HIERARCHICAL MULTIMODAL VARIATIONAL AUTOENCODERS

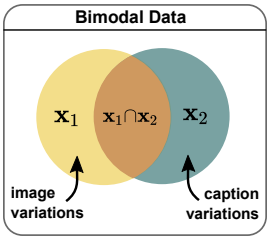
Anonymous authors
Paper under double-blind review

ABSTRACT

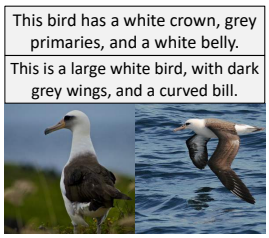
Humans find structure in natural phenomena by absorbing stimuli from multiple input sources such as vision, text, and speech. We study the use of deep generative models that generate multimodal data from latent representations. Existing approaches generate samples using a single shared latent variable, sometimes with marginally independent latent variables to capture modality-specific variations. However, there are cases where modality-specific variations depend on the kind of structure shared across modalities. To capture such heterogeneity, we propose a hierarchical multimodal VAE (HMVAE) that represents modality-specific variations using latent variables *dependent* on a shared top-level variable. Our experiments on the CUB and the Oxford Flower datasets show that the HMVAE can represent multimodal heterogeneity and outperform existing methods in sample generation quality and quantitative measures as the held-out log-likelihood.

1 INTRODUCTION

Data modalities represent different perspectives of the same concept. Generative models can learn from such data by reproducing it, which requires representing structures across modalities. Those models have a wide range of applications, such as zero-shot classification (Xian et al., 2017) or image caption generation (Vinyals et al., 2015). One difficulty in multimodal learning lies in the differing probabilistic structures across modalities. Fig. 1 shows an example where the common concept "*seabirds*" is described in two modalities, images and captions. Each modality contains specific variations: images reveal visual variations regarding texture, background, lighting, and pose, while captions characterize physical traits through various writing styles. Our goal is to develop generative models that capture variations in both modalities.



(a)



(b)

Figure 1: **Example of modality-specific unimodal variations.** (a) Venn diagram. (b) Image and text descriptions of *seabirds*.

One line of previous works (Zhu et al., 2017; Zhang et al., 2017; Reed et al., 2016) tackled this challenge with conditional generative adversarial networks (GANs) (Mirza and Osindero, 2014). These models generate samples for one modality conditioned on another modality and are optimized via competition between a generator and a discriminator. In contrast, Suzuki et al. (2016) used variational autoencoders (VAEs) to *jointly* generate M modalities from a learned latent representation. Both generative model families can be effective for either generating the data itself or generating feature vector representations of the data. The latter can be helpful for tasks such as zero-shot classification (Xian et al., 2018b) or reinforcement learning (Bruce et al., 2017). We focus on VAEs, which are explicit density estimators and can maximize the likelihood of all modality-specific variations. In contrast, GANs are implicit estimators, which can cause the generator to disproportionally favor specific variations in the data (Razavi et al., 2019).

Some multimodal VAEs (Wu and Goodman, 2018; Shi et al., 2019) assume conditional independence between modalities given a single latent variable $\boldsymbol{g}$ that captures all relevant information (Fig. 3a). This formulation may disregard unimodal variations. Other works (Huang et al., 2018;

Hsu and Glass, 2018; Mahajan et al., 2020; Sutter et al., 2020; Daunhawer et al., 2021; Lee and Pavlovic, 2021) have introduced disentangled latent variables $\boldsymbol{z}_{1:M}$, which are marginally independent of a shared latent variable $\boldsymbol{g}$ and represent structure specific to modality i (Fig. 3b). A disentangled latent representation may neglect the dependencies between unimodal and shared variations. Note that we discuss two inference network choices for the disentanglement paradigm in App. D.

Our work is motivated by data generating processes where the variations within a modality depend on the patterns shared across modalities. Consider captioning (modality 1) pictures of birds (modality 2). The images of seabirds can have sky or water in the background (Fig. 1), while those of warblers typically have forest backgrounds. Here, the bird species (shared pattern across modalities) dictates variations in the image modality. Consider a generative model where $\boldsymbol{g}$ represents structure shared across modalities and $\boldsymbol{z}_1$ variations specific to the image modality. When the two latent variables $\boldsymbol{g}$ and $\boldsymbol{z}_1$ are independent, the decoder that maps from latent variables onto images would effectively have to learn two entirely distinct functions - one for seabirds (where the independent variations determine the sky or water background) and another for warblers (where the independent variations determine the forest background). We argue that this aspiration is theoretically learnable but practically challenging because it may require a large model with disproportional capacity (that may generalize poorly, is challenging to train, or requires abundant data). In contrast, we suggest that a hierarchical latent representation is an inductive bias that captures realistic data variations and guides learning. Edges between $\boldsymbol{g}$ and $\boldsymbol{z}_{1:M}$ give the model the flexibility to decide which unimodal variations to capture given a shared latent concept. For example, a hierarchical model could *adaptively* learn that $\boldsymbol{z}_1$ for seabirds solely captures sky or water background variations while $\boldsymbol{z}_1$ for warblers captures forest background variations. This can help the decoder to share features between seabirds and warblers.

Contributions **(i)** We propose a hierarchical multimodal VAE (HMVAE) that incorporates a hierarchy between shared and modality-specific latent variables. **(ii)** We compare the HMVAE to several state-of-the-art baselines on the CUB and the Oxford Flower datasets. We report improved quantitative measures and diversely generated data of enhanced fidelity.

2 Background and Related Work

Variational Autoencoders (VAEs) (Kingma and Welling, 2013; Rezende et al., 2014) are deep generative models that represent the joint distribution $p_\theta(\boldsymbol{x}, \boldsymbol{z})$ using neural networks with parameters θ, where $\boldsymbol{x} \in \mathbb{R}^D$ is the observed vector and $\boldsymbol{z} \in \mathbb{R}^{D'}$ is the latent vector. As the true posterior $p_\theta(\boldsymbol{z}|\boldsymbol{x})$ is intractable, an approximate posterior $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ with parameters ϕ is used for inference. The parameters θ, ϕ are usually trained by maximizing the evidence lower bound (ELBO) for the marginal likelihood:

$$\mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log \frac{p_\theta(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right] \leq \log p_\theta(\boldsymbol{x}). \tag{1}$$

Many efforts have been made to increase the expressivity of VAEs, e.g., by improving the prior of $\boldsymbol{z}$ (Chen et al., 2016; Tomczak and Welling, 2018), and by introducing auxiliary latent variables (Maaløe et al., 2016). Our approach lies in the latter paradigm.

Hierarchical VAEs (HVAEs) (Rezende et al., 2014) have a hierarchical latent structure, where the topmost latent variable, drawn from an unconditional prior $p_\theta(\boldsymbol{z}_L)$, represents global features. The lower variables, drawn from conditional priors $p_\theta(\boldsymbol{z}_i|\boldsymbol{z}_{i+1})$, complement local characteristics in order to reconstruct the observed data via $p_\theta(\boldsymbol{x}|\boldsymbol{z}_1)$.

Sønderby et al. (2016) found the tendency for HVAEs to not effectively use higher-level latent variables when they are trained using inference networks of the form $q_\phi(\boldsymbol{z}_{i+1}|\boldsymbol{z}_i)$. They proposed to first infer the top-level variable $\boldsymbol{z}_L$ directly with $q_\phi(\boldsymbol{z}_L|\boldsymbol{x})$ and then infer the intermediate variables $\{\boldsymbol{z}_i\}$ with both bottom-up and top-down information through $q_{\phi,\theta}(\boldsymbol{z}_i|\boldsymbol{z}_{i+1}, \boldsymbol{x})$ for hierarchical level $i \in \{0, ..., L-1\}$. The bottom-up and top-down information for hierarchical level i are encoded as hidden variables through neural networks: $\boldsymbol{b}_i = f_{\phi,i}(\boldsymbol{x})$ and $\boldsymbol{t}_i = f_{\phi,\theta,i}(\boldsymbol{z}_{i+1})$. We make use of this idea in our work, concatenate both hidden states and pass the result to an MLP that parameterizes the respective posterior (see App. C.1 for further details). This inference procedure can result in improved density estimation and sample generation performance (Sønderby et al., 2016; Maaløe et al., 2019; Vahdat and Kautz, 2020; Child, 2020).

Multimodal VAEs represent M modalities $\boldsymbol{x}_{1:M} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_M\}$ that are assumed to be conditionally independent given a shared representation $\boldsymbol{g}$: $p_\theta(\boldsymbol{x}_{1:M}|\boldsymbol{g}) = \prod_{m=1}^{M} p_\theta(\boldsymbol{x}_i|\boldsymbol{g})$. Motivated by the factorization of the true posterior, Wu and Goodman (2018) used a product of experts (PoE) formulation to parameterize the approximate posterior distribution over the shared latent variable, where each expert characterizes information within a modality:

$$q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M}) \propto p_\theta(\boldsymbol{g}) \prod_{m=1}^{M} q_\phi(\boldsymbol{g}|\boldsymbol{x}_m). \tag{2}$$

When all experts are Gaussian, the product posterior of any combination of experts is easily computed.

Shi et al. (2019) showed that an inference model with a Gaussian PoE formulation can be miscalibrated, i.e., overconfident experts $q_\phi(\boldsymbol{g}|\boldsymbol{x}_i)$ with low densities dominate the product $q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})$. They instead used a mixture of experts (MoE) formulation:

$$q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M}) = \frac{1}{M} \sum_{m=1}^{M} q_\phi(\boldsymbol{g}|\boldsymbol{x}_m). \tag{3}$$

With the MoE formulation, optimization is analogous to a vote, which can avoid unreasonable dominance by a single modality.

Other generative approaches Mahajan et al. (2020) applied normalizing flows (Rezende and Mohamed, 2015) to map between the latent spaces of $\boldsymbol{g}_1$ and $\boldsymbol{g}_2$ in a VAE, where $q_\phi(\boldsymbol{g}_i|\boldsymbol{x}_i)$ constitutes the posterior over $\boldsymbol{g}_i$. Normalizing flows can be more expressive than other distribution choices, such as Gaussians, which can help to represent complex multimodal relationships. However, optimizing the ELBO requires tractable sampling and density estimation in *either* direction, for example, by using coupling layers (Dinh et al., 2014; 2016). This can be a challenging constraint in practice.

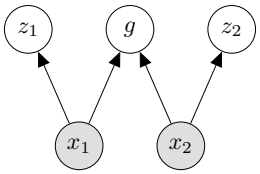


Figure 2: Inference model for MDVAE and MHVAE

Vasco et al. (2020) proposed a hierarchical multimodal generative model. Note that our works were developed independently and differ in the following aspects: first, Vasco et al. (2020) employ a non-hierarchical inference network (Fig. 17a). In contrast, we use a *"top-down"* hierarchical inference network that computes latent variables in the same order as the generative network (see this section's beginning and Figs. 3c and 3d). This architecture has the advantage that the importance distributions (posteriors) for the unimodal variables $\boldsymbol{z}$ can depend on all modalities (Fig. 3). In contrast, the respective importance distributions in the MHVAE only depend on single modalities (Fig. 2). We will discuss our method in detail in § 3. Second, Vasco et al. (2020) proposed an inference network that drops modality-specific hidden states at random. We use a mixture of experts posterior. Third, Vasco et al. (2020) evaluate their model on surjective data where single labels or attributes describe classes of images, i.e., there is not much variation in a single modality but lots in the other. We focus on a different data scenario where each modality has a large degree of variation.

