Multi-Facet Clustering Variational Autoencoders

Anonymous Author(s)
Affiliation
Address
email

Abstract

Work in deep clustering focuses on finding a *single* partition of data. However, high-dimensional data, such as images, typically feature *multiple* interesting characteristics one could cluster over. For example, images of objects against a background could be clustered over the shape of the object and separately by the colour of the background. In this paper, we introduce *Multi-Facet Clustering Variational Autoencoders (MFCVAE)*, a novel class of variational autoencoders with a hierarchy of latent variables, each with a Mixture-of-Gaussians prior, that learns multiple clusterings simultaneously, and is trained fully unsupervised and end-to-end. MFCVAE uses a progressively-trained ladder architecture which leads to highly stable performance. We provide novel theoretical results for optimising the ELBO analytically with respect to the categorical variational posterior distribution, and corrects earlier influential theoretical work. On image benchmarks, we demonstrate that our approach separates out and clusters over different aspects of the data in a disentangled manner. We also show other advantages of our model: the compositionality of its latent space and that it provides controlled generation of samples.

1 Introduction

Clustering is the task of finding structure by partitioning samples in a finite, unlabeled dataset according to statistical or geometric notions of similarity [1, 2, 3]. For example, we might group items along axes of empirical variation in the data, or maximise internal homogeneity and external separation of items within and between clusters with respect to a specified distance metric. The choice of similarity measure and how one consequently validates clustering quality is fundamentally a subjective one: it depends on what is useful for a particular task [2, 4]. In this work, we are interested in uncovering abstract, latent characteristics/facets/aspects/levels of the data to understand and characterise the data-generative process. We further assume a fully exploratory, unsupervised setting without prior knowledge on the data, which could be exploited while fitting the clustering algorithm, and in particular without given ground-truth partitions at training time.

When being faced with high-dimensional data such as images, speech or electronic health records, items typically have more than one abstract characteristic. Consider the example of the MNIST dataset [5]: MNIST images possess at least two such characteristics: The digit class, which might impose the largest amount of statistical variation, and the style of the digit (e.g. stroke width). This naturally raises a question: By which characteristic is a clustering algorithm supposed to partition the data? In MNIST, both digit class and (the sub-categories of) style would be perfectly reasonable candidates to answer this question. In our exploratory setting described above, there is not one "correct" partition of the data.

Deep learning based clustering algorithms, so-called *deep clustering*, were particularly successful in recent years in dealing with high-dimensional data by compressing the inputs into a lower-dimensional latent space in which clustering is computationally tractable [6, 7]. However, almost all of these deep clustering algorithms find only a *single* partition of the data, typically the one corresponding to the

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

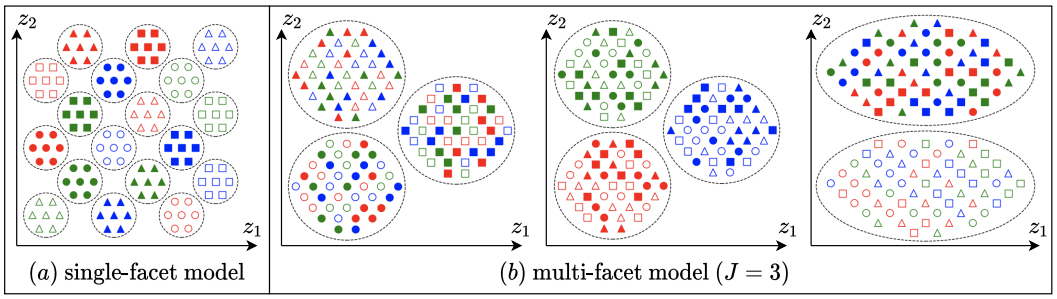

Figure 1: Latent space of a (a) single-facet model and a (b) multi-facet model ($J = 3$) with two dimensions (z_1, z_2) per facet. Both models perfectly separate the abstract characteristics of the data. However, the multi-facet model disentangles them into three sensible partitions (one per facet) and its required clusters scale linearly as opposed to exponentially w.r.t. the number of aspects in the data.

given class label in a supervised dataset [8, 9, 10, 11, 12, 13, 14, 15]. When evaluating their model, said approaches validate clustering performance by treating the one supervision label (e.g. digit class in the case of MNIST) as the de-facto "ground truth clustering". We argue that restricting our view to a single facet C_1 rather than all or at least multiple facets $(C_1, C_2, \ldots, C_J)$ is an arbitrary, incomplete choice of formulating the problem of clustering a high-dimensional dataset.

To this end, we propose *Multi-Facet Clustering Variational Autoencoders (MFCVAE)*, a principled, probabilistic model which finds multiple characteristics of the data simultaneously through its multiple Mixtures-of-Gaussians (MoG) prior structure. Our contributions are as follows: (a) *Multi-Facet Clustering Variational Autoencoders (MFCVAE)*, a novel class of probabilistic deep learning models for unsupervised, multi-facet clustering in high-dimensional data that can be optimised end-to-end. (b) Novel theoretical results for the optimisation of the corresponding ELBO, correcting and extending an influential, related paper for the single-facet case. (c) Demonstrating MFCVAE's stable empirical performance in terms of multi-facet clustering of various levels of abstraction, compositionality of facets, generative, unsupervised classification, and diversity of generation.

2 Multi-facet clustering

High-dimensional data are inherently structured according to a number of abstract characteristics, and in an exploratory setting, it is clear that arbitrarily clustering by one of them is insufficient. However, the question remains whether these multiple facets should also be explicitly *represented* by the model. In particular, one might argue that a single partition could be used to represent all cross-combinations[1] of facets $\mathcal{C} = C_1 \times C_2 \times \cdots \times C_J$ where $C_j = \{1, 2, \ldots, K_j\}$, as in Fig. 1 (a). In this work, we explain that explicitly representing and clustering by multiple facets, as we do in MFCVAE and illustrated in Fig. 1 (b), has the following four properties that are especially desirable in an unsupervised learning setting:

(a) Discovering a multi-facet structure. We adopt a probabilistically principled, unsupervised approach, specifying an independent, multiple Mixtures of Gaussians (MoG) prior on the latent space. This induces a disentangled representation across facets, meaning that in addition to examples assigned to certain clusters being homogeneous, the facets (and their corresponding clusters) represent different abstract characteristics of the data (such as digit class or digit style). Because of this multi-facet structure, the total number of clusters required to represent a given multi-partition structure of the data scales *linearly* w.r.t. the number of data characteristics. In comparison, the number of clusters required in a single-facet model scales *exponentially* (see Fig. 1, and Appendix A for details).

