A Decoder-Free Variational Deep Embedding for Unsupervised Clustering

Qiang Ji, Yanfeng Sun, *Member, IEEE*, Junbin Gao, Yongli Hu, *Member, IEEE*, and Baocai Yin, *Member, IEEE*

***Abstract*— In deep clustering frameworks, autoencoder (AE)- or variational AE-based clustering approaches are the most popular and competitive ones that encourage the model to obtain suitable representations and avoid the tendency for degenerate solutions simultaneously. However, for the clustering task, the decoder for reconstructing the original input is usually useless when the model is finished training. The encoder–decoder architecture limits the depth of the encoder so that the learning capacity is reduced severely. In this article, we propose a decoder-free variational deep embedding for unsupervised clustering (DFVC). It is well known that minimizing reconstruction error amounts to maximizing a lower bound on the mutual information (MI) between the input and its representation. That provides a theoretical guarantee for us to discard the bloated decoder. Inspired by contrastive self-supervised learning, we can directly calculate or estimate the MI of the continuous variables. Specifically, we investigate unsupervised representation learning by simultaneously considering the MI estimation of continuous representations and the MI computation of categorical representations. By introducing the data augmentation technique, we incorporate the original input, the augmented input, and their high-level representations into the MI estimation framework to learn more discriminative representations. Instead of matching to a simple standard normal distribution adversarially, we use end-to-end learning to constrain the latent space to be cluster-friendly by applying the Gaussian mixture distribution as the prior. Extensive experiments on challenging data sets show that our model achieves higher performance over a wide range of state-of-the-art clustering approaches.**

***Index Terms*— Augmented mutual information (MI), deep clustering, self-supervised learning (SSL), variational embedding.**

I. Introduction

THE need for large-scale labeled data sets is a major obstacle to the applicability of deep learning in many scenarios, where annotating data are expensive and time consuming. Unsupervised learning has the potential to scale up with this ever-increasing availability of data as it alleviates the need to carefully handcraft and annotates data sets. Clustering can be considered the most fundamental unsupervised learning task of discovering the inherent grouping structure, such that data points in the same group are more similar to each other and dissimilar to the data points in other groups. Unfortunately, traditional clustering algorithms [1]–[5] usually cannot handle complex raw data from, which some new features should be extracted by applying appropriate extractors for different problems. Thus, significant research interest in simultaneous representation learning and clustering the data arises.

Benefiting from the powerful unsupervised representation learning performance of deep learning, various clustering algorithms based on deep learning (deep clustering) have emerged [6]. Deep clustering algorithms usually consist of three essential components: deep neural network, network loss, and clustering loss. The deep neural network is the representation learning component that is employed to learn low-dimensional nonlinear embedding from the input data. The widely used network architectures include convolutional neural network (CNN) [7], [8], autoencoder (AE) [9], and variational AE (VAE) [10]. With the success of generative adversarial networks (GANs) [11], much work [12]–[14] has introduced the idea of GAN into representation learning and achieved good results. The objective function of deep clustering algorithms is generally a linear combination of unsupervised representation learning loss, here referred to as network loss $\mathcal{L}_R$, and a clustering oriented loss $\mathcal{L}_C$. They are formulated as

$$\mathcal{L} = \lambda \mathcal{L}_R + (1 - \lambda)\mathcal{L}_C \tag{1}$$

where λ is a hyperparameter between 0 and 1 that balances the impact of two loss functions. The network loss is independent of the clustering algorithm and usually enforces a desired constraint on the learned model. The network loss usually refers to the reconstruction loss of AE and VAE, or the adversarial loss of GAN, and is essential for the parameter initialization and avoiding collapsing clusters. Clustering loss is designed to encourage the model to learn cluster-friendly representations and complete data grouping.

Combining the neural network with traditional clustering algorithms is the most common and effective solution. Deep clustering network [15] utilizes an AE to learn representations that are amenable to the K-means algorithm. It pretrains the AE and then jointly optimizes the reconstruction loss and K-means loss with alternating cluster assignments. Joint unsupervised learning (JULE) [16] uses a CNN with agglomerative clustering loss to achieve impressive performance.

Manuscript received August 24, 2020; revised January 6, 2021; accepted March 25, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61772048, Grant U19B2039, Grant U1811463, Grant 61806014, and Grant 61632006; and in part by the Beijing Talents Project under Grant 2017A24. *(Corresponding authors: Yanfeng Sun; Baocai Yin.)*

Qiang Ji, Yanfeng Sun, Yongli Hu, and Baocai Yin are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: jiqiang64824@gmail.com; yfsun@bjut.edu.cn; huyongli@bjut.edu.cn; ybc@bjut.edu.cn).

Junbin Gao is with the Discipline of Business Analytics, The University of Sydney Business School, The University of Sydney, Sydney, NSW 2006, Australia (e-mail: junbin.gao@sydney.edu.au).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2021.3071275.

Digital Object Identifier 10.1109/TNNLS.2021.3071275

SpectralNet [17] learns a map that embeds the input into the eigenspace of their associated graph Laplacian matrix via a Siamese network and subsequently uses K-means to group them. Variational deep embedding (VaDE) [18] incorporates probabilistic clustering problems into the framework of VAE by imposing a Gaussian mixture distribution prior to the latent representation. StructAE [19] is a representative work that combines subspace clustering with deep learning for the first time. StructAE uses stacked AE to preserve the local and global subspace structure of the learned representations, which improves the limited representative capacity of the shallow models for subspace learning.

It is also a good idea to directly design a specific clustering loss according to the desired clustering assumption. Deep embedded clustering (DEC) [20] proposes a novel cluster assignment hardening loss that iteratively refines clusters with an auxiliary target distribution derived from the current soft cluster assignment. DEC is a pioneering work on deep clustering, and the proposed clustering loss is widely used in several related deep clustering models [21], [22]. Deep embedded regularized clustering (DEPICT) [22] consists of several tricks. It uses softmax layer stacked on top of convolutional autoencoder with a noisy encoder. It jointly optimizes reconstruction loss and cross entropy loss of softmax assignments and it's auxilliary assignments which leads to balanced cluster assignment loss. All the layers of the encoder and decoder also contribute to the reconstruction loss instead of just input and output layers. Inspired by the self-paced learning, deep adaptive clustering [23] transforms the clustering problem into a binary pairwise classification framework to judge whether pairs of data points belong to the same clusters. Information maximizing self-augmented training (IMSAT) [24] adopts the data augmentation technique to encourage the learned representations of augmented input to be close to those of the original input. Simultaneously, IMSAT maximizes the information-theoretic dependence between the input and its discrete representations. Invariant information clustering (IIC) [25] uses data augmentation to obtain a pair from each image and is simply to maximize the mutual information (MI) between the class assignments of each pair. Based on discovering a common invariance about different metrics used for clustering assignments, Peng *et al.* [26] propose a novel clustering method by minimizing the discrepancy between pairwise sample assignments for each data point. Li *et al.* [27] propose an end-to-end online clustering method that explicitly performs the instance- and cluster-level contrastive learning.

According to the above-mentioned summary, we can roughly categorize current deep clustering models into the following three categories based on the characteristics of network architecture and the nature of loss functions used.

The first category is the AE-based deep clustering algorithms that use AE as the basic framework. To achieve high performance steadily, most of the AE-based deep clustering algorithms [15], [22], [28] usually need to use a pretraining scheme, in which the network parameters are initialized with the reconstruction loss before the clustering loss is introduced. Although the reconstruction loss can help the model avoid degenerate solutions to a great extent, the symmetry of the encoder–decoder architecture severely limits the depth of the encoder for computational feasibility.

The second category is based on feed-forward networks trained only by specific clustering loss; thus, we refer to it as direct cluster optimization (DCO) [20], [21], [23]. DCO can easily use very deep neural networks or well-pretrained models to learn representations because no decoder is required to reconstruct the original input. However, compared with AE-based methods, as only is clustering loss used to optimize the model, the performance of the DCO method is more dependent on the robustness of clustering loss. When the clustering loss cannot effectively help the model avoid degenerate solutions, DCO suffers from the risk of obtaining a corrupted latent space. DEC [20] can be regarded as a representative DCO method, which jointly optimizes the cluster centroids $\mathbf{U}$ and the representations $\mathbf{Z}$ by backpropagation. Since DEC's clustering loss lacks effective constraints on the representations $\mathbf{Z}$, it is likely that both $\mathbf{Z}$ and $\mathbf{U}$ tend to the same solution to minimize the objective function during the training process. This situation means that although the clustering loss is sufficiently minimized, meaningless representations and cluster centroids are obtained. Improved DEC [29] finds the problem and solves it by introducing reconstruction loss throughout the training process. Another popular approach is to improve the robustness of clustering loss by introducing data augmentation technique [24], [25], [27], [30].

The third category is based on generative models, such as VAE and GAN, which can generate new samples while completing clustering. VAE-based algorithms [18], [31] have a good theoretical guarantee that makes the learned latent representations satisfy the desired distribution, but they inherit the same disadvantages of AE-based algorithms and suffer from high computational complexity. GAN-based algorithms [14], [32] are more flexible and diverse than VAE-based ones on generating images. However, mode collapse and slow convergence are the major obstacles to the applicability of GAN-based algorithms.

