Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 32(6): 3952–3972.
DOI: 10.3934/era.2024177
Received: 23 April 2024
Revised: 17 May 2024
Accepted: 07 June 2024
Published: 14 June 2024

Research article

A deep clustering framework integrating pairwise constraints and a VMF mixture model

He Ma[1,*] **and Weipeng Wu**[2,*]

[1] College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150000, China

[2] College of Software, Harbin Institute of Information Technology, Harbin 150431, China

* **Correspondence:** Email: mahe@hrbeu.edu.cn, wuwp@hrbiit.edu.cn.

Abstract: We presented a novel deep generative clustering model called Variational Deep Embedding based on Pairwise constraints and the Von Mises-Fisher mixture model (VDEPV). VDEPV consists of fully connected neural networks capable of learning latent representations from raw data and accurately predicting cluster assignments. Under the assumption of a genuinely non-informative prior, VDEPV adopted a von Mises-Fisher mixture model to depict the hyperspherical interpretation of the data. We defined and established pairwise constraints by employing a random sample mining strategy and applying data augmentation techniques. These constraints enhanced the compactness of intra-cluster samples in the spherical embedding space while improving inter-cluster samples' separability. By minimizing Kullback-Leibler divergence, we formulated a clustering loss function based on pairwise constraints, which regularized the joint probability distribution of latent variables and cluster labels. Comparative experiments with other deep clustering methods demonstrated the excellent performance of VDEPV.

Keywords: generative deep clustering; variational autoencoder; von Mises-Fisher mixture model; pairwise constraints; Kullback–Leibler divergence

1. Introduction

The clustering problem has undergone extensive research over the past seventy years, making significant progress in data mining, pattern recognition, image analysis, and other fields. However, when dealing with high-dimensional and large-scale data, the computational resources and time costs required by traditional clustering algorithms increase significantly. To address the challenges of dimensionality and computational complexity, researchers commonly turn to deep learning techniques to capture essential low-dimensional features in raw data and perform clustering analysis on these fea-

tures [1–3].

However, deep learning-based clustering faces several challenges, including: 1) Relatively weak modeling of complex data distributions, limiting deep exploration of data structures; 2) difficulty in obtaining continuous representations of the feature space, restricting accurate capturing of data continuity; 3) neglecting compactness within intra-cluster samples and separability between inter-cluster samples, leading to a less accurate understanding of global cluster structures and local structures; 4) lacking joint embedding and clustering processes, making it challenging to generate an embedding space best suited for clustering tasks.

To tackle these issues, we propose a novel generative deep clustering algorithm that fully exploits the advantages of a deep model based on variational autoencoders(VAE) and a clustering method based on pairwise constraints. As a generative model, the VAE can naturally handle uncertainty by modeling the data generation process within a probability framework. This model typically consists of two major parts: An encoder maps raw data to a continuous embedding space, and a decoder samples from the embedding space to reconstruct the original data. We adopt a prior distribution based on the von Mises-Fisher (VMF) mixture model, which models the embedding space and obtains cluster labels for the original data. The encoder of the VAE and the VMF mixture model can be viewed as a discriminative clustering model, trained by minimizing the KL divergence between the encoding distribution of the data and the distance of the VMF mixture model and maximizing the log term of the reconstruction distribution of the data. Our choice to adopt the VMF mixture model stems from the following considerations: A common assumption in the clustering approach using VAE is that the prior in the latent space follows a multivariate Gaussian distribution. During training, the model can leverage the mathematical convenience brought about by the properties of the Gaussian distribution. However, some studies have found that Gaussian distributions in models can violate non-informative prior assumptions in some cases, leading to unstable situations; therefore, it is recommended to consider VMF distributions as viable alternatives [4]. Additionally, latent variables in the embedding space undergo L2 norm normalization, exhibiting robust directional characteristics [5, 6]. To address this issue and effectively model the distribution of the latent space, we adopt the VMF mixture model as the prior for the VAE. In this paper, the training process adopts rejection sampling [4] to reparameterize the variational autoencoder.

While generative deep clustering models excel in handling high-dimensional and non-linear data, effectively modeling data distribution, and obtaining continuous representations in the embedding space, they often encounter scenarios where samples within the same cluster are sparsely distributed in the embedding space. The proximity between different clusters is also relatively close, potentially resulting in suboptimal cluster assignment outcomes. To better reflect the intrinsic structure of the data and enhance understanding and expressive power of data distributions, we employ data augmentation techniques and a random sample mining strategy to establish pairwise constraints. Using KL divergence, we calculate the regularization term between posterior distributions of pairwise constraints. These pairwise constraints include must-link and cannot-link constraints. The proposed pairwise constraints do not require additional supervised information. The sample set obtained by randomly mapping the original data forms the must-link constraints, while the high-confidence sample set defined by the VMF mixture model forms the cannot-link constraints. Introducing a clustering loss function based on pairwise constraints helps enhance compactness within intra-cluster samples and improve separability between inter-cluster samples, further enhancing clustering performance.