(b) Compositionality of facets. A multi-facet model has a compositional advantage: different levels of abstraction of the data are represented in separate latent variables. As we will show, this allows qualitatively diverse characteristics to be meaningfully combined.

[1]Note that in practice, not all cross-combinations of facets might be present. For example, in a dataset like MNIST, one might not observe 'right-tilted zeros', even though we observe 'right-tilted' digits and 'zeros'.

(c) Generative, unsupervised classification. Our method joins a myriad of single-facet clustering models in being able to accurately identify known class structures given by the label in standard supervised image datasets. However, in contrast to previous work, we are also able find interesting characteristics in other facets with homogeneous clusters. We stress that while we compare generative classification performance against other models to demonstrate statistical competitiveness, this task is *not* the main motivation for our fully unsupervised model.

(d) Diversity of generated samples. In a generative sense, the structure of the latent space allows us to compose new, synthetic examples by a set of J pairs of (continuous, discrete) latent variables. We can in particular intervene on each facet separately. This yields a rich set of options and fine-grained control for interventions and the diversity of generated examples.

We illustrate these four properties in our experiments in Section 4.

3 Multi-Facet Clustering Variational Autoencoders

Our model comprises J latent facets, each learning its own unique clustering of samples via a Mixture-of-Gaussians (MoG) distribution:

$$c_j \sim \mathrm{Cat}(\boldsymbol{\pi}_j), \quad \mathbf{z}_j \mid c_j \sim \mathcal{N}(\boldsymbol{\mu}_{c_j}, \boldsymbol{\Sigma}_{c_j}) \tag{1}$$

where $\boldsymbol{\pi}_j$ is the jth facet's K_j-dimensional vector of mixing weights, and $(\boldsymbol{\mu}_{c_j}, \boldsymbol{\Sigma}_{c_j})$ are the mean and covariance of the c_jth mixture component in facet j ($\boldsymbol{\Sigma}_{c_j}$ can be either diagonal or full).

The multi-facet generative model (Fig. 2 [Right]) is thus structured as

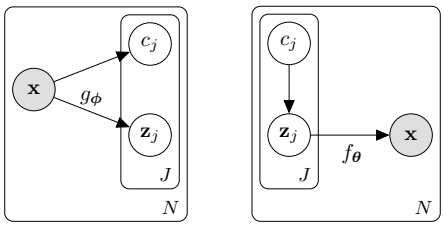

Figure 2: Graphical model of MFCVAE. [Left] Amortised variational posterior, $q_\phi(\vec{\mathbf{z}}, \mathbf{c}|\mathbf{x})$. [Right] Generative model, $p_\theta(\mathbf{x}, \vec{\mathbf{z}}, \mathbf{c})$.

$$p_\theta(\mathbf{x}, \vec{\mathbf{z}}, \mathbf{c}) = p_\theta(\mathbf{x}|\vec{\mathbf{z}})p_\theta(\vec{\mathbf{z}}|\mathbf{c})p_\theta(\mathbf{c}) = p_\theta(\mathbf{x}|\vec{\mathbf{z}}) \prod_{j=1}^{J} p_\theta(\mathbf{z}_j|c_j)p_\theta(c_j), \tag{2}$$

where $\mathbf{c} = \{c_1, c_2, ..., c_J\}$, $\vec{\mathbf{z}} = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_J\}$, and $p_\theta(\mathbf{x}|\vec{\mathbf{z}})$ is a user-defined likelihood model, for example a product of Bernoulli or Gaussian distributions, which is parameterised with a deep neural network $f(\vec{\mathbf{z}}; \theta)$. Importantly, this structure in Eq. (2) encodes prior independence across facets, i.e. $p_\theta(\vec{\mathbf{z}}, \mathbf{c}) = \prod_j p_\theta(\mathbf{z}_j, c_j)$, thereby encouraging facets to learn clusterings that span distinct subspaces of $\vec{\mathbf{z}}$. The overall marginal prior $p_\theta(\vec{\mathbf{z}})$ can be interpreted as a product of independent MoGs.

3.1 VaDE tricks

To train this model, we wish to optimise the evidence lower bound (ELBO) using an amortised variational posterior $q_\phi(\vec{\mathbf{z}}, \mathbf{c}|\mathbf{x})$ (Fig. 2 [Left]), parameterised by a neural network $g(\mathbf{x}; \phi)$, within which we will perform Monte Carlo (MC) estimation where necessary to approximate expectations

$$\log p(\mathcal{D}) \geq \mathcal{L}(\mathcal{D}; \theta, \phi) = \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\mathbb{E}_{q_\phi(\vec{\mathbf{z}},\mathbf{c}|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}, \vec{\mathbf{z}}, \mathbf{c})}{q_\phi(\vec{\mathbf{z}}, \mathbf{c}|\mathbf{x})}]\right]. \tag{3}$$

What should we choose for $q_\phi(\vec{\mathbf{z}}, \mathbf{c}|\mathbf{x})$? Training deep generative models with discrete latent variables can be challenging, as reparameterisation tricks so far developed, such as the Gumbel-Softmax trick [16, 17], necessarily introduce bias. Our setting, where we have multiple discrete latent variables, is potentially even more challenging. In addition, Gumbel-Softmax approaches become unstable above ≈ 200 discrete classes – this is a problem as in our experiments, we sometimes wish to have a large number of cluster components in certain facets. Also, we wish to make optimisation as light-weight as possible. We would prefer to avoid introducing additional neural networks whenever possible – as this simplifies both training and neural specification.