From the above-mentioned analysis, we conclude that a good deep clustering model should have three elements simultaneously: scalability of network capacity, robustness of loss function, and smoothness of latent space. In this article, we propose a novel deep clustering framework, called decoder-free VaDE for unsupervised clustering (DFVC). Inspired by the fact that the reconstruction error is strongly related to the MI [33], we prove that maximizing the MI between the input and its output minimizes the reconstruction error under ideal conditions in VAE. Moreover, Deep InfoMax (DIM) [34] provides a useful tool for us to estimate the MI of continuous variables. Thus, these facts provide theoretical feasibility for us to discard the decoder and to satisfy the scalability of network capacity. Different from DIM, we simultaneously estimate and maximize the MI between input data and the high-level representations that are composed of categorical variables and continuous variables. To make the model more robust and discriminative, we also incorporate the augmented data into the MI estimation framework by introducing the data augmentation technique. In order to ensure the smoothness

of the latent space, it is very effective to apply a certain smooth distribution to the learned representations. Gaussian distribution is widely used in representation learning due to its excellent mathematical properties [35]–[37]. However, a single Gaussian distribution excessively limits the expression capacity of the model and is not conducive to obtaining cluster-friendly representations. Thus we model the latent representation space with a Gaussian mixture distribution. The main contributions of this article can be summarized as follows.

1) We propose a novel decoder-free variational deep clustering framework that maximizes the MI between input data and the high-level representations instead of the decoder for reconstructing the original input.
2) We propose augmented MI (AMI) that extends DIM by introducing the MI between continuous representations extracted from independently augmented input and the MI between categorical representations. The proposed model achieves jointly learning representations and clustering assignments in an end-to-end manner.

The rest of this article is organized as follows. In Section II, we overview some related work on self-supervised representation learning and mutual-information estimation. In Section III, we propose a DFVC. Section IV shows the experimental results on the six challenging data sets. Finally, we draw a conclusion in Section V.

II. Related Works

A. Self-Supervised Representation Learning

A broad class of contemporary machine learning methods relies on manual labels as the only form of learning signal used during the training process. This overreliance on direct-semantic supervision often makes the models converge to brittle solutions because the data usually have much richer structures than what sparse labels can provide. Self-supervised learning (SSL) techniques are a promising class of representation learning methods that help us to exploit a variety of labels that come with the data for free. We can get supervision from the data itself by setting an appropriate self-supervised task (i.e., pretext task). An excellent pretext task is essential to obtain the latent representations that carry useful semantic or structural meanings and are beneficial to a variety of practical downstream tasks. SSL methods can be roughly divided into two categories: generative [38]–[40] and contrastive methods [41]–[43]. Contrastive SSL (CSSL) methods learn representations by discriminating or comparing between samples from different distributions and have led to great empirical success in computer vision tasks with unsupervised contrastive pretraining. CSSL differs from the traditional generative methods, which focus on reconstruction error in the pixel space to learn representations. Using pixel-level losses can lead to generative-based methods being overly focused on pixel-based details, rather than high-level latent characteristics. Moreover, pixel-based objectives often assume independence between each pixel, thereby reducing their ability to model correlations or complex structures.

More formally, given a set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ of N random samples, CSSL methods aim to learn an encoder $f_\theta(\cdot)$ such that $\mathbf{score}(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}^+)) \gg \mathbf{score}(f_\theta(\mathbf{x}), f_\theta(\mathbf{x}^-))$. We refer to $\mathbf{x}^+$ and $\mathbf{x}^-$ as a positive and negative sample, respectively. The score function is a metric that measures the similarity between two representations. $\mathbf{x}$ is commonly referred to as an anchor sample. We can construct a softmax classifier yielding a score function in favor of spatially or temporally congruent samples

$$\mathcal{L}_N = -\mathbb{E}_{\mathbf{X}}\left[\log \frac{\exp(f_\theta(\mathbf{x})^T f_\theta(\mathbf{x}^+))}{\sum_{i=1}^{N} \exp(f_\theta(\mathbf{x})^T f_\theta(\mathbf{x}_i))}\right]. \tag{2}$$

The denominator terms consist of one positive and $N-1$ negative samples. Here, we use the dot product as the score function. The loss function (2) is the familiar InfoNCE loss [41]. What is more, the InfoNCE objective is related to MI. Specifically, minimizing the InfoNCE loss maximizes a lower bound on the MI [44]. Based on noise-contrastive estimation principle [45], many CSSL methods are derived. DIM [34] learns representations of images by leveraging the local structure present in an image. The contrastive task behind DIM is to classify whether a pair of global and local representations are from the same image or not. Each local feature map has a limited receptive field. Thus, intuitively this means that global representations must capture information from all the different local regions.

DIM has been extended to other domains, such as graphs [46], and reinforcement learning environments [47]. A follow-up to DIM, augment multiscale DIM [48], uses standard data augmentation techniques as the set of transformations that a representation should be invariant to. Contrastive multiview coding [49] uses different views of the same image (depth, luminance, chrominance, surface normal, and semantic labels) as the set of transformations that the representation should be invariant to.

Contrastive predictive coding (CPC) [41] is a contrastive method that can be applied to any form of data that can be expressed in an ordered sequence. CPC learns representations by encoding information that is shared across data points multiple time steps apart, discarding local information. Momentum contrast [42] provides a framework of unsupervised learning visual representation as a dynamic dictionary look-up. The dictionary is structured as a large FIFO queue of encoded representations of data samples. SimCLR [43] proposes a simple framework for contrastive learning of visual representations. It learns representations for visual inputs by maximizing agreement between differently augmented views of the same sample via a contrastive loss in the latent space.

B. Mutual-Information Estimation

MI is a Shannon entropy-based measure that quantifies the amount of information obtained about one random variable through observing the other random variable. Inspired by the InfoMax principle [50], there has been a revival of representation learning approaches that advocate maximizing MI between the input and its representation. MI neural estimation [33] is an MI estimation technique of continuous variables, which is linearly scalable in dimensionality and in sample size, trainable through backpropagation, and strongly consistent. DIM [34] adopts a simple alternative based on the

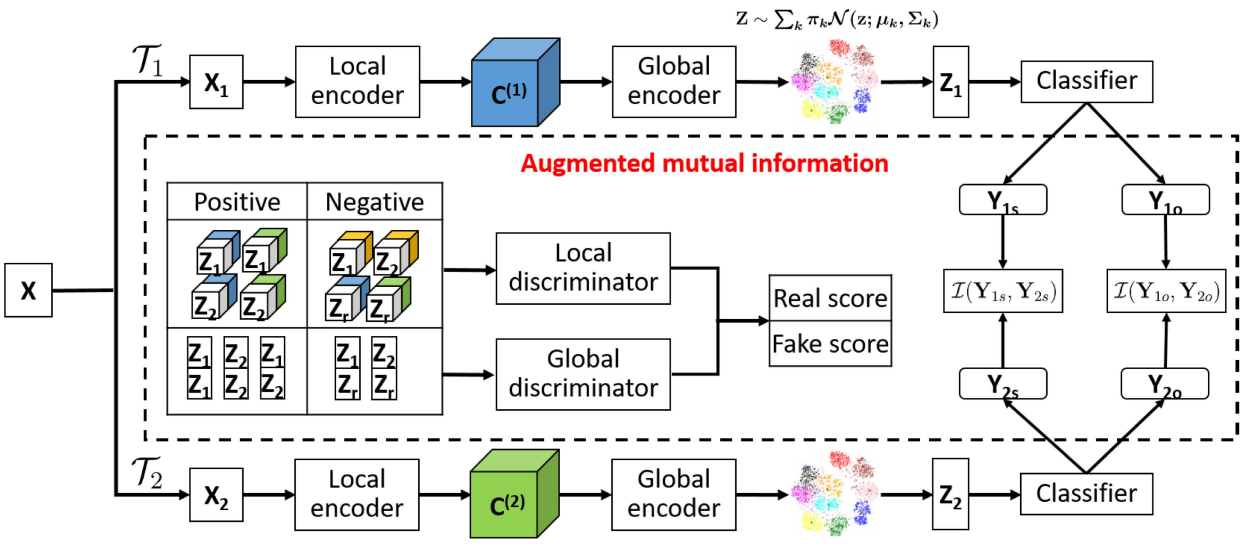


Fig. 1. Framework of the proposed DFVC model. We obtain a pair from one input by using two data augmentations and then use a shared deep neural network to learn the local, global, and categorical representations. The local and global discriminator networks are used to estimate the local and global MI among the input, local, and global continuous representations, respectively. Two softmax branches can be directly computed to obtain the MI between categorical representations and cluster assignments. To obtain a cluster-friendly latent space, we constrain the global representations to follow a Gaussian mixture distribution.

Jensen–Shannon divergence (JSD) that is an unbiased estimator and insensitive to the number of negative samples. Most significantly, DIM can leverage local structure in the input to improve the suitability of representations for classification. Bridle *et al.* [51] is the first to propose maximizing MI to learn probabilistic classifiers without supervision. However, they find that MI may be trivially optimized by a conditional model that classifies each data point into its own category. Regularized information maximization (RIM) [52] introduces a regularizing term that penalizes conditional models with complex decision boundaries in order to yield sensible clustering solutions. Inspired by RIM, IMSAT [24] combines MI constraint along with a self-augmentation training scheme to learn discrete representations of the input. Unsupervised clustering and segmentation functions are attainable by directly maximizing the MI between an image and its augmentations [25]. Deep comprehensive correlation mining (DCCM) [30] extends the instance-level MI to triplet-level and comes up with triplet MI loss to learn more discriminative features.