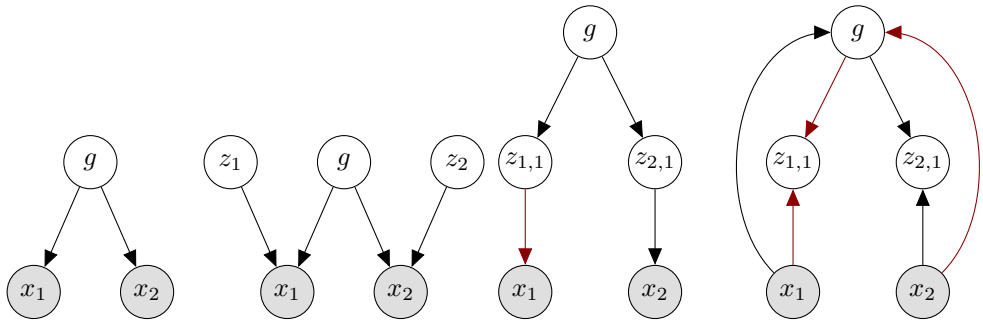
3 Multimodal Latent Hierarchies

We propose a hierarchical multimodal VAE (HMVAE) that captures the generative process of multiple modalities. The hyperparameter $L(m) \geq 2$ defines the number of hierarchical levels for modality m. If $L(m) = 2$, the hierarchy solely expands between the shared and unimodal variables. If $L(m) > 2$, the model incorporates additional unimodal hierarchies.

Generative model We assume conditional independence of modalities given a shared variable $\boldsymbol{g}$ (see Fig. 3c portraying the case of two modalities and two hierarchical levels):

$$p_{\phi,\theta}(\boldsymbol{x}, \boldsymbol{g}, \boldsymbol{z}) = \prod_{m=1}^{M} p_\theta(\boldsymbol{x}_m|\boldsymbol{z}_{m,1}) \cdot \Big(\prod_{i=1}^{L(m)-2} p_{\phi,\theta}(\boldsymbol{z}_{m,i}|\boldsymbol{z}_{m,i+1}) \Big) p_{\phi,\theta}(\boldsymbol{z}_{m,L(m)-1}|\boldsymbol{g}) \;\; p_\theta(\boldsymbol{g}). \tag{4}$$

The prior for the shared variable $\boldsymbol{g}$ is isotropic Gaussian. All conditional distributions for the intermediate variables $\{\boldsymbol{z}_{m,1:L(m)-1}\}$ for $m \in \{1, \dots M\}$ are also isotropic Gaussian, where mean and variance are parameterized using neural networks. We use the hierarchical formulation introduced in § 2 where some parameters are shared across the generative and inference networks. The conditional distributions for the observed variables $\{\boldsymbol{x}_m\}$ can, for example, be parameterized using isotropic Gaussian distributions (for continuous-valued data) or Bernoulli distributions (for binary data).



(a) MVAE, MMVAE (b) Multimodal disentanglement VAEs (c) HMVAE (proposed) (d) HMVAE inference

Figure 3: **Generative and inference models for two modalities.** Related work either represents shared structure in a single variable $\boldsymbol{g}$ (Fig. 3a) or additionally employs independent variables $\boldsymbol{z}_m$ for unimodal variations (Fig. 3b). App. D discusses respective inference network choices for Fig. 3b. In contrast, the HMVAE represents unimodal variations in conditional variables $\boldsymbol{z}_m$ (Fig. 3c). Figure 3d visualizes the proposed inference network, which is inspired by the unimodal inference network from Sønderby et al. (2016). The red edges visualize how an observed modality $\boldsymbol{x}_2$ can affect the inference of latent variable $\boldsymbol{z}_{1,1}$ in the other modality, and thus the representation of modality-specific variations for $\boldsymbol{x}_1$.

Inference model We extend Sønderby et al. (2016)'s hierarchical approach described in § 2 to the multimodal case (see Fig. 3c depicting the case of two modalities and two hierarchical levels):

$$q_{\phi,\theta}(\boldsymbol{g},\boldsymbol{z}|\boldsymbol{x}_{1:M}) = q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M}) \cdot \prod_{m=1}^{M} q_{\phi,\theta}(\boldsymbol{z}_{m,L(m)-1}|\boldsymbol{g},\boldsymbol{x}_m) \prod_{i=1}^{L(m)-2} q_{\phi,\theta}(\boldsymbol{z}_{m,i}|\boldsymbol{z}_{m,i+1},\boldsymbol{x}_m). \quad (5)$$

All distributions except the first factor $q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})$ are isotropic Gaussian with mean and variance inferred through neural networks from the conditional variables. The network employs skip connections from $\boldsymbol{x}_{1:M}$ to $\boldsymbol{g}$. The inference and generative networks share some parameters in the top-down networks that point from $\boldsymbol{g}$ to the lowest unimodal variable $\boldsymbol{z}_{m,i=0}$. This architecture can reinforce hierarchical decomposition (Sønderby et al., 2016). Furthermore, it ensures that the unimodal latent variables are conditioned on *all* modalities (as indicated by the red edges in Fig. 3) which helps learn crossmodal relationships. Section 2 and App. C.1 provide further details on the hierarchical architecture.

We parameterize the posterior distribution over the shared latent variable $q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})$ using a mixture of experts formulation (Eq. 3). We follow Shi et al. (2019) and assume that several modalities can entail modality-exclusive variations, i.e., some variations for modality i do not correlate with variations in modality $j \neq i$.

Optimization We train the HMVAE by maximizing the ELBO with stochastic backpropagation:

$$ELBO := \mathbb{E}_{q_{\phi,\theta}(\boldsymbol{g},\boldsymbol{z}|\boldsymbol{x}_{1:M})} \left[\log \frac{p_{\phi,\theta}(\boldsymbol{x}_{1:M},\boldsymbol{g},\boldsymbol{z})}{q_{\phi,\theta}(\boldsymbol{g},\boldsymbol{z}|\boldsymbol{x}_{1:M})} \right] \leq \log p_\theta(\boldsymbol{x}_{1:M}). \quad (6)$$

4 Experiments

We study the utility of our proposed HMVAE, showing that it outperforms state-of-the-art competitors. We will focus on two tasks: unconditional generation and crossmodal generation (e.g., caption-to-image generation).

We evaluate generated samples mainly in terms of semantic coherence and heterogeneity. Semantic coherence indicates whether a sample is related to its condition, e.g., an image $\boldsymbol{x}_1$ from $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ should be related to a caption $\boldsymbol{x}_2$. Heterogeneity describes diversity across samples for a single setting of the shared latent variable $\boldsymbol{g}$ and is essential to ensure that the generated samples capture the diversity of the data. As for the latent representations, we are particularly interested in the HMVAE:

can the model achieve latent hierarchical decomposition, i.e., does $\boldsymbol{z}_m$ encode conditionally independent variations for modality m given the shared latent representation $\boldsymbol{g}$? We also study classification tasks.

4.1 Experimental design

Models We reimplement the MVAE (Wu and Goodman, 2018), and the MMVAE (Shi et al., 2019). We keep the architecture and optimization hyperparameters similar across models. For the HMVAE, we set $L(m) = 2$ for all modalities $m \in \{1, 2\}$, i.e., the hierarchy solely expands between the shared and unimodal variables. We do not set $L(m) > 2$ to ensure that improvements in performance stem from hierarchical decomposition between $\boldsymbol{g}$ and $\boldsymbol{z}_{1:M}$ and not from more capacity in the unimodal networks. App. C describes our implementation in detail. All evaluated models train in less than five hours using a single GPU. The use of larger models may produce photorealistic images with a higher computational cost. For example, Child (2020) train a unimodal hierarchical VAE for around 2.5 weeks using 32 *V100* GPUs on datasets such as *FFHQ*. However, our goal is not photorealism but to investigate whether the proposed HMVAE represents multimodal data differently than the baselines.

Data We evaluate on the *CUB* (Wah et al., 2011) and the *Oxford Flower* (Nilsback and Zisserman, 2008) datasets. The modalities in both datasets are images and captions (text). We follow the common practice in the multimodal learning literature (Xian et al., 2018b; Sariyildiz and Cinbis, 2019; Schonfeld et al., 2019; Shi et al., 2019; 2020) and preprocess the data by extracting feature vectors for each modality in most of our experiments. This is because we are primarily interested in learning multimodal representations – and not state-of-the-art sample quality of generated images or captions. To visualize the image and caption features, we follow Shi et al. (2019; 2020), search for the nearest neighbor feature vector in the test set using the mean from $p(\boldsymbol{x}_i|\cdot)$, and visualize the respective image or caption. Using this practice, we can evaluate semantic coherence (e.g., are image and caption semantically related in caption-to-image generation?) and heterogeneity (how diverse are the visualized images or captions?). Section 4.3 additionally presents experiments where we maximize the likelihood of images (and not their features) to ensure that the HMVAE can represent data beyond feature vector space.

Across all experiments, we use the caption feature vectors provided by Zhang et al. (2017). These feature vectors $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$ were extracted using a CNN-RNN (Reed et al., 2016). Each image is paired with ten captions. We average the caption features for each image (except for the qualitative experiments where visualize captions explicitly). We find that this procedure can stabilize training and significantly reduce computational requirements. For § 4.2, we extract features $\boldsymbol{x}_1 \in \mathbb{R}^{2048}$ from a ResNet-101 trained on ImageNet. For § 4.3, we use images $\boldsymbol{x}_1 \in \mathbb{R}^{3\times64\times64}$ directly. We refer to App. C for details.

4.2 Feature generation

In this section, we maximize the likelihood of image features and caption features on the CUB and the Oxford Flower datasets. As described above, we use image features $\boldsymbol{x}_1 \in \mathbb{R}^{2048}$ and caption features $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$.

Image feature generation Figure 4 focuses on $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$. For the baselines, we sample $\boldsymbol{g}^{1:K} \sim q(\boldsymbol{g}|\boldsymbol{x}_2)$ to generate image features $p(\boldsymbol{x}_1|\boldsymbol{g}^k)$. For the proposed HMVAE, we sample $\boldsymbol{g} \sim q(\boldsymbol{g}|\boldsymbol{x}_2)$ and $\boldsymbol{z}_1^{1:K} \sim p(\boldsymbol{z}_1|\boldsymbol{g})$ to generate $p(\boldsymbol{x}_1|\boldsymbol{z}_1^k)$. Note that we cannot use $q(\boldsymbol{z}_1|\boldsymbol{g}, \boldsymbol{x}_1)$, because $\boldsymbol{x}_1$ is not observed (see Fig. 3d). We discover that all models generate image features that are mainly semantically coherent with the conditioning caption. However, the HMVAE generates samples of higher diversity than the baseline. Because we sample once over $\boldsymbol{g}$ and several times over $\boldsymbol{z}_1$, this diversity must be represented in $\boldsymbol{z}_1$.

Caption feature generation Figure 5 focuses on $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$. We reverse the procedure from the above paragraph to create the figure. For the baselines, we sample $\boldsymbol{g}^{1:K}$ from $q(\boldsymbol{g}|\boldsymbol{x}_1)$ to generate caption features $p(\boldsymbol{x}_2|\boldsymbol{g}^k)$. For the proposed HMVAE, we sample $\boldsymbol{g}$ from $q(\boldsymbol{g}|\boldsymbol{x}_1)$ and $\boldsymbol{z}_2^{1:K}$ from $p(\boldsymbol{z}_2|\boldsymbol{g})$ to generate caption feature vectors $p(\boldsymbol{x}_2|\boldsymbol{z}_2^k)$. We display the first 85 characters of each caption to ensure that the subplots do not overlap. We observe that all models can generate caption features that are coherent with the conditioning image feature. The proposed HMVAE, however, can

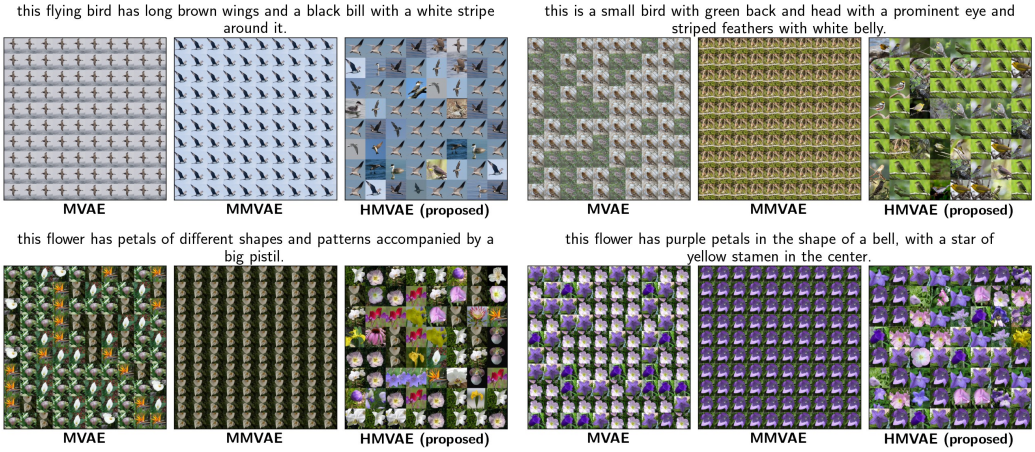


Figure 4: **CUB/Oxford Flower: Generating image feature vectors from caption feature vectors.** We generate image features, look up the nearest-neighbor feature in the test set, and visualize the associated image.

