Hierarchical VAEs Know What They Don't Know
Supplementary Material

Jakob D. Havtorn [1 2] **Jes Frellsen** [1] **Søren Hauberg** [1] **Lars Maaløe** [1 2]

A. Datasets

Table 1 lists the datasets used in the paper. We use the predefined train/test splits for the datasets.

For SmallNORB and Omniglot we resize the original grey-scale images to 28×28 with ordinary bi-linear interpolation. For each of these datasets, we also create a version where the grey-scale is inverted. We do this because, the overall white nature of the images tends to make detecting them as OOD from FashionMNIST artificially easy. The inversion is done via the simple transformation $\mathbf{x}_{\text{inverted}} = 255 - \mathbf{x}_{\text{original}}$ since images are encoded as 8 bit unsigned integers.

Dataset	Dimensionality	Examples
FashionMNIST (Xiao et al., 2017)	$28 \times 28 \times 1$	70,000
MNIST (LeCun et al., 1998)	$28 \times 28 \times 1$	70,000
notMNIST (Bulatov, 2011)	$28 \times 28 \times 1$	547,838
KMNIST (Clanuwat et al., 2018)	$28 \times 28 \times 1$	70,000
Omniglot (Lake et al., 2015)	$28 \times 28 \times 1$	32,460
SmallNORB (LeCun et al., 2004)	$28 \times 28 \times 1$	97,200
CIFAR10 (Krizhevsky, 2009)	$32 \times 32 \times 3$	60,000
SVHN (Netzer et al., 2011)	$32 \times 32 \times 3$	99,289

Table 1. Overview of the used datasets.

B. Model details

In Table 2 we specify the hyperparameters used when training our models.

We make our source code available at `https://github.com/JakobHavtorn/hvae-oodd`.

B.1. Hierarchical VAE

Our Hierarchical VAE (HVAE) model uses bottom-up inference and top-down generative paths as specified in the paper. For grey-scale images, the output is parameterized by a Bernoulli distribution while for natural images we use a Discretized Logistic Mixture (Salimans et al., 2017). The latent variables are parameterized by stochastic layers that output the mean and log-variance of a diagonal covariance Gaussian. The prior distribution on the top-most latent is a standard Gaussian. For grey-scale images, the lowest latent space is parameterized by a convolutional neural network and has dimensions $14 \times 14 \times 8$ interpreted as (height $\times$ width $\times$ latent dimension). The highest two latent variables are parameterized by dense transformations with 16 and 8 units, respectively. For natural images, all latent variables are parameterized by convolutional neural networks and have dimensions $(16 \times 16) \times 8$, $(8 \times 8) \times 16$ and $(4 \times 4) \times 32$, respectively for $\mathbf{z}_1, \mathbf{z}_2$ and $\mathbf{z}_3$ given as (height $\times$ width) $\times$ dim).

Each stochastic layer is preceeded by a determininistic transformation. For both grey-scale and natural images, each deterministic transformation consists of three residual blocks of the same type used by Maaløe et al. (2019). The structure of a residual block is:

$$\mathbf{y} = \text{Conv}\left(\text{Act}\left(\text{Conv}_s\left(\text{Act}(\mathbf{x})\right)\right)\right) + \mathbf{x} \ ,$$

where "Conv" refers to a same-padded convolution and "Act" to the activation function. Within a residual block, the first convolution always has stride 1 while the second convolution has stride s. In a deterministic transformation, any non-unit stride is performed in the third residual block. For grey-scale images, we stride by 2 in the first and second deterministic transformations but not the third. For natural images, we stride by 2 in all three deterministic blocks. In both cases, the first deterministic block uses a kernel size of 5 and the latter two a kernel of size 3. In all cases we use 64 channels We use the ReLU activation function (Fukushima, 1980; Nair & Hinton, 2010).

Since the benefits and drawbacks of using batch normalization (Ioffe & Szegedy, 2015) in hierarchical VAEs is still the matter of some debate (Sønderby et al., 2016; Vahdat & Kautz, 2020; Child, 2021) we choose to use weight normalization (Salimans & Kingma, 2016) as in other work (Maaløe et al., 2019) and initialize the model using the originally proposed data-dependent initialization. To have the

[1] Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby, Denmark [2] Corti AI, Copenhagen, Denmark. Correspondence to: Jakob D. Havtorn <jdh@corti.ai/jakob.havtorn@gmail.com>, Lars Maaløe <lm@corti.ai>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

stochastic layers initialize to standard Gaussian distributions (zero mean, unit variance), with this initialization, we select the activation function for the variance as a Softplus,

$$\mathrm{Softplus}(\mathbf{x}) = \frac{1}{\beta}\log\left(1+\exp(\beta\mathbf{x})\right) \ ,$$

with $\beta = \log(2) \approx 0.693$ to output 1 for $\mathbf{x} = 0$.