III. Proposed Method

A. Preliminary: Variational Autoencoder

The framework of VAEs provides a computationally efficient way for optimizing deep latent-variable models jointly with a corresponding inference model using stochastic gradient descent. Let $p_d(\mathbf{x})$ be the true unknown distribution and $\tilde{p}(\mathbf{x})$ be the empirical distributions defined by the samples in $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$. Consider a latent variable model with an observed variable $\mathbf{x} \in \mathcal{X}$ and a latent variable $\mathbf{z} \in \mathcal{Z}$. VAE learns stochastic mappings between an observed $\mathbf{x}$-space, whose empirical distribution $\tilde{p}(\mathbf{x})$ is typically complicated, and a latent $\mathbf{z}$-space, whose distribution can be relatively simple (e.g., Gaussian distribution). The joint distribution $p_\theta(\mathbf{x}, \mathbf{z})$ is often factorized as $p_\theta(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$, with a prior distribution over latent space $p(\mathbf{z})$, and a generative model $p_\theta(\mathbf{x}|\mathbf{z})$. The inference model $q_\phi(\mathbf{z}|\mathbf{x})$ approximates the true but intractable posterior $p(\mathbf{z}|\mathbf{x})$. Given the samples $\mathcal{X}$, the parameters can be estimated by maximizing the marginal log-likelihood

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_{\tilde{p}(\mathbf{x})}[\log p_\theta(\mathbf{x})] \\ &= \mathbb{E}_{\tilde{p}(\mathbf{x})}\left[\log \int_{\mathcal{Z}} p(\mathbf{z})\, p_\theta(\mathbf{x}|\mathbf{z}) d\mathbf{z}\right]. \end{aligned} \tag{3}$$

To avoid the difficult computation of the integral in (3), the idea behind VAE is to instead maximize the evidence lower bound (ELBO) of the log-likelihood with the reparameterization trick

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{\tilde{p}(\mathbf{x})}[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \text{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]. \tag{4}$$

We decompose the ELBO loss (4) into two terms and transform the maximization problem into a minimization problem. The final loss for VAE can be rewritten as

$$\begin{aligned}\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_{\tilde{p}(\mathbf{x})}[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z})]] \\ + \mathbb{E}_{\tilde{p}(\mathbf{x})}[\text{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]. \end{aligned} \tag{5}$$

The first term in (5) is the reconstruction error and the KL-divergence term can be interpreted as a regularization term, where the variational posterior $q_\phi(\mathbf{z}|\mathbf{x})$ should be matched to the prior distribution $p(\mathbf{z})$.

B. Reconstruction and Mutual Information

In generative models that rely on reconstruction (e.g., denoising AE [38], VAE [10], and adversarial AE [36]), the reconstruction error can be related to the MI as follows:

$$\mathcal{I}(\mathbf{x}, \mathbf{z}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\mathbf{z}) \geq \mathcal{H}(\mathbf{x}) - \mathcal{R}(\mathbf{x}|\mathbf{z}) \tag{6}$$

where $\mathbf{x}$ and $\mathbf{z}$ denote the input and its output of an encoder which is applied to inputs sampled from some source distribution. $\mathcal{R}(\mathbf{x}|\mathbf{z})$ denotes the expected reconstruction error of $\mathbf{x}$ given the latent variable $\mathbf{z}$. $\mathcal{H}(\mathbf{x})$ and $\mathcal{H}(\mathbf{x}|\mathbf{z})$, respectively, denote the marginal and conditional entropy of $\mathbf{x}$ in the distribution formed by applying the encoder to inputs sampled from the source distribution. Thus, in typical settings, models with reconstruction-type objectives provide some guarantees on the amount of information encoded in their intermediate representations.

From the perspective of the joint distribution $q_\phi(\mathbf{x}, \mathbf{z}) = \tilde{p}(\mathbf{x})q_\phi(\mathbf{z}|\mathbf{x})$, the KL divergence of $q_\phi(\mathbf{x}, \mathbf{z})$ from $p_\theta(\mathbf{x}, \mathbf{z})$ can be written as the negative ELBO (5), plus a constant

$$\begin{aligned}&\mathrm{KL}(q_\phi(\mathbf{x}, \mathbf{z})||p_\theta(\mathbf{x}, \mathbf{z}))\\&\quad= \mathbb{E}_{\tilde{p}(\mathbf{x})}[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}|\mathbf{z})]] + \mathbb{E}_{\tilde{p}(\mathbf{x})}[\log \tilde{p}(\mathbf{x})]\\&\qquad+ \mathbb{E}_{\tilde{p}(\mathbf{x})}[\mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]\\&\quad= \mathcal{L}_{\theta,\phi}(\mathbf{x}) + \text{constant}\end{aligned} \tag{7}$$

where constant $= \mathbb{E}_{\tilde{p}(\mathbf{x})}[\log \tilde{p}(\mathbf{x})] = -\mathcal{H}(\tilde{p}(\mathbf{x}))$. Thus, we can conclude that minimizing the VAE loss (5) is equivalent to minimize the KL-divergence of joint distribution $\mathrm{KL}(q_\phi(\mathbf{x}, \mathbf{z})||p_\theta(\mathbf{x}, \mathbf{z}))$. One goal of VAE is to do inference and to learn a good generative model. Theoretically, reconstruction is one desirable property of a model that does both inference and generation. However, VAE may reconstruct less faithfully than desired in practice. Due to the new insight of VAE, we demonstrate the relationship between reconstruction and MI.

Corollary 1: If ϕ^* and θ^* are the optimal solutions to the VAE loss (7). Subsequently, maximizing the MI $\mathcal{I}(\mathbf{x}, \mathbf{z})$ minimizes the expected reconstruction error.

Proof: MI can be decomposed into an entropy and a conditional entropy term: $\mathcal{I}(\mathbf{x}, \mathbf{z}) = \mathcal{H}(\mathbf{x}) - \mathcal{H}(\mathbf{x}|\mathbf{z})$. Since observed input $\mathbf{x}$ comes from an unknown distribution $\tilde{p}(\mathbf{x})$ on which θ has no influence, $\mathcal{H}(\mathbf{x})$ is an unknown constant. Thus, the infomax principle reduces to

$$\begin{aligned}\arg\max_\phi \mathcal{I}(\mathbf{x}, \mathbf{z}) &= -\arg\max_\phi \mathcal{H}(\mathbf{x}|\mathbf{z})\\&= \arg\max_\phi \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}|\mathbf{z})]\end{aligned} \tag{8}$$

where $q_\phi(\mathbf{x}|\mathbf{z})$ can be regarded as the optimal decoder. Given a parametric distribution $p_\theta(\mathbf{x}|\mathbf{z})$ parameterized by θ, we will have

$$\mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \le \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[\log q(\mathbf{x}|\mathbf{z})] \tag{9}$$

as can easily be obtained according to the property that $\mathrm{KL}(q(\mathbf{x}|\mathbf{z})||p_\theta(\mathbf{x}|\mathbf{z})) \ge 0$. Thus, we obtain the following optimization problem:

$$\min_{\phi,\theta} \mathbb{E}_{q_\phi(\mathbf{x},\mathbf{z})}[-\log p_\theta(\mathbf{x}|\mathbf{z})]. \tag{10}$$

From the VAE loss (5) and (9), we can conclude that minimizing the reconstruction error corresponds to maximizing a lower bound on the MI $\mathcal{I}(\mathbf{x}, \mathbf{z})$. We will end up maximizing the exact MI provided $\exists\theta^*$, s.t. $q(\mathbf{x}|\mathbf{z}) = p_{\theta^*}(\mathbf{x}|\mathbf{z})$. Since $p_\theta(\mathbf{x}|\mathbf{z})$ and $q_\phi(\mathbf{z}|\mathbf{x})$ both be parameterized using deep neural networks, which have very powerful nonlinear mapping ability. Theoretically, the encoder $q_\phi(\mathbf{z}|\mathbf{x})$ can push the posterior distribution of $\mathbf{z}$ to the prior distribution $p(\mathbf{z})$ infinitely and the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ is capable of perfectly reconstructing the original input $\mathbf{x}$. Thus, maximizing the MI $\mathcal{I}(\mathbf{x}, \mathbf{z})$ minimizes the expected reconstruction error when the joint distributions $q_\phi(\mathbf{x}, \mathbf{z})$ and $p_\theta(\mathbf{x}, \mathbf{z})$ are matched, i.e., ϕ^* and θ^* are the optimal solutions to the VAE loss (7). ■

According to **Corollary** 1, the VAE loss (7) that need to be optimized can be rewritten as follows:

$$\mathrm{KL}(q_\phi(\mathbf{x}, \mathbf{z})||p_\theta(\mathbf{x}, \mathbf{z})) = \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) - \mathcal{I}(\mathbf{x}, \mathbf{z}). \tag{11}$$

Inspired by **Corollary** 1, we can discard the decoder if we have a way to directly calculate or estimate the MI between $\mathbf{x}$ and $\mathbf{z}$. Because obtaining the discriminative or cluster-friendly representation is the most critical, rather than minimizing reconstruction error for unsupervised learning, especially clustering. The decoder limits the depth of the neural network and wastes a lot of hardware resources. Furthermore, in practice, we usually make the decoder follow the Gaussian distribution that will reduce the fitting ability of the decoder. Thus, how to directly calculate or estimate MI may become the key to unsupervised representation learning.

C. Decoder-Free Variational Autoencoder

According to **Corollary** 1, we know that perfect reconstruction can indeed help the model extract useful representation with exclusive information. However, we have no better choice than the Gaussian distribution for reconstruction in terms of practical feasibility and universal applicability, which makes it difficult for the decoder $p_\theta(\mathbf{x}|\mathbf{z})$ to fit the input infinitely. Even if the model can achieve perfect reconstruction, maximizing MI between the complete input and the encoder output (i.e., global MI) is often insufficient for learning useful representations [34]. The average MI between the representation and local regions of the input (i.e., local MI) plays a stronger role in improving the representation's quality than the global MI. Inspired by the DIM, we propose a decoder-free VAE by directly estimating the global and local MI instead of reconstructing the input. Unlike the Deep DIM, we provide an end-to-end training mode by variational divergence minimization (VDM) instead of using adversarial training. Especially, our model is easy to impose more flexible statistical constraints (e.g., Gaussian mixture distribution) onto learned representations for clustering tasks.

