which is then transformed into the latent variable using a fully connected layer. In the decoder, the latent variable is first transformed to a 4096-dimensional vector using a fully connected layer and then reshaped to $2 \times 2 \times 1024$. Again in each spatial scale, we use 1 transpose convolution layer to upscale the feature map and halve the number of channels followed by $t - 1$ convolution layers. Each convolution and transpose convolution layer is followed by a ReLU activation layer. When the spatial scale reaches that of the original image, we use a convolution layer to transofrm the feature map to 3 channels.

Residual Network: The network structure of the residual network is similar to that of a convolution network described above. We simply replace the convolution layer with a residual block. Inside the residual block, we use different numbers of convolution numbers. (The typical number of convolution layers inside a residual block is 2 or 3. In our experiments, we try 2, 3, 4 and 5.)

Training Details: All the experiments with different network structures and datasets are trained in the same procedure. We use the Adam optimization method and the default optimizer hyper parameters in Tensorflow. The batch size is 64 and we train the model for $250K$ iterations. The initial learning rate is 0.0002 and it is halved every $100K$ iterations.

Additional Results on ImageNet: We also show the reconstruction error for convolution networks with increasing depth trained on ImageNet in Figure 1. The trend is the same as that in Figure M.1.

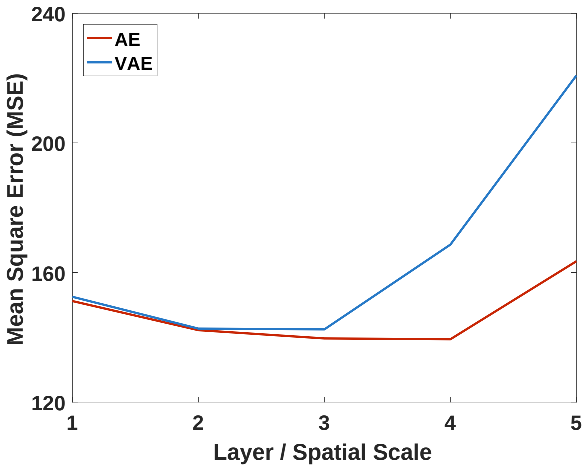

Figure 1: Reconstruction error for Convolution networks with increasing depth/# of spatial scales trained on ImageNet.

Additional Results with Error Bars: As suggested by a reviewer, we re-run the experiments from Figure M.1(*top*) for 5 trials and plot the mean reconstruction error with error bars in Figure 2. The basic trend is the same as that in the main paper, further supporting our conclusions.

Additional FID Score Evaluations: To complement the MSE-based reconstruction errors from all previous experiments, we compare fully connected networks (i.e., as in Figure M.1(*top*)) with different depth using the FID score, a metric that is widely believed to be at least somewhat reflective of perceptual realism. The FID scores of model recon-

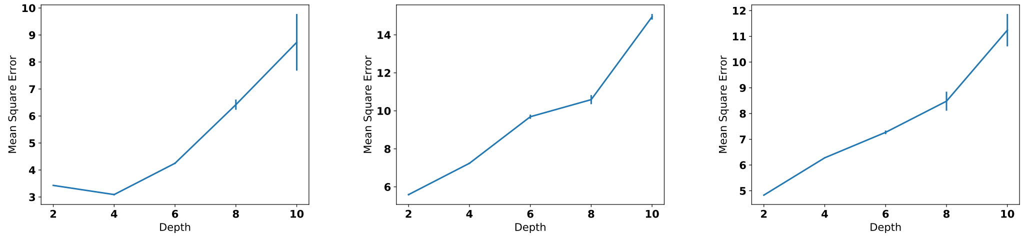

Figure 2: Reconstruction error with error bars corresponding to Figure M.1(*top*). The models are AE, VAE and VAE with KL annealing from left to right.

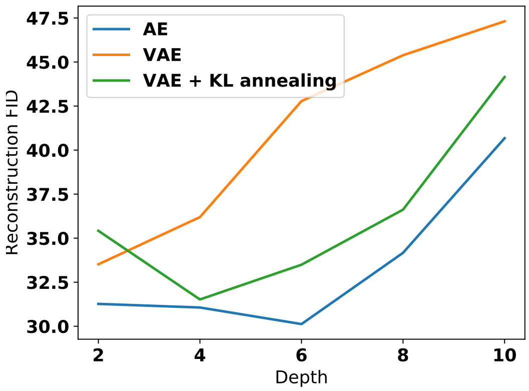

Figure 3: FID scores of model reconstructions. The trend is similar as when using MSE-based reconstruction error.

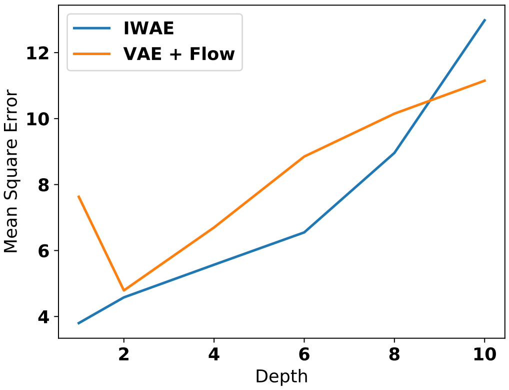

Figure 4: Reconstruction error for fully connected networks with increasing depth using an IWAE and a VAE with Sylvestor normalizing flows.

structions are shown in Figure 3. The trend is similar to that of the reconstruction MSE as expected.

Additional Results with More Flexible Non-Gaussian VAE Variants: We validate that non-Gaussian VAE models will behave similarly to Gaussian baselines as depth increases, again using the same setup from Figure M.1(*top*). We try two non-Gaussian VAE variants: an importance weighted autoencoder (IWAE) [1] and a VAE with Sylvestor normalizing flows [3]. The reconstruction errors (MSE-based) for both models are shown in Figure 4. From depth 2 onwards to depth 10, the errors monotonically increase from 4.58 to 12.98 for the IWAE, and from 4.79 to 11.15 for the flow model.