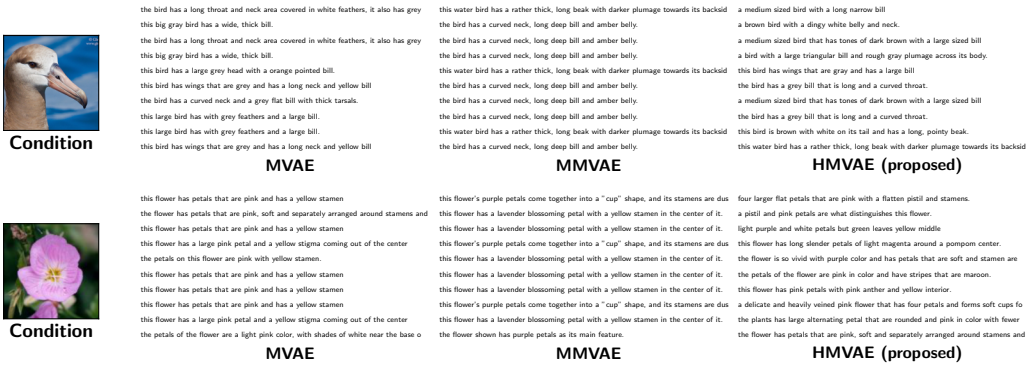


Figure 5: **CUB/Oxford Flower: Generating caption feature vectors from image feature vectors.** We generate caption features, look up the nearest-neighbor feature from the test set, and visualize the respective caption.

generate caption features of more variability. This heterogeneity must be encoded in $\boldsymbol{z}_2$, because we fix $\boldsymbol{g}$ and vary over $\boldsymbol{z}_2$.

Likelihood estimates Table 1 shows that the HMVAE achieves state-of-the-art performance across most evaluated likelihoods for all datasets.

Table 1: **Likelihood estimates.** We approximate the joint, marginal, and crossmodal likelihoods (defined in App. B) of test samples using 500 importance weighted samples. All models maximize the likelihood of image feature vectors and caption feature vectors. We report average and standard deviation over five runs per model. (higher is better)

Dataset and Model	$\log p(\boldsymbol{x}_1, \boldsymbol{x}_2)$	$\log p(\boldsymbol{x}_1)$	$\log p(\boldsymbol{x}_2)$	$\log p(\boldsymbol{x}_1 \vert \boldsymbol{x}_2)$	$\log p(\boldsymbol{x}_2 \vert \boldsymbol{x}_1)$
CUB					
MVAE (Wu and Goodman, 2018)	-2046.0 $\pm$ 6.6	-1902.3 $\pm$ 8.8	142.8 $\pm$ 16.1	-3594.0 $\pm$ 337.0	-3024.9 $\pm$ 297.4
MMVAE (Shi et al., 2019)	-2633.5 $\pm$ 47.7	-2075.5 $\pm$ 8.2	180.9 $\pm$ 10.5	-3227.2 $\pm$ 145.1	-2022.2 $\pm$ 77.5
HMVAE **(this work)**	**-1424.8** $\pm$ **14.8**	**-1854.8** $\pm$ **14.2**	**439.6** $\pm$ **2.0**	**-2382.9** $\pm$ **19.9**	**-879.7** $\pm$ **31.8**
Oxford Flower					
MVAE (Wu and Goodman, 2018)	-2471.3 $\pm$ 59.4	**-2433.4** $\pm$ **22.5**	215.4 $\pm$ 21.9	-3067.7 $\pm$ 67.5	-5282.8 $\pm$ 835.1
MMVAE (Shi et al., 2019)	-3029.9 $\pm$ 50.5	-2575.5 $\pm$ 8.6	141.2 $\pm$ 26.0	-3179.3 $\pm$ 44.2	-3454.0 $\pm$ 218.4
HMVAE **(this work)**	**-2069.1** $\pm$ **40.0**	-2497.6 $\pm$ 21.7	**442.2** $\pm$ **27.7**	**-3009.3** $\pm$ **38.0**	**-1400.0** $\pm$ **77.1**

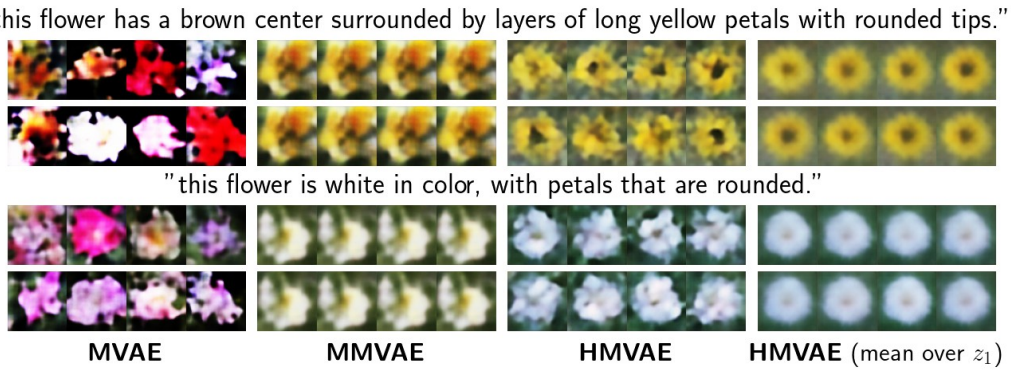


Figure 6: **Oxford Flower: crossmodal generation.** The first row presents unconditional image generation $p(\boldsymbol{x}_1|\boldsymbol{g})$. The second row displays caption-to-image generation $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$.

4.3 Image generation

In this section, we maximize the likelihood of images $\boldsymbol{x}_1 \in \mathbb{R}^{3\times64\times64}$ and caption feature vectors $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$. We use the same caption feature vectors as in the previous section.

Image generation Figure 6 focuses on $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ and $p(\boldsymbol{x}_1|\boldsymbol{g})$ using a convolutional network. We sample $\boldsymbol{g}^{1:K}$ from $q(\boldsymbol{g}|\boldsymbol{x}_2)$. For the baselines, we generate $p(\boldsymbol{x}_1|\boldsymbol{g}^k)$. For the HMVAE, we either sample conventionally (third column) or use the mean (last column) from $p(\boldsymbol{z}_1|\boldsymbol{g}^k)$ to generate $p(\boldsymbol{x}_1|\boldsymbol{z}_1^k)$. The MVAE's conditional image generations (second row) are reminiscent of the image distribution but display a noticeable lack of fidelity and semantic coherence. This observation is in line with the analysis by Shi et al. (2019), who argued that the MVAE struggles when faced with two modalities that each contain exclusive variations. The MMVAE displays high degrees of semantic coherence between generated images and the conditioning captions. However, the generated images lack visual detail and heterogeneity. The proposed HMVAE injects such missing details and heterogeneity. When sampling the mean over $\boldsymbol{z}_1$, the generated images display average heterogeneity. This observation indicates that $\boldsymbol{z}_1$ represents visual variations not in $\boldsymbol{g}$, i.e., hierarchical decomposition.

Table 2: **Oxford Flower:** Note that the MVAE improves FIDs, but lacks semantic coherence (see Fig. 6). We report average and standard deviation over five runs per model.

Model	Class-cond. FID for $p(\boldsymbol{x}_1\|\boldsymbol{x}_2)$	Zero-shot accuracy in latent space
MVAE	**273.89** $\pm$ **3.18**	10.04% $\pm$ 1.80%
MMVAE	319.13 $\pm$ 11.11	10.55% $\pm$ 1.59%
HMVAE	280.72 $\pm$ 2.48	**21.99%** $\pm$ **3.57%**

Table 2 presents class-conditional Fréchet Inception Distances (FIDs) (Heusel et al., 2017; Seitzer, 2020), which compare the distributions of true images and generated images. We compute the scores separately for each class and then average. Note that the MVAE's generates unconditional image samples of high quality (Fig. 11) which indicates that the conditional image samples may not be too far from their true manifold.

Latent representations Finally, we want to assess the usefulness of the latent representations without focusing on generative capability. Table 2 presents zero-shot classification performance in latent space, where we want to perform image classification solely given caption data. We perform classification in latent space (see App. A.2 for details) and observe that the proposed HMVAE's latent representations are most useful for zero-shot classification.

4.4 Comparison with the MDVAE and the MHVAE

Deeper hierarchies Multimodal disentanglement VAEs (MDVAEs) and the MHVAE (Vasco et al., 2020) employ additional unimodal variables $\boldsymbol{z}$ via a non-hierarchical inference network (Fig. 2). We argue that the proposed *hierarchical* inference network of the HMVAE (Fig. 3d) is a more natural choice to capture hierarchical multimodal structure. We assume that this advantage especially mate-

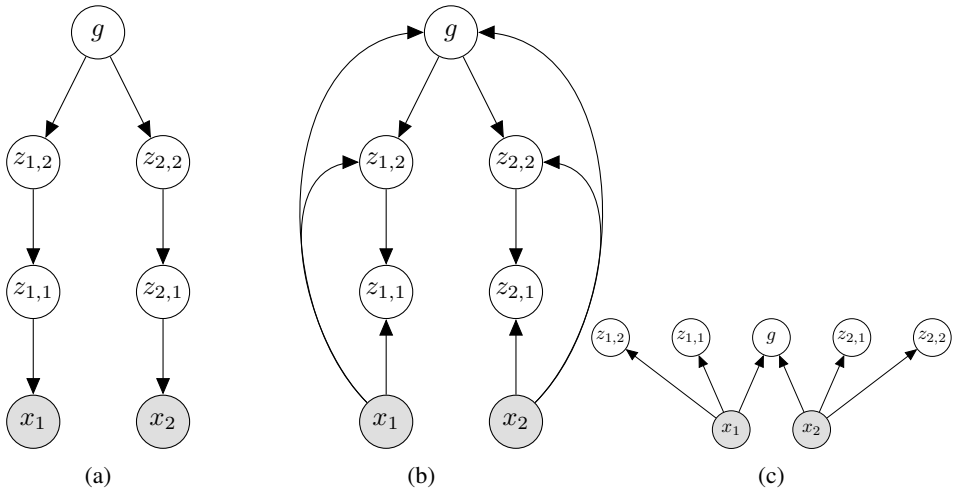


Figure 7: **Deeper hierarchies:** We assume that the proposed inference model of the HMVAE (b) naturally captures a deeper hierarchical structure (a). In contrast, a non-hierarchical inference model as in the MDVAE and the MHVAE (c) grows dissimilar from the generative model when increasing the number of unimodal variables. Note that this figure exemplarily shows the case of three hierarchical layers per modality for improved readability. The models can have an arbitrary number of unimodal variables.

rializes when using deeper hierarchies because a hierarchical inference network can explicitly capture dependencies between different levels of abstraction (Fig. 7). Therefore, we train the HMVAE with $L(1) = 5$ layers for the image modality (and $L(2) = 2$ layers for the caption feature modality).