For the objective function (11), the first term is easily calculated by giving the prior distribution $q(\mathbf{z})$. The essence of the second term is to increase the distance between the distribution $q_\phi(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})$ and the distribution $\tilde{p}(\mathbf{x})\tilde{p}(\mathbf{z})$. Considering that KL divergence has no upper bound theoretically, we use the JSD for stable optimization. Nevertheless, we still cannot calculate the JSD because $p(\mathbf{z})$ is unknown. Benefit from the VDM method [53], the JSD can be estimated by

$$\begin{aligned}\mathrm{JS}(Q, P) = \max_T (&\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log \sigma(T(\mathbf{x}))]\\&+ \mathbb{E}_{\mathbf{x}\sim q(\mathbf{x})}[\log(1 - \sigma(T(\mathbf{x})))\end{aligned} \tag{12}$$

where $\sigma(T(\mathbf{x}))$ is a discriminator network. Substituting $q_\phi(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})$ and $\tilde{p}(\mathbf{x})\tilde{p}(\mathbf{z})$ into (12), the estimation of global

MI can be obtained by maximizing the following loss function:

$$\mathcal{L}_{\mathrm{GMI}}(\phi, \omega; \mathbf{x}) = \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim q_\phi(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})}[\log \sigma(T_\omega(\mathbf{x}, \mathbf{z}))] + \mathbb{E}_{(\mathbf{x},\mathbf{z})\sim \tilde{p}(\mathbf{x})\tilde{p}(\mathbf{z})}[\log(1 - \sigma(T_\omega(\mathbf{x}, \mathbf{z}))]. \quad (13)$$

The essence of JSD estimation (13) is negative sample estimation. $\mathbf{x}$ and its corresponding $\mathbf{z}$ are regarded as a positive sample pair, $\mathbf{x}$ and randomly selected $\mathbf{z}$ are regarded as negative samples, and then the likelihood function is maximized. Referring to DIM, we also take into account local MI. Let $\{C_{ij}(\mathbf{x})|i = 1, 2, \ldots, h;\ j = 1, 2, \ldots, w\}$ be a feature map of an intermediate convolutional layer of $\mathbf{x}$. We measure the local MI by estimating the JSD of $C_{ij}(\mathbf{x})$ and $\mathbf{z}_\mathbf{x}$. Similar to the global MI estimation, we concatenate $C_{ij}(\mathbf{x})$ and $\mathbf{z}_\mathbf{x}$ to get $[C_{ij}(\mathbf{x}), \mathbf{z}_\mathbf{x}]$, which is equivalent to getting a larger feature map. The estimation of local MI can be obtained by maximizing the following loss function:

$$\mathcal{L}_{\mathrm{LMI}}(\phi, \psi; \mathbf{x}) = \sum_{i,j}(\mathbb{E}_{(\mathbf{x},\mathbf{z})\sim q_\phi(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})}[\log \sigma(T_\psi(C_{ij}, \mathbf{z}))]) + \sum_{i,j}(\mathbb{E}_{(\mathbf{x},\mathbf{z})\sim \tilde{p}(\mathbf{x})\tilde{p}(\mathbf{z})}[\log(1 - \sigma(T_\psi(C_{ij}, \mathbf{z}))]). \quad (14)$$

Thus, we have the final estimation of MI $\mathcal{I}(\mathbf{x}, \mathbf{z})$ as follows:

$$\mathcal{I}(\mathbf{x}, \mathbf{z}) = \mathcal{L}_{\mathrm{GMI}}(\phi, \omega; \mathbf{x}) + \mathcal{L}_{\mathrm{LMI}}(\phi, \psi; \mathbf{x}). \quad (15)$$

Bringing the MI estimation (15) into the VAE loss (11), we can obtain a decoder-free VAE. In order to obtain abundant negative samples, our model extends DIM by maximizing MI between the augmented input $\mathbf{x}'$ and its latent representation $\mathbf{z}'$. According to the definition of $\mathcal{I}(\mathbf{x}, \mathbf{z})$ (15), we can simply obtain $\mathcal{I}(\mathbf{x}', \mathbf{z}')$, which can be denoted as

$$\mathcal{I}(\mathbf{x}', \mathbf{z}') = \mathcal{L}_{\mathrm{GMI}}(\phi, \omega; \mathbf{x}') + \mathcal{L}_{\mathrm{LMI}}(\phi, \psi; \mathbf{x}'). \quad (16)$$

However, it is not sufficient to only consider the MI between the input and its representation to obtain cluster-friendly representations. Because $\mathcal{I}(\mathbf{x}, \mathbf{z})$ or $\mathcal{I}(\mathbf{x}', \mathbf{z}')$ focuses on investigating the unique information of each input, which tends to weaken the correlation between samples. We also need to explore the invariant information among samples, which is very important for obtaining cluster-friendly representations. It is well known that the input $\mathbf{x}$ and its transformed version $\mathbf{x}'$ essentially have the same semantic information. This reliable prior information is very helpful for the model to discover the similarity among samples. Thus, we also incorporate the MI $\mathcal{I}(\mathbf{z}, \mathbf{z}')$ into the DIM MI estimation framework, where $\mathbf{z}$ is the latent representation of the input $\mathbf{x}$ and $\mathbf{z}'$ is the latent representation of the augmented input $\mathbf{x}'$. The loss of MI estimator $\mathcal{I}(\mathbf{z}, \mathbf{z}')$ on global/local pairs can also be used with DIM by maximizing

$$\mathcal{L}_{\mathrm{GMI}}(\phi, \omega; \mathbf{x}, \mathbf{x}') = \mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim q_\phi(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{z}'|\mathbf{x}')}[\log \sigma(T_\omega(\mathbf{z}, \mathbf{z}'))] + \mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim \tilde{p}(\mathbf{z})\tilde{p}(\mathbf{z}')}[\log(1 - \sigma(T_\omega(\mathbf{z}, \mathbf{z}'))] \quad (17)$$

$$\mathcal{L}_{\mathrm{LMI}}(\phi, \psi; \mathbf{x}, \mathbf{x}') = \sum_{i,j}(\mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim q_\phi(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{z}'|\mathbf{x}')}[\log \sigma(T_\psi(C'_{ij}, \mathbf{z}))]) + \sum_{i,j}(\mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim q_\phi(\mathbf{z}|\mathbf{x})q_\phi(\mathbf{z}'|\mathbf{x}')}[\log \sigma(T_\psi(C_{ij}, \mathbf{z}'))]) + \sum_{i,j}(\mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim \tilde{p}(\mathbf{z})\tilde{p}(\mathbf{z}')}[\log(1 - \sigma(T_\psi(C'_{ij}, \mathbf{z}))]) + \sum_{i,j}(\mathbb{E}_{(\mathbf{z},\mathbf{z}')\sim \tilde{p}(\mathbf{z})\tilde{p}(\mathbf{z}')}[\log(1 - \sigma(T_\psi(C_{ij}, \mathbf{z}'))]) \quad (18)$$

$$\mathcal{I}(\mathbf{z}, \mathbf{z}') = \mathcal{L}_{\mathrm{GMI}}(\phi, \omega; \mathbf{x}, \mathbf{x}') + \mathcal{L}_{\mathrm{LMI}}(\phi, \psi; \mathbf{x}, \mathbf{x}'). \quad (19)$$

We denote $\mathcal{L}_{\mathrm{DMI}}(\phi, \omega, \psi)$ as the MI loss that is composed of $\mathcal{I}(\mathbf{x}, \mathbf{z})$, $\mathcal{I}(\mathbf{x}', \mathbf{z}')$, and $\mathcal{I}(\mathbf{z}, \mathbf{z}')$

$$\mathcal{L}_{\mathrm{DMI}}(\phi, \omega, \psi) = \mathcal{I}(\mathbf{x}, \mathbf{z}) + \mathcal{I}(\mathbf{x}', \mathbf{z}') + \mathcal{I}(\mathbf{z}, \mathbf{z}'). \quad (20)$$

D. Variational Deep Embedding Clustering

VAEs usually model the prior distribution of the latent representations as a single-multivariate Gaussian. In this article, we are focusing on clustering task—a single Gaussian distribution is obviously not the best choice. Naturally, we choose the Gaussian mixture distribution with clustering characteristics as the prior distribution of the latent representation $\mathbf{z}$. Reviewing the loss function (5) for VAE, we redefine the latent variable as $(\mathbf{z}, y)$, where $\mathbf{z}$ is still a continuous variable representing the encoding feature and y is a discrete variable representing the category. To replace $\mathbf{z}$ with $(\mathbf{z}, y)$, the loss function (5) can be rewritten as

$$\mathrm{KL}(q_\phi(\mathbf{x}, \mathbf{z}, \mathbf{y})||p_\theta(\mathbf{x}, \mathbf{z}, \mathbf{y})) = \sum_y \iint q_\phi(\mathbf{z}, y|\mathbf{x})\tilde{p}(\mathbf{x}) \log \frac{q_\phi(\mathbf{z}, y|\mathbf{x})\tilde{p}(\mathbf{x})}{p_\theta(\mathbf{x}|\mathbf{z}, y)p(\mathbf{z}, y)} d\mathbf{z} d\mathbf{x}. \quad (21)$$