Training of a HVAE model took approximately two days on a single NVIDIA GTX 1080 Ti graphics card.

B.2. BIVA

For the BIVA model (Maaløe et al., 2019), we use a specification that is very similar to that of the HVAE above, and to that of the original paper. The model has 10 latent variables the lowest 3 of which are spatial and the rest are densely connected in order to have an architecture similar to the HVAE. The model uses an overall stride of 8, achieved by striding by 2 in the first, fourth and sixth deterministic transformations. From $\mathbf{z}_1$ to $\mathbf{z}_{10}$, the latents have the following dimensions: The lowest three latents are spatial $(16 \times 16) \times 8$, $(16 \times 16) \times 16$ and $(16 \times 16) \times 32$, given as (height $\times$ width) $\times$ dim), while the rest are dense vectors with dimensions of $42, 40, 38, 36, 34, 32, 30$.

Training of a BIVA model took approximately a week on a single NVIDIA GTX 1080 Ti graphics card.

C. Analysis of the influence of latent variables on the marginal likelihood

In the paper, we argue that the lowest level latent variables, which have the highest dimensionality, contribute the most to the approximate likelihood. Here, we provide a stringent mathematical argument that generalizes this to the exact marginal likelihood in a model with a deterministic decoder.

C.1. Model specification

For an arbitrary hierarchical latent variable model, we have a prior $p(\mathbf{z}_L)$ and a generative mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^D$, such that $\mathbf{x} = f(\mathbf{z}_L)$ and $D > d$. Note that we will assume that f is deterministic, such that we are effectively working with $p(\mathbf{x}|\mathbf{z}) = \delta_{f(\mathbf{z})}(\mathbf{x})$. This is a limiting assumption, but it allows working through the following. For shorthand we will simply write $\mathbf{z} = \mathbf{z}_L$.

Let f have a bottleneck architecture, i.e.

$$f(\mathbf{z}) = f_1(\ldots f_{L-1}(f_L(\mathbf{z}))) \ , \tag{1}$$

where

$$f_i : \mathbb{R}^{d_i} \rightarrow \mathbb{R}^{d_{i-1}}, \qquad i = L, \ldots, 1 \ . \tag{2}$$

Here we use the notation $d_0 = D = |\mathbf{x}|$ and $d_L = d = |\mathbf{z}|$ and further assume $d_0 \geq d_1 \geq \ldots \geq d_{L-1} \geq d_L$ which gives the bottleneck.

Hyperparameter	Setting/Range
All	
Optimization	Adam (Kingma & Ba, 2015)
Learning rate	$3e-4$
Batch size	128
Epochs	2000
Free bits	2 nats shared among all $\mathbf{z}_i$
Free bits constant	200 epochs
Free bits annealed	200 epochs
Activation	ReLU
Initialization	Data-dependent (Salimans & Kingma, 2016)
HVAE	
Latent dimensionality	8-16-32 (natural) / 8-16-8 (grey)
Convolution kernel	5-3-3
Stride	2-2-2 (natural) / 2-2-1 (grey)
Warmup anneal period	200 epochs
BIVA	
Latent dimensionality	10-8-6 (spatial) 42-40-38-36-34-32-30 (dense)
Convolution kernel	5-3-3-3-3-3-3-3-3-3
Stride	2-1-1-2-1-2-1-1-1-1

Table 2. Selection of most important hyperparameters and their setting. Convolutional kernels are square and latent dimensions are given without spatial dimensions which are given in the text. See Appendix B for more details.