Implementation details There are many implementation choices for MDVAEs (Huang et al., 2018; Hsu and Glass, 2018; Mahajan et al., 2020; Sutter et al., 2020; Daunhawer et al., 2021; Lee and Pavlovic, 2021). However, many related works employ a product of experts posterior (Eq. 2). This posterior choice can produce poor results given the type of data considered in this work (where there is significant variation in both modalities) (Shi et al., 2019). Therefore, we use a mixture of experts posterior (Eq. 3) as in the HMVAE. The implemented MDVAE is almost identical to the HMVAE–except that the unimodal variables $\boldsymbol{z}$ are marginally independent from $\boldsymbol{g}$ (Figs. 2 and 3b). We train the model by maximizing the ELBO:

$$\log p_\theta(\boldsymbol{x}_{1:M}) \geq \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(\boldsymbol{g}) \prod_{i=1}^{M} p_\theta(\boldsymbol{z}_i)}{q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M}) \prod_{i=1}^{M} q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} + \log \prod_{i=1}^{M} p_\theta(\boldsymbol{x}_i|\boldsymbol{g}, \boldsymbol{z}_i)\right], \tag{7}$$

where $\boldsymbol{g} \sim q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})$ and $\boldsymbol{z}_i \sim q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)$ with $i \in \{1, ..., M\}$. We use $\boldsymbol{z}_1 \in \mathbb{R}^8$, $\boldsymbol{z}_1 \in \mathbb{R}^{100}$, $\boldsymbol{z}_2 \in \mathbb{R}^8$ and find that increasing the unimodal latent sizes tends to decrease crossmodal coherence.

As for the HMVAE (Vasco et al., 2020), we implement the networks from Figs. 2 and 3c. We follow the authors and employ *domain dropout* to learn representations over multiple modalities. We use a uniform domain dropout distribution, i.e., maximize three ELBOs given importance samples from $q(\boldsymbol{g}|\boldsymbol{x}_1)$, $q(\boldsymbol{g}|\boldsymbol{x}_2)$, and $q(\boldsymbol{g}|\boldsymbol{x}_{1:M})$, respectively (similar to Wu and Goodman (2018), Eq. 19). The source code in the supplementary material provides details.

Results For the MDVAE, we generate images $p(\boldsymbol{x}_1|\boldsymbol{z}_1, \boldsymbol{g})$ from $\boldsymbol{z}_1 \sim p(\boldsymbol{z}_1)$ and $\boldsymbol{g} \sim q(\boldsymbol{g}|\boldsymbol{x}_2)$. Table 3 shows that the MDVAE only captures crossmodal dependencies to some extend. The qualitative results from Fig. 8 reveal that the MDVAE provides some semantic coherence in the generated images but fails to align some image variations with the shared concept[1]. This is expected because the generative network from Fig. 3b cannot capture dependencies between $\boldsymbol{z}_i$ and $\boldsymbol{g}$. Note that the Oxford Flower dataset contains many image variations independent from the caption which should be in favor of MDVAEs. In contrast, the proposed HMVAE does not struggle when sampling from the priors over $\boldsymbol{z}$ (which we do across all crossmodal experiments).

[1]MDVAEs have no problems to generate images from themselves because this allows sampling from the posterior (and not the prior) over $\boldsymbol{z}_1$.

(a) (b) (c) (d)

Figure 8: **Oxford Flower:** (a) MDVAE: image-to-image generation (a) MDVAE: image-to-caption generation (a) MHVAE: image-to-caption generation (misses semantic coherence) (a) HMVAE: image-to-caption generation

For the MHVAE and HMVAE, we generate images $p(\boldsymbol{x}_1|\boldsymbol{z}_1)$ from $\boldsymbol{z}_1 \sim p(\boldsymbol{z}_1|\boldsymbol{g})$ and $\boldsymbol{g} \sim q(\boldsymbol{g}|\boldsymbol{x}_2)$ Table 3 and Fig. 8 show that the MHVAE and the HMVAE capture crossmodal dependencies well. The HMVAE improves semantic coherence for some classes relative to the MHVAE. Note that the HMVAE has fewer trainable parameters (tbd:1.5mio vs. 5mio?) and maximizes the *"pure"* ELBO during training. In contrast, the MHVAE requires a warmup scheme for the KL-divergence loss factors (Vasco et al., 2020; Sønderby et al., 2016). To our knowledge, the HMVAE is the first hierarchical multimodal VAE that scales beyond two hierarchical levels for a modality and still captures crossmodal dependencies.

Table 3: **Oxford Flower: Fréchet Inception Distances (FIDs)** (lower is better)

Model	Class-cond. FID for $p(\boldsymbol{x}_1\|\boldsymbol{x}_2)$
MDVAE	286.0
MHVAE	271.0
HMVAE	**251.7**

5 Conclusion

We have proposed a hierarchical multimodal VAE (HMVAE) where unimodal latent variables depend on a shared latent variable. We have demonstrated that the model improves generative modeling performance on multimodal data. Future work may further improve sample quality, for example, by using unimodal hierarchies (Maaløe et al., 2019; Vahdat and Kautz, 2020; Child, 2020). Another approach towards this goal may be the study of different ways to parameterize conditional densities: for example, normalizing flows can improve the posterior's flexibility (Kingma et al., 2016), autoregressive decoders can explicitly represent the conditional dependencies across observed dimensions (Chen et al., 2018), and the transformer architecture may be a good candidate to replace the convolutional neural networks (Dosovitskiy et al., 2020). Note that we discuss societal implications in App. E.

Limitations Our experimental evaluation is restricted to dense modalities such as images and text. It is important to continue studying the model on sparser data often found in the real world. Furthermore, our goal was the representation of heterogeneity within modalities. Although the proposed HMVAE outperforms our baselines, the model is not perfect. For example, when generating images by sampling from the conditional distribution $p(\boldsymbol{x}_1|\boldsymbol{z}_1)$, where $\boldsymbol{z}_1 \sim p(\boldsymbol{z}_1|\boldsymbol{g})$ and $\boldsymbol{g} \sim q(\boldsymbol{g}|\boldsymbol{x}_2)$, the HMVAE must hallucinate a coherent set of visual representations (possibly conditional on a given caption). This can be challenging when the caption is vague and does not contain sufficient information for the inference network to infer the shared underlying structure correctly. In other words, the additional expressivity that the HMVAE obtains through the conditional latent variables also makes learning harder since $p(\boldsymbol{x}_1|\cdot)$ must represent a larger manifold for the proposed HMVAE than it does for the baselines.

References

J. Bruce, N. Sünderhauf, P. Mirowski, R. Hadsell, and M. Milford. One-shot reinforcement learning for robot navigation with interactive replay. *arXiv preprint arXiv:1711.10137*, 2017.

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

X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

X. Chen, N. Mishra, M. Rohaninejad, and P. Abbeel. Pixelsnail: An improved autoregressive generative model. In *International Conference on Machine Learning*, pages 864–872. PMLR, 2018.

R. Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.

I. Daunhawer, T. M. Sutter, R. Marcinkevičs, and J. E. Vogt. Self-supervised disentanglement of modality-specific and shared factors improves multimodal generative models. *Pattern Recognition*, 12544:459, 2021.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

L. Dinh, D. Krueger, and Y. Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500*, 2017.

W.-N. Hsu and J. Glass. Disentangling by partitioning: A representation learning framework for multimodal sensory data. *arXiv preprint arXiv:1805.11264*, 2018.

X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 172–189, 2018.

D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in neural information processing systems*, pages 4743–4751, 2016.

M. Lee and V. Pavlovic. Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2021.

L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. In *International conference on machine learning*, pages 1445–1453. PMLR, 2016.

L. Maaløe, M. Fraccaro, V. Liévin, and O. Winther. Biva: A very deep hierarchy of latent variables for generative modeling. In *Advances in neural information processing systems*, pages 6548–6558, 2019.

S. Mahajan, I. Gurevych, and S. Roth. Latent normalizing flows for many-to-many cross-domain mappings. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJxE8erKDH.

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems*, pages 14837–14847, 2019.

S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv preprint arXiv:1505.05770*, 2015.

D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1278–1286, Bejing, China, 22–24 Jun 2014. PMLR. URL http://proceedings.mlr.press/v32/rezende14.html.

M. B. Sariyildiz and R. G. Cinbis. Gradient matching generative networks for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2178, 2019.

E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata. Generalized zero-and few-shot learning via aligned variational autoencoders. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8247–8255, 2019.

M. Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.1.1.

Y. Shi, N. Siddharth, B. Paige, and P. Torr. Variational mixture-of-experts autoencoders for multimodal deep generative models. In *Advances in Neural Information Processing Systems*, pages 15692–15703, 2019.

Y. Shi, B. Paige, P. H. Torr, and N. Siddharth. Relating by contrasting: A data-efficient framework for multimodal generative models. *arXiv preprint arXiv:2007.01179*, 2020.

C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther. Ladder variational autoencoders. In *Advances in neural information processing systems*, pages 3738–3746, 2016.

T. M. Sutter, I. Daunhawer, and J. E. Vogt. Multimodal generative learning utilizing jensen-shannon-divergence. *arXiv preprint arXiv:2006.08242*, 2020.

M. Suzuki, K. Nakayama, and Y. Matsuo. Joint multimodal learning with deep generative models. *arXiv preprint arXiv:1611.01891*, 2016.

J. Tomczak and M. Welling. Vae with a vampprior. In *International Conference on Artificial Intelligence and Statistics*, pages 1214–1223. PMLR, 2018.

G. Tucker, D. Lawson, S. Gu, and C. J. Maddison. Doubly reparameterized gradient estimators for monte carlo objectives. *arXiv preprint arXiv:1810.04152*, 2018.

A. Vahdat and J. Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.

M. Vasco, F. S. Melo, and A. Paiva. Mhvae: a human-inspired deep hierarchical generative model for multimodal representation learning. *arXiv preprint arXiv:2006.02991*, 2020.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

M. Wu and N. Goodman. Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5575–5585, 2018.

Y. Xian, B. Schiele, and Z. Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4582–4591, 2017.

Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2251–2265, 2018a.

Y. Xian, T. Lorenz, B. Schiele, and Z. Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551, 2018b.

H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915, 2017.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

APPENDIX

We partition the Appendix into five sections:

A. **Additional experiments:** We visualize unconditional caption feature samples from the HMVAE for the Oxford Flower dataset. We then quantitatively evaluate the HMVAE's crossmodal capability for $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$. Both experiments are complementary to the ones from § 4, where we had evaluated unconditional image generation (Fig. 6) and quantitatively assessed $p(\boldsymbol{x}_1|\boldsymbol{x}_2)$ (Table 2). Furthermore, we provide additional results for the experiments from the main paper.

B. **Likelihood estimation:** App. B presents the estimators used to compute the likelihoods reported in Table 1.

C. **Implementation:** App. C describes our implementation in detail. App. C.1 complements the background section § 2 by describing details of the employed hierarchical VAE architecture. This section also presents dataset-agnostic model specifications for the baselines. We then focus on dataset-specific implementation specifications, where App. C.2 covers the CUB dataset and App. C.3 covers the Oxford Flower dataset. These sections discuss dataset statistics, model architectures, and training.

D. **MDVAE: inference networks:** This section discusses inference networks choices for multimodal disentanglement VAEs (MDVAEs).

E. **Societal impact:** This section discusses the societal impact of the proposed model.

A ADDITIONAL EXPERIMENTS

This section presents additional experimental results that complement § 4 from the main paper.