Thus, we sidestep these problems, bias from relaxations of discrete variables *and* the downsides of additional amortised-posterior neural networks for the discrete latent variables, by developing the hierarchical version of the *VaDE trick*. This trick was first developed for clustering VAEs with a *single* Gaussian mixture in the generative model [10]. Informally, the idea (for a single-facet model) is to define a Bayes-optimal posterior for the discrete latent variable using the responsibilities of the

constituent components of the mixture model; these responsibilities are calculated using samples taken from the amortised posterior for the continuous latent variable.

Estimating the ELBO for models of this form does not require us to take MC samples from discrete distributions – the data likelihood is conditioned only on the continuous latent variable $\vec{\mathbf{z}}$, which we sample using the reparameterization trick [18], and the posterior for $\vec{\mathbf{z}}$ is conditioned only on $\mathbf{x}$. Thus, when calculating the ELBO, we can cheaply marginalise out discrete latent variables where needed. In other words, we do not have to perform multiple forward passes through the decoder as neither it nor the $\vec{\mathbf{z}}$ samples we feed it depend on c.

As it is integral to our method, we now briefly recapitulate the original VaDE trick for VAEs with a single latent mixture (correcting a misapprehension in the original form of this idea) and will then cover our hierarchical extension[2].

Single-Facet VaDE Trick: Consider a single facet model, so the generative model is $p_\theta(\mathbf{x}, \mathbf{z}, c) = p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}|c)p_\theta(c)$. Introduce a posterior $q_\phi(\mathbf{z}, c|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x})q_\phi(c|\mathbf{x})$ where $q_\phi(\mathbf{z}|\mathbf{x})$ is a multivariate Gaussian with diagonal covariance. The ELBO for this model for one datapoint is

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z},c|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z}|c)p_\theta(c)}{q_\phi(\mathbf{z}|\mathbf{x})q_\phi(c|\mathbf{x})}] = \mathbb{E}_{q_\phi(\mathbf{z},c|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\theta(\mathbf{z})p_\theta(c|\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})q_\phi(c|\mathbf{x})}], \quad (4)$$

where we have chosen to rewrite the generative model factorisation, $p_\theta(\mathbf{z}) = \sum_c p_\theta(\mathbf{z}|c)p_\theta(c)$ is the marginal mixture of Gaussians, and $p_\theta(c|\mathbf{z}) = p_\theta(\mathbf{z}|c)p_\theta(c)/p_\theta(\mathbf{z})$ is the Bayesian posterior for c.

Expanding out the ELBO, we get

$$\mathcal{L}(\mathbf{x}; \theta, \phi) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - \mathrm{KL}\left[q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z})\right] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right]. \quad (5)$$

We can *define* $q_\phi(c|\mathbf{x})$ such that $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right]$ is *minimal*, by construction, which is the case if we choose $q_\phi(c|\mathbf{x}) \propto \exp\left(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(c|\mathbf{z})\right)$ as we will show in Theorem 1. This means that we can simply use samples from the posterior for $\mathbf{z}$ to define the posterior for c, using Bayes' rule within the latent mixture model.

Remark: We note, however, that in the original description of this idea in [10], it was claimed that $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right]$ could, in general, be set to *zero*, which is not the case. Rather, this KL can be minimised, in general, to a *non-zero* value. We discuss this misapprehension in more detail and why the empirical results in [10] are still valid in Appendix B.1.1.

Theorem 1. *(Single-Facet VaDE Trick) For any probability distribution $q_\phi(\mathbf{z}|\mathbf{x})$, the distribution $q_\phi(c|\mathbf{x})$ that minimises $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right]$ in (5) is*

$$\operatorname*{argmin}_{q_\phi(c|\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right] = \boldsymbol{\pi}(c|q_\phi(\mathbf{z}|\mathbf{x})) \quad (6)$$

with the minimum value attained being

$$\min_{q_\phi(c|\mathbf{x})} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \mathrm{KL}\left[q_\phi(c|\mathbf{x})||p_\theta(c|\mathbf{z})\right] = -\log Z(q_\phi(\mathbf{z}|\mathbf{x})) \quad (7)$$

$$\textit{where} \qquad \boldsymbol{\pi}(c|q_\phi(\mathbf{z}|\mathbf{x})) := \frac{\exp\left(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p(c|\mathbf{z})\right)}{Z(q_\phi(\mathbf{z}|\mathbf{x}))} \ \textit{for } c = 1, \ldots, K \quad (8)$$

$$Z(q_\phi(\mathbf{z}|\mathbf{x})) := \sum_{c=1}^{K} \exp\left(\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log p(c|\mathbf{z})\right) . \quad (9)$$

Proof: See Appendix B.1. □

Multi-facet VaDE Trick: In this work, we consider the case of having J facets, each with its own pair of variables $(\mathbf{z}_j, c_j)$. Perhaps surprisingly, we do *not* have to make a mean-field assumption *between* the J facets for $\mathbf{c}$ once we have made one for $\vec{\mathbf{z}}$. In other words, once we have chosen that $q_\phi(\vec{\mathbf{z}}, \mathbf{c}|\mathbf{x}) = q_\phi(\mathbf{c}|\mathbf{x}) \prod_{j=1}^{J} q_\phi(\mathbf{z}_j|\mathbf{x})$, where $q_\phi(\mathbf{z}_j|\mathbf{x})$ is defined to be a multivariate Gaussian with diagonal covariance for each j, the optimal $q_\phi(\mathbf{c}|\mathbf{x})$ similarly factorises[3]. We formalise this:

[2] We note that the original VaDE paper, besides the misapprehension discussed in Section 3.1 and Appendix B.1.1, proposed a highly complex training algorithm with various pre-training heuristics which we significantly simplified while maintaining or increasing performance (details in Appendix D.5).

[3] We also provide the VaDE trick for the general form of the posterior for $\vec{\mathbf{z}}$, i.e. without assuming the factorisation $q_\phi(\vec{\mathbf{z}}|\mathbf{x}) = \prod_{j=1}^{J} q_\phi(\mathbf{z}_j|\mathbf{x})$, in Appendix B.3.