Assuming $\mathbf{x}$ is such that a corresponding latent variable $\mathbf{z}$ exists, i.e. that there exists $\mathbf{z}$ such that $\mathbf{x} = f(\mathbf{z})$, then we can write the likelihood of $\mathbf{x}$ through a standard change of variables (similar to flow-based models),

$$p(\mathbf{x}) = p(\mathbf{z}) \prod_{i=1}^{L} \left(\sqrt{\det \mathbf{J}_i^T \mathbf{J}_i} \right)^{-1} \ , \tag{3}$$

where $\mathbf{J}_i$ is the Jacobian of f_i, i.e.

$$\mathbf{J}_i = \frac{\partial f_i}{\partial \mathbf{z}_i} \in \mathbb{R}^{d_i \times d_{i-1}} \ . \tag{4}$$

Here we use the notation that $\mathbf{z}_i$ is the representation at layer i. Note that $\mathbf{J}_i^T \mathbf{J}_i$ is a $d_{i-1} \times d_{i-1}$ symmetric positive semidefinite matrix (determinant ≥ 0).

The log-likelihood can be written as

$$\log p(\mathbf{x}) = \log p(\mathbf{z}) - \frac{1}{2} \sum_{i=1}^{L} \log \det \mathbf{J}_i^T \mathbf{J}_i \ . \tag{5}$$

By construction of determinants, we can generally expect these determinants to grow with the dimensionality of the matrix. We should expect the determinant of a $d \times d$ matrix

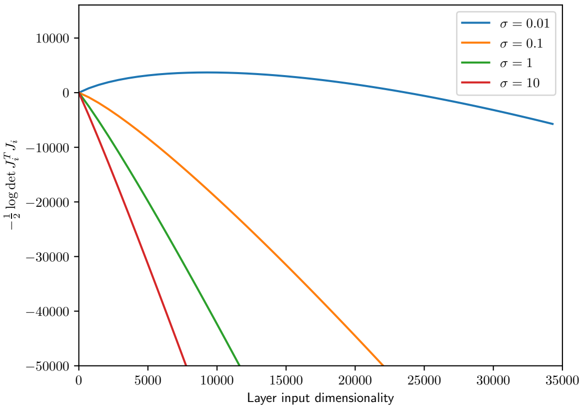

Figure 1. The expected inverse volume change for Gaussian Jacobians (7) on a log-scale.

to be of the order $\mathcal{O}(\lambda^d)$ for some number $\lambda > 0$. With that in mind, we should generally expect that

$$\det \mathbf{J}_{i+1}^T \mathbf{J}_{i+1} > \det \mathbf{J}_i^T \mathbf{J}_i \,, \tag{6}$$

due to the bottleneck assumption. If so, we see that the marginal likelihood $p(\mathbf{x})$ will be dominated by $\left(\sqrt{\det \mathbf{J}_1^T \mathbf{J}_1}\right)^{-1}$, i.e. low-level features have a higher influence on the likelihood than more important semantic ones.

C.2. The Gaussian case

The previous remarks can be made more precise if we make distributional assumptions on the Jacobians. Here we will assume that the Jacobians of each layer follow a Gaussian distribution. Specifically, we will assume that each entry in $\mathbf{J}_i$ is distributed as $\mathcal{N}(0, \sigma^2)$. The analysis below extends to nonzero means and more general covariance structure, but this comes with a cost of less transparent notation. In this setting, $\mathbf{J}_i^T \mathbf{J}_i$ follows a Wishart distribution (in the general setting it would follow a non-central Wishart distribution). Muirhead (2009) tells us that the expected multiplicative contribution to the likelihood of each layer is

$$\begin{aligned}\mathbb{E}\left[\left(\sqrt{\det \mathbf{J}_i^T \mathbf{J}_i}\right)^{-1}\right] &= \sigma^{-d_{i-1}} 2^{-\frac{d_{i-1}}{2}} \frac{\Gamma_{d_{i-1}}\left(\frac{1}{2}d_i - \frac{1}{2}\right)}{\Gamma_{d_{i-1}}\left(\frac{1}{2}d_i\right)} \\ &= \sigma^{-d_{i-1}} 2^{-\frac{d_{i-1}}{2}} \frac{\Gamma\left(\frac{1}{2}(d_i - d_{i-1})\right)}{\Gamma\left(\frac{1}{2}d_i\right)}\end{aligned} \tag{7}$$

where Γ_d is the multivariate Gamma function. Assuming that the increase in layer dimension $d_i - d_{i-1}$ is constant, then we see that (7) goes to zero as d_i goes to infinity as the Γ function grows super-exponentially to infinity. This super-exponential growth further implies that the first layers dominate the marginal likelihood $p(\mathbf{x})$. This is also visually evident in Figure 1.

