Multi-VAE: Learning Disentangled View-common and View-peculiar Visual Representations for Multi-view Clustering

Jie Xu[1], Yazhou Ren[1*], Huayi Tang[1], Xiaorong Pu[1], Xiaofeng Zhu[1], Ming Zeng[2], Lifang He[3]
[1]School of Computer Science and Engineering,
University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Department of Electrical Computer Engineering, Carnegie Mellon University, PA 15213, USA
[3]Department of Computer Science and Engineering, Lehigh Univerisity, PA 18015, USA
jiexuwork@outlook.com, {yazhou.ren,puxiaor}@uestc.edu.cn
{tangh4681,seanzhuxf,mingzcmu}@gmail.com, lih319@lehigh.edu

Abstract

Multi-view clustering, a long-standing and important research problem, focuses on mining complementary information from diverse views. However, existing works often fuse multiple views' representations or handle clustering in a common feature space, which may result in their entanglement especially for visual representations. To address this issue, we present a novel VAE-based multi-view clustering framework (Multi-VAE) by learning disentangled visual representations. Concretely, we define a view-common variable and multiple view-peculiar variables in the generative model. The prior of view-common variable obeys approximately discrete Gumbel Softmax distribution, which is introduced to extract the common cluster factor of multiple views. Meanwhile, the prior of view-peculiar variable follows continuous Gaussian distribution, which is used to represent each view's peculiar visual factors. By controlling the mutual information capacity to disentangle the view-common and view-peculiar representations, continuous visual information of multiple views can be separated so that their common discrete cluster information can be effectively mined. Experimental results demonstrate that Multi-VAE enjoys the disentangled and explainable visual representations, while obtaining superior clustering performance compared with state-of-the-art methods.

1. Introduction

Clustering analysis is a fundamental research topic in many fields, such as computer vision, machine learning, and data mining, etc. Its goal is to partition data items with similar patterns or characteristics into the same group. With the unprecedented growth of deep learning, deep clustering methods [9, 37, 44, 47] overcome the shortcomings of shallow models and make considerable progress in clustering performance. In real-world applications, however, visual data is often collected from multiple views or diverse sources, e.g., 1) various writing styles of one digit written by different people, 2) multiple views of an object captured from cameras in multiple directions. Compared with single-view clustering, accordingly, multi-view clustering (MVC) can access to more comprehensive characteristics contained in multi-view data and thus attracts increasing attention.

Existing MVC methods can be roughly divided into three categories: 1) The first category is multi-view spectral clustering [18, 23, 32, 33], where multiple graph structures are constructed for clustering. 2) The second category [25, 52] uses non-negative matrix factorization to decompose the feature matrix and obtain cluster assignments. 3) The third category is based on subspace clustering [21, 53], which conducts self-representation on a subspace shared by multiple views. More researches on MVC can be found in [49].

For many MVC methods, the central bottleneck is their high complexity that makes it unrealistic for handling large-scale data clustering tasks. Recent approaches have achieved inspirational progress by applying deep models [3, 7, 34, 45, 50, 55]. However, most of them learn the cluster structures by exploring common representations or fusing features of all views. Although complementary information can be fetched in this way, the interference caused by the entanglement among multiple views is also ignored.

We are inspired by two observations: 1) Cluster information is discrete, which is an abstraction of the maximum common visual information of all views. 2) Each view's peculiar visual information is often continuous, which has different effects on clustering. For example, the observations from multiple sides of an object are conducive to bet-

*Corresponding author

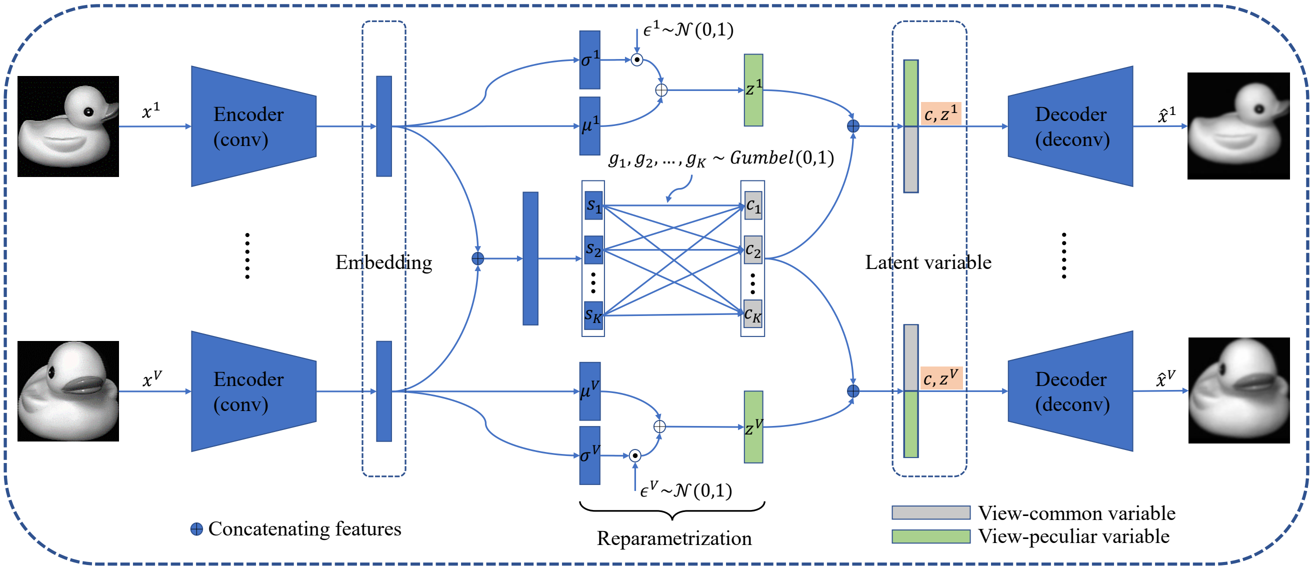


Figure 1. The framework of Multi-VAE. Inference process: $\boldsymbol{z}^v$ extracts the v-th view's peculiar visual information that is contained in the embedding transformed by the corresponding encoder. $\boldsymbol{c}$ represents the cluster information among all views' embeddings. Generative process: the v-th view's latent variable is composed of $\boldsymbol{z}^v$ and $\boldsymbol{c}$, which is fed into the corresponding decoder to generate samples.

ter describe itself. Nevertheless, the two writing styles of a digit have no complementary effect for clustering, instead, which may even cause interference. How to disentangle them and learn explainable multi-view visual representations? This is an interesting but challenging problem. Fortunately, some advancements about disentangled representation learning [1] have been made. Some generative models, such as variational autoencoders (VAE) [2] and generative adversarial networks (GAN) [4], are used to learn the explainable representations, each unit of which corresponds to a single factor of variation of the data. However, *learning disentangled visual representations has been rarely studied for multi-view clustering*.