To create an embedding space optimized for clustering tasks, we propose a unified loss function that addresses both embedding and clustering objectives simultaneously, enhancing the efficacy of the learning process. This unified loss function incorporates the loss functions of our VAE-based model as well as the clustering objectives. We then use a jointly optimized approach to update parameters and estimate cluster assignments.

In addition, experimental results demonstrate that our proposed method performs better in handling multiple high-dimensional and large-scale datasets. Therefore, the contributions of this paper include:

• Providing a more suitable nonlinear embedding space for clustering tasks via adopting a von Mises-Fisher mixture model as the prior distribution in the variational autoencoder.

• Introducing a unified loss function, integrating data augmentation techniques and a random sample mining strategy as pairwise constraints, simultaneously addressing both embedding and clustering objectives.

• Achieving competitive results through multiple experiments, compared with state-of-the-art methods in the field. The experimental outcomes demonstrate the notable superiority of our method across various datasets and scenarios, confirming its outstanding performance.

2. Related work

There are several studies on deep clustering. Deep clustering is a methodology that merges deep learning with clustering techniques to extract meaningful representations from unlabeled data and segregate the data into distinct clusters. Compared to traditional clustering methods such as k-means, hierarchical, and bisecting k-means clustering, deep clustering handles large-scale data more effectively and does not require manually prepared features. Deep clustering can be categorized into three types based on the organization of modules and the design of functionality: multi-stage, simultaneous, and generative [2].

In multi-stage deep clustering, representation learning and clustering are divided into two sequential independent modules. A standard model is to train autoencoders using an unsupervised manner, then stack the encoding part of the autoencoders for data representation, and finally apply traditional clustering methods [7]. Zhang et al. used autoencoders and the subspace clustering method to obtain the self-expression representation and then performed clustering through spectral clustering [8]. Tao et al. proposed a representation learning method that combines instance discrimination and feature decorrelation, which is suitable for learning the latent space of clustering [9]. Dang et al. introduced a nearest neighbor matching method to maintain consistent cluster assignments at the local and global levels [10]. Most multi-stage deep clustering methods are simple to deploy and have relatively intuitive structures [11, 12].

In simultaneous deep clustering, representation learning and clustering algorithms participate in training and clustering together, interacting with each other. The DEC proposed by Xie et al. simultaneously learned the low-dimensional representation space and clustered centroids by pre-training multi-layer autoencoders [13]. Ye et al. simultaneously learned the representation and the embedding of spectral clustering through an autoencoder and introduced sparsity constraints [14]. Thirumoorthy et al. proposed a shrinking autoencoder and used the Frobenius norm as a penalty term to enhance the stability of the data representation retrieved for the training input [15]. Cai et al. introduced EDESC, which breaks away from the self-expressive framework, enabling the learning of subspace bases from

deep representations to enhance more powerful representations [16]. In simultaneous deep clustering, the model can learn clustering-oriented representation [17–20].

In generative deep clustering, models like generative adversarial networks (GAN) or VAE are employed for representation learning and clustering. The variational deep embedding (VaDE) proposed by Jiang et al. adopted the concept of DEC and applied it to VAE. Dilokthanakul et al. proposed a method similar to VaDE. Ji et al. replaced VAE's decoder [21] by calculating the mutual information of representation and clustering. Wang et al. decoupled input data into specific and standard features through VAE and then clustered [22]. VAE and its deformations have been widely studied and applied [23–27].

The prior distribution is essential for VAE, and many studies are related to priors. Some studies have verified that changing the standard Gaussian distribution prior can produce some benefits. Davidson et al. pointed out that the standard Gaussian distribution breaks the assumption of uninformative priors and proposed using VMF prior [4]. VMF can better explain some data types, such as directional data, and has obvious advantages over the standard Gaussian distribution in low-dimensional space [28–30]. As the data dimension increases, researchers have found that in the context of unsupervised clustering, the Gaussian Mixture Model (GMM) gradually emerges as a preferable choice over the standard Gaussian distribution for characterizing the distribution of complex latent spaces [21–26, 31–34]. Research on VAE algorithms based on GMM priors is currently relatively mature. Some studies have pointed out that in the training of VAE, the model may benefit from L2 normalization of the latent space [35]. Compared with GMM, the VMF mixture model is more suitable for describing the distribution of normalized data. There are relatively few studies on VAE based on VMF mixture distribution prior [36–38].