2. Proof of Proposition M.1

While the following analysis could in principle be extended to more complex datasets, for our purposes it is sufficient to consider the following simplified case for ease of exposition. Specifically, we assume that $n > 1, d > \kappa$, set $d = 2, n = 2, \kappa = 1$, and $\boldsymbol{x}^{(1)} = (1, 1), \boldsymbol{x}^{(2)} = (-1, -1)$.

Additionally, we will use the following basic facts about the Gaussian tail. Note that (2)-(3) below follow from integration by parts; see [2].

Lemma 1 *Let $\epsilon \sim \mathcal{N}(0, 1), A > 0$; $\phi(x), \Phi(x)$ be the pdf and cdf of the standard normal distribution, respectively. Then*

$$\begin{aligned} 1 - \Phi(A) &\leq e^{-A^2/2}, && (1)\\ \mathbb{E}[\epsilon \mathbf{1}_{\{\epsilon > A\}}] &= \phi(A), && (2)\\ \mathbb{E}[\epsilon^2 \mathbf{1}_{\{\epsilon > A\}}] &= 1 - \Phi(A) + A\phi(A). && (3) \end{aligned}$$

2.1 Suboptimality of (M.6)

Under the specificed conditions, the energy from (M.6) has a value of nd. Thus to show that it is not the global minimum, it suffices to show that the following VAE, parameterized by δ, has energy $\to -\infty$ as $\delta \to 0$:

$$\begin{aligned} &\mu_z^{(1)} = 1, \mu_z^{(2)} = -1,\\ &\boldsymbol{W}_x = (\alpha + 1, \alpha + 1), \boldsymbol{b}_x = 0,\\ &\sigma_z^{(1)} = \sigma_z^{(2)} = \delta,\\ &\quad \gamma = \mathbb{E}_{\mathcal{N}(\varepsilon|0,1)} 2(1 - \pi_\alpha((\alpha + 1)(1 + \delta\varepsilon)))^2. \end{aligned}$$

This follows because, given the stated parameters, we have that

$$\begin{aligned} \mathcal{L}(\theta, \phi) =& \sum_{i=1}^{2} (1 + 2 \log \mathbb{E}_{\mathcal{N}(\varepsilon|0,1)} 2(1 - \pi_\alpha((\alpha + 1)(1 + \delta\varepsilon)))^2 - 2 \log \delta + \delta^2 + 1)\\ =& \sum_{i=1}^{2} (\Theta(1) + 2 \log \mathbb{E}_{\mathcal{N}(\varepsilon|0,1)} (1 - \pi_\alpha(\alpha + 1 + (\alpha + 1)\delta\varepsilon))^2 - 2 \log \delta)\\ \leq^{(i)}& 4 \log \delta + \Theta(1). \end{aligned}$$

(i) holds when $\delta < \frac{1}{\alpha+1}$; to see this, denote $x := \alpha + 1 + (\alpha + 1)(\delta\varepsilon)$. Then

$$\begin{aligned} &\mathbb{E}_{\mathcal{N}(\varepsilon|0,1)} (1 - \pi_\alpha(x))^2\\ =& \mathbb{E}_\varepsilon[(1 - \pi_\alpha(x))^2 \mathbf{1}_{\{x \geq \alpha\}}] + \mathbb{E}_\varepsilon[(1 - \pi_\alpha(x))^2 \mathbf{1}_{\{|x| < \alpha\}}] + \mathbb{E}_\varepsilon[(1 - \pi_\alpha(x))^2 \mathbf{1}_{\{x < -\alpha\}}]\\ \leq& \underbrace{\mathbb{E}_\varepsilon[(1 - (x - \alpha))^2]}_{(a)} + \underbrace{\mathbb{P}(|x| < \alpha)}_{(b)} + \underbrace{\mathbb{E}_\varepsilon((1 - x - \alpha)^2 \mathbf{1}_{\{x < -\alpha\}})}_{(c)}. \end{aligned}$$

In the RHS above $(a) = [(\alpha+1)\delta]^2$; using (1)-(3) we then have

$$\begin{aligned}
(b) &< \mathbb{P}(x < \alpha) = \mathbb{P}\left(\varepsilon < \frac{-1}{(\alpha+1)\delta}\right) \le \exp\left(-\frac{1}{2[(\alpha+1)\delta]^2}\right). \\
(c) &< \mathbb{E}_\varepsilon((2\alpha + (\alpha+1)\delta\varepsilon)^2 \mathbf{1}_{\{x<\alpha\}}) \\
&= \int_{-\infty}^{\frac{-1}{(\alpha+1)\delta}} (2\alpha + (\alpha+1)\delta\varepsilon)^2 \frac{1}{\sqrt{2\pi}} e^{-\varepsilon^2/2} d\varepsilon \\
&< \int_{-\infty}^{\frac{-1}{(\alpha+1)\delta}} (4\alpha^2 + [(\alpha+1)\delta\varepsilon]^2) \frac{1}{\sqrt{2\pi}} e^{-\varepsilon^2/2} d\varepsilon \\
&< \left\{4\alpha^2 + ((\alpha+1)\delta)^2 \left[1 + \frac{1}{\sqrt{2\pi}}\right]\right\} \exp\left(-\frac{1}{2[(\alpha+1)\delta]^2}\right)
\end{aligned}$$

when $\delta < \frac{1}{\alpha+1}$. Thus

$$\lim_{\delta\to 0} \frac{\mathbb{E}_{\mathcal{N}(\varepsilon|0,1)}(1-\pi_\alpha(x))^2}{[(\alpha+1)\delta]^2} = 1,$$

and

$$\lim_{\delta\to 0} \{\log \mathbb{E}_{\mathcal{N}(\varepsilon|0,1)}(1-\pi_\alpha(x))^2 - 2\log\delta\} = 2\log(\alpha+1),$$

or

$$2\log \mathbb{E}_\epsilon(1-\pi_\alpha(x))^2 = 4\log\delta + \Theta(1),$$

and we can see (i) holds.