To optimize the loss function (21), we need to design a feasible solution for the model generative process and the inference process. The generative process is defined as

$$p(\mathbf{z}, y) = p(\mathbf{z}|y)p(y) \quad (22)$$

$$p_\theta(\mathbf{x}|\mathbf{z}, y) = p_\theta(\mathbf{x}|\mathbf{z}) \quad (23)$$

$$p(y_k = 1) = \frac{1}{k} \quad (24)$$

$$p(\mathbf{z}|y_k = 1) = \mathcal{N}(\boldsymbol{\mu}_k, \mathbf{I}) \quad (25)$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \begin{cases} \mathbf{Ber}(\boldsymbol{\mu}_x) & \mathbf{x} \text{ is binary} \\ \mathcal{N}(\boldsymbol{\mu}_x, \lambda\mathbf{I}) & \mathbf{x} \text{ is real-valued} \end{cases} \quad (26)$$

where $\boldsymbol{\mu}_y$ and $\boldsymbol{\sigma}_k{}^2$ denote the mean and variance of the kth Gaussian component, respectively, and $\boldsymbol{\mu}_x = g(\mathbf{z}_\mathbf{x}; \phi)$ denotes the decoder output. The inference process is defined as

$$q_\phi(\mathbf{z}, y|\mathbf{x}) = q_{\phi_2}(y|\mathbf{z})q_{\phi_1}(\mathbf{z}|\mathbf{x}), \mathbf{x} \sim \tilde{p}(\mathbf{x}) \quad (27)$$

$$q_{\phi_1}(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z{}^2\mathbf{I}) \quad (28)$$

$$q_{\phi_2}(y|\mathbf{z}) = \mathbf{Multinomial}(\boldsymbol{\pi}) \quad (29)$$

$$[\boldsymbol{\mu}_z, \log \boldsymbol{\sigma}_z^2] = f_1(\mathbf{x}; \phi_1) \quad (30)$$

$$\boldsymbol{\pi} = f_2(\mathbf{z}; \phi_2) \quad (31)$$

where f_1 and f_2 denote the encoder network for learning the representation distributions and the classifier for computing the probability of which Gaussian component the learned representations belong to, respectively. According to

the above-mentioned definition, the loss function (21) can be rewritten as

$$\begin{aligned}&\mathrm{KL}\big(q_\phi(\mathbf{x},\mathbf{z},\mathbf{y})||\,p_\theta(\mathbf{x},\mathbf{z},\mathbf{y})\big)\\&=\sum_y\iint q_{\phi_2}(y|\mathbf{z})q_{\phi_1}(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})\ln\frac{q_{\phi_2}(y|\mathbf{z})q_{\phi_1}(\mathbf{z}|\mathbf{x})\tilde{p}(\mathbf{x})}{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}|y)p(y)}d\mathbf{z}d\mathbf{x}.\end{aligned}\tag{32}$$

Based on **Corollary** 1, the loss function (32) can be rewritten as

$$\begin{aligned}\mathcal{L}&=-\mathcal{I}(\mathbf{x},\mathbf{z})+\mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}\left[\sum_y q_{\phi_2}(y|\mathbf{z})\log\frac{q_{\phi_1}(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|y)}\right]\\&\quad+\mathbb{E}_{\mathbf{x}\sim\tilde{p}(\mathbf{x})}[\mathrm{KL}(q_{\phi_2}(y|\mathbf{z})\|\,p(y))],\quad\mathbf{z}\sim q_{\phi_1}(\mathbf{z}|\mathbf{x})\\&=-\mathcal{I}(\mathbf{x},\mathbf{z})+\mathcal{L}_{\mathrm{VEC}}+\mathcal{L}_{\mathrm{REG}}\end{aligned}\tag{33}$$

where

$$\begin{aligned}q_{\phi_1}(\mathbf{z}|\mathbf{x})&=\frac{1}{\prod_{i=1}^d\sqrt{2\pi{\sigma_z}^2(i)}}\exp\left\{-\frac{1}{2}\left\|\frac{\mathbf{z}-\boldsymbol{\mu}_z}{\boldsymbol{\sigma}_z}\right\|^2\right\}\\p(\mathbf{z}|y)&=\frac{1}{(2\pi)^{d/2}}\exp\left\{-\frac{1}{2}\|\mathbf{z}-\boldsymbol{\mu}_{\boldsymbol{k}}\|^2\right\}\end{aligned}\tag{34}$$

$$\begin{aligned}&\log\frac{q_{\phi_1}(\mathbf{z}|\mathbf{x})}{p(\mathbf{z}|y)}\\&\quad=-\frac{1}{2}\sum_{i=1}^d\log{\sigma_z}^2(i)-\frac{1}{2}\left\|\frac{\mathbf{z}-\boldsymbol{\mu}_z}{\boldsymbol{\sigma}_z}\right\|^2+\frac{1}{2}\|\mathbf{z}-\boldsymbol{\mu}_{\boldsymbol{k}}\|^2\\&\quad\approx-\frac{1}{2}\sum_{i=1}^d\log{\sigma_z}^2(i)+\frac{1}{2}\|\mathbf{z}-\boldsymbol{\mu}_{\boldsymbol{k}}\|^2\end{aligned}\tag{35}$$

$$\begin{aligned}&\mathrm{KL}(q_{\phi_2}(y|\mathbf{z})\|\,p(y))\\&\quad=\sum_y q_{\phi_2}(y|\mathbf{z})\log\frac{q_{\phi_2}(y|\mathbf{z})}{p(y)}\\&\quad=\sum_y q_{\phi_2}(y|\mathbf{z})\log q_{\phi_2}(y|\mathbf{z})-\sum_y q_{\phi_2}(y|\mathbf{z})\log p(y)\\&\quad\approx\sum_y q_{\phi_2}(y|\mathbf{z})\log q_{\phi_2}(y|\mathbf{z})+\text{constant}.\end{aligned}\tag{36}$$

The loss function (33) has a very intuitive meaning in its form. $\mathcal{I}(\mathbf{x},\mathbf{z})$ denotes the MI between the sample $\mathbf{x}$ and the representation $\mathbf{z}$. In practice, $\mathcal{I}(\mathbf{x},\mathbf{z})$ includes $\mathcal{L}_{\mathrm{GMI}}$ and $\mathcal{L}_{\mathrm{LMI}}$. The second term is to hope that $\mathbf{z}$ can best fit the exclusive Gaussian component of a certain category. The third term is a regularizer for avoiding degenerate solutions. Therefore, we obtain a cluster-friendly loss function by imposing a Gaussian mixture distribution onto the $\mathbf{z}$.

E. Augmented Mutual Information

Although the objective function (33) can already achieve good performance for clustering tasks, the prior information between $\mathbf{x}$ and $\mathbf{x}'$ is not fully utilized for the categorical variable y. Furthermore, the above-mentioned MI $\mathcal{I}(\mathbf{x},\mathbf{z})$ indeed extracts the discriminative representation $\mathbf{z}$ highly related to the input $\mathbf{x}$. It is easy to weaken the correlation between the samples from the same category due to the randomness of negative sample estimation. In this section, we propose a novel learning strategy that encourages the model to discover the discriminative representations highly relevant to input samples from the same category. It is difficult to find the samples from the same category for unsupervised learning. For this issue, we use the online data augmentation technique to generate positive pairs consisting of input $\mathbf{x}$ and its randomly perturbed version $\mathbf{x}'=g(\mathbf{x})$. Let $(\mathbf{x},\mathbf{x}')$ be a positive pair from a joint probability distribution $P(\mathbf{x},\mathbf{x}')$. Since we are focusing on clustering, we hope that the model can finally give a probabilistic cluster assignment result. Inspired by IIC [25], to discover the common information between $\mathbf{x}$ and $\mathbf{x}'$, we can simply maximize MI between probabilistic output parameterized by a neural network $f:\mathcal{X}\to\mathcal{Y}$

$$\mathcal{L}_{\mathrm{CMI}}=\max_\phi I\left(f_\phi(\mathbf{x}),\,f_\phi(\mathbf{x}')\right)\tag{37}$$

where ϕ consists of ϕ_1 and ϕ_2. Different from the MI loss function $\mathcal{L}_{\mathrm{DMI}}$ mentioned earlier, $I(f_\phi(\mathbf{x}),f_\phi(\mathbf{x}'))$ can be directly computed without variational estimating since $y=f_\phi(\mathbf{x})\in[0,1]^K$ can be interpreted as the distribution of a discrete random variable y over K category. Consider now a pair of such cluster assignment variables (y,y') for the inputs $(\mathbf{x},\mathbf{x}')$. Their conditional joint distribution is given by $P(y=k,y'=k|\mathbf{x},\mathbf{x}')=f_\phi^k(\mathbf{x})\cdot f_\phi^k(\mathbf{x}')$. After marginalization over the data batch, the joint probability distribution is given by the $K\times K$ matrix $\mathbf{P}$, where each element at row k and column k' constitutes $\mathbf{P}_{kk'}=P(y=k,y'=k')$

$$\mathbf{P}=\frac{1}{n}\sum_{i=1}^n f_\phi(\mathbf{x}_i)\cdot f_\phi\left(\mathbf{x}'_i\right)^T.\tag{38}$$

The marginals $\mathbf{P}_k$ and $\mathbf{P}_{k'}$ can be obtained by summing over the rows and columns of this matrix $\mathbf{P}$. Thus, we have the marginals $\mathbf{P}_k$ and $\mathbf{P}_{k'}$ can be obtained by summing over the rows and columns of this matrix $\mathbf{P}$. Thus, we have

$$I(y,y')=\sum_{k=1}^K\sum_{k'=1}^K\mathbf{P}_{kk'}\cdot\log\frac{\mathbf{P}_{kk'}}{\mathbf{P}_k\cdot\mathbf{P}_{k'}}\tag{39}$$

where $\mathbf{P}$ is symmetrized using $\mathbf{P}=(\mathbf{P}+\mathbf{P}^T)/2$. In this article, we propose the AMI which combines $\mathcal{L}_{\mathrm{DMI}}$ and $\mathcal{L}_{\mathrm{CMI}}$ to discover the shared and distinguished information contained in the data. We denote the AMI by $\mathcal{L}_{\mathrm{AMI}}=\mathcal{L}_{\mathrm{DMI}}+\alpha\mathcal{L}_{\mathrm{CMI}}$. Based on the above-mentioned analysis and exploration, the final loss function is summarized as

$$\mathcal{L}_{\mathrm{DFVC}}=-\mathcal{L}_{\mathrm{AMI}}+\beta\mathcal{L}_{\mathrm{VEC}}+\gamma\,\mathcal{L}_{\mathrm{REG}}\tag{40}$$

where α, β, and γ are constants to balance the contributions of different terms. We summarize the overall training process in **Algorithm** 1. Fig. 1 shows the framework of the proposed DFVC model.