Theorem 2. *(Multi-Facet VaDE Trick for factorized $q_\phi(\vec{\mathbf{z}}|\mathbf{x})$, $p(\vec{\mathbf{z}}, \mathbf{c})$) For any factorized probability distribution $q_\phi(\vec{\mathbf{z}}|\mathbf{x}) = \prod_j q_\phi(\mathbf{z}_j|\mathbf{x})$, the distribution $q_\phi(\mathbf{c}|\mathbf{x})$ that minimises $\mathbb{E}_{q_\phi(\vec{\mathbf{z}}|\mathbf{x})} \mathrm{KL}\left[q_\phi(\mathbf{c}|\mathbf{x})||p_\theta(\mathbf{c}|\vec{\mathbf{z}})\right]$ under factorized prior $p(\vec{\mathbf{z}}, \mathbf{c}) = \prod_j p(\mathbf{z}_j, c_j)$ of (2) is*

$$\operatorname*{argmin}_{q_\phi(\mathbf{c}|\mathbf{x})} \mathbb{E}_{q_\phi(\vec{\mathbf{z}}|\mathbf{x})} \mathrm{KL}\left[q_\phi(\mathbf{c}|\mathbf{x})||p_\theta(\mathbf{c}|\vec{\mathbf{z}})\right] = \prod_j \boldsymbol{\pi}_j(c_j|q_\phi(\mathbf{z}_j|\mathbf{x})) \tag{10}$$

where the minimum value is attained at

$$\min_{q_\phi(\mathbf{c}|\mathbf{x})} \mathbb{E}_{q_\phi(\vec{\mathbf{z}}|\mathbf{x})} \mathrm{KL}\left[q_\phi(\mathbf{c}|\mathbf{x})||p_\theta(\mathbf{c}|\vec{\mathbf{z}})\right] = -\sum_j \log Z_j(q_\phi(\mathbf{z}_j|\mathbf{x})) \tag{11}$$

$$\textit{where} \qquad \boldsymbol{\pi}_j(c_j|q_\phi(\mathbf{z}_j|\mathbf{x})) := \frac{\exp(\mathbb{E}_{q_\phi(\mathbf{z}_j|\mathbf{x})} \log p_\theta(c_j|\mathbf{z}_j))}{Z_j(q_\phi(\mathbf{z}_j|\mathbf{x}))}, \textit{for } c_j = 1, \ldots, K_j \tag{12}$$

$$Z_j(q_\phi(\mathbf{z}_j|\mathbf{x})) := \sum_{c_j=1}^{K_j} \exp(\mathbb{E}_{q_\phi(\mathbf{z}_j|\mathbf{x})} \log p_\theta(c_j|\mathbf{z}_j)) \,. \tag{13}$$

Proof: See Appendix B.2. □

Note that we use Eq. (12) as the probability distribution of assigning input $\mathbf{x}$ to clusters of facet j.

Armed with these theoretical results, we can now write the ELBO for our model, with the optimal posterior for $\mathbf{c}$, in a form that trivially admits stochastic estimation and does not necessitate extra recognition networks for $\mathbf{c}$,

$$\begin{aligned}\mathcal{L}^{\mathrm{MFCVAE}}(\mathcal{D}; \theta, \phi) = &\mathbb{E}_{\mathbf{x}\sim\mathcal{D}} \Big[\mathbb{E}_{q_\phi(\vec{\mathbf{z}}|\mathbf{x})} \log p_\theta(\mathbf{x}|\vec{\mathbf{z}}) \\ &- \sum_{j=1}^{J} \left[\mathbb{E}_{q_\phi(c_j|\mathbf{x})} \mathrm{KL}(q_\phi(\mathbf{z}_j|x)||p_\theta(\mathbf{z}_j|c_j)) + \mathrm{KL}(q_\phi(c_j|\mathbf{x})||p(c_j))\right] \Big]\end{aligned} \tag{14}$$

where the optimal $q_\phi(c_j|\mathbf{x})$ is given by Eq. (12) for each j.

To obtain the posterior distributions for $\mathbf{c}$, we take MC samples from $q_\phi(\vec{\mathbf{z}}|\mathbf{x})$ and use these to construct the posterior as in Eq. (12). We found one MC sample ($L = 1$; for each facet and for each $\mathbf{x}$) to be sufficient. We derive the complete MC estimator which we use as the loss function of our model and ablations on two alternative forms in Appendix C.

3.2 Neural implementation and training algorithm

It is worth pausing here to consider what neural architecture best suits our desire for learning multiple disentangled facets, and then further how we can best train our model to robustly elicit from it well-separated facets. In the introduction, we discussed the different plausible ways to cluster high-dimensional data, such as in MNIST digits by stroke thickness and class identity. These different aspects intuitively correspond to different levels of abstraction about the image. It is thus natural that these levels would be best captured by different depths of the neural networks in each amortised posterior. These ideas have motivated the use of *ladder networks* in deep generative models that aim to learn different facets of the input data into different layers of latent variables. Here, we take inspiration from *Variational Ladder Autoencoders (VLAEs)* [19]: A VLAE architecture has a deterministic "backbone" in both the recognition and generative model. The different layers of latent variables branch out from these at different depths along. This inductive bias naturally leads to stratification and does so without having to bear the computational cost of training a completely separate encoder (say) for each layer. We refer to Appendix D.2 for details on how we implement our VLAE architecture.

Further, we found *progressive training* [20], previously shown to help VLAEs learn layer-by-layer disentangled representations, to be of great use in making each facet consistently represent the same aspects of data. The general idea of progressive training is to start with training a *single* facet (typically the one corresponding to the deepest recognition and generative neural networks) for a certain number of epochs, and progressively and smoothly loop in the other facets one after the other. We discuss the details of our progressive training schedule in Appendix D.3. We find that both the VLAE architecture and progressive training are jointly important to stabilise training and get robust qualitative and quantitative results as we show in Appendix E.1.