A.1 FEATURE GENERATION

In this section, we maximize the likelihood of image features $\boldsymbol{x}_1 \in \mathbb{R}^{2048}$ and caption features $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$ on the CUB dataset and the Oxford Flower dataset. We use the same experimental setup as in § 4.

Image feature generation Figure 9a presents generated samples where the captions are from the training set. We observe that the MMVAE still lacks variety in its generations, which indicates that this tendency is not caused by poor generalization. Figure 9b presents generated samples where the captions are from the test set. The HMVAE can represent different backgrounds (e.g., various sea or forest settings), even though this information is missing in the conditioning captions.

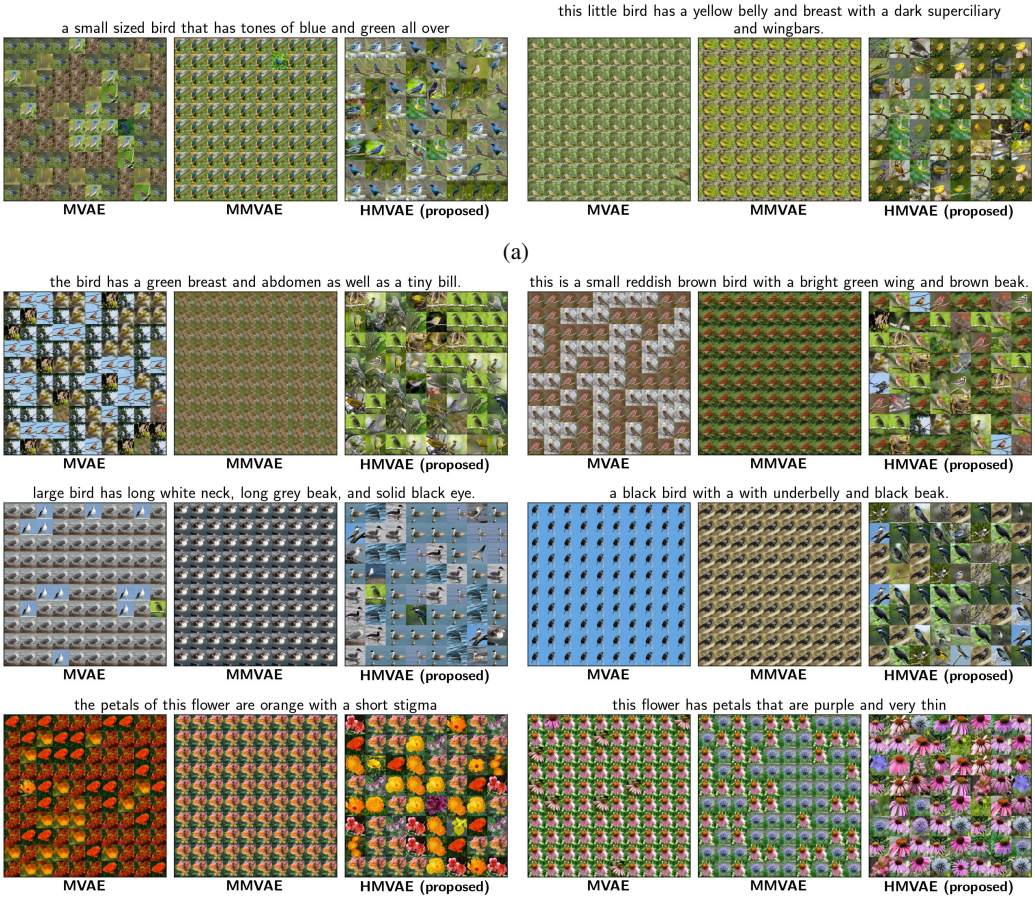


Figure 9: **CUB/Oxford Flower: Generating image feature vectors from caption feature vectors.** Plot (a) represents the training set, plot (b) the test set. We generate image features, look up the nearest-neighbor feature in the test set, and visualize the associated image.

Caption feature generation Figure 10 presents additional results for caption feature generation which confirm our findings from § 4.

Figure 10: **CUB/Oxford Flower: Generating caption feature vectors from image feature vectors.** We generate caption features, look up the nearest-neighbor feature from the test set, and visualize the respective caption.

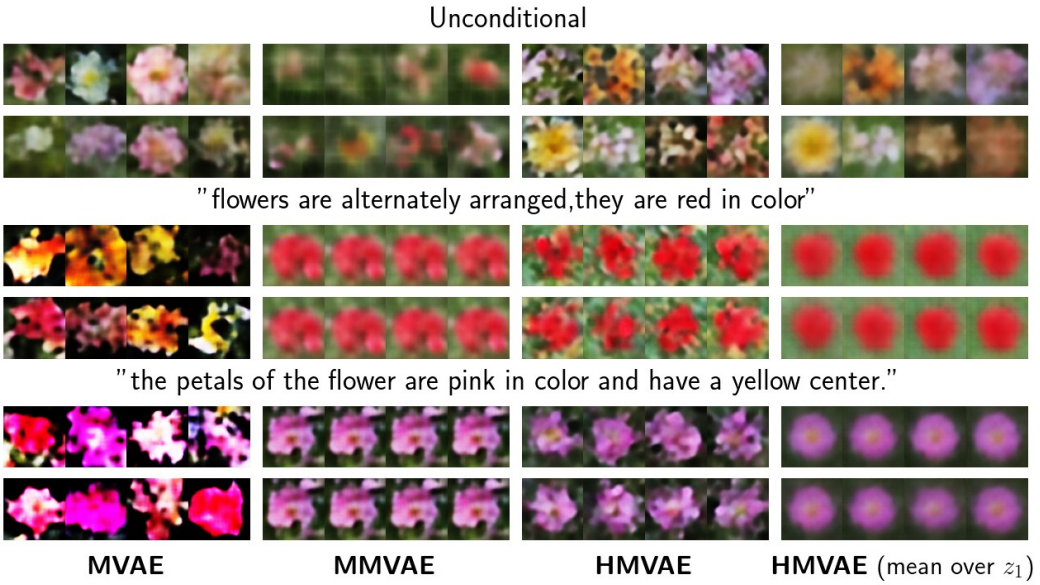


Figure 11: **Oxford Flower: image generation**

A.2 Oxford Flower: image generation and caption feature generation

In this section, we maximize the likelihood of images $\boldsymbol{x}_1 \in \mathbb{R}^{3\times64\times64}$ and caption feature vectors $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$. We use the same setup as in the main paper.

Image generation Figure 11 complements the results from Fig. 6. For the unconditional generation (first row), we sample $\boldsymbol{g}^{1:K}$ from the prior $p(\boldsymbol{g})$ and generate the images analogously to the conditional generation.

Caption feature generation Fig. 12 presents results for $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$. Figure 13 visualizes unconditional caption feature vector samples from the HMVAE. We first sample from $p(\boldsymbol{g})$ and then gener-

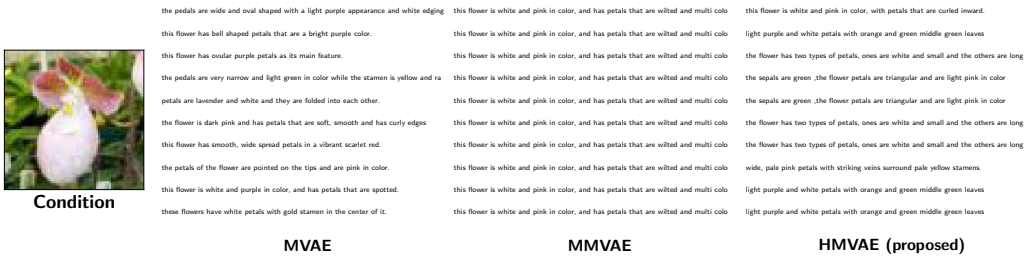


Figure 12: **Oxford Flower: caption feature generation from images.** We generate caption features, look up the nearest-neighbor feature from the test set, and visualize the respective caption.

the petals are bright yellow and curled in on each other, and the stem is thick and green.
these flowers have shredded yellow petals with orange stamen in the center of it.
the stigma are yellow and green, the petals are a pinkish color.
this flower has a ring formation of light purple oblong petals.
this flower's purple petals come together into a "cup" shape, and its stamens are dusted with yellowish pollen.
reminiscent of a gorgeous oil painting bursting with color this flower has petals and stamen that appear to explode outward and grab your attention.
an orange and yellow large flower with long pedals and a tan center.
this flower has purple petals as well as a green pistil.
this particular flower has petals that are dark pink on the outside
with the purple's flower petal falling off you can see the pedical underneath.
the flower is so vivid with purple color and has petals that are soft and stamen are white and sticking out from the centre
the petals are round and pink and white and the sepals are oval shaped and green.
the petals of the flower are a vibrant red color, with shades of orange throughout.
this flower has red petals as well as a white stamen.
this flower has petals that are pink and has a yellow stamen
this flower has petals that are purple with string stamen
this bright flower has three large pointed striped pink petals, and one rounded.
this flower has petals that are red and yellow and folded together
an orange and yellow large flower with long pedals and a tan center.
the pedicel has many flowers growing from it and the petals being pink.
this flower has petals that have a color combination of pink,white, yellow, and brown.
a bright gold, white, and black flowers with spiky look.
this flower has yellow petals, with brown stamen in the center.
this flower has petals that are orange and folded in together
this flower has petals that are purple and has yellow stamen
the pedicel has many flowers growing from it and the petals being pink.
the petals are red and fade fo fuschia and cover the stamen and pistil.
this flower is white and purple in color, and has petals that are wavy and thin.
this flower has a yellow-orange ovule and yellow and brown petals surrounding it, with grass like leaves.
the flower is oddly shaped and has a large round white petal and yellow accent petals.
light purple and white petals with orange and green middle green leaves
this orange and yellow flower has thin rounded petals and green sepals.
flower with an orange and yellow spiky center and large, droopy petals.
this flower is purple in color, and has petals that are wilted in appearance.
this flower has petals that are white with red points and large yellow center.
the pedicel has many flowers growing from it and the petals being pink.
the dark orange color on the wide, sturdy petals graduate to yellow in the center.
this flower has spiny, pink petals, and appears rough and sharp.
outer petals are greewn in color,inner petals are white pink in color
this flower has a brown pistil and pink downward facing petals.
this flower has a set of bold, white petals and a delicate framework.
the flower is pink with petals that are soft, thin and separately arranged around stamens
the flower is so big that has petals that are so thick, pink and has disc of stamens above it
this purple flower looks like a ball and has individual florets that have their own stamens.
this flower is yellow and white petals with the green stems
a flower with orange petals and red anther filaments in the center.
this flower has white petals with a stamen that is yellow
these flowers have shredded yellow petals with orange stamen in the center of it.
this flower has purple petals with maroon pistils and anther filaments.
this flower has large petals that range from yellow to pink.

Figure 13: **Oxford Flower: Unconditional caption feature generation for the caption modality.** We generate caption features, look up the nearest-neighbor feature from the test set, and visualize the respective caption.

ate $p(\boldsymbol{x}_2|\boldsymbol{z}_1)$ via $\boldsymbol{z}_1 \sim p(\boldsymbol{z}_1|\boldsymbol{g})$. We observe that the HMVAE can generate diverse samples without relying on any conditioning. Note that Fig. 6 from the main paper presents the analogous experiment for $p(\boldsymbol{x}_1|\boldsymbol{g})$.

Quantitative measures Section 4 already provided class-conditional FID scores to assess semantic coherence and heterogeneity for caption-to-image generation. In the following, we will quantitatively assess image-to-caption generation.