IV. Experiments and Analysis

In this section, we conduct some experiments to verify the effectiveness of DFVC by comparing it against other state-of-the-art algorithms. After that, we conduct more ablation studies by controlling several influence factors. Finally, we do a series of analysis experiments to verify the effectiveness of the unified model training framework. All codes are implemented on a PC machine installed in a 64-bit operating system with two Nvidia GeForce GTX 1080 Ti GPUs with 11-GB Video memory. The proposed model and all compared

Algorithm 1 DFVC

Input: Unlabelled dataset $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$
Parameter: Class number K, α, β and γ
Output: Cluster assignment

```
Initialize the neural network parameters φ1, φ2 randomly;
while epoch ⩽ Maxiter do
  for each minibatch x_b in X do
    Generate x' via data augmentation;
    Computing I(x, z) by Eq. (15)
    Computing I(x', z') by Eq. (16)
    Computing I(z, z') by Eq. (19)
    Computing L_DMI by Eq. (20);
    Computing L_VEC and L_REG by Eq. (33);
    Computing L_CMI by Eq. (39);
    Computing L_DFVC by Eq. (40);
    Update model parameters by backpropagation;
  end for
end while
return cluster assignment
```

TABLE I
PARAMETER SETTING OF THE DIFFERENT DATA SETS

Dataset	α	β	γ
MNIST	2×10^{0}	5×10^{-1}	1×10^{-1}
CIFAR-10	3×10^{0}	2×10^{-1}	2×10^{-1}
CIFAR-100	3×10^{0}	2×10^{-1}	2×10^{-1}
STL-10	2×10^{0}	3×10^{-1}	2×10^{-1}
ImageNet-10	2×10^{0}	2×10^{-1}	2×10^{-2}
ImageNet-dog	2×10^{0}	2×10^{-1}	2×10^{-2}

TABLE II
STATISTICS OF DIFFERENT DATA SETS

Dataset	Images	Image size	Clusters
MNIST	70000	28×28	10
CIFAR-10	60000	$32 \times 32 \times 3$	10
CIFAR-100	60000	$32 \times 32 \times 3$	20
STL-10	13000	$96 \times 96 \times 3$	10
ImageNet-10	13000	$96 \times 96 \times 3$	10
Imagenet-dog	19500	$96 \times 96 \times 3$	15

models are coded in Python 3.6. The code is available at https://github.com/jqwylb/deep-clustering.

A. Data Sets

We select six challenging image data sets for deep unsupervised learning and clustering, including the MNIST [54], CIFAR-10 [55], CIFAR-100 [55], STL-10 [56], ImageNet-10 [57], and Imagenet-dog [57]. We adopt the same setting as that in [23], where the training and validation data of each data set are jointly utilized, and the 20 superclasses are considered for the CIFAR-100 data set in experiments. For MNIST, we use the full data (train and test) consisting of ten handwritten digits, total 70000 samples. For ImageNet-10 and Imagenet-dog, we randomly choose ten and 15 kinds of dog images from the ImageNet data set and resize these images to $96 \times 96 \times 3$ as the same in [23] for fairness. For evaluating the performance of clustering, we adopt three commonly used metrics, including normalized MI, accuracy, and adjusted rand index.

B. Experimental Settings

The network architecture used in our model is the Resnet18 except for the MNIST data set. For MNIST, we set four convolutional layers and three pooling layers, followed by two fully connected layers. All network parameters are randomly initialized with HeNormal initializer [58]. Batch normalization [59] and ReLU activation are used on all hidden layers. ReLU activation is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. The reason we use ReLU activation is that it overcomes the vanishing gradient problem, allowing models to learn faster and perform better. For the local MI discriminator network, we use a similar setting as in DIM [34]. For the global MI objective, we first encode the input into a feature map $\mathbf{C}$, which, in this case, is the output of the last convolutional layer. We then encode the feature map further using fully connected layers to get the global representation $\mathbf{z}$. We directly use $(\mathbf{z}, \mathbf{z})$ as the input of the global discriminator, which can be understood as the global discriminator contains a shared global and local encoder. For the local MI objective, we concatenate the global representation with the feature map at every location. A 1×1 convolutional discriminator is then used to score the pair. The last layer of our model contains two-headed softmax branches, which are a standard clustering head and an auxiliary overclustering head. The standard clustering head is used to output clustering assignment, and its number of nodes is equal to the number of ground-truth classes. The overclustering head is only used for auxiliary learning, and we always set its number of nodes to twice the number of ground-truth classes. We adopt the Adam optimizer with $lr = 1e-4$. The small perturbations used in the experiments include rotation, shift, rescale, color adjustment, and so on. The setting of the hyperparameters α, β, and γ, which are relatively stable within a certain range, are shown in Table I. We summarize the statistics of these data sets in Table II.

C. Compared Methods

In the experiment, we adopt both traditional methods and deep learning-based methods, including K-means [1], SC [2], AE [9], GAN [60], deconvolutional networks [61], DIM [34], DEPICT [22], VAE [10], DEC [20], JULE [16], deep adaptive image clustering [23], DCCM [30], IIC [25], and partition confidence maximization [62].

D. Results and Analysis

1) Comparisons With State-of-the-Art Methods: In Table III, we report the quantitative results of these clustering algorithms on the experimental data sets. In most cases, we can intuitively see that DFVC has achieved better performance compared with other state-of-the-art clustering algorithms on all three evaluation metrics. For the ImageNet-10 data set, although the clustering performance of DFVC is not the best, it is still very competitive to PICA and surpasses other competitors by

TABLE III

CLUSTERING PERFORMANCE OF DIFFERENT METHODS ON SIX CHALLENGING DATA SETS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Dataset	MNIST			CIFAR-10			CIFAR-100			STL-10			ImageNet-10			Imagenet-dog		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI	NMI	ACC	ARI
Kmeans	0.501	0.572	0.365	0.087	0.229	0.049	0.084	0.130	0.028	0.125	0.192	0.061	0.119	0.241	0.057	0.055	0.105	0.020
SC	0.662	0.695	0.521	0.103	0.247	0.085	0.090	0.136	0.022	0.098	0.159	0.048	0.151	0.274	0.076	0.038	0.111	0.013
AE	0.725	0.812	0.613	0.239	0.314	0.169	0.100	0.165	0.048	0.250	0.303	0.161	0.210	0.317	0.152	0.104	0.185	0.073
DIM	0.843	0.892	0.827	0.252	0.374	0.191	0.133	0.192	0.098	0.269	0.343	0.182	0.292	0.354	0.174	0.124	0.206	0.104
JULE	0.913	0.964	0.927	0.192	0.272	0.138	0.103	0.137	0.033	0.182	0.277	0.164	0.175	0.300	0.138	0.054	0.138	0.028
DEC	0.771	0.843	0.741	0.257	0.301	0.161	0.136	0.185	0.050	0.276	0.359	0.186	0.282	0.381	0.203	0.122	0.195	0.079
VAE	0.876	0.945	0.049	0.245	0.291	0.167	0.108	0.152	0.040	0.200	0.282	0.146	0.193	0.334	0.168	0.107	0.179	0.078
DEPICT	0.917	0.965	0.904	0.237	0.279	0.171	0.094	0.137	0.041	0.229	0.312	0.166	0.242	0.363	0.197	0.128	0.219	0.081
GAN	0.763	0.736	0.827	0.265	0.315	0.176	0.121	0.151	0.045	0.212	0.298	0.139	0.225	0.346	0.157	0.121	0.174	0.078
DeCNN	0.757	0.817	0.669	0.240	0.282	0.174	0.092	0.133	0.038	0.227	0.299	0.162	0.186	0.313	0.142	0.098	0.175	0.073
DAC	0.935	0.977	0.948	0.396	0.522	0.306	0.185	0.238	0.088	0.366	0.470	0.257	0.394	0.527	0.302	0.219	0.275	0.111
DDC	0.951	0.980	0.967	0.424	0.524	0.329	-	-	-	0.371	0.489	0.267	0.433	0.577	0.345	-	-	-
IIC	-	0.992	-	-	0.617	-	-	0.257	-	-	0.596	-	-	-	-	-	-	-
DCCM	0.951	0.982	0.954	0.496	0.623	0.408	0.285	0.327	0.173	0.376	0.482	0.262	0.608	0.710	0.555	0.321	0.383	0.182
PICA	-	-	-	0.591	0.696	0.512	0.310	0.337	0.171	0.611	0.713	0.531	**0.802**	**0.870**	**0.761**	0.352	0.352	0.201
DFVC	**0.987**	**0.995**	**0.966**	**0.643**	**0.756**	**0.615**	**0.435**	**0.472**	**0.261**	**0.642**	**0.731**	**0.598**	0.753	0.847	0.736	**0.375**	**0.391**	**0.184**