In this paper, we propose a novel VAE-based framework for multi-view clustering (dubbed Multi-VAE), which can learn disentangled and explainable visual representations and tackle large-scale data clustering problems. Different from the existing multi-view clustering methods, as shown in Figure 1, we introduce a view-common variable $\boldsymbol{c}$ and multiple view-peculiar variables $\{\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^V\}$ in a multiple VAEs architecture. In order to learn the common visual representation across views (i.e., cluster information), the view-common variable $\boldsymbol{c}$ is inferred from all views' embeddings. Meanwhile, each view-peculiar variable $\boldsymbol{z}^v$ is only inferred from the corresponding view's embedding so as to learn peculiar visual representations (like angle, styles, and size, etc). For each view, its latent variable is made up of $\boldsymbol{c}$ and $\boldsymbol{z}^v$ and is used to generate examples. Since the cluster information is discrete and peculiar visual information is continuous, the prior distributions of $\boldsymbol{c}$ and $\boldsymbol{z}^v$ we selected are Gumbel Softmax distribution and Gaussian distribution, respectively. By controlling the mutual information capacity of KL divergence between the posterior of the latent variables and their prior during training, the common and peculiar visual representations of multiple views can be disentangled, which are further used for clustering.

Specifically, the contributions of this work include:

- We propose a novel multi-view VAE framework, namely Multi-VAE, where the view-common and view-peculiar variables are introduced to mine the discrete clusters and continuous visual factors.
- Our model can disentangle all views' common cluster representation and each view's peculiar visual representations. In this way, the interference of multiple views' superfluous information is reduced when mining their complementary information for clustering.
- Multi-VAE shows clearly superior clustering performance compared with other methods. Moreover, its complexity is linear to data size. To our knowledge, this is the first attempt to implement MVC by learning disentangle and explainable representations.

2. Related Work

Autoencoder-based Clustering. In recent years, autoencoder (AE) has shown impressive performance in representation of high-dimensional data. DEC [44] is the well-known method that utilizes AE to perform clustering. Its improved version (IDEC) [11] introduced a reconstruction term to address the distortion of embedded space. The convolutional autoencoder was applied in [9] to deal with image clustering. More clustering works based on AE can be found in [12, 35]. The combination of variational inference and autoencoder leads to the birth of variational autoencoder (VAE) [17]. The VAE-based deep clustering framework is first proposed in [16], where the generative procedure of data is modeled with a Gaussian mixture model [29]. The Gaussian prior is also used in subsequent VAE-based clustering models [5, 22]. Yang *et al.* [48] proposed graph embedding in a Gaussian mixture variational autoencoder. Although there are already some VAE-based multi-view or multi-modal learning methods, such as [8, 20, 42, 50], our work is the first attempt to give a disentangled multi-view VAE framework in view-common and view-peculiar representation learning perspectives.

Multi-view Clustering. Spectral clustering [31] is a popular traditional method. In [18], spectral clustering was extended to perform multi-view clustering. A parameter-free method was proposed in [33], which was an auto-weighted multiple graph learning framework. Non-negative matrix factorization, which is equivalent to the relaxed K-means, is also applied in some multi-view clustering methods. For example, Liu *et al.* [25] explored multi-view common latent factors via matrix factorization. Zhao *et al.* [52] presented a deep matrix factorization structure for multi-view clustering. Much attention is paid to multi-view subspace clustering, which assumes the data of multiple views share a common subspace. In [21], the authors took self-representation layers to obtain subspace hierarchically and utilized encoding layers to achieve multi-view consistency. The work [3] simultaneously learned cluster assignments and multi-view embeddings. Recently, multi-view clustering were discussed with more techniques, e.g., binary coding [51] and self-paced learning [36]. Deep model based multi-view clustering [24, 40, 45, 46, 54] also attracted increasing attention in recent years.

Disentangled Representation Learning. In contrast to ordinary representation learning, disentangled representation learning aims to obtain explainable factors hidden in data [1]. InfoGAN [4] and β-VAE [13] are two most prominent methods for disentanglement in unsupervised manner. InfoGAN can learn both discrete and continuous representations, but it suffers from unstable training and reduced diversity of generated samples. In β-VAE, the ELBO contains likelihood term and KL divergence (D_{KL}) term:

$$\mathcal{L}_{ELBO}(\boldsymbol{x}) = \mathbb{E}_{q(\boldsymbol{z}|\boldsymbol{x})}\left[\log p(\boldsymbol{x}|\boldsymbol{z})\right] - \beta D_{KL}(q(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})), \tag{1}$$

where the observed sample $\boldsymbol{x}$ is generated from the latent variable $\boldsymbol{z}$. People give a higher weight on the KL divergence term (i.e., $\beta > 1$) to increase the pressure of the posterior $q(\boldsymbol{z}|\boldsymbol{x})$ to match the prior $p(\boldsymbol{z})$, which is conducive to learn disentangled representations. VAE-based frameworks to separate discrete and continuous representations were given in [6, 39]. To achieve the balance between the reconstruction quality and disentanglement, those works [2, 6] proposed to gradually increase the upper bound of the KL divergence term during training.

3. The Proposed Method

Problem Statement. Given a multi-view image dataset $\{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \dots, \boldsymbol{x}_i^V\}_{i=1}^N$, each sample has V views that contain different visual information and N is the data size. Multi-view clustering aims to group them into K clusters.

3.1. Architecture

Since our motivation is to learn disentangled representations of multiple views via VAE, we introduce independent view-common variable $\boldsymbol{c} \in \mathbb{R}^K$ and view-peculiar variables $\{\boldsymbol{z}^v \in \mathbb{R}^{Z_v}\}_{v=1}^V$ to model the multi-view data. We consider the following generative model (i.e., joint probability):

$$\begin{aligned} p(\boldsymbol{x}^v, \boldsymbol{z}^v, \boldsymbol{c}) &= p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})p(\boldsymbol{z}^v, \boldsymbol{c}) \\ &= p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})p(\boldsymbol{z}^v)p(\boldsymbol{c}), \end{aligned} \tag{2}$$

where the view-common variable $\boldsymbol{c}$ is shared by all views and represents their cluster information. For the v-th view, the view-peculiar variable $\boldsymbol{z}^v$ represents its peculiar visual information such as angle, size, style, etc. Without loss of generality, cluster information should be obtained from all views and the peculiar information should only be extracted from the v-th view. Let $\{\boldsymbol{x}^v\}$ denote all views' data, i.e., $\{\boldsymbol{x}^1, \boldsymbol{x}^2, \dots, \boldsymbol{x}^V\}$. So, the posterior of $\boldsymbol{c}$ and $\boldsymbol{z}^v$ are written as $p(\boldsymbol{c}|\{\boldsymbol{x}^v\})$ and $p(\boldsymbol{z}^v|\boldsymbol{x}^v)$, respectively. Considering it is intractable to calculate the integral of posterior in VAE, we use $q_\phi(\boldsymbol{c}|\{\boldsymbol{x}^v\})$ and $q_{\phi^v}(\boldsymbol{z}^v|\boldsymbol{x}^v)$ parameterized by ϕ and ϕ^v to approximate the true posterior.