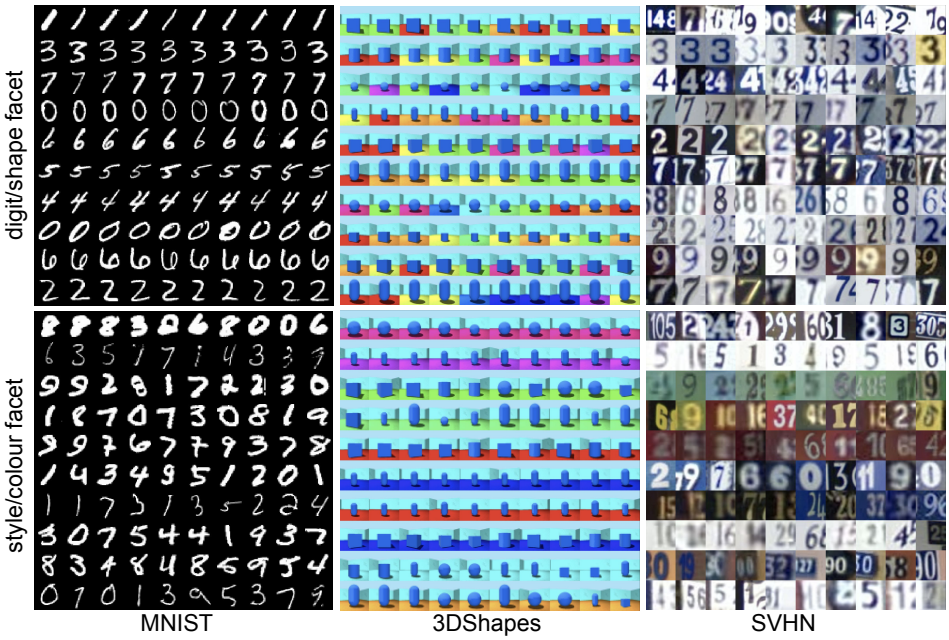

Figure 3: Input examples for clusters of MFCVAE with two-facets ($J = 2$) trained on MNIST, 3DShapes and SVHN. Clusters (rows) in each facet j are sorted in decreasing order by the average assignment probability of test inputs over each cluster. Inputs (columns) are sorted in decreasing order by their assignment probability $\max_{c_j} \boldsymbol{\pi}_j(c_j|q_\phi(\mathbf{z}_j|\mathbf{x}))$. We visualise the first 10 clusters and inputs from the test set (see Appendix E.3 for all clusters).

4 Experiments

In the following, we demonstrate the usefulness of our model and its prior structure in four experimental analyses: (a) discovering a multi-facet structure (b) compositionality of latent facets (c) generative, unsupervised classification, and (d) diversity of generated samples from our model. We train our model on three image datasets: MNIST [5], 3DShapes (two configurations) [21] and SVHN [22]. We refer to Appendices D and E for experimental details and further results. We also provide our code implementing MFCVAE, using *PyTorch Distributions* [23], and reproducing our results at `<URL_removed_during_double_blind_review_see_ZIP_in_supplementary_instead>`.

4.1 Discovering a multi-facet structure

We start by demonstrating that our model can discover a multi-facet structure in data. Fig. 3 visualises input examples representative of clusters in a two-facet ($J = 2$) model. For each facet j, input examples $\boldsymbol{x}$ with latent variable $\boldsymbol{z}_j$ are assigned to latent cluster $c_j = \text{argmax}_{c_j} \boldsymbol{\pi}_j(c_j|q_\phi(\mathbf{z}_j|\mathbf{x}))$ according to Eq. (12). Surprisingly, we find that we can represent the two most striking data characteristics – digit class and style (mostly in the form of stroke width, e.g. 'bold', 'thin') in MNIST, object shape and floor colour in 3DShapes (configuration 1), and digit class and background colour in SVHN – in two separate facets of the data. In each facet, clusters are homogeneous w.r.t. a value from the represented characteristic. When comparing our results on MNIST with LTVAE [24], the model closest to ours in its attempt to learn a clustered latent space of multiple facets, LTVAE struggles to separate data characteristics into separate facets (c.f. [24] Fig. 5; in particular, both facets learn digit class, i.e. this characteristic is not properly disentangled between facets), whereas MFCVAE better isolates the two.

4.2 Compositionality of latent facets

A unique advantage of the prior structure of MFCVAE compared to other unsupervised generative models, say a VAE with an isotropic Gaussian prior, is that it allows different abstract characteristics

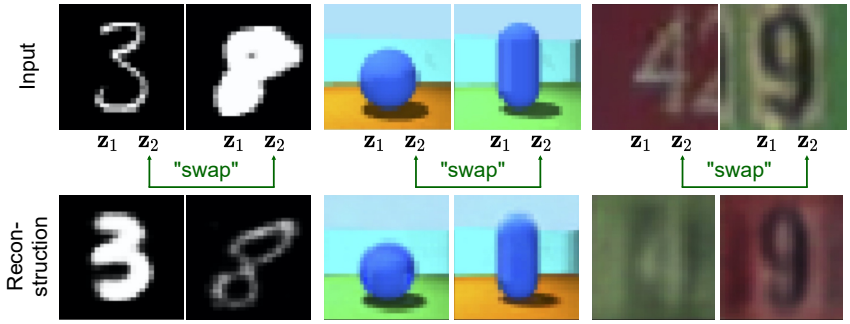

Figure 4: Reconstructions of two input examples when swapping their latent style/colour.

to be composed in the separated latent space. Here, we show how this enables interventions on a per-facet basis, illustrated with a two-facet model where we assume style/colour learned in facet $j = 1$ and digit/shape learned in facet $j = 2$. Let us have two inputs assigned to two different style clusters according to Eq. (12) (and two different digit clusters). For both inputs, we obtain their latent representation $\mathbf{z}_j$ as the modes of $q_\phi(\mathbf{z}_j|\mathbf{x})$, respectively. Now, we swap the style/colour facet's representation, i.e. $\mathbf{z}_1$ of both inputs, and pass these together with their unchanged digit/shape representation $\mathbf{z}_2$ through the decoder $f(\vec{\mathbf{z}};\theta)$ to get reconstructions which we visualise in Fig. 4 (see Appendix E.4 for a more rigorous explanation of this swapping procedure).