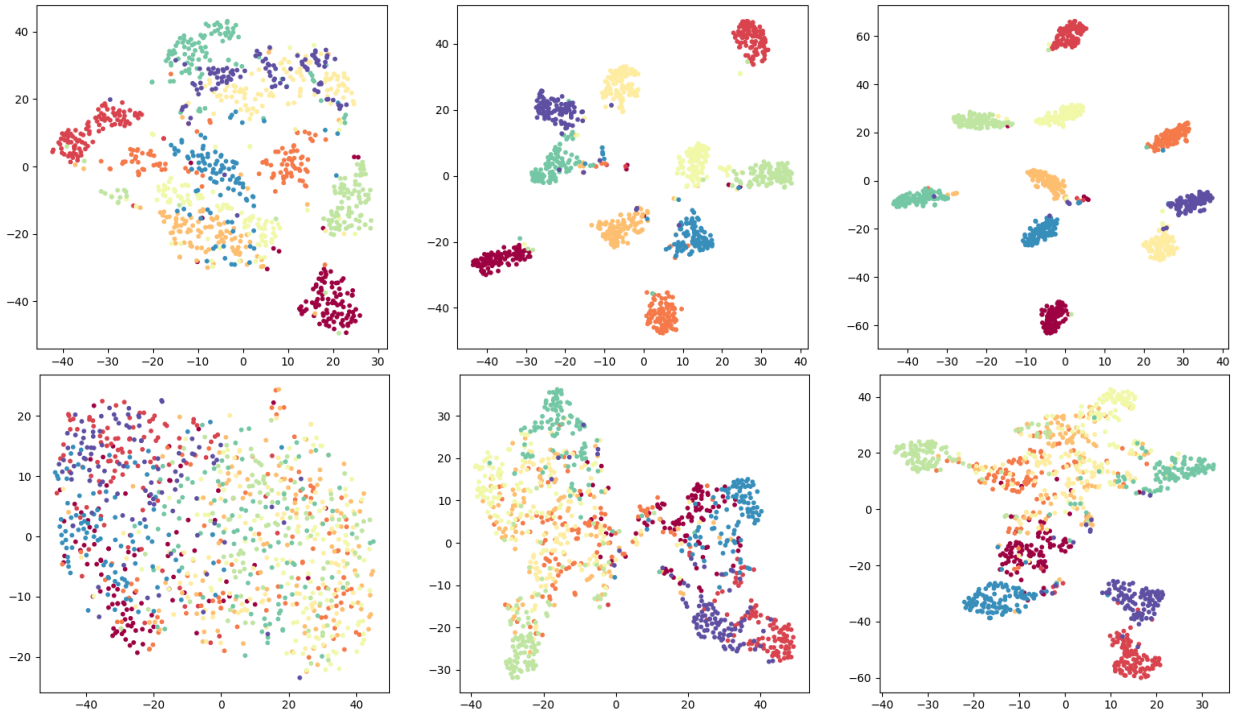


Fig. 2. Visualization of the latent representations for different training stages of the proposed DFVC on the MNIST and CIFAR-10 data sets: The plots top refer to the MNIST data set and the bottom plots to the CIFAR-10. Different colors represent different clusters. From left to right: DFVC tends to progressively learn more discriminative and cluster-friendly representations with the increasing of epochs.

a large margin. Several tendencies can be observed from the clustering results with further analysis.

First, the performance of the clustering methods based on deep learning is generally superior to the traditional methods (e.g., K-means, SC). It indicates that extracting useful representation is more crucial than selecting clustering techniques.

Second, AE and DIM are both representation learning algorithms independent of downstream tasks. The representations from DIM are more suitable for clustering than ones from AE. However, compared with other clustering-driven methods, DIM's performance is obviously insufficient. This result indicates that AE and DIM are very good as upstream pretraining models, but they are not enough to be directly used for downstream tasks.

Third, the performance of DCCM, PICA, and DFVC using the data augmentation technique is better than that of other algorithms. It implies that introducing data augmentation technology into unsupervised clustering can help the model to be optimized more reasonably and to avoid degenerate solutions. More importantly, both DCCM and DFVC are dedicated to finding discriminative representations by maximizing MI between the input and its representation. Different from DCCM, DFVC combines the MI estimation between continuous variables and the exact MI computation between discrete variables to efficiently obtain the unique and invariant

TABLE IV
CLUSTERING PERFORMANCE SENSITIVITY TO AUGMENTATION MODE

Dataset	CIFAR-10			ImageNet-10		
Mode	NMI	ACC	ARI	NMI	ACC	ARI
No	0.279	0.327	0.228	0.212	0.284	0.198
Single	0.636	0.749	0.608	0.749	0.841	0.730
Double	**0.643**	**0.756**	**0.615**	**0.753**	**0.847**	**0.736**

TABLE V
$\mathcal{L}_{\text{DMI}}$ SENSITIVITY TO THE GLOBAL AND LOCAL MI

Dataset	CIFAR-10			ImageNet-10		
G & L	NMI	ACC	ARI	NMI	ACC	ARI
$\mathcal{L}_{GMI}$	0.185	0.247	0.153	0.214	0.297	0.184
$\mathcal{L}_{LMI}$	0.211	0.291	0.182	**0.274**	**0.363**	**0.251**
$\mathcal{L}_{DMI}$	**0.213**	**0.294**	**0.184**	0.273	0.361	**0.251**

TABLE VI
$\mathcal{L}_{\text{AMI}}$ SENSITIVITY TO THE GLOBAL AND LOCAL MI

Dataset	CIFAR-10			ImageNet-10		
G & L	NMI	ACC	ARI	NMI	ACC	ARI
$\mathcal{L}_{AMI}$ (G)	0.446	0.605	0.382	0.581	0.684	0.512
$\mathcal{L}_{AMI}$ (L)	0.494	0.629	0.398	0.608	0.724	0.550
$\mathcal{L}_{AMI}$ (G+L)	**0.497**	**0.632**	**0.401**	**0.611**	**0.726**	**0.553**

TABLE VII
$\mathcal{L}_{\text{DFVC}}$ SENSITIVITY TO THE GLOBAL AND LOCAL MI

Dataset	CIFAR-10			ImageNet-10		
G & L	NMI	ACC	ARI	NMI	ACC	ARI
$\mathcal{L}_{DFVC}$ (G)	0.622	0.729	0.584	0.732	0.819	0.719
$\mathcal{L}_{DFVC}$ (L)	0.640	0.751	0.611	0.750	0.843	0.732
$\mathcal{L}_{DFVC}$ (G+L)	**0.643**	**0.756**	**0.615**	**0.753**	**0.847**	**0.736**

information of the representations. Moreover, we constrain the latent representations to follow the Gaussian mixture distribution so that they have the clustering property.

Fig. 2 visualizes the latent representations of the DFVC on the MNIST and CIFAR10 data sets using t-SNE [63] at the different training stages. We randomly select 1000 data points and compute their latent representations using the learned encoder. We can see that DFVC exhibits more cluster-friendly embedding space, and the latent representations spread more compactly, and the clusters are purer and well-separated. The above-mentioned results can sufficiently verify the effectiveness and superiority of our proposed DFVC. To further evaluate the quality of the learned representations, we conduct experimental analysis from five aspects: correlation analysis, the effect of overclustering, the effect of reconstruction, avoiding degenerate solutions, and sensitivity to initialization.

2) Correlation Analysis: We carried out four more ablation experiments to further analyze the effect of MI on different pairs and the necessity of data augmentation. For the ablation study of data augmentation, we set three combination modes of the input augmentation: no input augmentation, single-input augmentation, and double-input augmentation.

As shown in Table IV, we can obviously see that the data augmentation technique can significantly improve the clustering performance of the model. No data augmentation means that our model degenerates to DIM [34] that only considers the MI between the input and its representation, which leads to poor performance.

TABLE VIII
EFFECT OF OVERCLUSTERING

Dataset	CIFAR-10			STL-10		
OC	NMI	ACC	ARI	NMI	ACC	ARI
No	0.564	0.591	0.607	0.602	0.648	0.614
Yes	**0.643**	**0.756**	**0.615**	**0.642**	**0.731**	**0.598**

For the ablation study of MI, we subdivide $\mathcal{L}_{\text{DMI}}$ into two parts: global (G) $\mathcal{L}_{\text{GMI}}$ and local (L) $\mathcal{L}_{\text{LMI}}$ MI loss. To comprehensively examine the effect of global and local MI on the clustering performance of the model, we carried out three experiments with double-input augmentation: $\mathcal{L}_{\text{DMI}}$, $\mathcal{L}_{\text{AMI}}$, and $\mathcal{L}_{\text{DFVC}}$ sensitivity to the global and local MI. The experimental results are shown in Tables V–VII, respectively.

From the results of Table V, we can see that local MI plays a more important role in clustering performance than global MI. Even the performance based on local MI is better than that based on both (G + L) MI on the ImageNet-10 data set. Combining the results of Tables V–VII, we can obviously see that the clustering performance based on both (G + L) MI is generally better than that based on single MI. These results show that incorporating knowledge about a locality in the input into the objective can indeed improve a representation's suitability for clustering. Compare the results of Tables V and VI, we can see that the introduction of MI between categorical representations can significantly improve the clustering performance of the model. The result indicates that the proposed AMI can indeed help the model extract discriminative representations. Compare the results of Tables VI and VII, we can see that imposing the Gaussian mixture distribution on the representations can further improve the clustering performance. The result indicates that Gaussian mixture distribution can indeed help the model extract cluster-friendly representations.

3) Effect of Overclustering: The experiment about the effect of overclustering is to verify whether the auxiliary overclustering head is helpful in the standard clustering performance. "No" means that the model does not contain the overclustering head, and "Yes" means that the model contains both the standard clustering head and overclustering head. From the results in Table VIII, we can obviously see that the overclustering head plays a very important role in the clustering performance of the model. Also note that, without using the overclustering head, our DFVC can still achieve competitive performance.

4) Effect of Reconstruction: In order to verify that the proposed AMI can completely replace the decoder, we carried out one experiment that combining the reconstruction loss of a decoder with the AMI loss on CIFAR-10 and ImageNet-10 data sets. We set two hyperparameters $\lambda_1 = 2.0$ (DFVC_REC1) and $\lambda_2 = 0.2$ (DFVC_REC2) for weighting reconstruction loss. The setting of the hyperparameters α, β, and γ are the same as the original ones.