D. Derivation of the $\mathcal{L}^{>k}$ bound

In this section we present the derivation of $\mathcal{L}^{>k}$ and show that it is a lower bound on the marginal likelihood.

First, we consider a two-layered VAE with bottom-up inference. We proceed very similarly to the derivation of the regular ELBO and also use Jensen's inequality.

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int\!\!\int p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_1|\mathbf{z}_2)p(\mathbf{z}_2)\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2 \\ &= \log \int\!\!\int \frac{q(\mathbf{z}_2|\mathbf{x})}{q(\mathbf{z}_2|\mathbf{x})} p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_1|\mathbf{z}_2)p(\mathbf{z}_2)\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2 \\ &= \log \int\!\!\int q(\mathbf{z}_2|\mathbf{x})p(\mathbf{z}_1|\mathbf{z}_2)\frac{p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{x})}\mathrm{d}\mathbf{z}_1\mathrm{d}\mathbf{z}_2 \\ &\geq \mathbb{E}_{p(\mathbf{z}_1|\mathbf{z}_2)q(\mathbf{z}_2|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z}_1)p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{x})}\right] \equiv \mathcal{L}^{>1} \,.\end{aligned} \tag{8}$$

Here, we have introduced the variational distribution $q(\mathbf{z}_2|\mathbf{x})$ which, naively, is different from any of the available variational distributions $q(\mathbf{z}_1|\mathbf{x})$ and $q(\mathbf{z}_2|\mathbf{z}_1)$. However, it's easy to see that we can simply define $q(\mathbf{z}_2|\mathbf{x}) = q(\mathbf{z}_2|d_1(\mathbf{x}))$ where $d_1(\mathbf{x}) = \mathbb{E}\left[q(\mathbf{z}_1|\mathbf{x})\right]$. I.e. we compute the distribution over $\mathbf{z}_2$ via the mode of $q(\mathbf{z}_1|\mathbf{x})$. This is possible since we exclusively manipulate the variational proposal distribution without altering the generative model $p(\mathbf{x}, \mathbf{z})$.

In general, the derivation of $\mathcal{L}^{>k}$ for an L-layered hierarchical VAE with $\mathbf{z} = \mathbf{z}_1, \ldots, \mathbf{z}_L$ is as follows:

$$\begin{aligned}\log p(\mathbf{x}) &= \log \int p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \\ &= \log \int \frac{q(\mathbf{z}_{>k}|\mathbf{x})}{q(\mathbf{z}_{>k}|\mathbf{x})} p(\mathbf{x}|\mathbf{z})p(\mathbf{z})\mathrm{d}\mathbf{z} \\ &= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z})\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}_{>k}|\mathbf{x})}\mathrm{d}\mathbf{z} \\ &= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k})p(\mathbf{z}_{>k})\frac{p(\mathbf{x}|\mathbf{z})}{q(\mathbf{z}_{>k}|\mathbf{x})}\mathrm{d}\mathbf{z} \\ &= \log \int q(\mathbf{z}_{>k}|\mathbf{x})p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k})\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}\mathrm{d}\mathbf{z} \\ &\geq \mathbb{E}_{p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k})}\left[\log q(\mathbf{z}_{>k}|\mathbf{x})\frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}\right] \\ &\geq \mathbb{E}_{p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k})q(\mathbf{z}_{>k}|\mathbf{x})}\left[\log \frac{p(\mathbf{x}|\mathbf{z})p(\mathbf{z}_{>k})}{q(\mathbf{z}_{>k}|\mathbf{x})}\right] \equiv \mathcal{L}^{>k} \,.\end{aligned} \tag{9}$$

Similar to the $L = 2$ case above, we have defined

$$q(\mathbf{z}_{>k}|\mathbf{x}) = q(\mathbf{z}_{>k}|d_k(\mathbf{x}))$$

with d_k defined recursively as

$$d_k(\mathbf{x}) = \mathbb{E}\left[q(\mathbf{z}_k|d_{k-1}(\mathbf{x}))\right] \,, \qquad d_0(\mathbf{x}) = \mathbf{x} \,.$$

That is, we simply consider the inference network below $\mathbf{z}_{k+1}$ to be a deterministic encoder and forward pass the mode of each preceding variational distribution.