Inference Process. As shown in Figure 1, all views' embeddings are concatenated for the purpose of learning their common information in the inference process. Then, K neurons (denoted as $\boldsymbol{s} = \{s_1, s_2, \dots, s_K\}$) are set to obtain the view-common variable $\boldsymbol{c}$. Concretely, in order to easily represent the cluster assignment of a datum, we expect $\boldsymbol{c}$ is a one-hot representation. However, discrete random variables are non-differentiable for neural networks' parameters. Its differentiable relaxation is discussed in [15, 28]. Based on this, the prior of the view-common variable we selected is a product of independent uniform Gumbel Softmax distributions, i.e., $p(\boldsymbol{c}) = p(c_1)p(c_2)\dots p(c_K)$, where $p(c_k) \sim Gumbel(0, 1)$. Consequently, the approximate posterior $q_\phi(\boldsymbol{c}|\{\boldsymbol{x}^v\})$ is written as

$$q_\phi(\boldsymbol{c}|\{\boldsymbol{x}^v\}) = \prod_{k=1}^{K} q_\phi(c_k|\{\boldsymbol{x}^v\}). \tag{3}$$

According to the Gumbel-Max reparameterization trick [10], we can further get the following expression:

$$q_\phi(c_k|\{\boldsymbol{x}^v\}) = \mathcal{G}(\boldsymbol{s}) = \frac{\exp((\log s_k + g_k)/\tau)}{\sum_{i=1}^{K}\exp((\log s_i + g_i)/\tau)}, \tag{4}$$

where $g_k \sim Gumbel(0,1)$ and τ is the temperature parameter to control the relaxation. Except for the cluster information, we assume other visual information is continuous, and the prior of the view-peculiar variable is standard normal distribution, i.e., $p(\boldsymbol{z}^v) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $q_{\phi^v}(\boldsymbol{z}^v|\boldsymbol{x}^v)$ is parameterized by a factorized Gaussian:

$$q_{\phi^v}(\boldsymbol{z}^v|\boldsymbol{x}^v) = \prod_{i=1}^{Z_v} q_{\phi^v}(z_i^v|\boldsymbol{x}^v). \tag{5}$$

According to the reparameterization trick [17, 38], we have the following elementwise equality:

$$q_{\phi^v}(z_i^v|\boldsymbol{x}^v) = \mathcal{N}(\mu_i^v, (\sigma_i^v)^2) = \mu_i^v + \sigma_i^v \epsilon_i^v, \tag{6}$$

where $\epsilon_i^v \sim \mathcal{N}(0,1)$. μ_i^v and σ_i^v are parameterized with neural networks, whose input is the v-th view's embedding.

Generative Process. Each view's latent variable contains the view-common variable $\boldsymbol{c}$ and the view-peculiar variable $\boldsymbol{z}^v$. In the generative process, they are concatenated to generate examples. Further, the likelihood or decoder of the v-th view can be expressed as

$$\hat{\boldsymbol{x}}^v = p_{\theta^v}(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c}). \tag{7}$$

In the architecture, the parameters ϕ, $\{\phi^1, \phi^2, \ldots, \phi^V\}$, and $\{\theta^1, \theta^2, \ldots, \theta^V\}$ are partially shared, which are omitted in the subsequent derivation for convenience.

3.2. Variational Lower Bound

The objective of variational inference is to maximize the likelihood function of the observed multi-view data. By using Jensen's inequality, the log-likelihood of our proposed model is formulated as

$$\begin{aligned}\sum_{v=1}^{V} \log p(\boldsymbol{x}^v) &= \sum_{v=1}^{V} \log \int_{\boldsymbol{z}^v} \sum_{\boldsymbol{c}} p(\boldsymbol{x}^v, \boldsymbol{z}^v, \boldsymbol{c}) \mathrm{d}\boldsymbol{z}^v \\ &\geq \sum_{v=1}^{V} \mathbb{E}_{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} \left[\log \frac{p(\boldsymbol{x}^v, \boldsymbol{z}^v, \boldsymbol{c})}{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} \right] \\ &= \sum_{v=1}^{V} \mathcal{L}_{ELBO}(\boldsymbol{x}^v),\end{aligned} \tag{8}$$

where $\mathcal{L}_{ELBO}(\boldsymbol{x}^v)$ is the evidence lower bound (ELBO) of the v-th view. In variational inference, maximizing the likelihood is equal to maximizing the ELBO. Given $p(\boldsymbol{x}^v, \boldsymbol{z}^v, \boldsymbol{c}) = p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})p(\boldsymbol{z}^v, \boldsymbol{c})$, each view's ELBO can be written as

$$\begin{aligned}\mathcal{L}_{ELBO}(\boldsymbol{x}^v) = &\mathbb{E}_{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} [\log p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})] \\ &- D_{KL}(q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{z}^v, \boldsymbol{c})).\end{aligned} \tag{9}$$

We assume the view-common and view-peculiar variables are conditionally independent, i.e., $q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\}) = q(\boldsymbol{z}^v|\boldsymbol{x}^v)q(\boldsymbol{c}|\{\boldsymbol{x}^v\})$ and the prior $p(\boldsymbol{z}^v, \boldsymbol{c}) = p(\boldsymbol{z}^v)p(\boldsymbol{c})$. The KL divergence ($D_{KL}$) can be factored into two parts:

$$\begin{aligned}&D_{KL}(q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{z}^v, \boldsymbol{c})) \\ &= \mathbb{E}_{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} \left[\log \frac{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})}{p(\boldsymbol{z}^v, \boldsymbol{c})} \right] \\ &= \mathbb{E}_{q(\boldsymbol{z}^v|\boldsymbol{x}^v)} \mathbb{E}_{q(\boldsymbol{c}|\{\boldsymbol{x}^v\})} \left[\log \frac{q(\boldsymbol{z}^v|\boldsymbol{x}^v)q(\boldsymbol{c}|\{\boldsymbol{x}^v\})}{p(\boldsymbol{z}^v)p(\boldsymbol{c})} \right] \\ &= \mathbb{E}_{q(\boldsymbol{z}^v|\boldsymbol{x}^v)} \mathbb{E}_{q(\boldsymbol{c}|\{\boldsymbol{x}^v\})} \left[\log \frac{q(\boldsymbol{z}^v|\boldsymbol{x}^v)}{p(\boldsymbol{z}^v)} \right] \\ &+ \mathbb{E}_{q(\boldsymbol{z}^v|\boldsymbol{x}^v)} \mathbb{E}_{q(\boldsymbol{c}|\{\boldsymbol{x}^v\})} \left[\log \frac{q(\boldsymbol{c}|\{\boldsymbol{x}^v\})}{p(\boldsymbol{c})} \right] \\ &= D_{KL}(q(\boldsymbol{z}^v|\boldsymbol{x}^v)||p(\boldsymbol{z}^v)) + D_{KL}(q(\boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{c})).\end{aligned} \tag{10}$$