TABLE IX
EFFECT OF RECONSTRUCTION LOSS ON CLUSTERING PERFORMANCE

Dataset	CIFAR-10			ImageNet-10		
Method	NMI	ACC	ARI	NMI	ACC	ARI
DFVC_REC1	0.619	0.723	0.577	0.721	0.802	0.706
DFVC_REC2	0.632	0.741	0.601	0.742	0.831	0.721
DFVC	**0.643**	**0.756**	**0.615**	**0.753**	**0.847**	**0.736**

TABLE X
NECESSITY OF AVOIDING UNDERCLUSTERING USING THE ENTROPY REGULARIZATION OF THE CLUSTER SIZE DISTRIBUTION

Dataset	CIFAR-10			CIFAR-100		
Entropy	NMI	ACC	ARI	NMI	ACC	ARI
No	0.365	0.486	0.583	0.195	0.281	0.242
Yes	**0.643**	**0.756**	**0.615**	**0.435**	**0.472**	**0.261**

TABLE XI
MODEL PERFORMANCE SENSITIVITY TO NETWORK INITIALIZATION

Initialization	CIFAR-10			ImageNet-10		
Metrics	NMI	ACC	ARI	NMI	ACC	ARI
Gaussian	0.619	0.726	0.558	0.714	0.837	0.698
Xavier	0.628	0.737	0.571	0.731	0.842	0.709
HeNormal	**0.643**	**0.756**	**0.615**	**0.753**	**0.847**	**0.736**

From the results of Table IX, the addition of a decoder for reconstruction does not further improve the clustering performance. Conversely, increasing the weight of reconstruction loss reduces clustering performance. It indicates that the proposed AMI is qualified to replace the decoder to help the model extract more discriminative representations.

5) Avoiding Degenerate Solutions: We examined how important DFVC needs to solve the generic degenerate solutions problems (assigning most samples to one cluster) in our context. The results in Table X indicate that it is highly necessary to take into account this problem in the model design; otherwise, the model will be trivially guided to such undesired solutions. This also verifies that the proposed DFVC idea is compatible well with the entropy regularization of the cluster size distribution, enabling to eliminate simply trivial results without resorting to complex designs or tricks.

6) Sensitivity to Initialization: Model initialization is an important part of both deep neural networks and clustering. We tested its sensitivity in our DFVC w.r.t. model performance on CIFAR-10 and ImageNet-10. We evaluated the three most widely used initialization ways: Gaussian, Xavier, and HeNormal. Table XI shows that DFVC can work stably without clear variation in the overall performance when using different initialization methods and the three initialization methods can achieve good performance. This verifies that our method is insensitive to network initialization. Moreover, the results show that HeNormal can achieve the best performance on both CIFAR-10 and ImageNet-10, indicating that the combination of HeNormal and the activation function ReLU is more suitable for DFVC.

V. CONCLUSION

In this article, we developed an end-to-end clustering framework, i.e., a DFVC. To extract the useful representations, we propose the AMI that combines the MI variational estimation of continuous representations and the MI exact computation of categorical representations. By introducing the data augmentation technique, we incorporate the original input, the augmented input, and their high-level representations into the MI estimation framework to learn more discriminative representations. While achieving excellent clustering performance, the DFVC improves the robustness and avoids degenerate solutions. Extensive experiments on several challenging image data sets show that DFVC achieves significant improvement over the state-of-the-art methods.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers and editors for their constructive comments and suggestions.

REFERENCES

[1] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, 1967, vol. 1, no. 14, pp. 281–297.
[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.
[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, 1996, vol. 96, no. 34, pp. 226–231.
[4] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Germany: Springer, 2006.
[5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[6] E. Min, X. Guo, Q. Liu, G. Zhang, J. Cui, and J. Long, "A survey of clustering with deep learning: From the perspective of network architecture," *IEEE Access*, vol. 6, pp. 39501–39514, 2018.
[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[9] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.
[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. 2nd Int. Conf. Learn. Represent.*, 2014, pp. 1–14.
[11] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[12] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 1558–1566.
[13] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–18.
[14] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan, "ClusterGAN: Latent space clustering in generative adversarial networks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 4610–4617.
[15] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards k-means-friendly spaces: Simultaneous deep learning and clustering," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3861–3870.
[16] J. Yang, D. Parikh, and D. Batra, "Joint unsupervised learning of deep representations and image clusters," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5147–5156.
[17] U. Shaham, K. Stanton, H. Li, R. Basri, B. Nadler, and Y. Kluger, "SpectralNet: Spectral clustering using deep neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–21.
[18] Z. Jiang, Y. Zheng, H. Tan, B. Tang, and H. Zhou, "Variational deep embedding: An unsupervised and generative approach to clustering," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1965–1972.
[19] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J. T. Zhou, and S. Yang, "Structured AutoEncoders for subspace clustering," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5076–5086, Oct. 2018.
[20] J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis," in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, 2016, pp. 478–487.

[21] F. Li, H. Qiao, and B. Zhang, "Discriminatively boosted image clustering with fully convolutional auto-encoders," *Pattern Recognit.*, vol. 83, pp. 161–173, Nov. 2018.

[22] K. G. Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang, "Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5736–5745.

[23] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan, "Deep adaptive image clustering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5879–5887.

[24] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama, "Learning discrete representations via information maximizing self-augmented training," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1558–1567.

[25] X. Ji, A. Vedaldi, and J. Henriques, "Invariant information clustering for unsupervised image classification and segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9865–9874.

[26] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4857–4868, Nov. 2020.

[27] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," 2020, *arXiv:2009.09687*. [Online]. Available: http://arxiv.org/abs/2009.09687

[28] D. Chen, J. Lv, and Y. Zhang, "Unsupervised multi-manifold clustering by learning deep representation," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 1–7.

[29] X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 1753–1759.

[30] J. Wu *et al.*, "Deep comprehensive correlation mining for image clustering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8150–8159.

[31] L. Yang, N.-M. Cheung, J. Li, and J. Fang, "Deep clustering by Gaussian mixture variational autoencoders with graph embedding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6440–6449.

[32] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[33] M. I. Belghazi *et al.*, "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 531–540.

[34] R. D. Hjelm *et al.*, "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–24.

[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.

[36] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5810–5818.

[37] Y. Pu *et al.*, "Variational autoencoder for deep learning of images, labels and captions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2352–2360.

[38] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[39] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2536–2544.

[40] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 649–666.

[41] O. J. Hénaff *et al.*, "Data-efficient image recognition with contrastive predictive coding," 2019, *arXiv:1905.09272*. [Online]. Available: http://arxiv.org/abs/1905.09272

[42] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.

[43] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020, *arXiv:2002.05709*. [Online]. Available: http://arxiv.org/abs/2002.05709

[44] B. Poole, S. Ozair, A. V. D. Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97, 2019, pp. 5171–5180.

[45] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, vol. 9, 2010, pp. 297–304.

[46] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–17.

[47] A. Anand, E. Racah, S. Ozair, Y. Bengio, M.-A. Côté, and R. D. Hjelm, "Unsupervised state representation learning in Atari," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8769–8782.

[48] P. Bachman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 15535–15545.

[49] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," 2019, *arXiv:1906.05849*. [Online]. Available: http://arxiv.org/abs/1906.05849

[50] R. Linsker, "Self-organization in a perceptual network," *Computer*, vol. 21, no. 3, pp. 105–117, Mar. 1988.

[51] J. S. Bridle, A. J. Heading, and D. J. MacKay, "Unsupervised classifiers, mutual information and phantom targets," in *Proc. Adv. Neural Inf. Process. Syst.*, 1992, pp. 1096–1101.

[52] A. Krause, P. Perona, and R. G. Gomes, "Discriminative clustering by regularized information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 775–783.

[53] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 271–279.

[54] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[55] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Citeseer, Univ. Toronto, Toronto, ON, Canada, 2009.

[56] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

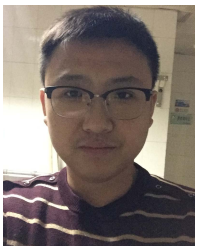
[59] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, vol. 37, 2015, pp. 448–456.

[60] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[61] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.

[62] J. Huang, S. Gong, and X. Zhu, "Deep semantic clustering by partition confidence maximisation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8849–8858.

[63] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.



Qiang Ji received the B.S. degree from the Shandong University of Finance and Economics, Jinan, China, in 2013. He is currently pursuing the Ph.D. degree with the Beijing Municipal Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology, Beijing, China.

His current research interests include computer vision, pattern recognition, deep learning, and clustering methods.

Yanfeng Sun (Member, IEEE) received the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 1993.

She is currently a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing, China. She is also a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing. Her current research interests include pattern recognition, machine learning, and image analysis.

Junbin Gao received the B.Sc. degree in computational mathematics from the Huazhong University of Science and Technology (HUST), Wuhan, China, in 1982, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 1991.

He was a Professor in computer science with the School of Computing and Mathematics, Charles Sturt University, Bathurst, NSW, Australia. From 1982 to 2001, he was an Associate Lecturer, a Lecturer, an Associate Professor, and a Professor with the Department of Mathematics, HUST. He was a Senior Lecturer and a Lecturer in computer science with the University of New England, Armidale, NSW, Australia, from 2001 to 2005. He is currently a Professor of big data analytics with The University of Sydney Business School, The University of Sydney, Sydney, NSW, Australia. His current research interests include machine learning, data analytics, Bayesian learning and inference, and image analysis.

Yongli Hu (Member, IEEE) received the Ph.D. degree from the Beijing University of Technology, Beijing, China, in 2005.

He is currently a Professor of computer science and technology with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing University of Technology. His current research interests include computer graphics, pattern recognition, and multimedia technology.

Baocai Yin (Member, IEEE) received the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 1993.

He is currently a Researcher with the Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing, China. He is also a Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing. His current research interests include multimedia, multifunctional perception, and virtual reality, and computer graphics.

Dr. Yin is a member of the China Computer Federation.