Additionally, we obtain $p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k})p(\mathbf{z}_{>k})$ by splitting

$$p(\mathbf{z}) = p(\mathbf{z}_L)p(\mathbf{z}_{L-1}|\mathbf{z}_L)\cdots p(\mathbf{z}_1|\mathbf{z}_2)$$

at index k. Importantly, we then evaluate

$$p(\mathbf{z}_{>k}) = p(\mathbf{z}_L)p(\mathbf{z}_{L-1}|\mathbf{z}_L)\cdots p(\mathbf{z}_{k+1}|\mathbf{z}_{k+2})$$

with samples from $q(\mathbf{z}_{>k}|\mathbf{x})$ while

$$p(\mathbf{z}_{\leq k}|\mathbf{z}_{>k}) = p(\mathbf{z}_k|\mathbf{z}_{k+1})p(\mathbf{z}_{k-1}|\mathbf{z}_k)\cdots p(\mathbf{z}_1|\mathbf{z}_2)$$

is evaluated for $\mathbf{z}_k$ with $\mathbf{z}_{k+1} \sim q(\mathbf{z}_{>k}|\mathbf{x})$ and for $\mathbf{z}_{<k}$ with $\mathbf{z}_{>k}$ obtained conditionally from itself.

E. The complementary $\mathcal{L}^{<l}$ bound

We can generalize the $\mathcal{L}^{>k}$ bound by introducing the flipped version, $\mathcal{L}^{<l}$, which compared to $\mathcal{L}^{>k}$, instead samples the $L-l$ *highest* latent variables in the hierarchy from the prior $\mathbf{z}_l, \ldots, \mathbf{z}_L \sim p_\theta(\mathbf{z}_{\geq l}) = p_\theta(\mathbf{z}_l|\mathbf{z}_{l+1})\cdots p_\theta(\mathbf{z}_L)$ and the remaining lower latents from the approximate posterior $\hat{\mathbf{z}}_1, \ldots, \hat{\mathbf{z}}_{l-1} \sim q_\phi(\mathbf{z}_{<l}|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x})q_\phi(\mathbf{z}_2|\mathbf{z}_1)\cdots q_\phi(\mathbf{z}_{l-1}|\mathbf{z}_{l-2})$,

$$\mathcal{L}^{<l} = \mathbb{E}_{p_\theta(\mathbf{z}_{\geq l})q_\phi(\mathbf{z}_{<l}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})p_\theta(\mathbf{z}_{<l})}{q_\phi(\mathbf{z}_{<l}|\mathbf{x})}\right]. \quad (10)$$

Similar to $\mathcal{L}^{>k}$, we recover the regular ELBO for $l = L$. Contrary to $\mathcal{L}^{>k}$, this bound puts as much emphasis on the lowest latent variables as the regular ELBO but keeps track of large deviation from the unconditional prior in the top $L-l$ KL-terms since it is not guided by the approximate posterior for $\mathbf{z}_{>l}$. We hypothesize that this bound might be useful for OOD detection in cases where the discriminating factor is to be found in low-level statistics rather than high-level features.

Additionally, we can incorporate it in a generalized log likelihood-ratio between $\mathcal{L}^{<l}$ and $\mathcal{L}^{>k}$

$$LLR^{>k}_{<l} = \mathcal{L}^{<l} - \mathcal{L}^{>k}. \quad (11)$$

We hypothesize that this score, or the other possible permutations of it, might be useful for OOD detection but leave further examination to future work.

F. Note on the KL-term of hierarchical VAEs

In this research we choose model parameterizations relying on bottom-up inference (Burda et al., 2016),

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_1|\mathbf{x})\prod_{i=2}^{L} q_\phi(\mathbf{z}_i|\mathbf{z}_{i-1})\,. \quad (12)$$

We do this because bottom-up inference enables the model to learn covariance between the latent variables in the hierarchy. In the inference model, any latent variable is dependent on the latent variables below it in the hierarchy and, importantly, the top most latent variable is dependent on all other latent variables.