2.2 Local Optimality of (M.6)

We will now show that at (M.6), the Hessian of the energy has structure

$$\begin{array}{ccccc}
 & (\boldsymbol{W}_x) & (\boldsymbol{b}_x) & (\sigma_z^{(i)}, \mu_z^{(i)}) & (\gamma) \\
(\boldsymbol{W}_x) & 0 & 0 & 0 & 0 \\
(\boldsymbol{b}_x) & 0 & \frac{2}{\gamma}I & 0 & 0 \\
(\sigma_z^{(i)}, \mu_z^{(i)}) & 0 & 0 & (p.d.) & 0 \\
(\gamma) & 0 & 0 & 0 & (p.d.)
\end{array}$$

where p.d. means the corresponding submatrix is positive definite and independent of other parameters. While the Hessian is 0 in the subspace of $\boldsymbol{W}_x$, we can show that for VAEs that are only different from (M.6) by $\boldsymbol{W}_x$, the gradient always points back to (M.6). Thus (M.6) is a strict local minima.

First we compute the Hessian matrix block-wise. We will identify $\boldsymbol{W}_x \in \mathbb{R}^{2\times 1}$ with the vector $(W_j)_{j=1}^2$, and use the shorthand notations $\boldsymbol{x}^{(i)} = (x_j^{(i)})_{j=1}^2$, $\boldsymbol{b}_x = (b_j)_{j=1}^2$, $z^{(i)} = \mu_z^{(i)} + \sigma_z^{(i)}\varepsilon$, where $\varepsilon \sim \mathcal{N}(0,1)$ (recall that $z^{(i)}$ is a scalar in this proof).

1. The second-order derivatives involving $\boldsymbol{W}_x$ can be expressed as

$$\frac{\partial \mathcal{L}}{\partial W_j} = \frac{-2}{\gamma} \sum_{i=1}^n \mathbb{E}_\varepsilon[(\pi'_\alpha(W_j z^{(i)}) z^{(i)}) \cdot (x_j^{(i)} - \pi_\alpha(W_j z^{(i)}) - b_j)], \tag{4}$$

and therefore all second-order derivatives involving W_j will have the form

$$\mathbb{E}_\epsilon[\pi'_\alpha(W_j z^{(i)})F_1 + \pi''_\alpha(W_j z^{(i)})F_2], \tag{5}$$

where F_1, F_2 are some arbitrary functions that are finite at (M.6). Since $\pi'_\alpha(0) = \pi''_\alpha(0) = W_j = 0$, the above always evaluates to 0 at $\boldsymbol{W}_x = 0$.

2. For second-order derivatives involving $\boldsymbol{b}_x$, we have

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{b}_x} = \frac{-2}{\gamma}\mathbb{E}_\varepsilon[\boldsymbol{x}^{(i)} - \pi_\alpha(\boldsymbol{W}_x z^{(i)}) - \boldsymbol{b}_x]$$

and

$$\begin{aligned} \frac{\partial^2\mathcal{L}}{\partial(\boldsymbol{b}_x)^2} &= \frac{2}{\gamma}I, \\ \frac{\partial^2\mathcal{L}}{\partial\gamma\partial\boldsymbol{b}_x} &= \frac{2}{\gamma^2}\frac{\partial\mathcal{L}}{\partial\boldsymbol{b}_x} = 0, && \text{(since } \boldsymbol{W}_x = 0\text{);} \end{aligned}$$

and $\frac{\partial^2\mathcal{L}}{\partial\mu_z^{(i)}\partial\boldsymbol{b}_x}$ and $\frac{\partial^2\mathcal{L}}{\partial\mu_z^{(i)}\partial\sigma_z^{(i)}}$ will also have the form of (5), thus both equal 0 at $\boldsymbol{W}_x = 0$.

3. Next consider second-order derivatives involving $\mu_z^{(i)}$ or $\sigma_k^{(i)}$. Since the KL part of the energy, $\sum_{i=1}^n \mathrm{KL}(q_\phi(\boldsymbol{z}|\boldsymbol{x}^{(i)})|p(\boldsymbol{z}))$, only depends on $\mu_z^{(i)}$ and $\sigma_k^{(i)}$, and have p.d. Hessian at (M.6) independent of other parameters, it suffices to calculate the derivatives of the reconstruction error part, denoted as $\mathcal{L}_{\text{recon}}$. Since

$$\begin{aligned} \frac{\partial\mathcal{L}_{\text{recon}}}{\partial\mu_z^{(i)}} &= \frac{-2}{\gamma}\sum_{i,j}\mathbb{E}_\epsilon\left[(x_j^{(i)} - \pi_\alpha(W_j z^{(i)}) - b_j)W_j\pi'_\alpha(W_j z^{(i)})\right], \\ \frac{\partial\mathcal{L}_{\text{recon}}}{\partial\sigma_z^{(i)}} &= \frac{-2}{\gamma}\sum_{i,j}\mathbb{E}_\epsilon\left[(x_j^{(i)} - \pi_\alpha(W_j z^{(i)}) - b_j)W_j\epsilon\pi'_\alpha(W_j z^{(i)})\right], \end{aligned}$$

all second-order derivatives will have the form of (5), and equal 0 at $\boldsymbol{W}_x = 0$.