Surprisingly, by construction of this intervention in our multi-facet model, we observe reconstructions that "swap" their style/background colour, yet in most cases preserve their digit/shape. This intervention is successful across a wide set of clusters on MNIST and 3DShapes. It works less so on SVHN where we hypothesise that this is due to the much more diverse dataset and (consequently) the model reaching a lower fit (see Section 4.3). We show further examples including failure cases in Appendix E.4 which show that our model learns a multi-facet structure allowing complex inventions.

4.3 Generative, unsupervised classification

Recall our fully unsupervised, exploratory setting of clustering where the goal is to identify and characterise multiple meaningful latent structures *de novo*. In practice, we have no ground-truth data partition – if labels were available, the task would be better formulated as a supervised classification in the first place. That said, it is often reasonable to assume that the class label in a supervised dataset represents a semantically meaningful latent structure that contributes to observed variation in the data. Indeed, this assumption underlies the traditional approach for benchmarking clustering models on labelled data: the class label is hidden during training; afterwards it is revealed as a pseudo ground-truth partitioning of the data for assessing clustering "accuracy". MFCVAE aims to capture multiple latent structures and can be deployed as a multi-facet generative classifier, as distinct from standard single-facet discriminative classifiers [1, p.30]. But we emphasise that high classification accuracy is attained as a by-product, and is *not* our core goal – we do not explicitly target label accuracy, nor does high label accuracy necessarily correspond to the "best" multi-facet clustering.

Following earlier work, we report classification performance in terms of *unsupervised clustering accuracy* on the test set, which we define in Appendix D.4, comparing against commonly used single-facet (SF) and multi-facet (MF) deep clustering approaches (we use results as reported) of both deterministic and probabilistic nature. We report the mean and standard deviation (if available) of accuracy over T runs with different random seeds, where $T = 10$ for MFCVAE. Models marked with $^\eta$ explicitly state that they instead report the best result obtained from R restarts with different random seeds (DEC: $R = 20$, VaDE: $R = 10$). Both of these types of reporting in previous work – not providing error bars over several runs and picking the best run (while not providing error bars) – ignore stability of the model w.r.t. initialisation. We further discuss this issue and the importance of stability in deep clustering approaches in Appendix E.1.

On MNIST, MFCVAE (*ours*, MF) achieves $92.02\% \pm 3.02\%$, compared to DEC with 84.03% ([8] $^\eta$, SF), DCN with 83.0% ([9], SF), IMSAT with $98.4\% \pm 0.4\%$ ([11], SF), ACOL-GAR with $98.32\% \pm 0.08\%$ ([15], SF), VaDE with 94.5% ([10] $^\eta$, SF), and LTVAE with 86.3% ([24], MF). On 3DShapes, MFCVAE obtains $99.46\% \pm 1.10\%$ for floor colour and $88.47\% \pm 1.82\%$ for object shape

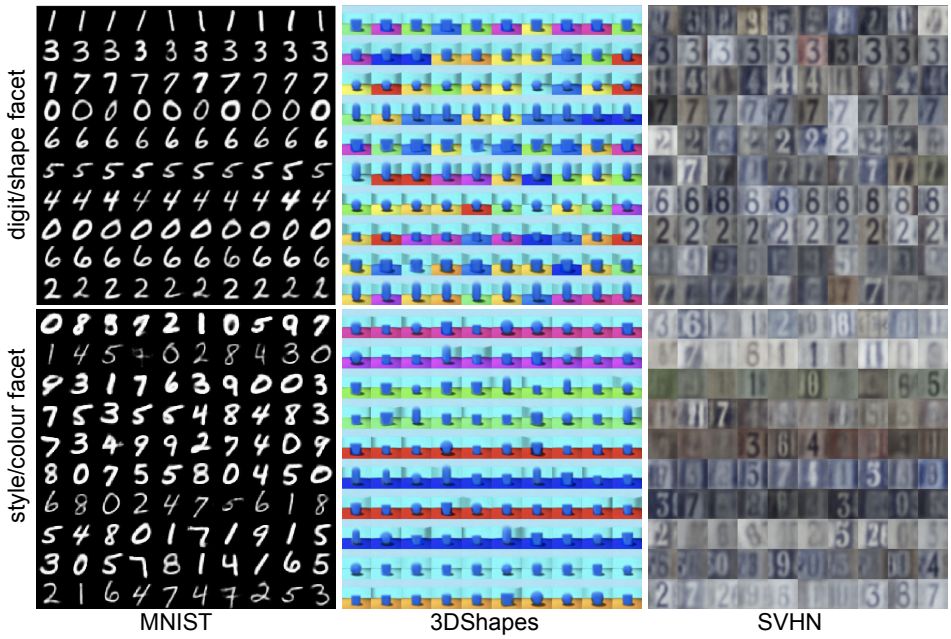

Figure 5: Synthetic samples generated from MFCVAE with two facets ($J = 2$) trained on MNIST, 3DShapes, and SVHN. For each cluster c_j in facet j, $\mathbf{z}_j$ is sampled from $p(\mathbf{z}_j|c_j)$ and $\mathbf{z}_{j'}$ is sampled from $p(\mathbf{z}_{j'})$ for the other facet $j' \neq j$. Each row corresponds to 10 random samples from a cluster. Clusters (rows) are sorted and selected (and are from the same trained model) as in Fig. 3 (see Appendix E.6 for all clusters).

on configuration 1, and $100.00\% \pm 0.00\%$ for wall colour and $90.05\% \pm 2.65\%$ for object shape on configuration 2. On SVHN, MFCVAE achieves $56.25\% \pm 0.88\%$ digit class accuracy, compared to IMSAT with $57.3\% \pm 3.9\%$, ACOL-GAR with $76.80\% \pm 1.30\%$ and VLAC with $37.8\% \pm 2.2\%$ ([25], MF).

In summary, we are able to recover the assumed ground-truth clustering in 3DShapes stably and with high accuracy, and achieve competitive performance on MNIST and SVHN, in particular compared to other probabilistic deep clustering models (IMSAT and ACOL-GAR are single-facet, deterministic models). It is worth noting that we report classification performance for the same hyperparameter configurations and training runs of our model that are used in all experimental sections and in particular for Fig. 3, 4 and 5, i.e. our trained model has a pronounced multi-facet characteristic. In contrast, while it is somewhat unclear, LTVAE seems to report its clustering performance when trained with only a single facet, not when performing multi-facet clustering [24].