In contrast, a top-down inference model (Sønderby et al., 2016) has a topmost latent variable $\mathbf{z}_L$ that is independent of the other latent variables and is directly given by $\mathbf{x}$.

$$q_\phi(\mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{z}_L|\mathbf{x})\prod_{i=L-1}^{1} q_\phi(\mathbf{z}_i|\mathbf{z}_{i+1})\,. \quad (13)$$

This, in essence, makes $\mathbf{z}_L$ a mean-field approximation without any covariance structure tying it to the other latent variables, $\mathrm{Cov}(z_{L,i}, z_{k,j}) = 0$ for $k < L$. Furthermore, since the approximate posterior (and the prior) typically have diagonal covariance, $\mathbf{z}_L$ is also mean-field within its own elements, $\mathrm{Cov}(z_{L,i}, z_{L,j}) = 0$ for $i \neq j$.

We hypothesize that the covariance of latent variables towards the top of the hierarchy with other latent variables is important for learning semantic representations. However, top-down inference models are easier to optimize as has recently been demonstrated (Sønderby et al., 2016; Vahdat & Kautz, 2020; Child, 2021).

In the following, we inspect the differences between the ELBO used for bottom-up inference and the ELBO used for top-down inference and show that it is not generally possible to decompose the total KL-divergence into separate KL-divergences per latent variable. Specifically, for top-down inference it is possible to obtain KL-divergence at the top-most latent variable and an expectation of a KL-divergence for the other latent variables. For bottom-up inference, the resulting terms are no longer KL-divergences except at the top-most latent variable.

We ask the question whether models relying on top-down inference are impeded in their use for semantic OOD detection, or whether they still learn to assign a more semantic representation in the top-most variables simply due to the flexibility of the deterministic neural network layers. This remains an open research question.

F.1. Bottom-up inference

By splitting up the expectation, we can write the ELBO of a two-layer bottom-up hierarchical VAE as

$$\begin{aligned}\log p(\mathbf{x}) \geq\ & \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p(\mathbf{x}|\mathbf{z}_1)\right] \quad (14)\\ &+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p(\mathbf{z}_1|\mathbf{z}_2) - \log q(\mathbf{z}_1|\mathbf{x})\right]\\ &+ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})}\left[\log p(\mathbf{z}_2) - \log q(\mathbf{z}_2|\mathbf{z}_1)\right]\ .\end{aligned}$$

We can write out the expectations in order to derive the KL-divergence terms of the bottom-up ELBO:

$$\begin{aligned}\log p(\mathbf{x}) \geq & \int\int \log p(\mathbf{x}|\mathbf{z}_1) \mathrm{d}\mathbf{z}_2\mathbf{z}_1 \\ & + \int q(\mathbf{z}_1|\mathbf{x}) \int q(\mathbf{z}_2|\mathbf{z}_1) \log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{x})} \mathrm{d}\mathbf{z}_2\mathbf{z}_1 \\ & + \int q(\mathbf{z}_1|\mathbf{x}) \int q(\mathbf{z}_2|\mathbf{z}_1) \log \frac{p(\mathbf{z}_2)}{q(\mathbf{z}_2|\mathbf{z}_1)} \mathrm{d}\mathbf{z}_2\mathbf{z}_1 \ .\end{aligned} \tag{15}$$

From the above, we can see that since the decomposition is in a reverse order, we cannot derive the KL-divergence for the second term. This will hold in general for L-layered models for any latent variables $\mathbf{z}_1, ..., \mathbf{z}_{L-1}$:

$$\begin{aligned}\log p(\mathbf{x}) \geq & \ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1)\right] \\ & + \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \left[\mathbb{E}_{q(\mathbf{z}_2|\mathbf{z}_1)} \left[\log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{x})}\right]\right] \\ & + \mathbb{E}_{q(\mathbf{z}_1|\mathbf{x})} \left[-D_{\mathrm{KL}}[q(\mathbf{z}_2|\mathbf{z}_1)||p(\mathbf{z}_2)]\right] \ .\end{aligned} \tag{16}$$

F.2. Top-down inference

By splitting up the expectation, we can write the ELBO of a two-layer top-down hierarchical VAE as

$$\begin{aligned}\log p(\mathbf{x}) \geq & \ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1)\right] \\ & + \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_2|\mathbf{x}) - \log q(\mathbf{z}_2|\mathbf{x})\right] \\ & + \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{z}_1|\mathbf{z}_2) - \log q(\mathbf{z}_1|\mathbf{z}_2)\right] \ .\end{aligned} \tag{17}$$