4. For γ, we can calculate that $\partial^2\mathcal{L}/\partial\gamma^2 = 4/\gamma^2 > 0$ at (M.6).

Now, consider VAE parameters that are only different from (M.6) in $\boldsymbol{W}_x$. Plugging $\boldsymbol{b}_x = \bar{\boldsymbol{x}}, \mu_z^{(i)} = 0, \sigma_k^{(i)} = 1$ into (4), we have

$$\frac{\partial\mathcal{L}}{\partial W_j} = \frac{-2}{\gamma}\sum_{i=1}^n \mathbb{E}_\varepsilon[(\pi'_\alpha(W_j\varepsilon)\varepsilon)\cdot(-\pi_\alpha(W_j\varepsilon))].$$

As $(\pi'_\alpha(W_j\varepsilon)\varepsilon)\cdot(-\pi_\alpha(W_j\varepsilon)) \leq 0$ always holds, we can see that the gradient points back to (M.6). This concludes our proof of (M.6) being a strict local minima. ■

3. Proof of Proposition (M.2)

We begin by assuming an arbitrarily complex encoder for convenience. This allows us to remove the encoder-sponsored amortized inference and instead optimize independent parameters $\boldsymbol{\mu}_z^{(i)}$ and $\boldsymbol{\sigma}_z^{(i)}$ separately for each data point. Later we will show that this capacity assumption can be dropped and the main result still holds.

We next define

$$\boldsymbol{m}_z \triangleq \left[\left(\boldsymbol{\mu}_z^{(1)}\right)^\top, \ldots, \left(\boldsymbol{\mu}_z^{(n)}\right)^\top\right]^\top \in \mathbb{R}^{\kappa n} \;\text{ and }\; \boldsymbol{s}_z \triangleq \left[\left(\boldsymbol{\sigma}_z^{(1)}\right)^\top, \ldots, \left(\boldsymbol{\sigma}_z^{(n)}\right)^\top\right]^\top \in \mathbb{R}^{\kappa n}, \quad (6)$$

which are nothing more than the concatenation of all of the decoder means and variances from each data point into the respective column vectors. It is also useful to decompose the assumed non-degenerate decoder parameters via

$$\theta \;\equiv\; [\psi, w], \quad \psi \;\triangleq\; \theta \backslash w, \quad (7)$$

where $w \in [0,1]$ is a scalar such that $\mu_x(\boldsymbol{z};\theta) \equiv \mu_x(w\boldsymbol{z};\psi)$. Note that we can always reparameterize an existing deep architecture to extract such a latent scaling factor which we can then hypothetically optimize separately while holding the remaining parameters ψ fixed. Finally, with slight abuse of notation, we may then define the function

$$f\left(w\boldsymbol{m}_z, w\boldsymbol{s}_z\right) \;\triangleq \quad (8)$$
$$\sum_{i=1}^{n} f\left(\boldsymbol{\mu}_z^{(i)}, \boldsymbol{\sigma}_z^{(i)}, [\tilde{\psi}, w], \boldsymbol{x}^{(i)}\right) \equiv \sum_{i=1}^{n} \mathbb{E}_{\mathcal{N}\left(\boldsymbol{z}|\boldsymbol{\mu}_z^{(i)}, \text{diag}\left[\boldsymbol{\sigma}_z^{(i)}\right]^2\right)} \left[\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x\left(w\boldsymbol{z};\tilde{\psi}\right)\|_2^2\right].$$

This is basically just the original function f summed over all training points, with ψ fixed at the corresponding values extracted from $\tilde{\theta}$ while w serves as a free scaling parameter on the decoder.

Based on the assumption of Lipschitz continuous gradients, we can always create the upper bound

$$f(\boldsymbol{u}, \boldsymbol{v}) \;\leq\; f(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \quad (9)$$
$$+ \;(\boldsymbol{u} - \tilde{\boldsymbol{u}})^\top \nabla_u f(\boldsymbol{u}, \boldsymbol{v})\big|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} \;+\; \tfrac{L}{2}\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_2^2 \;+\; (\boldsymbol{v} - \tilde{\boldsymbol{v}})^\top \nabla_v f(\boldsymbol{u}, \boldsymbol{v})\big|_{\boldsymbol{v}=\tilde{\boldsymbol{v}}} \;+\; \tfrac{L}{2}\|\boldsymbol{v} - \tilde{\boldsymbol{v}}\|_2^2,$$

where L is the Lipschitz constant of the gradients and we have adopted $\boldsymbol{u} \triangleq w\boldsymbol{m}_z$ and $\boldsymbol{v} \triangleq w\boldsymbol{\sigma}_z$ to simplify notation. Equality occurs at the evaluation point $\{\boldsymbol{u}, \boldsymbol{v}\} = \{\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}\}$. However, this bound does not account for the fact that we know $\nabla_v f(\boldsymbol{u}, \boldsymbol{v}) \geq 0$ (i.e., $f(\boldsymbol{u}, \boldsymbol{v})$ is increasing w.r.t. $\boldsymbol{v}$) and that $\boldsymbol{v} \geq 0$. Given these assumptions, we can produce the refined upper bound

$$f^{ub}(\boldsymbol{u}, \boldsymbol{v}) \;\geq\; f(\boldsymbol{u}, \boldsymbol{v}), \quad (10)$$

where $\quad f^{ub}(\boldsymbol{u}, \boldsymbol{v}) \;\triangleq$

$$f(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}) \;+\; (\boldsymbol{u} - \tilde{\boldsymbol{u}})^\top \nabla_u f(\boldsymbol{u}, \boldsymbol{v})\big|_{\boldsymbol{u}=\tilde{\boldsymbol{u}}} \;+\; \tfrac{L}{2}\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\|_2^2 + \sum_{j=1}^{nd} g\left(v_j, \tilde{v}_j, \nabla_{v_j} f(\boldsymbol{u}, \boldsymbol{v})\big|_{v_j=\tilde{v}_j}\right) \quad (11)$$

and the function $g : \mathbb{R}^3 \to \mathbb{R}$ is defined as

$$g\left(v, \tilde{v}, \delta\right) \triangleq \begin{cases} \left(v - \tilde{v}\right)\delta + \frac{L}{2}\left(v - \tilde{v}\right)_2^2 & \text{if } \; v \geq \tilde{v} - \frac{\delta}{L} \; \text{ and } \; \{v, \tilde{v}, \delta\} \geq 0, \\ \frac{-\delta^2}{2L} & \text{if } \; v < \tilde{v} - \frac{\delta}{L} \; \text{ and } \; \{v, \tilde{v}, \delta\} \geq 0, \\ \infty & \text{otherwise.} \end{cases} \tag{12}$$