4.4 Diversity of generated samples

We lastly show that MFCVAE enables diverse generation of synthetic examples for each given cluster in the different facets, as a downstream task in addition to clustering. To obtain synthetic examples for a cluster c_j in facet j, we sample $\mathbf{z}_j$ from $p(\mathbf{z}_j|c_j)$, and sample $\mathbf{z}_{j'}$ from $p(\mathbf{z}_{j'})$ for all other facets $j' \neq j$. We then take the modes of $p_\theta(\mathbf{x}|\bar{\mathbf{z}})$ where $\bar{\mathbf{z}} = (\mathbf{z}_1, \ldots, \mathbf{z}_j, \ldots, \mathbf{z}_J)$ as the generated images. Fig. 5 shows synthetic examples generated from the models ($J = 2$) trained on MNIST, 3DShapes and SVHN.

For all three datasets, we observe synthetic samples that are homogeneous w.r.t. the characteristic value (e.g. 'red background') of a cluster in the chosen facet (as we sample this continuous latent variable from the conditional distribution), but heterogeneous and diverse w.r.t. all other facets (as we sample all other continuous latent variables from their marginal distribution). For example, in MNIST, when fixing a cluster in the digit facet, we observe generated samples that have the same digit class (e.g. all '1'), but are diverse in style (e.g. different 'stroke width'). Conversely, when fixing a cluster in the style facet, we get samples homogeneous in style, but heterogeneous in digit class. Likewise, in 3DShapes, fixing a cluster in the wall colour facet produces generations diverse

in shape, but having the same wall color, and conversely when fixing the shape facet. Besides, in all clusters, generated samples are diverse w.r.t. other factors of variation in 3DShapes, such as orientation and scale. In SVHN, while less strong than in Fig. 3, these patterns extend here for the two facets style (background colour is particularly distinct) and digit class. These results are consistent with and underline the observed disentanglement of facets that we found in our previous experimental analyses.

5 Related work

Within the deep generative framework, various deep clustering models have been proposed. VaDE [10] is the most important prior work, a probabilistic model which has been highly influential in deep clustering. Related approaches, GM-VAEs [26] and GM-DGMs [27, 28, 29], have similar overall performance and explicitly represent the discrete clustering latent variable during training. Non-parametric approaches include DLDPMMs [27], and HDP-VAEs [30]. Further, many non-generative methods for clustering have been proposed that use neural components [8, 9, 14, 11, 15, 26, 12]. All these approaches, however, view the aim of a clustering algorithm as partitioning a given standard dataset into the same class structure imposed by a supervised label that is given with the data, such as the supervised digit label in MNIST.

Hierarchical VAEs can both be a way to learn more powerful models [31, 32, 33, 34, 35], but can also enable to separate out representations where each layer of latent variables represents a different aspect of the data. Variational Ladder Autoencoders (VLAEs) [19] aim to do the latter: to learn independent sets of latent variables, each representing some part of the data; but each group of latent variables within this set has a $\mathcal{N}(\mathbf{0}, \mathbf{1})$ prior, so it does not perform clustering. Recently, progressive training for VLAEs has been proposed [20] which sharpens the separation between layers.

Learning multiple clusterings simultaneously has been studied in the case of very low dimensional datasets [36, 37, 38, 39] under the names *alternative clusterings* and *non-redundant clustering*. However, when it comes to clustering high-dimensional data, like images, approaches are rare. The recently proposed LTVAE [24] aims to perform this task, proposing a variational autoencoder with a latent tree model prior for a set of continuous latent variables $\vec{\mathbf{z}}$, of which each $\mathbf{z}_j$ has a GMM prior. The neural components are trained via stochastic gradient ascent under the ELBO; this is interleaved with a heuristic (hill-climbing) search algorithm to grow or prune the tree structure and message-passing to learn its nodes' GMM parameters of the current structure of the tree prior in a manner reminiscent of SVAEs [40], rendering the entire training algorithm *not* end-to-end differentiable (in contrast to MFCVAE). LTVAE learns multiple clusterings over the data, however, lacks a proper disentanglement of facets, as discussed in Section 4.1.

6 Conclusion

We introduced Multi-Facet Clustering Variational Autoencoders (MFCVAE), a novel class of probabilistic deep learning models for unsupervised, multi-partition clustering in high-dimensional data that is end-to-end differentiable. We provided novel theoretical results for optimising its ELBO, correcting and extending an influential related paper for the single-facet case. We demonstrated MFCVAE's empirical performance in terms of multi-facet clustering of various levels of abstraction, and the usefulness of its prior structure for composing, classifying and generating samples, achieving state-of-the-art performance among deep probabilistic multi-facet models.

An important limitation of our work shared with many other deep clustering algorithms is the lack of a procedure to find good hyperparameters through a metric known at training time. Future work should explore: MFCVAE with $J > 2$; automatic tuning of hyperparameters J and K_j; application to large-scale datasets of other modalities; and regularising the model facet-wise to further enforce disentangled representations in the latent space [41]. While we successfully stabilized model fitting by combining a VLAE architecture [19] with progressive training [20], further work on unifying and extending these approaches to stable learning will be key to harnessing the full potential of deep clustering models.

References

[1] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[2] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[3] Pierre Hansen and Brigitte Jaumard. Cluster analysis and mathematical programming. *Mathematical programming*, 79(1-3):191–215, 1997.

[4] Ulrike Von Luxburg, Robert C Williamson, and Isabelle Guyon. Clustering: Science or art? In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 65–79. JMLR Workshop and Conference Proceedings, 2012.

[5] Yann LeCun, Corinna Cortes, and CJ Burges. MNIST handwritten digit database, 2010.

[6] Erxue Min, Xifeng Guo, Qiang Liu, Gen Zhang, Jianjing Cui, and Jun Long. A survey of clustering with deep learning: From the perspective of network architecture. *IEEE Access*, 6:39501–39514, 2018.