There are also many studies on constrained clustering. In constrained clustering, background knowledge is represented as a set of instance-level constraints. Some studies based on the K-means algorithm, such as PCKmeans [39] and COPKmeans [40], have proven that pairwise constraints help improve clustering performance. Goschenhofer et al. studied methods of using unconstrained data and proposed a pseudo-constraint mechanism to overcome confirmation bias [41]. Manduchi et al. proposed DC-GMM based on VAE to guide clustering through pairwise constraints [42]. Hajjar et al. achieved multi-view clustering by integrating non-negative latent and spectral latent and proposing consistent smoothness and orthogonality constraints for all views [43]. Lv et al. employ pairwise similarity to weight the reconstruction loss and guide supervised similarity learning using pseudo-graphs and pseudo-labels [44], similar to the idea of constructing pairwise constraints in this paper. Most of the pairwise constraints are exogenous knowledge [45, 46], while in this paper, the pairwise constraints are generated by the output of the clustering process, such as pseudo-labels and new data generated through data augmentation. The main differences between our proposed method and existing methods lie in several fundamental aspects:

Our approach is built on the VaDE architecture, which is generative deep clustering. By leveraging VaDE, we can take advantage of the inherent advantages of variational autoencoders while accommodating the unique characteristics of the Von Mises-Fisher mixture model. Unlike deep clustering models such as VaDE, GMVAE, and DC-GMM that rely on Gaussian mixture model prior distributions within the VAE framework, our method introduces a prior based on the Von Mises-Fisher mixture model. This choice allows us to capture directional data distributions more efficiently, especially if the data exhibits circular or spherical patterns. A significant enhancement introduced in our approach is

the incorporation of unsupervised pairwise constraints for cluster analysis. By exploiting the pairwise relationships between data points without requiring class labels, we enhance the effectiveness of the clustering process and improve the overall quality of the clustering results, which is different from clustering models that use label information.

3. Proposed method

In this section, we first introduce the VDEPV architecture based on VAE, which includes a generative and inference process. Then, we introduce pairwise constraints based on data augmentation and random sample mining. Finally, we show the VDEPV framework and provide a joint learning framework to train the entire network layer using the unified representation learning and clustering loss function.

3.1. VDEPV Architecture

In this section, we use VAE to construct our VDEPV. The prior distribution describes the distribution of data embedding; however, it may undermine non-informative assumptions and lead to unstable situations. We adopted a VMF mixture model instead of the more common standard Gaussian distribution or GMM to avoid this instability.

Since VDEPV is a generative deep model, we describe its generation process first. Let's consider a dataset X with N samples, where each sample $\{x_i\}_{i=1}^{N}$ is a D-dimensional observation vector. In the generation process, K clusters are defined, and for each data point x_i, the model learns a latent variable $z \in \mathbb{R}^J$. For each observed sample x_i, the procedure begins by selecting a cluster $c \in \{0,1\}^{K\times 1}$ from the prior probability distribution, parameterized by π. Subsequently, a latent vector z is generated from the VMF associated with cluster c, with the VMF parameterized by the mean direction μ_k and the concentration γ_k. The generation of the latent variable is based on a specific reparameterization technique of rejection sampling [4]. Next, we use the neural network g to compute μ_x and γ_x. The sample is then generated from the multivariate VMF $\mathcal{V}(x|\mu_x, \gamma_x)$.

$$p(c) = \prod_{k=1}^{K} \pi_k^{c_k}\,, \tag{3.1}$$

$$p(z|c) = \mathcal{V}(z|\mu_k, \gamma_k)\,, \tag{3.2}$$

$$p_\theta(x|z) = \mathcal{V}(x|\mu_x, \gamma_x)\,, \tag{3.3}$$

$$[\mu_x; \ln \gamma_x] = g(z;\theta)\,, \tag{3.4}$$

where

$$\mathcal{V}(z|\mu_k, \gamma_k) = C_d(\gamma_k)\exp(\gamma_k {\mu_k}^T z)\,, \tag{3.5}$$

$$C_d(\gamma_k) = \frac{\gamma_k^{d/2-1}}{(2\pi)^{d/2} I_{d/2-1}(\gamma_k)}\,, \tag{3.6}$$

where π_k satisfies $\sum_{k=1}^{K} \pi_k = 1$, c_k and π_K denote the kth entry of c and π, respectively. $\|\mu_k\|_2^2 = 1$. g is a neural network with trainable parameters θ. The generative process of VDEPV refers to Figure 1.

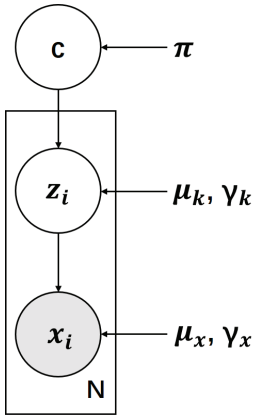


Figure 1. The generative process of VDEPV.