In Table 4, we classify caption features given image data[2]. For the baselines, we sample $\boldsymbol{g}$ from $q(\boldsymbol{g}|\boldsymbol{x}_1)$ to generate $p(\boldsymbol{x}_2|\boldsymbol{g})$. For the HMVAE, we sample $\boldsymbol{z}_2 \sim p(\boldsymbol{z}_2|\boldsymbol{g})$ and $\boldsymbol{g} \sim q(\boldsymbol{g}|\boldsymbol{x}_1)$ to generate $p(\boldsymbol{x}_2|\boldsymbol{z}_2)$. We use the means from $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$ as the classification support, and the caption features $p(\boldsymbol{x}_2)$ as the query. For each sample in the query, we search for the nearest neighbor in the

[2]Note that Table 2 from the main paper evaluated the opposite case, i.e., support and query are now switched. Because the caption data is arguably less informative than the images, we expect lower accuracies here.

support (using Euclidean distance) and predict the respective label. The reported scores represent the average classification accuracy. This measure is an indication of semantic coherence.

Table 4: **Oxford Flower: Miscellaneous measures.** Both the MMVAE and the HMVAE perform well in terms of semantic coherence (zero-shot classification accuracy). The MMVAE tends to produce homogeneous representations, which can enhance classification performance at the expense of heterogeneity (variance column). We report average and standard deviation over five runs per model.

Model	Zero-Shot Accuracy $p(\boldsymbol{x}_2\|\boldsymbol{x}_1)$ vs. $p(\boldsymbol{x}_2)$	Variances $p(\boldsymbol{x}_2\|\boldsymbol{x}_1)$
MVAE (Wu and Goodman, 2018)	8.68% $\pm$ 3.64%	0.5925 $\pm$ 0.0923
MMVAE (Shi et al., 2019)	**32.68% $\pm$ 3.00%**	0.0020 $\pm$ 0.0002
HMVAE **(this work)**	30.94% $\pm$ 2.08%	0.0270 $\pm$ 0.0039

As expected from the qualitative results from § 4, the MVAE's classification performance is lowest across all evaluated models, which indicates the insufficient representation of semantic coherence. Though the HMVAE performs well, the MMVAE obtains the highest classification accuracy. Based on the experiments from § 4, we assume that the MMVAE's homogeneous representations may be more suitable for classification tasks but often lack both heterogeneity and fidelity.

The last column of Table 4 presents variances over $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$. For each image $\boldsymbol{x}_1^k$, we sample $\boldsymbol{g}^{1:100} \sim q(\boldsymbol{g}|\boldsymbol{x}_2^k)$ to generate $p(\boldsymbol{x}_2|\boldsymbol{x}_1)$. We average over all means $\bar{\boldsymbol{x}_{\mathbf{2}}}^{1:100}$ to obtain a prototype representation $\boldsymbol{x}_2^p$. We then compute the pairwise Euclidean distances for $\{\bar{\boldsymbol{x}_{\mathbf{2}}}^k, \boldsymbol{x}_2^p\}$ for $k \in \{1, ..., 100\}$, calculate the per-dimension variances across all distances, and finally report the average across all dimensions. High variance indicates diverse representations. Importantly, diversity is only valuable when paired with semantic coherence, e.g., high classification performance. We observe that the MVAE scores highest on our variance measure but lacks semantic coherence (see classification performance). The HMVAE exceeds the MMVAE by an order of magnitude in our variance measure – while representing only slightly less semantic coherence. This demonstrates that the HMVAE improves heterogeneity while not (significantly) sacrificing semantic coherence.

Experimental setup for zero-shot experiment from § 4.3 We classify images given caption feature vectors. For the baselines, we compute the means of $q(\boldsymbol{g}|\boldsymbol{x}_1)$ and $q(\boldsymbol{g}|\boldsymbol{x}_2)$. For the HMVAE, we generate the mean of $p(\boldsymbol{z}_1|\boldsymbol{g})$ where $\boldsymbol{g}$ is the mean from either $q(\boldsymbol{g}|\boldsymbol{x}_1)$ or $q(\boldsymbol{g}|\boldsymbol{x}_2)$. Note that using $p(\boldsymbol{z}_1|\boldsymbol{g})$ is a harder task than using $q(\boldsymbol{z}_1|\boldsymbol{x}_1, \boldsymbol{g})$. However, using the latter would violate the zero-shot assumption. We then use the latent representations conditioning on the caption features as the classification support and the latent representations conditioned on the images as the query. For each sample in the query, we search for the nearest neighbor in the support (using Euclidean distance) and predict the respective label. The reported scores represent the average classification accuracy.

B LIKELIHOOD ESTIMATION

In this section, we formalize the estimators that we use to evaluate the HMVAE, the MVAE (Wu and Goodman, 2018), and the MMVAE (Shi et al., 2019). We report all likelihoods in *nats*.

B.1 JOINT LIKELIHOOD

We now focus on the joint likelihood over M modalities. The special case with $M = 1$ describes the unimodal likelihoods $p(\boldsymbol{x}_i)$.

Single latent variable We now discuss models that employ a single shared variable $\boldsymbol{g}$. We approximate the likelihood using K importance samples:

$$\log p(\boldsymbol{x}_{1:M}) \approx \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(\boldsymbol{x}_{1:M}, \boldsymbol{g}^k)}{q(\boldsymbol{g}^k|\boldsymbol{x}_{1:M})} \qquad \text{where} \quad \boldsymbol{g}^{1:K} \sim q(\boldsymbol{g}|\boldsymbol{x}_{1:M}) \tag{8}$$

We follow Shi et al. (2019), evaluate the mixture of experts VAE using stratified sampling, and estimate a tighter estimate, which averages over modalities inside the logarithm. This objective should only be used during the evaluation because it may lead to an outsized impact of more informative modalities during optimization. Notably, K samples of the joint posterior are equivalent to $T = K/M$ samples from each modality-specific posterior.

$$\log p(\boldsymbol{x}_{1:M}) \approx \log \frac{1}{M} \sum_{m=1}^{M} \frac{1}{T} \sum_{t=1}^{T} \frac{p(\boldsymbol{g}_m^t, \boldsymbol{x}_{1:M})}{q(\boldsymbol{g}_m^t|\boldsymbol{x}_m)} \qquad \text{where} \quad \boldsymbol{g}_m^{1:T} \sim q(\boldsymbol{g}|\boldsymbol{x}_m) \tag{9}$$

Latent hierarchy We now discuss models that employ both a shared variable $\boldsymbol{g}$ and conditionally independent, modality-specific variables $\boldsymbol{z}_{m,i}$. We approximate the likelihood using K importance samples and omit the super- and subscripts in the sampling definition for improved readability:

$$\begin{aligned} \log p(\boldsymbol{x}_{1:M}) &\approx \log \frac{1}{K} \sum_{k=1}^{K} \frac{p(\boldsymbol{x}_{1:M}, \boldsymbol{g}^k, \boldsymbol{z})}{q(\boldsymbol{g}^k, \boldsymbol{z}|\boldsymbol{x}_{1:M})} \\ &\text{where} \quad \boldsymbol{g}, \boldsymbol{z} \sim q(\boldsymbol{g}, \boldsymbol{z}|\boldsymbol{x}_{1:M}) \end{aligned} \tag{10}$$

The factorized generative and inference networks for the proposed latent hierarchy of the HMVAE are formalized in Eqs. 4 and 5 of the main paper. Note that both product and mixture of experts formulations of the shared posterior can be used within these hierarchical formulations.

B.2 CROSSMODAL LIKELIHOOD

In this section, we discuss estimators for the crossmodal likelihood $p(\boldsymbol{x}_t|\boldsymbol{x}_c)$ given a target modality t and a conditioning modality c. Analogously to Eq. 8, we evaluate the likelihoods using K importance samples.

Single latent variable

We first formalize the conditional likelihood similarly to Suzuki et al. (2016) and Wu and Goodman (2018):

$$\begin{aligned} \log p(\boldsymbol{x}_t|\boldsymbol{x}_c) &= \log \int_{\boldsymbol{g}} p(\boldsymbol{x}_t, \boldsymbol{g}|\boldsymbol{x}_c) d\boldsymbol{g} = \log \int_{\boldsymbol{g}} \frac{p(\boldsymbol{x}_t, \boldsymbol{x}_c, \boldsymbol{g})}{p(\boldsymbol{x}_c)} d\boldsymbol{g} \\ &= \underbrace{\log \mathbb{E}_{q(\boldsymbol{g}|\boldsymbol{x}_c)} \left[\frac{p(\boldsymbol{x}_t, \boldsymbol{x}_c, \boldsymbol{g})}{q(\boldsymbol{g}|\boldsymbol{x}_c)} \right]}_{①} - \underbrace{\log \mathbb{E}_{q(\boldsymbol{g}|\boldsymbol{x}_c)} \left[\frac{p(\boldsymbol{x}_c, \boldsymbol{g})}{q(\boldsymbol{g}|\boldsymbol{x}_c)} \right]}_{②} \end{aligned} \tag{11}$$

In Eq. 11, term ① constitutes an estimation of the joint likelihood $p(\boldsymbol{x}_{1:M})$ using an importance distribution that is conditioned on modality c. Term ② estimates the unimodal likelihood $p(\boldsymbol{x}_c)$.

Latent hierarchy We now generalize the estimator from Eq. 11 to the proposed latent hierarchy of the HMVAE. We first formalize the conditional likelihood:

$$p(\boldsymbol{x}_t|\boldsymbol{x}_c) = \int_{\boldsymbol{g},\boldsymbol{z}} p(\boldsymbol{x}_t, \boldsymbol{g}, \boldsymbol{z}|\boldsymbol{x}_c) d\boldsymbol{z} d\boldsymbol{g} = \int_{\boldsymbol{g},\boldsymbol{z}} \frac{p(\boldsymbol{x}_t, \boldsymbol{g}, \boldsymbol{z}, \boldsymbol{x}_c)}{p(\boldsymbol{x}_c)} d\boldsymbol{z} d\boldsymbol{g} \tag{12}$$

We assume that the modality-specific generative networks are conditionally independent given the shared representation (§ 3):

$$p(\boldsymbol{x}_t|\boldsymbol{x}_c) = \int_{\boldsymbol{g},\boldsymbol{z}} \frac{p(\boldsymbol{x}_t, \boldsymbol{z}_t|\boldsymbol{g}) p(\boldsymbol{x}_c, \boldsymbol{z}_c|\boldsymbol{g}) p(\boldsymbol{g})}{p(\boldsymbol{x}_c)} d\boldsymbol{z} d\boldsymbol{g} \tag{13}$$

We then define the importance distributions as the posterior over $\boldsymbol{g}$ and the conditional prior over $\boldsymbol{z}$. The method must learn $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ with $\boldsymbol{z}_t$ being solely conditioned on the *other* modality c. This formulation allows a fair comparison to the non-hierarchical baseline, which must do the same. In contrast, an importance distribution $q(\boldsymbol{z}_t|\boldsymbol{x}_t, \boldsymbol{g})$ would make the conditioning of $p(\boldsymbol{x}_t|\boldsymbol{z}_t)$ multimodal.

$$\log p(\boldsymbol{x}_t|\boldsymbol{x}_c) = \log \mathbb{E}_{q(\boldsymbol{g},\boldsymbol{z}_c|\boldsymbol{x}_c), p(\boldsymbol{z}_t|\boldsymbol{g})} \left[\underbrace{p(\boldsymbol{x}_t|\boldsymbol{z}_t, \boldsymbol{g})}_{1} \frac{p(\boldsymbol{x}_c, \boldsymbol{g}, \boldsymbol{z}_c)}{q(\boldsymbol{g}, \boldsymbol{z}_c|\boldsymbol{x}_c)} \right] - \log \mathbb{E}_{q(\boldsymbol{g},\boldsymbol{z}_c|\boldsymbol{x}_c)} \left[\frac{p(\boldsymbol{x}_c, \boldsymbol{g}, \boldsymbol{z}_c)}{q(\boldsymbol{g}, \boldsymbol{z}_c|\boldsymbol{x}_c)} \right] \tag{14}$$

In Eq. 14, we can amortize computation for the unimodal likelihood $p(\boldsymbol{x}_c)$ which was defined in App. B.1. Term 1 is the only one that requires additional forward passes through the network. Note that the remaining terms do not depend on the adjusted importance distribution $p(\boldsymbol{z}_t|\boldsymbol{g})$.