In this way, the KL divergence terms of $\boldsymbol{c}$ and $\boldsymbol{z}^v$ are separated, which is designed for disentangling the view-common and view-peculiar representations. For the v-th view, the objective to be maximized becomes

$$\begin{aligned}&\mathcal{L}_{ELBO}(\boldsymbol{x}^v) = \mathbb{E}_{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} [\log p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})] \\ &- D_{KL}(q(\boldsymbol{z}^v|\boldsymbol{x}^v)||p(\boldsymbol{z}^v)) - D_{KL}(q(\boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{c})).\end{aligned} \tag{11}$$

3.3. Learning Disentangled Representation

As analyzed in [6, 14], the KL divergence term is an upper bound of the mutual information between latent variables and data. For the purpose of disentangling the view-common and view-peculiar representations of our model, each latent variable should encode more information of variation. Therefore, the channel capacity of KL divergence terms in Eq. (11) should gradually increase. We define the controlled capacities C_c and C_z for the KL divergence terms of the view-common and view-peculiar variables, respectively. The ELBO of the v-th view is formulated as

$$\begin{aligned}\mathcal{L}_{ELBO}(\boldsymbol{x}^v) = &\mathbb{E}_{q(\boldsymbol{z}^v, \boldsymbol{c}|\{\boldsymbol{x}^v\})} [\log p(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})] \\ &- \beta |D_{KL}(q(\boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{c})) - C_c| \\ &- \beta |D_{KL}(q(\boldsymbol{z}^v|\boldsymbol{x}^v)||p(\boldsymbol{z}^v)) - C_z|,\end{aligned} \tag{12}$$

where β is a trade-off coefficient. In particular, when $p(\boldsymbol{c})$ is a uniform categorical distribution, the KL divergence about the view-common variable is bounded:

$$\begin{aligned}D_{KL}(q(\boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{c})) &= \sum_{i=1}^{K} q_i \log \frac{q_i}{p_i} = \sum_{i=1}^{K} q_i \log \frac{q_i}{1/K} \\ &= -H(q) + \log K \leq \log K,\end{aligned} \tag{13}$$

where H is the entropy. Based on this, we let $C_c = \log K$, which controls the maximum capacity of the variational information encoded in $\boldsymbol{c}$. Considering that different views have different scale of data reconstruction loss, we further introduce weights on β to balance the disentanglement of all views. For the v-th view, the weight is calculated by

$$\beta^v = \beta \frac{\mathbb{E}_{q(\boldsymbol{z}^v,\boldsymbol{c}|\{\boldsymbol{x}^v\})}\left[\log p(\boldsymbol{x}^v|\boldsymbol{z}^v,\boldsymbol{c})\right]}{\max_v \mathbb{E}_{q(\boldsymbol{z}^v,\boldsymbol{c}|\{\boldsymbol{x}^v\})}\left[\log p(\boldsymbol{x}^v|\boldsymbol{z}^v,\boldsymbol{c})\right]}. \tag{14}$$

Eventually, our total loss function contains three parts:

$$\begin{aligned} \mathcal{L}_{loss} &= -\sum_{v=1}^{V} \mathcal{L}_{ELBO}(\boldsymbol{x}^v) \\ &= \sum_{v=1}^{V} \beta^v |D_{KL}(q(\boldsymbol{c}|\{\boldsymbol{x}^v\})||p(\boldsymbol{c})) - C_c| \\ &+ \sum_{v=1}^{V} \beta^v |D_{KL}(q(\boldsymbol{z}^v|\boldsymbol{x}^v)||p(\boldsymbol{z}^v)) - C_z| \\ &- \sum_{v=1}^{V} \mathbb{E}_{q(\boldsymbol{z}^v,\boldsymbol{c}|\{\boldsymbol{x}^v\})}\left[\log p(\boldsymbol{x}^v|\boldsymbol{z}^v,\boldsymbol{c})\right], \end{aligned} \tag{15}$$

where the first and second terms are optimized to learn disentangled view-common and view-peculiar representations. The third term is the likelihood term, which is optimized to maintain the reconstruction quality of VAEs.

Multi-VAE-C: Review the framework in Figure 1, all views' features are separated into $\{\boldsymbol{c}, \boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^V\}$. Then each couple of $\{\boldsymbol{c}, \boldsymbol{z}^v\}$ is combined to reconstruct the features. In this way, each view's peculiar visual information is extracted by its view-peculiar representation (or variable) $\boldsymbol{z}^v$. Conversely, all views' common cluster information is extracted by the view-common representation $\boldsymbol{c}$. Since $\boldsymbol{c}$ is the approximation of one-hot representation, the clustering prediction of the i-th sample can be calculated by

$$y_i = \arg\max_j (\boldsymbol{c}_j) = \arg\max_j (q_\phi(\boldsymbol{c}_j|\{\boldsymbol{x}_i^v\})). \tag{16}$$

Multi-VAE-CZ: Given multiple views' visual information may be complementary for clustering, we scale the separated representations to $[0, 1]$ and concatenate them to form a global latent representation (denoted as $[\boldsymbol{c}; \{\boldsymbol{z}^v\}]$), which is fed into K-means to obtain another clustering prediction.

Complexity Analysis. We define K, V, N as the number of clusters, views, and data points, respectively. Let M denote the maximum number of neurons in autoencoders and Z denote the maximum dimensionality of view-peculiar variables. Generally, $V, K, Z \ll M$ holds. The optimization of Multi-VAE is just to minimize Eq. (15), which is summarized in Algorithm 1. In each iteration, the complexity to generate the prior distributions for the view-common variable is $O(NK)$ and for the view-peculiar variables is $O(VNZ)$. The complexity of autoencoders of all views is $O(VNM^2)$. Therefore, the total complexity of our method is linear to the data size N.

Algorithm 1 : The optimization of Multi-VAE

Input: Multi-view dataset $\{\boldsymbol{x}_i^1, \boldsymbol{x}_i^2, \ldots, \boldsymbol{x}_i^V\}_{i=1}^N$;
Number of clusters K; Trade-off coefficient β;
Maximum controlled capacity C_z; $C_c = \log K$.