The generative process can be decomposed through the joint probability $p_\theta(x, z, c)$, which can be described as

$$p_\theta(x, z, c) = p_\theta(x|z)p(z|c)p(c)\,. \tag{3.7}$$

Next, we introduce the inference process of VDEPV. In the inference process, the observed variable x is mapped by the encoder of VDEPV to a latent variable z in the latent space. This latent space is a spherical space that L2 has normalized. We use $p_\theta(z, c|x)$to describe the inference process. However, in model training, finding the posterior probability of the latent variable and the parameters for maximum likelihood estimation is tough. Therefore, a variational distribution $q_\phi(z, c|x)$ is introduced to approximate the posterior distribution of latent variables z and clusters c. Then we define

$$q_{\phi_1}(z|x) = \mathcal{V}(z|\tilde{\mu}, \tilde{\gamma})\,, \tag{3.8}$$

$$[\tilde{\mu}, \ln\tilde{\gamma}] = f_1(x; \phi_1)\,, \tag{3.9}$$

$$q_{\phi_2}(c|z) = \text{Multinomial}(\tilde{\pi})\,, \tag{3.10}$$

$$\tilde{\pi} = f_2(z; \phi_2)\,, \tag{3.11}$$

where, ϕ_1 and ϕ_2 are the parameters of the networks f_1 and f_2, respectively. For convenience, we use ϕ to collectively represent ϕ_1 and ϕ_2.

The Inference process can be decomposed by the joint probability $q_\phi(z, c|x)$ and can be described as

$$q_\phi(z, c|x) = q_{\phi_1}(z|x)q_{\phi_2}(c|z)\,. \tag{3.12}$$

In the case of using the VMF distribution $\mathcal{V}(\tilde{\mu}, \tilde{\gamma})$, traditional reparameterization techniques may fail because the probability density function of the VMF has a certain complexity, making it challenging to transform the sampling process into a differentiation operation concerning the parameters through

simple deterministic transformations. Therefore, rejection sampling schemes are used to obtain the latent variable z.

Firstly, a variation ω is sampled from the VMF distribution with given shape parameters $\tilde{\gamma}$. The sampling process is defined by the probability density function $g(\omega|\tilde{\gamma})$. This probability density function depends on the distribution of the variation ω, and its form is determined by the characteristics and parameters of the VMF distribution.

$$g(\omega|\tilde{\gamma}) = \frac{2(\pi^{\frac{d}{2}})}{\Gamma(\frac{d}{2})} C_d(\tilde{\gamma}) \frac{\exp(\omega\tilde{\gamma})(1-\omega^2)^{\frac{1}{2}(d-3)}}{\text{Beta}(\frac{1}{2}, \frac{1}{2}(d-1))} \tag{3.13}$$

where $g(\omega|\tilde{\gamma})$ is the probability density function for the sampling variable ω, $\Gamma(\cdot)$ denotes the gamma function, $C_d(\tilde{\gamma})$ is the normalization constant, which depends on the dimension d and the shape parameter $\tilde{\gamma}$, and $\text{Beta}(\cdot,\cdot)$ is the beta function.

Next, employing a specific transformation relationship, the mode vector e_1 (i.e., the first coordinate axis) is mapped to the mean vector $\tilde{\mu}$ of the VMF distribution through a Householder mapping. This mapping preserves the length and angles of vectors while transforming e_1 into the mean of the distribution.

Finally, the transformed mean vector $\tilde{\mu}$ is combined with the sampled v using a given formula to obtain the final sample z. This formula preserves the length and direction of the sample and depends on the previously sampled variation ω.

$$z = w\tilde{\mu} + v\sqrt{1-w^2} \tag{3.14}$$

This process ensures that the samples obtained from the VMF distribution are differentiable, making them convenient for tasks like variational inference that require gradient computations. Using these differentiable samples, one can efficiently perform gradient-based optimization or inference procedures, such as gradient ascent in variational inference, to approximate complex posterior distributions. This differentiability property enhances the applicability and effectiveness of the VMF distribution in various probabilistic modeling and machine learning tasks.

3.2. Pairwise constraints of VDEPV

In this paper, we propose pairwise constraints based on data augmentation and random sample mining strategies, aimed at improving the clustering performance of VDEPV. It is worth noting that we utilize these pairwise constraints in a fully unsupervised manner. Among these, the must-link constraints are constructed using data augmentation techniques, while the cannot-link constraints are built using random sample mining strategies. Next, we will describe in detail the components of these two types of pairwise constraints.

We employ data augmentation methods to construct the must-link constraints. Specifically, we apply random mappings T to each image x_i in the original image set X, including geometric and pixel transformations, to generate new images with different data but the same labels as the original images. This means that one original image can generate multiple derived images, as shown in the Figure 2. We denote these newly generated images as x_j. This mapping relationship can be represented by the following formula

$$x_j \leftarrow \text{T}(x_i)\,. \tag{3.15}$$

The set of must-link constraints is defined as $X_{ML} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ belong to the same cluster, where } 1 \leq i, j \leq N\}$. Data augmentation techniques introduce diversity into the original data by applying random transformations, thereby enhancing the scale of the dataset. We used basic data augmentation techniques, with common operations as follows:

Translation: Also known as shifting, it changes the position of the image by moving pixels along the horizontal and vertical directions of the image. The offset range is between -3.5 and 3.5.