Given that

$$\tilde{v} - \tfrac{\delta}{L} = \arg\min_v \left[\left(v - \tilde{v}\right)\delta + \tfrac{L}{2}\left(v - \tilde{v}\right)_2^2\right] \quad \text{and} \quad \tfrac{-\delta^2}{2L} = \min_v \left[\left(v - \tilde{v}\right)\delta + \tfrac{L}{2}\left(v - \tilde{v}\right)_2^2\right], \tag{13}$$

the function g is basically just setting all values of $\left(v - \tilde{v}\right)\delta + \frac{L}{2}\left\|v - \tilde{v}\right\|_2^2$ with negative slope to the minimum $\frac{-\delta^2}{2L}$. This change is possible while retaining an upper bound because $f\left(\boldsymbol{u}, \boldsymbol{v}\right)$ is non-decreasing in $\boldsymbol{v}$ by stated assumption. Additionally, g is set to infinity for all $v < 0$ to enforce non-negatively.

While it may be possible to proceed further using f^{ub}, we find it useful to consider a final modification. Specifically, we define the approximation

$$f^{appr}\left(\boldsymbol{u}, \boldsymbol{v}\right) \quad \approx \quad f^{ub}\left(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}\right), \tag{14}$$

where $\quad f^{appr}\left(\boldsymbol{u}, \boldsymbol{v}\right) \;\; \triangleq$

$$f\left(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{v}}\right) \; + \; \left(\boldsymbol{u} - \tilde{\boldsymbol{u}}\right)^\top \nabla_u f\left(\boldsymbol{u}, \boldsymbol{v}\right)\big|_{\boldsymbol{u} = \tilde{\boldsymbol{u}}} \; + \; \tfrac{L}{2}\left\|\boldsymbol{u} - \tilde{\boldsymbol{u}}\right\|_2^2 + \sum_{j=1}^{nd} g^{appr}\left(v_j, \tilde{v}_j, \nabla_{v_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\big|_{v_j = \tilde{v}_j}\right) \tag{15}$$

and

$$g^{appr}\left(v, \tilde{v}, \delta\right) \triangleq \begin{cases} \frac{-\delta^2}{2L} + \frac{\delta^2}{2L\tilde{v}^2}v^2 & \text{if } \; \tilde{v} - \frac{\delta}{L} \geq 0 \; \text{ and } \; \{v, \tilde{v}, \delta\} \geq 0, \\ \left(\frac{L\tilde{v}^2}{2} - \delta\tilde{v}\right) + \left(\frac{\delta}{\tilde{v}} - \frac{L}{2}\right)v^2 & \text{if } \; \tilde{v} - \frac{\delta}{L} < 0 \; \text{ and } \; \{v, \tilde{v}, \delta\} \geq 0, \\ \infty & \text{otherwise.} \end{cases} \tag{16}$$

While slightly cumbersome to write out, g^{appr} has a simple interpretation. By construction, we have that

$$\min_v g^{appr}\left(v, \tilde{v}, \delta\right) = g^{appr}\left(0, \tilde{v}, \delta\right) = \min_v g\left(v, \tilde{v}, \delta\right) = g\left(0, \tilde{v}, \delta\right) \tag{17}$$

and

$$g^{appr}\left(\tilde{v}, \tilde{v}, \delta\right) = g\left(\tilde{v}, \tilde{v}, \delta\right) = 0. \tag{18}$$

At other points, g^{appr} is just a simple quadratic interpolation but without any factor that is linear in v. And removal of this linear term, while retaining (17) and (17) will be useful for the analysis that follows below. Note also that although $f^{appr}\left(\boldsymbol{u}, \boldsymbol{v}\right)$ is no longer a strict bound on $f\left(\boldsymbol{u}, \boldsymbol{v}\right)$, it will nonetheless still be an upper bound whenever $v_j \in \{0, \tilde{v}_j\}$ for all j which will ultimately be sufficient for our purposes.

We now consider optimizing the function

$$h^{appr}(\boldsymbol{m}_z, \boldsymbol{s}_z, w) \triangleq \tfrac{1}{\gamma} f^{appr}\left(w\boldsymbol{m}_z, w\boldsymbol{s}_z\right) + \sum_{i=1}^{n} \left\|\boldsymbol{\mu}_z^{(i)}\right\|_2^2 + \left\|\boldsymbol{\sigma}_z^{(i)}\right\|_2^2 - \log\left|\operatorname{diag}\left[\boldsymbol{\sigma}_z^{(i)}\right]^2\right|. \tag{19}$$