C IMPLEMENTATION

C.1 GENERAL SPECIFICATIONS

Hierarchical models The following formalizes latent distributions assuming the unimodal hierarchical VAE inspired by Sønderby et al. (2016) from § 2. We start with the conditional isotropic Gaussian prior

$$p_{\phi,\theta}(\boldsymbol{z}_i|\boldsymbol{z}_{i+1}) = \mathcal{N}(\boldsymbol{z}_i|\mu_\theta(d_{\phi,\theta,i}(\boldsymbol{z}_{i+1})), \sigma_\theta(d_{\phi,\theta,i}(\boldsymbol{z}_{i+1}))), \tag{15}$$

where $i \in \{1, ..., L\}$ refers to the hierarchical level. The MLP $d_{\phi,\theta,i}$ represents the top-down network, which is partially shared across the inference and generative networks. Its output is passed to the stochastic layers μ_θ and σ_θ that parameterize the prior distribution.

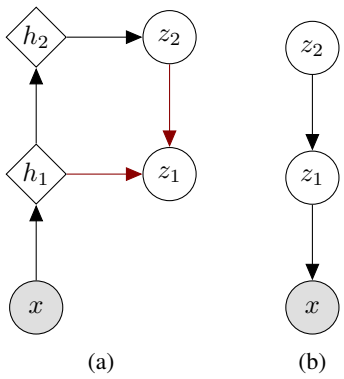


Figure 14: **Unimodal hierarchical VAEs** (Sønderby et al., 2016)
(a) Inference network. The red edges visualize the merging procedure of bottom-up and top-down information. (b) Generative network.

We then focus on the more complicated case of inferring the isotropic Gaussian posterior $q_{\phi,\theta}(\boldsymbol{z}_i|\boldsymbol{z}_{i+1}, \boldsymbol{x})$ where $i < L$ for L latent variables (Fig. 14). We first infer the hidden states for $\boldsymbol{z}_{i+1}$, $\boldsymbol{x}$ and for hierarchical level $i \in \{1, ..., L\}$, where $d_{\theta,\phi}$ is the top-down MLP and d_ϕ the bottom-up MLP:

$$\begin{aligned} \boldsymbol{h}_{i,t} &= d_{\phi,\theta,i}(\boldsymbol{z}_{i+1}), \\ \boldsymbol{h}_{i,b} &= d_{\phi,i}(\boldsymbol{x}). \end{aligned} \tag{16}$$

We concatenate both hidden states along the first dimension and pass the result through another neural network m_ϕ to create the joint representation. We denote the concatenation operation as $\langle \cdot, \cdot \rangle$:

$$\boldsymbol{h}_{i,j} = m_\phi(\langle \boldsymbol{h}_{i,t}, \boldsymbol{h}_{i,b} \rangle). \tag{17}$$

This joint representation is input to the stochastic layers μ_ϕ and σ_ϕ that parameterize the posterior distribution:

$$q_{\phi,\theta}(\boldsymbol{z}_i|\boldsymbol{z}_{i+1}, \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}_i|\mu_\phi(\boldsymbol{h}_{i,j}), \sigma_\phi(\boldsymbol{h}_{i,j})). \tag{18}$$

The remaining priors and posteriors are parameterized similarly. *Appendix A* from Maaløe et al. (2019) provides further background on unimodal hierarchical VAEs that inspire this work.

MVAE and MMVAE We follow Shi et al. (2019), *train* the MMVAE by maximizing a loose bound and *evaluate* using a tight estimate as described in App. B.1. The following describes exemplary differences in implementation compared to the original formulations:

- We employ a slightly different network architecture for the MVAE and the MMVAE (Wu and Goodman, 2018) relative to their original formulation, respectively. For example, we do not include dropout regularization across all implementations.
- For the MMVAE, we neither employ importance weighted autoencoders (Burda et al., 2015) nor use a doubly reparameterized gradient estimator (Tucker et al., 2018). Note that these techniques increase computational requirements. Neither the MVAE nor the HMVAE relies on such measures.
- We do not use Laplace latent distributions in the MMVAE, but Gaussian latent distributions as in all implemented methods.

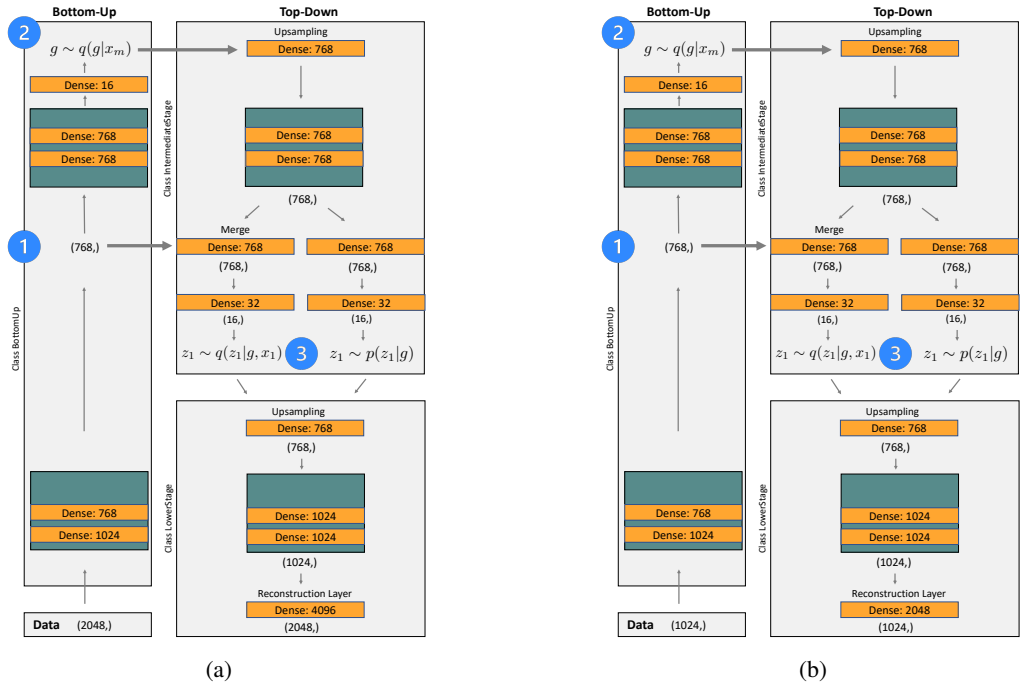


Figure 15: **CUB/Oxford Flower: Hierarchical VAE architectures for the HMVAE.** (a) Image feature vector VAE for the CUB and the Oxford Flower datasets. The decoder parameterizes a normal distribution. We do not display the *LeakyRelu* activation functions. (b) Caption feature vector VAE for the CUB and the Oxford Flower dataset (used in conjunction with an image feature vector VAE). The decoder parameterizes a normal distribution. We do not display the *LeakyRelu* activation functions.

C.2 CUB DATASET

Data The CUB dataset includes an image and a caption modality. For the images, we use the features vectors $\boldsymbol{x}_1 \in \mathbb{R}^{2048}$ provided by Xian et al. (2018a), who employed a ResNet-101 (He et al., 2016) trained on ImageNet (Deng et al., 2009). As described in § 4, we use caption features $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$. We divide the dataset into three splits that do not share classes. The training set consists of 111 classes. The validation split consists of 39 classes and the test split contains 50 classes. We train the model on the training set and tune hyperparameters on the validation set. We then evaluate the trained model on the test set, i.e., we do not use the validation set for training as in some previous works. We use the same test split as Reed et al. (2016) who provided the CNN-RNN feature extractor for the captions, which is essential to avoid test data leakage. We standardize the features for both modalities, i.e., we subtract the mean $\boldsymbol{\mu}$ and divide by the standard deviation $\boldsymbol{\sigma}$. We compute both parameters on the training set. We assume that such preprocessing can be helpful, because the models generate a Gaussian likelihood $p(\boldsymbol{x}_i|\cdot)$.

Model-agnostic specifications The generative models maximize the likelihood of the data under learned Gaussian distributions. We train the methods for 100 epochs with a learning rate of 1e-4. We gradually increase the KL-regularization weighting factor from zero to one for a warm-up period of 25 (Sønderby et al., 2016).

HMVAE The hierarchical VAEs are visualized in Fig. 15a and Fig. 15b. The model uses a batch size of 256. For each datapoint in a batch, we use ten samples from the respective shared posterior (i.e., $\boldsymbol{g}^{1:10} \sim q(\boldsymbol{g}|\boldsymbol{x}_{1:M})$) during training. The shared latent space g contains 16 dimensions.

MVAE and MMVAE The models are visualized in Table 5a and Table 5b. We share the architecture across baselines. Those models are similar in their architecture to the hierarchical method, but lack the hierarchical components. We tested a version with additional layers to compensate for this reduced capacity. However, models with more capacity (i.e., more weights) did not improve the

performance of the flat models. The models use a batch size of 256. For each datapoint in a batch, we use ten samples from the respective shared posterior (i.e., $\boldsymbol{g}^{1:10} \sim q(\boldsymbol{g}|\boldsymbol{x}_{1:M})$) during training. The shared latent spaces g contain eight dimensions.

Table 5: **CUB/Oxford Flower: MVAE and MMVAE architecture.** (a) Image feature vector VAE for the CUB and the Oxford Flower datasets. The decoder parameterizes a normal distribution. (b) Caption feature vector VAE for the CUB and the Oxford Flower datasets (used in conjunction with an image feature vector VAE). The decoder parameterizes a normal distribution.

(a)

Encoder	**Decoder**
Input $\in \mathbb{R}^{2048}$	Input $\in \mathbb{R}^{16}$
Dense(2048, 1024)	Dense(16, 768)
LeakyRelu()	LeakyRelu()
Dense(1024, 768)	Dense(768, 768)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(768, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(1024, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 32)	Dense(1024, 4096)

(b)

Encoder	**Decoder**
Input $\in \mathbb{R}^{1024}$	Input $\in \mathbb{R}^{16}$
Dense(1024, 1024)	Dense(16, 768)
LeakyRelu()	LeakyRelu()
Dense(1024, 768)	Dense(768, 768)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(768, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(1024, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 32)	Dense(1024, 2048)

C.3 Oxford Flower

Data The Oxford Flower dataset includes an image and a caption modality. For the images, we either (a) use the images directly, where $\boldsymbol{x}_1 \in \mathbb{R}^{3\times64\times64}$, or (b) use feature vectors, where $\boldsymbol{x}_1 \in \mathbb{R}^{2048}$ For (a), we preprocess the images by first making them quadratic (using *CenterCrop* in PyTorch) and then resizing them to a resolution of $(3, 64, 64)$. For (b), we standardize the features for both modalities, i.e., we subtract the mean $\boldsymbol{\mu}$ and divide by the standard deviation $\boldsymbol{\sigma}$. We compute both parameters on the training set. We assume that such preprocessing can be helpful, because the respective models generate a Gaussian likelihood $p(\boldsymbol{x}_i|\cdot)$. As described in § 4, we use feature representations for the caption modality $\boldsymbol{x}_2 \in \mathbb{R}^{1024}$. We divide the dataset into three splits that do not share classes. The training set consists of 62 classes. Both validation and test split contain 20 classes. We train the model on the training set and tune hyperparameters on the validation set. We then evaluate the trained model on the test set, i.e., we do not use the validation set for training as in some previous works. We use the same test split as Reed et al. (2016) who provided the CNN-RNN feature extractor for the captions, which is essential to avoid test data leakage.