Rotation: By rotating the image to change its angle, the diversity of the data is increased, making the model more robust. The range is between -25 and 25.

Elastic deformation: By introducing local non-linear distortions to the image, mimicking the deformations that occur due to the elastic properties of objects. The intensity of the deformation is controlled within the range of 100 to 300, and the spatial extent of the deformation is controlled within the range of 10 to 30.

Contrast enhancement: By increasing or decreasing the difference between adjacent pixels in the image, the contrast of the image is enhanced or reduced. Here, the enhancement factor is set to 1.2.

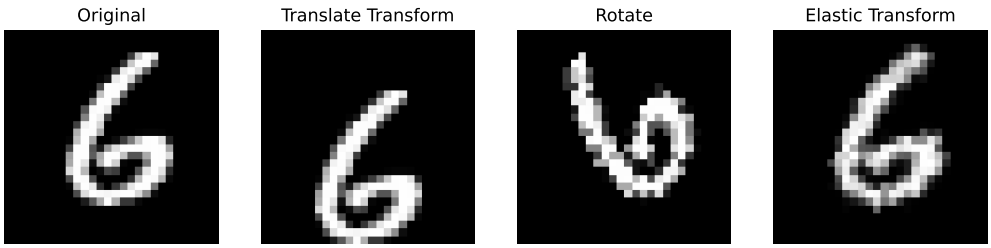


Figure 2. Data augmentation introduces diverse transformations to the original images, generating a collection of must-linkage constraints. The following showcases three effects: translation, rotation, and elastic deformation.

Although data augmentation techniques have been proven effective in enhancing data diversity, they are not suitable for building the cannot-link constraints in our proposed method. This is mainly due to the nature of the cannot-link constraints, which require identifying differences in labels between data points. Data augmentation techniques cannot create sample points with different labels. We construct the cannot-link constraints to address this challenge by adopting a random sample mining strategy. The set of cannot-link constraints is defined as $X_{CL} = \{(x_i, x_j) : x_i \text{ and } x_j \text{ belong to different clusters, where } 1 \leq i, j \leq N\}$. In each training process of VDEPV, the model's encoder is used to obtain representations of the input data. Then, the VMF (von Mises-Fisher) mixture distribution model generates pseudo-labels on the training data. For each image $x_i \in X$, we assign the corresponding pseudo-label y_i. Next, we randomly select some images $\{x_j \in X\}$ from the dataset, ensuring that the predicted probability of the pseudo-label y_j of the selected new image x_j has the maximum difference from the predicted probability of y_i for x_i. In this process, for each x_i, a set of new image data is generated. This process is called random sample mining (RSM), which can be described by the following formula:

$$x_j \leftarrow \text{RSM}(x_i)\,. \tag{3.16}$$

In this way, we are able to construct the cannot-link constraints, which will play a crucial role in the subsequent improvement of clustering performance.

3.3. Joint loss of VDEPV

In this section, we first provide a schematic diagram of the VDEPV architecture, then introduce the loss functions of VAE and pairwise constraints, respectively, and finally provide a unified loss function.

Compared to standard learning methods in generative deep learning, we simultaneously learn variational autoencoders and pairwise constrained clustering. As shown in Figure 3, VDEPV includes the following components: 1) The encoder uses a fully connected layer to map the input image into a spherical latent space. 2) When sampling latent vectors in a spherical latent space, pairwise constraints promote the compactness of latent vectors within clusters and the separation between clusters through posterior probabilities. 3) The decoder uses a fully connected layer to reconstruct the input image from the latent vectors. 4) Calculate the loss function using the reconstructed data and original input, including reconstruction loss, KL divergence, and pairwise constraint regularization terms. 5) Update the model parameters according to the gradient calculated by the loss function until the model converges or reaches a predetermined training round. 6) After training, the model can cluster input images.