If we define $\mathcal{L}\left(\boldsymbol{m}_z, \boldsymbol{s}_z, w\right)$ as the VAE cost from (M.3) under the current parameterization, then by design it follows that

$$h^{appr}(\tilde{\boldsymbol{m}}_z, \tilde{\boldsymbol{s}}_z, \tilde{w}) = \mathcal{L}\left(\tilde{\boldsymbol{m}}_z, \tilde{\boldsymbol{s}}_z, \tilde{w}\right) \tag{20}$$

and

$$h^{appr}(\boldsymbol{m}_z, \boldsymbol{s}_z, w) \geq \mathcal{L}\left(\boldsymbol{m}_z, \boldsymbol{s}_z, w\right) \tag{21}$$

whenever $w\sigma_j \in \{0, \tilde{w}\tilde{\sigma}_j\}$ for all j. Therefore if we find such a solution $\{\boldsymbol{m}'_z, \boldsymbol{s}'_z, w'\}$ that satisfies this condition and has $h^{appr}(\boldsymbol{m}'_z, \boldsymbol{s}'_z, w') < h^{appr}(\tilde{\boldsymbol{m}}_z, \tilde{\boldsymbol{s}}_z, \tilde{w})$, it necessitates that $\mathcal{L}(\boldsymbol{m}'_z, \boldsymbol{s}'_z, w') < \mathcal{L}(\tilde{\boldsymbol{m}}_z, \tilde{\boldsymbol{s}}_z, \tilde{w})$ as well. This then ensures that $\{\tilde{\boldsymbol{m}}_z, \tilde{\boldsymbol{s}}_z, \tilde{w}\}$ cannot be a local minimum.

We now examine the function h^{appr} more closely. After a few algebraic manipulations and excluding irrelevant constants, we have that

$$\begin{aligned} & h^{appr}(\boldsymbol{m}_z, \boldsymbol{s}_z, w) \equiv \\ & \quad \sum_{j=1}^{nd} \left\{ \tfrac{1}{\gamma} \left[w m_{z,j} \, \nabla_{u_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\Big|_{u_j = \tilde{w}\tilde{m}_{z,j}} + \tfrac{L}{2}\left(w^2 m_{z,j}^2 - 2 w m_{z,j} \tilde{w}\tilde{m}_{z,j}\right) + c_j w^2 s_{z,j}^2 \right] \right. \\ & \qquad\qquad \left. + \;\; m_{z,j}^2 + s_{z,j}^2 - \log s_{z,j}^2 \right\}, \end{aligned} \tag{22}$$

where c_j is the coefficient on the v^2 term from (16). After rearranging terms, optimizing out $\boldsymbol{m}_z$ and $\boldsymbol{s}_z$, and discarding constants, we can then obtain (with slight abuse of notation) the reduced function

$$h^{appr}(w) \triangleq \sum_{j=1}^{nd} \frac{y_j}{\gamma + \beta w^2} + \log(\gamma + c_j w^2), \tag{23}$$

where $\beta \triangleq \frac{L}{2}$ and $y_j \triangleq \frac{L}{2} \left\| \tilde{w}\tilde{m}_{z,j} - \frac{1}{L} \, \nabla_{u_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\Big|_{u_j = \tilde{w}\tilde{m}_{z,j}} \right\|_2^2$. Note that y_j must be bounded since $L \neq 0$[1] and $w \in [0, 1]$, $\nabla_{u_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\Big|_{u_j = \tilde{w}\tilde{m}_{z,j}} \leq L$, and $\tilde{\boldsymbol{m}}$ are all bounded. The latter is implicitly bounded because the VAE KL term prevents infinite encoder mean functions. Furthermore, c_j must be strictly greater than zero per the definition of a non-degenerate decoder; this guarantees that

$$g^{appr}\left(\tilde{w}\tilde{s}_j, \tilde{w}\tilde{s}_j, \nabla_{v_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\Big|_{v_j = \tilde{w}\tilde{s}_j}\right) > g^{appr}\left(0, \tilde{w}\tilde{s}_j, \nabla_{v_j} f\left(\boldsymbol{u}, \boldsymbol{v}\right)\Big|_{v_j = \tilde{w}\tilde{s}_j}\right), \tag{24}$$

which is only possible with $c_j > 0$. Proceeding further, because

$$\nabla_{w^2} h^{appr}(w) = \sum_{j=1}^{nd} \left(\frac{-\beta y_j}{\left(\gamma + \beta w^2\right)^2} + \frac{c_j}{\gamma + c_j w^2} \right), \tag{25}$$

we observe that if γ is increased sufficiently large, the first term will always be smaller than the second since β and all y_j are bounded, and $c_j > 0 \; \forall j$. So there can never be a point

[1] $L = 0$ would violate the stipulated conditions for a non-degenerate decoder since it would imply that no signal from $\boldsymbol{z}$ could pass through the decoder. And of course if $L = 0$, we would already be at a solution exhibiting posterior collapse.

whereby $\nabla_{w^2} h^{appr}(w) = 0$ when $\gamma = \gamma'$ sufficiently large. Therefore the minimum in this situation occurs on the boundary where $w^2 = 0$. And finally, if $w^2 = 0$, then the optimal $\boldsymbol{m}_z$ and $\boldsymbol{s}_z$ is determined solely by the KL term, and hence they are set according to the prior. Moreover, the decoder has no signal from the encoder and is therefore optimized by simply setting $\boldsymbol{\mu}_x \left(0; \tilde{\psi}\right)$ to the mean $\bar{\boldsymbol{x}}$ for all i.[2] Additionally, none of this analysis requires and arbitrarily complex encoder; the exact same results hold as long as the encoder can output a 0 for means and 1 for the variances.