1: Randomly initialize parameters ϕ and $\{\phi^v, \theta^v\}_{v=1}^V$.
2: **while** not reaching the maximal epochs **do**
3: Calculate $\{s^k\}_{k=1}^K$ and $\{\boldsymbol{\mu}^v, \boldsymbol{\delta}^v\}_{v=1}^V$ by encoders.
4: Infer $q_\phi(\boldsymbol{c}|\{\boldsymbol{x}^v\})$ and $\{q_{\phi^v}(\boldsymbol{z}^v|\boldsymbol{x}^v)\}_{v=1}^V$ by Eqs. (3) and (5) with reparameterization tricks.
5: Generate $\{p_{\theta^v}(\boldsymbol{x}^v|\boldsymbol{z}^v, \boldsymbol{c})\}_{v=1}^V$ by decoders.
6: Update ϕ and $\{\phi^v, \theta^v\}_{v=1}^V$ by minimizing Eq. (15).
7: **end while**

Output: Disentangled representations $\boldsymbol{c}$ and $\{\boldsymbol{z}^v\}_{v=1}^V$.

4. Experimental Setup

4.1. Datasets

MNIST [19] is a popular handwritten digital image dataset (0-9). Fashion [43] contains 10 kinds of fashionable products (such as T-shirt, dress, and coat, etc). COIL [30] is another image dataset about different poses of objects (like cup, duck and block, etc). In order to evaluate multi-view clustering performance and disentangled visual representations, we construct **Multi-MNIST**, **Multi-Fashion**, **Multi-COIL-10** and **Multi-COIL-20**. For those datasets, multiple views of each example are randomly sampled from a same category. Specifically, in Multi-MNIST, an example with different views implies a same digit written by different people. In Multi-Fashion, the different fashionable designs of one category of product denote different views. In Multi-COIL-10 ($K = 10$) and Multi-COIL-20 ($K = 20$), the different views of one object are various in poses but have the same cluster information. In some scenarios, different views of data may obey different distributions. To make the gap of different views' visual information larger, we construct **Digit-Product** and **Object-Digit-Product**. Concretely, in Digit-Product, view-1 is MNIST and view-2 is Fashion. In Object-Digit-Product, view-1 is COIL, view-2 is MNIST, and view-3 is Fashion. So the multi-view visual information is across datasets but their clusters are one-to-one correspondence. For example, cup corresponds to digit-0 and T-shirt; duck corresponds to digit-1 and trouser, etc.

4.2. Comparing Methods

We compare Multi-VAE against the following popular and state-of-the-art methods. *Single-view methods*: **K-means** [27] is a popular traditional method. **DEC** [44] and **IDEC** [11] are AE-based methods. **β-VAE** [2] and **JointVAE** [6] are VAE-based methods, the learned representations of which are used to perform clustering. *Multi-view methods*: **BMVC** [51] is binary multi-view clustering. **RMSL** [21] presents a reciprocal multi-layer sub-

	Datasets	Multi-COIL-10				Multi-COIL-20				Object-Digit-Product			
	Size	720 samples, 3 views				1,440 samples, 3 views				720 samples, 3 views			
	Metrics	ACC	NMI	ARI	Purity	ACC	NMI	ARI	Purity	ACC	NMI	ARI	Purity
Single-view	K-means (1967)	0.733	0.769	0.648	0.757	0.415	0.645	0.384	0.415	0.326	0.297	0.143	0.337
	DEC (2016)	0.740	0.774	0.656	0.765	0.651	0.784	0.587	0.677	0.317	0.344	0.168	0.334
	IDEC (2017)	0.736	0.772	0.651	0.763	0.657	0.784	0.591	0.679	0.327	0.343	0.167	0.337
	β-VAE (2018)	0.598	0.685	0.514	0.632	0.531	0.667	0.450	0.573	0.297	0.278	0.111	0.321
	JointVAE (2018)	0.649	0.724	0.553	0.681	0.537	0.678	0.456	0.548	0.320	0.254	0.126	0.331
Multi-view	BMVC (2018)	0.678	0.681	0.530	0.678	0.834	0.900	0.813	0.881	0.810	0.661	0.634	0.810
	RMSL (2019)	0.964	0.925	0.921	0.964	0.665	0.763	0.587	0.691	0.950	0.917	0.906	0.953
	MVC-LFA (2019)	0.860	0.868	0.799	0.871	0.801	0.852	0.738	0.802	0.926	0.880	0.849	0.926
	COMIC (2019)	0.796	0.916	0.729	0.799	0.496	0.770	0.309	0.500	0.201	0.419	0.146	0.203
	SAMVC (2020)	0.667	0.826	0.621	0.729	0.570	0.791	0.554	0.610	0.770	0.826	0.702	0.801
	DEMVC (2021)	0.891	0.948	0.897	0.900	0.850	0.965	0.860	0.850	0.801	0.901	0.784	0.801
	Multi-VAE-C (ours)	0.900	0.967	0.897	0.900	0.845	0.943	0.842	0.876	0.897	0.942	0.873	0.897
	Multi-VAE-CZ (ours)	0.993	0.989	0.985	0.993	0.980	0.976	0.961	0.980	0.977	0.971	0.954	0.977

Table 1. Comparison results on small-scale datasets. The best and the second best values are highlighted in red and blue, respectively.

	Datasets	Multi-MNIST				Mult-Fashion				Digit-Product			
	Size	70,000 samples, 2 views				10,000 samples, 3 views				30,000 samples, 2 views			
	Metrics	ACC	NMI	ARI	Purity	ACC	NMI	ARI	Purity	ACC	NMI	ARI	Purity
Single-view	K-means (1967)	0.539	0.482	0.360	0.577	0.476	0.513	0.348	0.551	0.349	0.346	0.187	0.390
	DEC (2016)	0.875	0.849	0.803	0.875	0.563	0.617	0.451	0.609	0.396	0.408	0.226	0.422
	IDEC (2017)	0.884	0.868	0.826	0.884	0.569	0.625	0.461	0.615	0.402	0.442	0.233	0.433
	β-VAE (2018)	0.493	0.436	0.291	0.519	0.513	0.510	0.337	0.513	0.343	0.317	0.174	0.385
	JointVAE (2018)	0.641	0.614	0.490	0.651	0.393	0.368	0.246	0.415	0.471	0.435	0.289	0.479
Multi-view	BMVC (2018)	0.893	0.902	0.856	0.897	0.779	0.756	0.682	0.782	0.548	0.442	0.379	0.570
	RMSL (2019)	–	–	–	–	0.376	0.342	0.204	0.391	–	–	–	–
	MVC-LFA (2019)	–	–	–	–	0.782	0.748	0.685	0.784	–	–	–	–
	COMIC (2019)	–	–	–	–	0.578	0.642	0.436	0.608	–	–	–	–
	SAMVC (2020)	–	–	–	–	0.622	0.688	0.557	0.661	0.649	0.619	0.499	0.674
	DEMVC (2021)	0.982	0.989	0.986	0.982	0.786	0.903	0.772	0.791	0.798	0.896	0.833	0.798
	Multi-VAE-C (ours)	0.989	0.996	0.989	0.989	0.816	0.856	0.762	0.818	0.853	0.832	0.810	0.853
	Multi-VAE-CZ (ours)	0.999	0.998	0.999	0.999	0.907	0.883	0.839	0.907	0.925	0.934	0.907	0.923