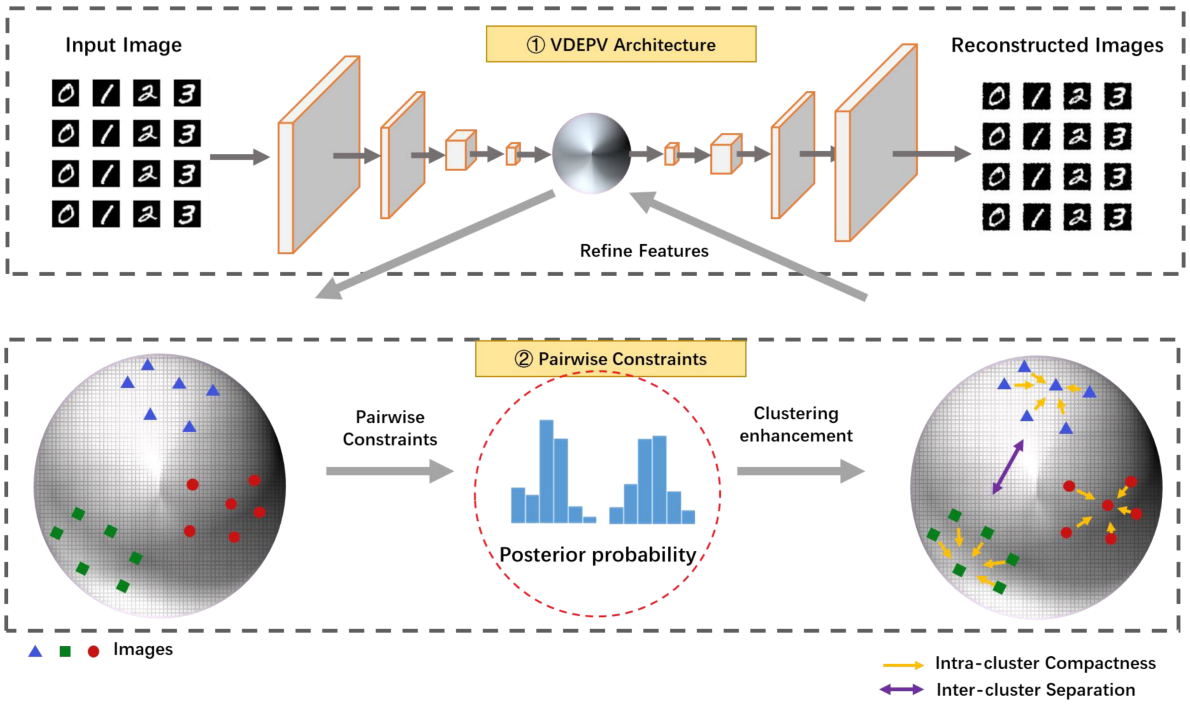


Figure 3. VDEPV architecture and pairwise constraints. VDEPV consists of variational autoencoders stacked on fully connected neural networks, and pairwise constraints are applied on a spherical latent space to promote intra-cluster compaction and inter-cluster separation.

In order to define the loss function of VDEPV, we introduce the loss function of VAE. The loss function of VAE usually includes reconstruction loss and KL divergence. The reconstruction loss describes the decoder's ability to reconstruct the input data. For VAE, this is usually calculated by comparing the difference between the original input and the reconstructed data generated by the decoder from vectors sampled from the latent space. KL divergence measures the difference between the learned latent space and the distribution of the latent space. KL divergence ensures that the learned latent representation is statistically close to the distribution of the latent space. It helps to make the latent space more regular,

prevent overfitting, and improve the model's generalization ability. The reconstruction loss and KL divergence can be expressed as

$$\mathcal{L}_{\text{VAE}} = E_{q_\phi(z,c|x)}\Big[\log p_\theta(x|z)\Big] + KL\Big[q_{\phi_1}(z|x)\|p(z,c)\Big], \tag{3.17}$$

the first term is the reconstruction loss, and the second is the KL divergence. The minimization goal of this loss function is to adjust the model parameters during the training process so that the model can achieve effective data reconstruction.

In order to define the loss function for pairwise constraints, we first introduce the similarity matrix. We use the matrix $A \in \mathbb{R}^{M\times M}$ to describe the interrelationship of pairs of constraints, where $M = M_{ML} + M_{CL}$. The representation of A is

$$\mathrm{A} = \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1M} \\ A_{12} & A_{22} & \cdots & A_{M2} \\ \vdots & \vdots & \ddots & \vdots \\ A_{1M} & A_{2M} & \cdots & A_{MM} \end{bmatrix}, \tag{3.18}$$

where, each element A_{ij} is used as the weight of the edge connecting samples i and j, indicating their degree of similarity. For must-link constraints, the obvious dependency between x_i and x_j within the same cluster determines $A_{ij} = 1$. In contrast, for cannot-link constraints, the clear difference between x_i and x_j in different clusters dictates $A_{ij} = -1$.