Oxford Flower: Image features vectors and caption features vectors

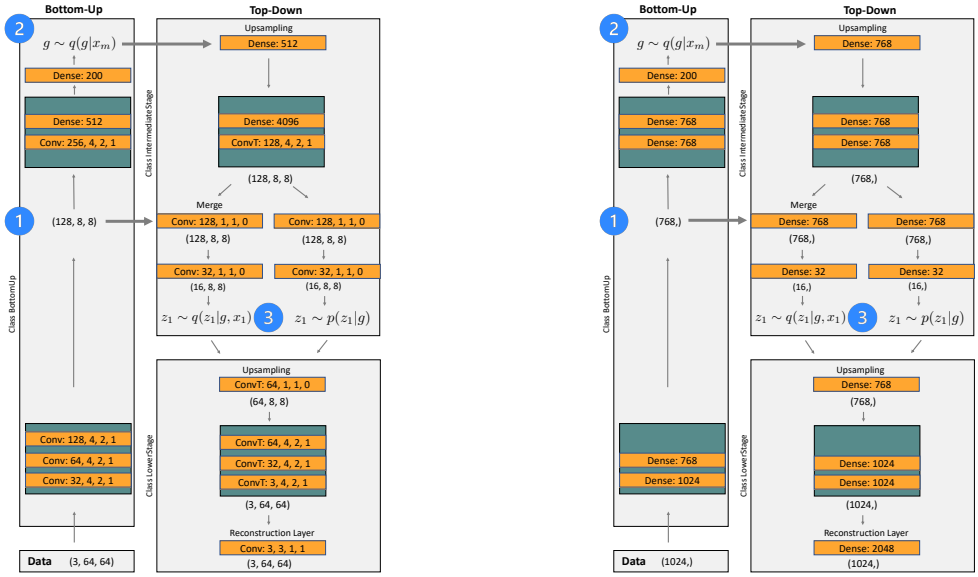
Model-agnostic specifications The VAEs maximize the likelihood of each modality under a learned Gaussian distribution, respectively. All models are trained for 100 epochs. We gradually increase the KL-regularization weighting factor from zero to one for a warm-up period of 25 (Sønderby et al., 2016). We train the models with a learning rate of 1e-4 and a batch size of 256. For each datapoint in a batch, we use ten samples from the respective shared posterior (i.e., $\boldsymbol{g}^{1:10} \sim q(\boldsymbol{g}|\boldsymbol{x}_{1:M})$) during training.

HMVAE Figure 15a displays the image VAE, Fig. 15b shows the caption feature vector VAE.

MVAE and MMVAE Table 5a displays the image VAE, Table 5b shows the caption feature vector VAE.

Oxford Flower: Images (not image feature vectors) and caption features vectors

Model-agnostic specifications The caption feature vector VAEs maximize the likelihood of the data under learned Gaussian distributions. The image VAEs minimize the binary cross-entropy between sigmoid-activated learned tensors and the ground truth. We train the models with a learning rate of



(a) (b)

Figure 16: **Oxford Flower: Hierarchical VAE architectures for the HMVAE.** (a) Image VAE for the Oxford Flower dataset. The decoder generates scalars, which are used to compute the binary cross-entropy with the ground truth. We do not display the *Swish* activation functions. (b) Caption feature vector VAE for the Oxford Flower dataset (used in conjunction with an image VAE). The decoder parameterizes a normal distribution. We do not display the *LeakyRelu* activation functions.

1e-4 and a batch size of 256. We gradually increase the KL-regularization weighting factor from zero to one for a warm-up period of 20 (Sønderby et al., 2016).

HMVAE Figure 16a portrays the hierarchical VAE for the image modality of our proposed HMVAE. The orange blocks denote either dense or (transposed) convolution layers. Dense blocks contain the output dimension, convolutional blocks the output channel dimension as well as (symmetrical) kernel size, stride, and padding. The large gray boxes *BottomUp*, *IntermediateStage*, and *LowerStage* represent classes in the code base.

Bottom-up and top-down passes partition the network. The former mostly consists of convolutional layers. However, the top-level stochastic variable (compare 2 in the figure) is a vector (and not spatial) to comply with the other modalities. Dense layers first process the flattened three-dimensional output from the convolutional layers and then generate a tensor containing the concatenated parameter vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ for the shared isotropic Gaussian posterior $q(\boldsymbol{g}|\boldsymbol{x}_1)$. The respective shared variable $\boldsymbol{g}$ has 100 dimensions. Samples from the product over modality experts $q(\boldsymbol{g}|\boldsymbol{x}_{1:M})$ are the input to the top-down network. Some top-down layers share weights across inference and generative network (Sønderby et al., 2016). Other layers are specific to the inference or the generative network.

The inference layers merge bottom-up information 1 and top-down information by concatenating both hidden states along the channel dimension and passing the result through a convolutional layer. The generative layers hallucinate variations coherent with the missing bottom-up information. The resulting lower-level latent isotropic Gaussian distributions 3 are spatial, i.e., the method generates a tensor containing the concatenated parameter vectors $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$. The respective variable $\boldsymbol{z}_1$ consists of 16 channel dimensions and eight spatial dimensions. This specification makes them predestined to model visual variations – as opposed to the flat top-level variable. Samples from one of these latent distributions are passed to the final top-down layers. A reconstruction layer concludes the pass by producing a sigmoid-activated tensor to compute the binary cross-entropy with the ground truth. Analogously, Fig. 16b presents the hierarchical VAE for the other modality. The model is trained for 750 epochs.

MVAE and MMVAE The MVAE and MMVAE share their basic architecture. Table 6 displays the image VAE, Table 7 shows the caption feature VAE. Both models are trained for 750 epochs. For the MMVAE, we find that training for 500 epochs improves zero-shot classification performance and decreases image sample quality (as measured by the Fréchet Inception Distances, bot measures were calculated on the validation set). Because we care most about visual quality, we use the version with 750 epochs.

Table 6: **Oxford Flower: MVAE and MMVAE architectures.** Image VAE for the Oxford Flower dataset. The decoder generates scalars, which are used to compute the binary cross-entropy with the ground truth. We report the output-channel dimensions.

Encoder	**Decoder**
Input $\in \mathbb{R}^{3\times64\times64}$	Input $\in \mathbb{R}^{100}$
Conv(c=32, k=4, s=2, p=1)	Dense(100, 512)
Swish()	Swish()
Conv(c=64, k=4, s=2, p=1)	Dense(512, 4096)
Swish()	Swish()
Conv(c=128, k=4, s=2, p=1)	ConvT(c=128, k=4, s=2, p=1, op=0)
Swish()	Swish()
Conv(c=128, k=1, s=1, p=0)	ConvT(c=64, k=1, s=1, p=0, op=0)
Swish()	Swish()
Conv(c=256, k=4, s=2, p=1)	ConvT(c=64, k=1, s=1, p=0, op=0)
Swish()	Swish()
Dense(4096, 512)	ConvT(c=64, k=4, s=2, p=1, op=0)
Swish()	Swish()
Dense(512, 200)	ConvT(c=32, k=4, s=2, p=0, op=0)
	Swish()
	ConvT(c=3, k=4, s=2, p=0, op=0)
	Swish()
	Conv(c=3, k=4, s=2, p=1)
	Sigmoid()

Table 7: **Oxford Flower: MVAE and MMVAE architectures.** Caption feature vector VAE for the Oxford Flower dataset (used in conjunction with an image VAE). The decoder parameterizes a normal distribution.

Encoder	**Decoder**
Input $\in \mathbb{R}^{1024}$	Input $\in \mathbb{R}^{16}$
Dense(1024, 1024)	Dense(16, 768)
LeakyRelu()	LeakyRelu()
Dense(1024, 768)	Dense(768, 768)
LeakyRelu()	LeakyRelu()
Dense(768, 1024)	Dense(768, 1024)
LeakyRelu()	LeakyRelu()
Dense(1024, 768)	Dense(1024, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(1024, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 768)	Dense(1024, 1024)
LeakyRelu()	LeakyRelu()
Dense(768, 32)	Dense(1024, 2048)

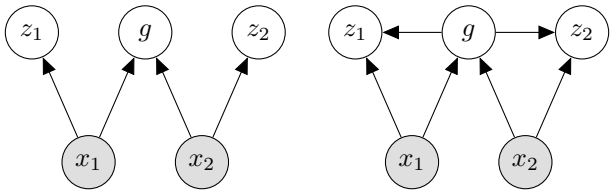


Figure 17: MDVAE: inference network choices

D MDVAE: INFERENCE NETWORKS

There are several inference network choices for multimodal disentanglement VAEs (MDVAEs): Fig. 17a depicts the natural inference network. Figure 17b modifies this network to include dependencies between $\boldsymbol{g}$ and $\boldsymbol{z}_{1:M}$. This modified architecture recovers our proposed hierarchical inference network for the special case of two hierarchical layers (Fig. 3d). Note that the edges between $\boldsymbol{g}$ and $\boldsymbol{z}_{1:2}$ are still absent in the generative model (Fig. 3b): the model must learn to represent *independent* variation in $\boldsymbol{z}_1$. For example, the model must generate $p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i, \boldsymbol{g})$ from $\boldsymbol{z}_i \sim p_\theta(\boldsymbol{z}_1)$ (not necessarily $\boldsymbol{z}_i \sim q_\phi(\boldsymbol{z}_1|\boldsymbol{x}_i, \boldsymbol{g})$!) and $\boldsymbol{g} \sim q_\phi(\boldsymbol{g}|\boldsymbol{x}_{j\neq i})$.

E SOCIETAL IMPACT

Generative models can have pernicious effects on society via creating and propagating synthetic data that mimics reality (e.g., DeepFakes). Extending these models to multiple modalities can strengthen their performance and hence amplify these challenges. Therefore, annotating and marking data as synthetically generated is vital to ensure that people can spot and identify synthetic data outside of contexts where their synthetic nature is clear. Furthermore, the training data for deep generative models rarely represents the rich diversity of people, images, and objects in the world. Biases towards overrepresented groups will inevitably creep into generative models, and if the model is used for solving classification tasks on underrepresented groups, it is unlikely to prove useful. Extending the training data to multiple modalities can mitigate these effects if the additional modalities are less biased. In general, building generative models that can better cope with heterogeneity, e.g., by improving models or collecting more realistic data, is a good step towards alleviating harmful biases.

F COMPARISON WITH THE MDVAE AND THE MHVAE: FURTHER DISCUSSION

Implementation: MHVAE (Vasco et al., 2020)

We follow Vasco et al. (2020) and maximize the sum over the multi- and unimodal ELBOs:

$$
\begin{aligned}
\mathcal{L} &= \mathcal{L}_{\text{joint}} + \mathcal{L}_{\text{images}} + \mathcal{L}_{\text{captions}} \\
\mathcal{L}_{\text{joint}} &\coloneqq \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log \frac{p_\theta(\boldsymbol{z}_i|\boldsymbol{g})}{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)}\right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})} \left[\log \frac{p_\theta(\boldsymbol{g})}{q_\phi(\boldsymbol{g}|\boldsymbol{x}_{1:M})}\right] \\
\mathcal{L}_{\text{images}} &\coloneqq \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log \frac{p_\theta(\boldsymbol{z}_i|\boldsymbol{g})}{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)}\right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{g}|\boldsymbol{x}_1)} \left[\log \frac{p_\theta(\boldsymbol{g})}{q_\phi(\boldsymbol{g}|\boldsymbol{x}_1)}\right] \\
\mathcal{L}_{\text{captions}} &\coloneqq \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log p_\theta(\boldsymbol{x}_i|\boldsymbol{z}_i)\right] - \sum_{i=1}^{M} \mathbb{E}_{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)} \left[\log \frac{p_\theta(\boldsymbol{z}_i|\boldsymbol{g})}{q_\phi(\boldsymbol{z}_i|\boldsymbol{x}_i)}\right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{g}|\boldsymbol{x}_2)} \left[\log \frac{p_\theta(\boldsymbol{g})}{q_\phi(\boldsymbol{g}|\boldsymbol{x}_2)}\right]
\end{aligned}
\tag{19}
$$