Table 2. Comparison results on large-scale datasets. "–" denotes the unknown result due to high complexity of the corresponding method.

space learning method. **MVC-LFA** [41] proposes late fusion alignment maximization for multi-view clustering. **COMIC** [34] performs clustering by matching cross-views. **SAMVC** [36] is an auto-weighted multi-view clustering method with self-paced learning. **DEMVC** [45] introduces a collaborative training trick in deep multi-view clustering.

4.3. Implementation Details

The convolutional (Conv) and fully connected (Fc) neural networks are adopted in our Multi-VAE[1]. The encoder is: Input $-$ Conv^4_{32} $-$ Conv^4_{64} $-$ Conv^4_{64} $-$ Fc_{256}. It means that convolution kernel sizes are 4-4-4, channels are 32-64-64, and the dimensionality of embedding is 256. The stride is set to 2. $\boldsymbol{s}$, $\boldsymbol{\mu}^v$ and $\boldsymbol{\sigma}^v$ are parameterized with linear layers. The decoders are symmetric with the encoders. All view-peculiar variables are 10-dimensional. The temperature parameter τ we adopted is 0.67 and the activation function is ReLU. On Multi-MNIST, Multi-Fashion, Multi-COIL-10, and Multi-COIL-20, an encoder and a decoder are shared for all views. We use Adam with the learning rate of 0.0005 to train the model for 500 epochs. β is set to 30. The maximum controlled capacity of view-peculiar variables is set to 5. For the comparing methods, we use open-source codes with the settings recommended by the authors.

[1] https://github.com/SubmissionsIn/Multi-VAE

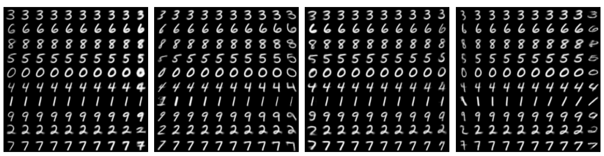
5. Experimental Results and Analysis

5.1. Comparison with State-of-the-Arts

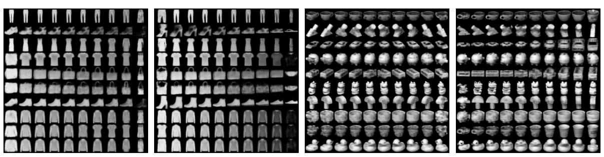
We use four quantitative metrics, including clustering accuracy (ACC), normalized mutual information (NMI), adjusted rand index (ARI), and Purity. The higher value indicates better clustering performance. The results on small-scale and large-scale datasets are reported in Tables 1 and 2, respectively, from which we obtain the following conclusions: 1) In general, the performance of multi-view methods

(a) $\{\boldsymbol{c}, \mathbf{0}\}$ (b) $\{\boldsymbol{c}, \boldsymbol{z}^1\}$ (c) $\{\boldsymbol{c}, \boldsymbol{z}^2\}$

Figure 2. View-common and view-peculiar variables represent different visual information.



(a) Thickness (b) Angle (c) Width (d) Height



(e) Size (f) Grayscale (g) Angle (h) Orientation

Figure 3. Disentangled view-peculiar visual representations of different datasets.

is better than that of single-view methods especially for the datasets with large visual gaps (e.g., Object-Digit-Product and Digit-Product). The reason is that, intuitively, multi-view clustering methods are designed for handling multi-view data which can exploit richer properties to improve the clustering performance. 2) On all datasets, our method consistently achieves the best performance in terms of most metrics. The intrinsic reason is that, by learning disentangled visual representations, Multi-VAE reduces the interference between each view's peculiar information and all views' common cluster information. Further, it becomes more effective to discover the complementary information and common cluster structures of multiple views. 3) The clustering performance on the global latent representation (Multi-VAE-CZ) is better than that on the view-common representation (Multi-VAE-C). The principal reason is that some visual information hidden in view-peculiar representations is complementary for multi-view clustering.

5.2. Disentangled Visual Representation Analysis

Multi-VAE can generate images by decoding the latent variable $\{\boldsymbol{c}, \boldsymbol{z}^v\}$. We show the samples generated by varying the view-common variable $\boldsymbol{c}$ (in form of one-hot) while setting $\boldsymbol{z}^v$ to $\mathbf{0}$. In Figure 2(a), each unit of the view-common variable generates the standard samples of one class, which are all views' common characteristics and precisely represent their cluster information. Then, for the v-th view, $\boldsymbol{c}$ is fixed and the continuous view-peculiar variable $\boldsymbol{z}^v$ is randomly sampled. As shown in Figure 2(b) and (c), the samples belonging to the same class contain different visual characteristics in different views. Further, we fix each cluster representation in $\boldsymbol{c}$ and show the results generated with traversals of certain components of view-peculiar variable $\boldsymbol{z}^v$. In Figure 3, it is discovered that the view-peculiar visual representations are also disentangled, such as thickness, angle, width, and height for MNIST; size and grayscale for Fashion; angle and orientation for COIL. Those visual factors are continuous.

Intuitively, the negative effects are likely to occur if people do not disentangle each view's continuous visual factors from its cluster factor. Different from other methods (e.g., fusion of features or learning common subspace), in Multi-VAE, all views' common cluster information is encoded in the view-common representation, and each view's peculiar or superfluous information is encoded in the respective view-peculiar representation. Hence, our method can learn the disentangled and explainable representations, which is the most fundamental reason for its improvement compared to other approaches.

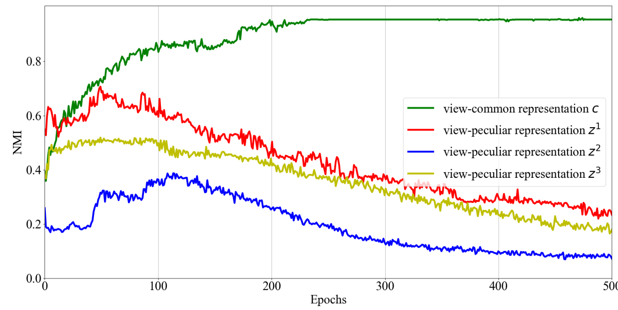
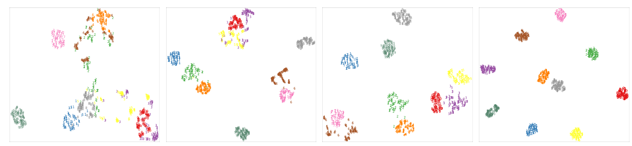


Figure 4. Learning process. The clustering performance of view-common representation gradually enhances, and that of view-peculiar representations is the opposite.