Note also that if we proceed through the above analysis using $\boldsymbol{w} \in \mathbb{R}^{\kappa}$ as parameterizing a separate w_j scaling factor for each latent dimension $j \in \{1, \dots, \kappa\}$, then a smaller γ value would generally force partial collapse. In other words, we could enforce nonzero gradients of $h^{appr}(w)$ along the indices of each latent dimension separately. This loosely criteria would then lead to $q_{\phi^*}(z_j|\boldsymbol{x}) = p(z_j)$ along some but not all latent dimensions as stated in the main text below Proposition M.2. ■

4. Representative Stationary Point Exhibiting Posterior Collapse in Deep VAE Models

Here we provide an example of a stationary point that exhibits posterior collapse with an arbitrary deep encoder/decoder architecture. This example is representative of many other possible cases. Assume both encoder and decoder mean functions $\boldsymbol{\mu}_x$ and $\boldsymbol{\mu}_z$, as well as the diagonal encoder covariance function $\boldsymbol{\Sigma}_z = \text{diag}[\boldsymbol{\sigma}_z^2]$, are computed by standard deep neural networks, with layers composed of linear weights followed by element-wise nonlinear activations (the decoder covariance satisfies $\boldsymbol{\Sigma}_x = \gamma \boldsymbol{I}$ as before). We denote the weight matrix from the first layer of the decoder mean network as $\boldsymbol{W}^1_{\mu_x}$, while $\boldsymbol{w}^1_{\mu_x, \cdot j}$ refers to the corresponding j-th column. Assuming ρ layers, we denote $\boldsymbol{W}^{\rho}_{\mu_z}$ and $\boldsymbol{W}^{\rho}_{\sigma_z^2}$ as weights from the last layers of the encoder networks producing $\boldsymbol{\mu}_z$ and $\log \boldsymbol{\sigma}_z^2$ respectively, with j-th rows defined as $\boldsymbol{w}^{\rho}_{\mu_z, j\cdot}$ and $\boldsymbol{w}^{\rho}_{\sigma_z^2, j\cdot}$. We then characterize the following key stationary point:

Proposition 2 *If* $\boldsymbol{w}^1_{\mu_x, \cdot j} = \left(\boldsymbol{w}^{\rho}_{\mu_z, j\cdot}\right)^{\top} = \left(\boldsymbol{w}^{\rho}_{\sigma_z^2, j\cdot}\right)^{\top} = \boldsymbol{0}$ *for any* $j \in \{1, 2, \dots, \kappa\}$*, then the gradients of (M.3) with respect to* $\boldsymbol{w}^1_{\mu_x, \cdot j}$*,* $\boldsymbol{w}^{\rho}_{\mu_z, j\cdot}$*, and* $\boldsymbol{w}^{\rho}_{\sigma_z^2, j\cdot}$ *are all equal to zero.*

If the stated weights are zero along dimension j, then obviously it must be that $q_{\phi}(z_j|\boldsymbol{x}) = p(z_j)$, i.e., a collapsed dimension for better or worse. The proof is straightforward; we provide the details below for completeness.

Proof: First we remind that the variational upper bound is defined in (M.1). We define $\mathcal{L}(\boldsymbol{x}; \theta, \phi)$ as the loss at a data point $\boldsymbol{x}$, *i.e.*

$$\mathcal{L}(\boldsymbol{x}; \theta, \phi) = -\mathbb{E}_{q_{\phi}(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})\right] + \mathbb{KL}\left[q_{\phi}(\boldsymbol{z}|\boldsymbol{x}) || p(\boldsymbol{z})\right]. \tag{26}$$

[2] We are assuming here that the decoder has sufficient capacity to model any constant value, e.g., the output layer has a bias term.

The total loss is the integration of $\mathcal{L}(\boldsymbol{x};\theta,\phi)$ over $\boldsymbol{x}$. Further more, we denote $\mathcal{L}_{kl}(\boldsymbol{x};\theta)$ and $\mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)$ as the KL loss and the generation loss at $\boldsymbol{x}$ respectively, *i.e.*

$$\begin{aligned}
\mathcal{L}_{kl}(\boldsymbol{x};\phi) &= \mathbb{KL}\left[q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right] = \sum_{i=1}^{\kappa} \mathbb{KL}\left[q_\phi(z_j|\boldsymbol{x})||p(z_j)\right], \\
&= \frac{1}{2}\sum_{j=1}^{\kappa}\left(\mu_{z,j}^2 + \sigma_{z,j}^2 - \log\sigma_{z,j}^2 - 1\right) &(27)\\
\mathcal{L}_{gen}(\boldsymbol{x};\phi,\theta) &= -\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})\right]. &(28)
\end{aligned}$$

The second equality in (27) holds because the covariance of $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ and $p(\boldsymbol{z})$ are both diagonal. The last encoder layer and the first decoder layer are denoted as $\boldsymbol{h}_e^\rho$ and $\boldsymbol{h}_d^1$. If $\boldsymbol{w}_{\mu_z,j\cdot}^\rho = 0, \boldsymbol{w}_{\sigma_z^2,j\cdot}^\rho = 0$, then we have

$$\mu_{z,j} = \boldsymbol{w}_{\mu_z,j\cdot}^\rho \boldsymbol{h}_e^\rho = 0, \quad \sigma_{z,j}^2 = \exp\left(\boldsymbol{w}_{\sigma_z^2,j\cdot}\right) = 1, \quad q(z_j|\boldsymbol{x}) = \mathcal{N}(0,1). \tag{29}$$

The gradient of $\mu_{z,j}$ and $\sigma_{z,j}$ from $\mathcal{L}_{kl}(\boldsymbol{x};\phi)$ becomes

$$\frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{\partial\mu_{z,j}} = \mu_{z,j} = 0, \quad \frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{\partial\sigma_{z,j}} = 1 - \sigma_{z,j}^{-1} = 0. \tag{30}$$

So the gradient of $\boldsymbol{w}_{\mu_z,j\cdot}^\rho$ and $\boldsymbol{w}_{\sigma_z^2,j\cdot}^\rho$ from $\mathcal{L}_{kl}$ is

$$\frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{\partial\boldsymbol{w}_{\mu_z,j\cdot}^\rho} = \frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{\partial\mu_{z,j}}\boldsymbol{h}_e^{\rho\top} = 0, \tag{31}$$

$$\frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{\partial\boldsymbol{w}_{\sigma_z^2,j\cdot}^\rho} = \frac{\partial\mathcal{L}_{kl}(\boldsymbol{x};\phi)}{2\sigma_{z,j}\cdot\partial\sigma_{z,j}}\boldsymbol{h}_e^{\rho\top} = 0. \tag{32}$$