We can write out the expectations in order to derive the KL-divergence terms:

$$\begin{aligned}\log p(\mathbf{x}) \geq & \int\int \log p(\mathbf{x}|\mathbf{z}_1) d\mathbf{z}_1\mathbf{z}_2 \\ & + \int q(\mathbf{z}_2|\mathbf{x}) \log \frac{p(\mathbf{z}_2|\mathbf{x})}{q(\mathbf{z}_2|\mathbf{x})} d\mathbf{z}_2 \\ & + \int q(\mathbf{z}_2|\mathbf{x}) \int q(\mathbf{z}_1|\mathbf{z}_2) \log \frac{p(\mathbf{z}_1|\mathbf{z}_2)}{q(\mathbf{z}_1|\mathbf{z}_2)} d\mathbf{z}_1\mathbf{z}_2 \ .\end{aligned} \tag{18}$$

The KL-divergence terms can now easily be computed by:

$$\begin{aligned}\log p(\mathbf{x}) \geq & \ \mathbb{E}_{q(\mathbf{z}_1,\mathbf{z}_2|\mathbf{x})} \left[\log p(\mathbf{x}|\mathbf{z}_1)\right] \\ & - D_{\mathrm{KL}}[q(\mathbf{z}_2|\mathbf{x})||p(\mathbf{z}_2)] \\ & - \mathbb{E}_{q(\mathbf{z}_2|\mathbf{x})} \left[D_{\mathrm{KL}}[q(\mathbf{z}_1|\mathbf{z}_2)||p(\mathbf{z}_1|\mathbf{z}_2)]\right] \ .\end{aligned} \tag{19}$$

Note that the KL-divergence in the second layer is not exact since it is dependent on the sample-noise from the layer below. An exact solution can only be derived if the latent variables $\mathbf{z}$ are all conditionally independent. However, this comes at the cost of not learning a covariance structure.

G. Additional results

We provide additional results for a model trained on FashionMNIST in Table 5, a model trained on MNIST in Table 6, a model trained on CIFAR10 in Table 4 and a model trained on SVHN in Table 3.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
		Trained on SVHN		
CIFAR10	$L^{>0}$	0.992	0.993	0.004
CIFAR10	$L^{>1}$	0.988	0.990	0.002
CIFAR10	$L^{>2}$	0.746	0.756	0.468
CIFAR10	$LLR^{>1}$	0.939	0.950	0.052
SVHN	$L^{>0}$	0.599	0.587	0.702
SVHN	$L^{>1}$	0.555	0.543	0.755
SVHN	$L^{>2}$	0.403	0.431	0.869
SVHN	$LLR^{>1}$	0.489	0.484	0.799

Table 3. Additional results for the HVAE model trained on SVHN. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
		Trained on CIFAR10		
SVHN	$L^{>0}$	0.083	0.318	0.974
SVHN	$L^{>1}$	0.097	0.320	0.972
SVHN	$L^{>2}$	0.693	0.725	0.599
SVHN	$LLR^{>2}$	0.811	0.837	0.394
CIFAR10	$L^{>0}$	0.485	0.488	0.817
CIFAR10	$L^{>1}$	0.467	0.476	0.822
CIFAR10	$L^{>2}$	0.411	0.433	0.869
CIFAR10	$LLR^{>1}$	0.469	0.479	0.835

Table 4. Additional results for the HVAE model trained on CIFAR10. All results computed with 1000 importance samples.