[7] Elie Aljalbout, Vladimir Golkov, Yawar Siddiqui, Maximilian Strobel, and Daniel Cremers. Clustering with deep learning: Taxonomy and new methods. *arXiv preprint arXiv:1801.07648*, 2018.

[8] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487. PMLR, 2016.

[9] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning*, pages 3861–3870. PMLR, 2017.

[10] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1965–1972, 2017.

[11] Weihua Hu, Takeru Miyato, Seiya Tokui, Eiichi Matsumoto, and Masashi Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *International Conference on Machine Learning*, pages 1558–1567. PMLR, 2017.

[12] Uri Shaham, Kelly Stanton, Henry Li, Boaz Nadler, Ronen Basri, and Yuval Kluger. SpectralNet: Spectral clustering using deep neural networks. In *International Conference on Learning Representations*, 2018.

[13] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4066–4075, 2019.

[14] Sudipto Mukherjee, Himanshu Asnani, Eugene Lin, and Sreeram Kannan. ClusterGAN: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 4610–4617, 2019.

[15] Ozsel Kilinc and Ismail Uysal. Learning latent representations in neural networks for clustering through pseudo supervision and graph-based activity regularization. In *International Conference on Learning Representations*, 2018.

[16] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *International Conference on Learning Representations*, 2017.

[17] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[19] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In *International Conference on Learning Representations*, 2017.

[20] Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. In *International Conference on Learning Representations*, 2020.

[21] Chris Burgess and Hyunjik Kim. 3D shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[22] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

[24] Xiaopeng Li, Zhourong Chen, Leonard K.M. Poon, and Nevin L Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. In *International Conference on Learning Representations*, 2019.

[25] Matthew Willetts, Stephen Roberts, and Chris Holmes. Disentangling to cluster: Gaussian mixture variational ladder autoencoders. In *4th Workshop on Bayesian Deep Learning (NeurIPS 2019)*, 2019.

[26] Nat Dilokthanakul, Pedro A M Mediano, Marta Garnelo, Matthew C H Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture VAE. *arXiv preprint arXiv:1611.02648*, 2017.

[27] Eric Nalisnick, Lars Hertel, and Padhraic Smyth. Approximate inference for deep latent gaussian mixtures. In *Workshop on Bayesian Deep Learning Workshop (NIPS 2016)*, 2016.

[28] Matthew Willetts, Stephen J Roberts, and Christopher C Holmes. Semi-unsupervised learning using deep generative models. In *3rd Workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018.

[29] Matthew Willetts, Stephen J Roberts, and Christopher C Holmes. Semi-unsupervised learning: Clustering and classifying using ultra-sparse labels. In *IEEE Big Data Workshop*, 2020.

[30] Prasoon Goyal, Zhiting Hu, Xiaodan Liang, Chenyu Wang, and Eric P Xing. Nonparametric variational auto-encoders for hierarchical representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-Octob, pages 5104–5112, 2017.

[31] Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 2016.

[32] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, 2016.

[33] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *Advances in Neural Information Processing Systems*, 2019.

[34] Arash Vahdat and Jan Kautz. NVAE: A deep hierarchical variational autoencoder. In *Advances in Neural Information Processing Systems*, 2020.

[35] Rewon Child. Very deep VAEs generalize autoregressive models and can outperform them on images. In *International Conference on Learning Representations*, 2021.

[36] Ying Cui, Xiaoli Z. Fern, and Jennifer G. Dy. Non-redundant multi-view clustering via orthogonalization. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 3:133–142, 2007.

[37] Ian Davidson and Zijie Qi. Finding alternative clusterings using constraints. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 773–778, 2008.

[38] Zijie Qi and Ian Davidson. A principled and flexible framework for finding alternative clusterings. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 717–725, 2009.

[39] Emmanuel Muller, Stephan Gunnemann, Ines Farber, and Thomas Seidl. Discovering multiple clustering solutions: Grouping objects in different views of the data. In *2012 IEEE 28th International Conference on Data Engineering*, pages 1207–1210. IEEE, 2012.

[40] Matthew James Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams. Composing graphical models with neural networks for structured representations and fast inference. In *Advances in Neural Information Processing Systems*, 2016.

[41] William Peebles, John Peebles, Jun-Yan Zhu, Alexei A. Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2020.

[42] Charles R Harris, K Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

[43] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com.

[44] J. D. Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[45] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[46] Guido Van Rossum. *The Python Library Reference, release 3.8.2*. Python Software Foundation, 2020.

[47] Andrew Collette. *Python and HDF5*. O'Reilly, 2013.

[48] G. Bradski. The OpenCV library. *Dr. Dobb's Journal of Software Tools*, 2000.

[49] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[50] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.

[51] Harold W Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[52] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

[53] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064. PMLR, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We set out our theoretical and methodological contributions in the abstract and introduction. We then motivate the problem and state the properties of our model in Section 2, discuss our theoretical contributions in Section 3, and demonstrate the model's properties empirically in Section 4.
   (b) Did you describe the limitations of your work? [Yes] We describe the limitations of our approach both explicitly, and implicitly as part of the future work in Section 6.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] We address this question in Appendix D.4.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] Prior independence across facets and the mean-field assumption of the posterior required for Theorems 1, 2 and 3 are stated in Section 3.
   (b) Did you include complete proofs of all theoretical results? [Yes] The complete proofs of Theorems 1, 2 and 3 in Section 3.1 are shown in Appendix B.

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We provide our code, instructions on how to download the data, and on how to reproduce the main experimental results in a ZIP file.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section D.4.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Throughout the paper (Section 4.3, Appendices E.1, E.5) and across all datasets, we report error bars (standard deviation over 10 runs with different random seeds) for our model (MFCVAE). Specifically, we discuss the importance in deep clustering research of considering error bars to ensure stability of results in Appendix E.1, and discuss the shortcomings of not reporting error bars and picking the best of multiple runs with different random seeds among previous work in Section 4.3.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We address this question in Section D.4.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We discuss all existing assets used in Appendix D.4 and briefly refer to key ones in Section 4.
   (b) Did you mention the license of the assets? [Yes] We mention the license of the assets (if known) in Appendix D.4.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our code as a ZIP file.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] We address this question in Appendix D.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We address this question in Appendix D.1.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]