In cases where samples belong to the must-link constraint set, we expect similarities in their attributes and cluster assignments. In contrast, we expect significant differences in their attributes and cluster assignments for samples not included in the cannot-link constraint set. Given the effectiveness of the VAE model in capturing data distribution, we adopt the KL distance to quantify the closeness between the posterior distribution of the augmented data and the true posterior distribution of the original data. Combining Eq (3.18) and the KL distance, we design a loss function for pairwise constraints as follows:

$$\mathcal{L}_{\text{PC}} = \frac{1}{2}\sum_{j=1,j\neq i}^{M} A_{ij} KL(q_\phi(z,c|x_j)\|p_\theta(z,c|x_i)). \tag{3.19}$$

Finally, by combining the Eqs (3.17) and (3.19), we propose a joint loss function

$$\begin{aligned} \mathcal{L} =& \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{PC}} \\ =& - E_{q_\phi(z,c|x)}\Big[\log p_\theta(x|z)\Big] + KL\Big[q_{\phi_1}(z|x)\|p(z,c)\Big] \\ &+ \frac{1}{2}\sum_{j=1,j\neq i}^{M} A_{ij} KL(q_\phi(z,c|x_j)\|p_\theta(z,c|x_i)). \end{aligned} \tag{3.20}$$

However, in Eq (3.20), the third term to the right of the equal sign is difficult to calculate. Next, we propose a solution to this problem. According to the properties of conditional variational autoencoders, minimizing the loss function (3.20) is equivalent to minimizing the following expression

$$\min_{\phi,\theta}\left(-\log p(x_i) + \frac{1}{2}\sum_{j=1,j\neq i}^{M} A_{ij}\mathrm{KL}(q_\phi(z,c|x_j)\|p_\theta(z,c|x_i))\right), \tag{3.21}$$

The above expression can be simplified to

$$\min_{\phi,\theta}\left(-\log p(x_i)+\frac{1}{2}\sum_{j=1,j\neq i}^{M}A_{ij}\mathrm{G}(\phi,\theta,x_i,x_j)\right), \tag{3.22}$$

where

$$\mathrm{G}(\phi,\theta,x_i,x_j)=\frac{1}{2}(\mathrm{KL}(q_\phi(z,c|x_i)||p_\theta(z,c|x_i)))+\frac{1}{2}(\mathrm{KL}(q_\phi(z,c|x_j)||p_\theta(z,c|x_i)))\,. \tag{3.23}$$

That is, minimizing

$$\min_{\phi,\theta}\left(-\log p(x_i)+\sum_{j=1,j\neq i}^{M}A_{ij}(\log p(x_i)-\mathrm{ELBO}(\phi,\theta,x_i)-\mathrm{ELBO}(\phi,\theta,x_i,x_j))\right), \tag{3.24}$$

and

$$\mathrm{ELBO}(\phi,\theta,x_i)=-\mathrm{KL}(q_\phi(z,c|x_i)||p(z,c))+E_{q_\phi(z,c|x_i)}\left[\log p_\theta(x_i|z)\right], \tag{3.25}$$

$$\mathrm{ELBO}(\phi,\theta,x_i,x_j)=-\mathrm{KL}(q_\phi(z,c|x_j)||p(z,c))+E_{q_\phi(z,c|x_j)}\left[\log p_\theta(x_i|z)\right]. \tag{3.26}$$

Specifying

$$\sum_{j=1,j\neq i}^{M}A_{ij}=1\,, \tag{3.27}$$

Therefore, we finally obtain the simplified unified loss function.

$$\mathcal{L}=-\sum_{j=1,j\neq i}^{M}A_{ij}(\mathrm{ELBO}(\phi,\theta,x_i)+\mathrm{ELBO}(\phi,\theta,x_i,x_j))\,. \tag{3.28}$$

The pseudo-code of the entire algorithm is shown in Algorithm 1.

4. Experiments

In this section, we first introduce three benchmark handwritten digit image datasets. Subsequently, we present the details of the VDEPV model's structure and parameters. We then provide metrics for evaluating the model's performance. Finally, we conduct comparisons with state-of-the-art clustering methods on all datasets to assess the performance of VDEPV.

4.1. Datasets

We use MNIST, QMNIST, and EMNIST to evaluate the clustering performance of VDEPV. MNIST originates from the National Institute of Standards and Technology in the United States, constituting a collection of handwritten digit images with 60,000 training samples and 10,000 testing samples, each depicting a 28x28-pixel handwritten digit image. QMNIST is an extended variant of MNIST developed by Google Research, utilizing additional label information to enhance the dataset. QMNIST comprises 60,000 training samples and 60,000 testing samples. EMNIST is another extension of MNIST, featuring a broader range of characters and font styles. Compared to MNIST letters, the letters in EMNIST occupy a larger proportion of the image. As the clustering task is entirely unsupervised, we concatenate training and testing samples where applicable. Table 1 provides an overview of the quantities, feature dimensions, and labels of these datasets used in our experiments.