(a) Epoch 10 (b) Epoch 100 (c) Epoch 200 (d) Epoch 300

Figure 5. Visualization of the global latent representation $[\boldsymbol{c}; \{\boldsymbol{z}^v\}]$ with t-SNE [26].

5.3. Multi-view Clustering Process Analysis

The learning process on Object-Digit-Product is shown in Figure 4. In the beginning, the view-common variable $\boldsymbol{c}$ has no representation capability for cluster information of multiple views. The view-common and view-peculiar representations are mixed in latent variables, which corre-

Variants	ACC	NMI	ARI
Multi-VAE (vanilla VAE)	0.872	0.907	0.862
Multi-VAE (β-VAE)	0.683	0.780	0.632
Multi-VAE (C_z)	0.704	0.726	0.592
Multi-VAE (C_c)	0.902	0.959	0.880
Multi-VAE ($C_c + C_z$)	0.993	0.989	0.985

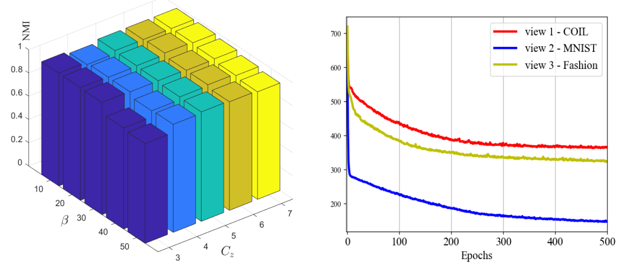
Table 3. Ablation study on the variants of Multi-VAE.

sponds to low clustering performance and entanglement of representations, as shown in Figure 5(a). Gradually, the clustering performance of view-common representation is improved and that of view-peculiar representations is reduced. The global latent representations are also separated correspondingly as shown in Figure 5 (b)–(d).

Accordingly, we can conclude the mechanism of Multi-VAE to improve clustering performance: 1) The view-common variable captures the common cluster information of multiple views, which plays a major role in clustering. Disentangling the view-common variable from all latent variables facilitates the view-common representation to learn better cluster structures. 2) Except for the cluster information, each view's peculiar visual information is learned by its view-peculiar variable. Although the view-peculiar visual representation has no clear cluster structures, it may be complementary for other views. This is in accord with the conclusion 3 obtained in Section 5.1.

5.4. Ablation Study

In this subsection, four variants are tested to examine the effect of our proposed framework: 1) Multi-VAE (vanilla VAE) consists of multiple vanilla VAEs with the proposed architecture. 2) Multi-VAE (β-VAE) denotes the variant applying β-VAE without setting any controlled capacity. 3) Multi-VAE (C_z) is the model with the controlled capacity only for view-peculiar variables. Similarly, 4) Multi-VAE (C_c) denotes the controlled capacity is only set for view-common variable. Multi-VAE ($C_c + C_z$) represents the complete framework. Table 3 shows the results on Multi-COIL-10 and interesting validations are made as follows. Compared with using vanilla VAE, directly improving the weight of KL divergence terms (i.e., using β-VAE) is not helpful to learn the multi-view clustering information. The worst performance comes from applying controlled capacity only to view-peculiar variables, which makes the model's learning focus on the peculiar information among multiple views. The application of controlled capacity to view-common variable results in considerable improvement, because the view-common variable emphasizes learning common cluster information of all views. The optimal setting is both controlled capacities are adopted during learning the disentangled representations. Hence, the model's different parts have distinct contributions that are in accordance with our motivation.



(a) (b)

Figure 6. (a) Parameter sensitivity analysis of β and C_z. (b) Training loss on Object-Digit-Product.

5.5. Parameter and Convergence Analysis

The hyper-parameters of Multi-VAE include the trade-off coefficient β and the maximum controlled capacity C_z for view-peculiar variables. As shown in Figure 6(a), we employ the grid search strategy and test the mean clustering performance on them. Even though most frameworks for disentanglement are sensitive to the choice of hyper-parameters [6], β and C_z are insensitive to the clustering performance of our method. The possible reason is, in the proposed multiple VAEs architecture, the inference that comes from all views increases the robustness compared to single-view architectures. The training loss is shown in Figure 6(b), from which we know that although different views have large gaps in visual information, Multi-VAE has good convergence property.

6. Conclusion

In this paper, we have presented a novel generative model (Multi-VAE) that can learn disentangled visual representations for multi-view clustering. In Multi-VAE, all views' cluster representation and each view's specific visual representations are disentangled by the proposed view-common variable and view-peculiar variables, respectively. Extensive experiments demonstrate that Multi-VAE achieves state-of-the-art clustering performance. In addition, Multi-VAE has linear complexity to data size. Its framework to learn disentangled and explainable visual representations is instructive for multi-view learning.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (No. 61806043), Sichuan Science and Technology Program (Nos. 2021YFS0172 and 2020YFS0119), Fundamental Research Funds for the Central Universities of China (No. ZYGX2021YGLH022), NSF ONR (No. N00014-18-1-2009), and Lehigh's Accelerator Grant (No. S00010293).

References

[1] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *TPAMI*, 35(8):1798–1828, 2013. 2, 3

[2] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in $beta$-vae. *NeurIPS*, 2018. 2, 3, 5

[3] Mansheng Chen, Ling Huang, Chang-Dong Wang, and Dong Huang. Multi-view clustering in latent embedding space. In *AAAI*, pages 3513–3520, 2020. 1, 3

[4] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*, pages 2180–2188, 2016. 2, 3

[5] Nat Dilokthanakul, Pedro AM Mediano, Marta Garnelo, Matthew CH Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016. 3

[6] Emilien Dupont. Learning disentangled joint continuous and discrete representations. *NeurIPS*, 2018. 3, 4, 5, 8

[7] Shaohua Fan, Xiao Wang, Chuan Shi, Emiao Lu, Ken Lin, and Bai Wang. One2multi graph autoencoder for multi-view graph clustering. In *Proceedings of The Web Conference 2020*, pages 3070–3076, 2020. 1

[8] Marco Federici, Anjan Dutta, Patrick Forré, Nate Kushman, and Zeynep Akata. Learning robust representations via multi-view information bottleneck. *ICLR*, 2020. 3

[9] Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *ICCV*, pages 5736–5745, 2017. 1, 2

[10] Emil Julius Gumbel. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 33, 1954. 3