Now we consider the gradient from $\mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)$. We have

$$\frac{-\partial\log p_\theta(\boldsymbol{x}|\boldsymbol{z})}{\partial z_j} = \frac{-\partial\log p_\theta(\boldsymbol{x}|\boldsymbol{z})}{\partial\boldsymbol{h}_d^1}\frac{\partial\boldsymbol{h}_d^1}{\partial z_j}. \tag{33}$$

Since

$$\boldsymbol{h}_d^1 = \text{act}\left(\sum_{j=1}^{\kappa}\boldsymbol{w}_{\mu_x,\cdot j}^1 z_j\right), \tag{34}$$

where act$(\cdot)$ is the activation function, we can obtain

$$\frac{\partial\boldsymbol{h}_d^1}{\partial z_j} = \text{act}'\left(\sum_{j=1}^{\kappa}\boldsymbol{w}_{\mu_x,\cdot j}^1 z_j\right)\boldsymbol{w}_{\mu_x,\cdot j}^1 = 0. \tag{35}$$

Plugging this back into (33) gives

$$\frac{-\partial\log p_\theta(\boldsymbol{x}|\boldsymbol{z})}{\partial z_j} = 0. \tag{36}$$

According to the chain rule, we have

$$\frac{\partial \mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)}{\partial \boldsymbol{w}^{\rho}_{\mu_z,j\cdot}} = \mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})}{\partial z_j}\frac{\partial z_j}{\partial \boldsymbol{w}^{\rho}_{\mu_z,j\cdot}}\right] = 0, \tag{37}$$

$$\frac{\partial \mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)}{\partial \boldsymbol{w}^{\rho}_{\sigma_z^2,j\cdot}} = \mathbb{E}_{\boldsymbol{z}\sim q_{\phi}(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})}{\partial z_j}\frac{\partial z_j}{\partial \boldsymbol{w}^{\rho}_{\sigma_z^2,j\cdot}}\right] = 0. \tag{38}$$

After combining these two equations with (31) and (32) and then integrating over $\boldsymbol{x}$, we have

$$\frac{\partial \mathcal{L}(\theta,\phi)}{\partial \boldsymbol{w}^{\rho}_{\mu_z,j\cdot}} = 0, \tag{39}$$

$$\frac{\partial \mathcal{L}(\theta,\phi)}{\partial \boldsymbol{w}^{\rho}_{\sigma_z^2,j\cdot}} = 0. \tag{40}$$

Then we consider the gradient with respect to $\boldsymbol{w}^{1}_{\mu_x,\cdot j}$. Since $\boldsymbol{w}_{\mu_x,\cdot j}$ is part of θ, it only receives gradient from $\mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)$. So we do not need to consider the KL loss. If $\boldsymbol{w}^{1}_{\mu_x,\cdot j} = 0$, $\boldsymbol{h}^{1}_{d} = \sum_{j=1}^{\kappa} \boldsymbol{w}^{1}_{\mu_x,\cdot j} z_j$ is not related to $\boldsymbol{z}_j$. So $p_{\theta}(\boldsymbol{x}|\boldsymbol{z}) = p_{\theta}(\boldsymbol{x}|\boldsymbol{z}_{\neg j})$, where $\boldsymbol{z}_{\neg j}$ represents $\boldsymbol{z}$ without the j-th dimension. The gradient of $\boldsymbol{w}^{1}_{\mu_x,\cdot j}$ is

$$\begin{aligned}\frac{\partial \mathcal{L}_{gen}(\boldsymbol{x};\theta,\phi)}{\partial \boldsymbol{w}^{1}_{\mu_x,\cdot j}} &= \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})}{\partial \boldsymbol{w}^{1}_{\mu_x,\cdot j}}\right] = \mathbb{E}_{\boldsymbol{z}\sim q(\boldsymbol{z}|\boldsymbol{x})}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z})}{\partial \boldsymbol{h}^{1}_{d}} z_j\right] \qquad (41)\\ &= \mathbb{E}_{\boldsymbol{z}_{\neg j}\sim q(\boldsymbol{z}_{\neg j}|\boldsymbol{x})}\left[\mathbb{E}_{z_j\sim\mathcal{N}(0,1)}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z}_{\neg j})}{\partial \boldsymbol{h}^{1}_{d}} z_j\right]\right]\\ &= \mathbb{E}_{\boldsymbol{z}_{\neg i}\sim q(\boldsymbol{z}_{\neg i}|\boldsymbol{x})}\left[\frac{-\partial \log p_{\theta}(\boldsymbol{x}|\boldsymbol{z}_{\neg j})}{\partial \boldsymbol{h}^{1}_{d}}\mathbb{E}_{z_j\sim\mathcal{N}(0,1)}[\boldsymbol{z}_j]\right] = 0.\end{aligned}$$

The integration over $\boldsymbol{x}$ should also be 0. So we obtain

$$\frac{\partial \mathcal{L}(\theta;\phi)}{\partial \boldsymbol{w}^{1}_{\mu_x,\cdot j}} = 0. \tag{42}$$

■

References

[1] Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv preprint arXiv:1509.00519*, 2015.

[2] E. Orjebin. A recursive formula for the moments of a truncated univariate normal distribution. 2014.

[3] Rianne van den Berg, Leonard Hasenclever, Jakub M Tomczak, and Max Welling. Sylvester normalizing flows for variational inference. In *Uncertainty in Artificial Intelligence*, 2018.