We note that while the likelihood is highly unreliable across the datasets, the proposed log likelihood-ratio score is consistent and always allows correct OOD detection with high AUROC↑.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on FashionMNIST				
MNIST	$\mathcal{L}^{>0}$	0.268	0.363	0.882
MNIST	$\mathcal{L}^{>1}$	0.593	0.591	0.658
MNIST	$\mathcal{L}^{>2}$	0.712	0.750	0.548
MNIST	$LLR^{>1}$	0.986	0.987	0.011
notMNIST	$\mathcal{L}^{>0}$	0.916	0.932	0.116
notMNIST	$\mathcal{L}^{>1}$	0.983	0.986	0.000
notMNIST	$\mathcal{L}^{>2}$	0.997	0.997	0.000
notMNIST	$LLR^{>1}$	0.998	0.998	0.000
KMNIST	$\mathcal{L}^{>0}$	0.690	0.694	0.554
KMNIST	$\mathcal{L}^{>1}$	0.835	0.863	0.359
KMNIST	$\mathcal{L}^{>2}$	0.844	0.875	0.339
KMNIST	$LLR^{>1}$	0.974	0.977	0.017
Omniglot28x28	$\mathcal{L}^{>0}$	0.898	0.837	0.166
Omniglot28x28	$\mathcal{L}^{>1}$	0.991	0.989	0.011
Omniglot28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
Omniglot28x28	$LLR^{>2}$	1.000	1.000	0.000
Omniglot28x28Inverted	$\mathcal{L}^{>0}$	0.261	0.361	0.879
Omniglot28x28Inverted	$\mathcal{L}^{>1}$	0.450	0.431	0.709
Omniglot28x28Inverted	$\mathcal{L}^{>2}$	0.557	0.574	0.678
Omniglot28x28Inverted	$LLR^{>1}$	0.954	0.954	0.050
SmallNORB28x28	$\mathcal{L}^{>0}$	0.982	0.984	0.000
SmallNORB28x28	$\mathcal{L}^{>1}$	0.998	0.998	0.000
SmallNORB28x28	$\mathcal{L}^{>2}$	1.000	1.000	0.000
SmallNORB28x28	$LLR^{>2}$	0.999	0.999	0.002
SmallNORB28x28Inverted	$\mathcal{L}^{>0}$	0.965	0.971	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>1}$	0.997	0.992	0.000
SmallNORB28x28Inverted	$\mathcal{L}^{>2}$	0.981	0.985	0.000
SmallNORB28x28Inverted	$LLR^{>2}$	0.941	0.946	0.069
FashionMNIST	$\mathcal{L}^{>0}$	0.476	0.484	0.816
FashionMNIST	$\mathcal{L}^{>1}$	0.475	0.482	0.817
FashionMNIST	$\mathcal{L}^{>2}$	0.475	0.484	0.823
FashionMNIST	$LLR^{>1}$	0.488	0.496	0.811

Table 5. Additional results for the HVAE model trained on FashionMNIST. All results computed with 1000 importance samples.

OOD dataset	Metric	AUROC↑	AUPRC↑	FPR80↓
Trained on MNIST				
FashionMNIST	$L^{>0}$	1.000	1.000	0.000
FashionMNIST	$L^{>1}$	1.000	1.000	0.000
FashionMNIST	$L^{>2}$	0.981	0.983	0.003
FashionMNIST	$LLR^{>1}$	0.999	0.999	0.000
notMNIST	$L^{>0}$	1.000	1.000	0.000
notMNIST	$L^{>1}$	1.000	1.000	0.000
notMNIST	$L^{>2}$	1.000	1.000	0.000
notMNIST	$LLR^{>1}$	1.000	0.999	0.000
KMNIST	$L^{>0}$	1.000	1.000	0.000
KMNIST	$L^{>1}$	1.000	1.000	0.000
KMNIST	$L^{>2}$	0.987	0.987	0.011
KMNIST	$LLR^{>1}$	0.999	0.999	0.000
Omniglot28x28	$L^{>0}$	1.000	1.000	0.000
Omniglot28x28	$L^{>1}$	1.000	1.000	0.000
Omniglot28x28	$L^{>2}$	1.000	1.000	0.000
Omniglot28x28	$LLR^{>1}$	1.000	1.000	0.000
Omniglot28x28Inverted	$L^{>0}$	0.862	0.902	0.205
Omniglot28x28Inverted	$L^{>1}$	0.923	0.943	0.056
Omniglot28x28Inverted	$L^{>2}$	0.749	0.691	0.411
Omniglot28x28Inverted	$LLR^{>1}$	0.944	0.953	0.057
SmallNORB28x28	$L^{>0}$	1.000	1.000	0.000
SmallNORB28x28	$L^{>1}$	1.000	1.000	0.000
SmallNORB28x28	$L^{>2}$	1.000	1.000	0.000
SmallNORB28x28	$LLR^{>1}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$L^{>0}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$L^{>1}$	1.000	1.000	0.000
SmallNORB28x28Inverted	$L^{>2}$	0.977	0.980	0.001
SmallNORB28x28Inverted	$LLR^{>1}$	0.985	0.987	0.000
MNIST	$L^{>0}$	0.488	0.486	0.807
MNIST	$L^{>1}$	0.469	0.469	0.816
MNIST	$L^{>2}$	0.514	0.505	0.791
MNIST	$LLR^{>2}$	0.515	0.507	0.792

Table 6. Additional results for the HVAE model trained on MNIST. All results computed with 1000 importance samples.