[11] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017. 2, 5

[12] Xifeng Guo, En Zhu, Xinwang Liu, and Jianping Yin. Deep embedded clustering with data augmentation. In *ACML*, pages 550–565, 2018. 3

[13] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2016. 3

[14] Matthew D Hoffman and Matthew J Johnson. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *Workshop in Advances in Approximate Bayesian Inference, NIPS*, 2016. 4

[15] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *ICLR*, 2017. 3

[16] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: An unsupervised and generative approach to clustering. *IJCAI*, pages 1965–1972, 2017. 3

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *ICLR*, 2014. 3, 4

[18] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. *NIPS*, 24:1413–1421, 2011. 1, 3

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[20] Mihee Lee and Vladimir Pavlovic. Private-shared disentangled multimodal vae for learning of hybrid latent representations. *arXiv preprint arXiv:2012.13024*, 2020. 3

[21] Ruihuang Li, Changqing Zhang, Huazhu Fu, Xi Peng, Tianyi Zhou, and Qinghua Hu. Reciprocal multi-layer subspace learning for multi-view clustering. In *ICCV*, pages 8172–8180, 2019. 1, 3, 5

[22] Xiaopeng Li, Zhourong Chen, Leonard KM Poon, and Nevin L Zhang. Learning latent superstructures in variational autoencoders for deep multidimensional clustering. *ICLR*, 2019. 3

[23] Yeqing Li, Feiping Nie, Heng Huang, and Junzhou Huang. Large-scale multi-view spectral clustering via bipartite graph. In *AAAI*, pages 2750–2756, 2015. 1

[24] Yijie Lin, Yuanbiao Gou, Zitao Liu, Boyun Li, Jiancheng Lv, and Xi Peng. COMPLETER: Incomplete multi-view clustering via contrastive prediction. In *CVPR*, 2021. 3

[25] Jialu Liu, Chi Wang, Jing Gao, and Jiawei Han. Multi-view clustering via joint nonnegative matrix factorization. In *SDM*, pages 252–260. SIAM, 2013. 1, 3

[26] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9:2579–2605, 2008. 7

[27] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967. 5

[28] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *ICLR*, 2017. 3

[29] Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual Review of Statistics and Its Application*, 2000. 3

[30] Sameer A Nene, Shree K Nayar, Hiroshi Murase, et al. Columbia object image library (coil-100). 1996. 5

[31] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2002. 3

[32] Feiping Nie, Guohao Cai, and Xuelong Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *AAAI*, pages 2408–2414, 2017. 1

[33] Feiping Nie, Jing Li, Xuelong Li, et al. Parameter-free auto-weighted multiple graph learning: a framework for multi-view clustering and semi-supervised classification. In *IJCAI*, pages 1881–1887, 2016. 1, 3

[34] Xi Peng, Zhenyu Huang, Jiancheng Lv, Hongyuan Zhu, and Joey Tianyi Zhou. COMIC: Multi-view clustering without parameter selection. In *ICML*, pages 5092–5101, 2019. 1, 6

[35] Yazhou Ren, Kangrong Hu, Xinyi Dai, Lili Pan, Steven CH Hoi, and Zenglin Xu. Semi-supervised deep embedded clustering. *Neurocomputing*, 325:121–130, 2019. 3

[36] Yazhou Ren, Shudong Huang, Peng Zhao, Minghao Han, and Zenglin Xu. Self-paced and auto-weighted multi-view clustering. *Neurocomputing*, 383:248–256, 2020. 3, 6

[37] Yazhou Ren, Ni Wang, Mingxia Li, and Zenglin Xu. Deep density-based image clustering. *Knowledge-Based Systems*, 197:105841, 2020. 1

[38] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286. PMLR, 2014. 4

[39] Andriy Serdega and Dae-Shik Kim. VMI-VAE: Variational mutual information maximization framework for vae with discrete and continuous priors. *arXiv preprint arXiv:2005.13953*, 2020. 3

[40] Xiukun Sun, Miaomiao Cheng, Chen Min, and Liping Jing. Self-supervised deep multi-view subspace clustering. In *ACML*, pages 1001–1016, 2019. 3

[41] Siwei Wang, Xinwang Liu, En Zhu, Chang Tang, Jiyuan Liu, Jingtao Hu, Jingyuan Xia, and Jianping Yin. Multi-view clustering via late fusion alignment maximization. In *IJCAI*, pages 3778–3784, 2019. 6

[42] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. *NeurIPS*, 2018. 3

[43] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[44] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *ICML*, pages 478–487, 2016. 1, 2, 5

[45] Jie Xu, Yazhou Ren, Guofeng Li, Lili Pan, Ce Zhu, and Zenglin Xu. Deep embedded multi-view clustering with collaborative training. *Information Sciences*, 573:279–290, 2021. 1, 3, 6

[46] Jie Xu, Yazhou Ren, Huayi Tang, Zhimeng Yang, Lili Pan, Yang Yang, and Xiaorong Pu. Self-supervised discriminative feature learning for deep multi-view clustering. *arXiv preprint arXiv:2103.15069*, 2021. 3

[47] Bo Yang, Xiao Fu, Nicholas D Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *ICML*, pages 3861–3870, 2017. 1

[48] Linxiao Yang, Ngai-Man Cheung, Jiaying Li, and Jun Fang. Deep clustering by gaussian mixture variational autoencoders with graph embedding. In *ICCV*, pages 6440–6449, 2019. 3

[49] Yan Yang and Hao Wang. Multi-view clustering: A survey. *Big Data Mining and Analytics*, 1(2):83–107, 2018. 1

[50] Ming Yin, Weitian Huang, and Junbin Gao. Shared generative latent representation learning for multi-view clustering. In *AAAI*, pages 6688–6695, 2020. 1, 3

[51] Zheng Zhang, Li Liu, Fumin Shen, Heng Tao Shen, and Ling Shao. Binary multi-view clustering. *TPAMI*, 41(7):1774–1782, 2018. 3, 5

[52] Handong Zhao, Zhengming Ding, and Yun Fu. Multi-view clustering via deep matrix factorization. In *AAAI*, pages 2921–2927, 2017. 1, 3

[53] Qinghai Zheng, Jihua Zhu, Zhongyu Li, Shanmin Pang, Jun Wang, and Yaochen Li. Feature concatenation multi-view subspace clustering. *Neurocomputing*, 379:89–102, 2020. 1

[54] Runwu Zhou and Yi-Dong Shen. End-to-end adversarial-attention network for multi-modal clustering. In *CVPR*, pages 14619–14628, 2020. 3

[55] Pengfei Zhu, Binyuan Hui, Changqing Zhang, Dawei Du, Longyin Wen, and Qinghua Hu. Multi-view deep subspace clustering networks. *arXiv preprint arXiv:1908.01978*, 2019. 1