Algorithm 1: VDEPV Algorithm

Input: Dataset X, maximum iteration max_iter.
Output: Generated model parameters θ, inference models $\{\phi_1, \phi_2\}$.
while *Model parameters* $\{\theta, \phi_1, \phi_2\}$ *have not converged and iteration count is less than maximum max_iter* **do**
    Shuffle the dataset X and extract a batch of samples X';
    while *Dataset X has not been traversed* **do**
        Construct X_{ML} using data augmentation technique: $X_{\mathrm{ML}} = \{(x_i, x_j) : x_i$ and x_j belong to the same cluster, $x_i \in X'\}$;
        Construct X_{CL} using random sample mining technique: $X_{\mathrm{CL}} = \{(x_i, x_j) : x_i$ and x_j belong to different clusters, $x_i \in X'$, $x_j \in X'\}$;
        Construct similarity matrix A using Eq (3.18);
        Input X_{ML} and X_{CL} into the inference model from Eqs (3.8) to (3.11) to obtain mean $\tilde{\mu}_i$ and concentration $\tilde{\gamma}_i$ for x_i, mean $\tilde{\mu}_j$ and concentration $\tilde{\gamma}_j$ for x_j;
        Generate latent variables z_i and z_j using Eq (3.14) with $\{\tilde{\mu}_i, \tilde{\gamma}_i\}$ and $\{\tilde{\mu}_j, \tilde{\gamma}_j\}$ respectively;
        Input latent variables z_i and z_j into the generation model from Eqs (3.1) to (3.4) to obtain reconstructed samples x_i and x_j and cluster assignment labels;
        Compute joint loss function using Eq (3.28);
        Minimize joint loss function using Adam optimizer;
        Update model parameters $\{\theta, \phi_1, \phi_2\}$;
    end
end
Return: Generated model parameters θ, inference models $\{\phi_1, \phi_2\}$.

4.2. Implementation details

We employ fully connected layers in both the encoder and decoder for all datasets. The encoder architecture follows the pattern D-500-500-2000-10, while the decoder architecture is structured as 10-2000-500-500-D, where D represents the input dimension. ReLU activation functions are applied to all hidden layers. When constructing the set of must-links in pairwise constraints, we generate an appropriate number of augmented instances for each input data (empirically, we observed that setting M=20 produces the optimal results).

VaDE [34] adopts a pretraining strategy to alleviate the impact of undesirable local minima or saddle points that may arise during the initial training phase. Motivated by these considerations, we employ stacked autoencoders (AE) to pre-train our neural network.

We employ the Adam optimizer throughout the experiment, setting the learning rate to 0.001 ini-

Table 1. Datasets description.

Datasets	All instances	Training instances	Testing instances	Features	Classes
MNIST	70,000	60,000	10,000	784	10
QMNIST	120,000	60,000	60,000	784	10
EMNIST	70,000	60,000	10,000	784	10

tially and configuring the beta values as (0.3 and 0.5). The learning rate undergoes decay every 10 epochs, with a decay factor of 0.8. The algorithm uses PyTorch 1.10.0 and the Python 3.9.18 framework, running on an NVIDIA GeForce GTX 1060.

4.3. Evaluation

The experiments in this section employ three metrics widely used in clustering tasks to evaluate model performance: ACC, ARI, and NMI. The purpose of using these metrics is to understand how well our model performs in terms of clustering. The calculation formulas for these indicators are as follows:

Unsupervised Clustering Accuracy (ACC) measures the average correct clustering rate for clustered samples, and its specific calculation is as follows

$$ACC = \frac{1}{N}\sum_{i} N_i^j,$$

Here, N_i^j denotes the number of data samples in the i-th cluster that is correctly assigned to cluster j. N represents the total number of samples.

Normalized Mutual Information (NMI): NMI quantifies the mutual information between predicted and true labels, normalized to the range [0, 1], and

$$NMI = \frac{2I(X;Y)}{H(X)+H(Y)},$$

where

$$I(X;Y) = \sum_{x,y} p(x,y)\log\frac{p(x,y)}{p(x)p(y)},$$

and

$$H(X) = -\sum_{i} p\left(x_i\right)\log_b p\left(x_i\right),$$

Adjusted Rand Index (ARI): ARI is derived from the Rand Index (RI), which measures clustering as a series of pairwise decisions based on correct decision rates:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}},$$

Here n_{ij} represents each element in $X \times Y$.

4.4. Clustering results

We compare our clustering model, VDEPV, with several baseline and state-of-the-art clustering algorithms, including traditional clustering such as K-means, GMM, and BisectingKMeans; the multi-stage deep clustering like AE + Spectral Clustering (AE+SC), AE+VMF mixture model (AE+VMFMM), and AE+GMM, GAN+K-means (GAN-K), VAE+K-means (VAE-K), and SENet

[8]; the simultaneous deep clustering such as DEC [13] and EDESC [16]; the generative deep clustering like GMVAE [31], VaDE [34], and DC-GMM [42]; the constrained clustering such as PCKmeans [39], COPKmeans [40], and PSSC [44].

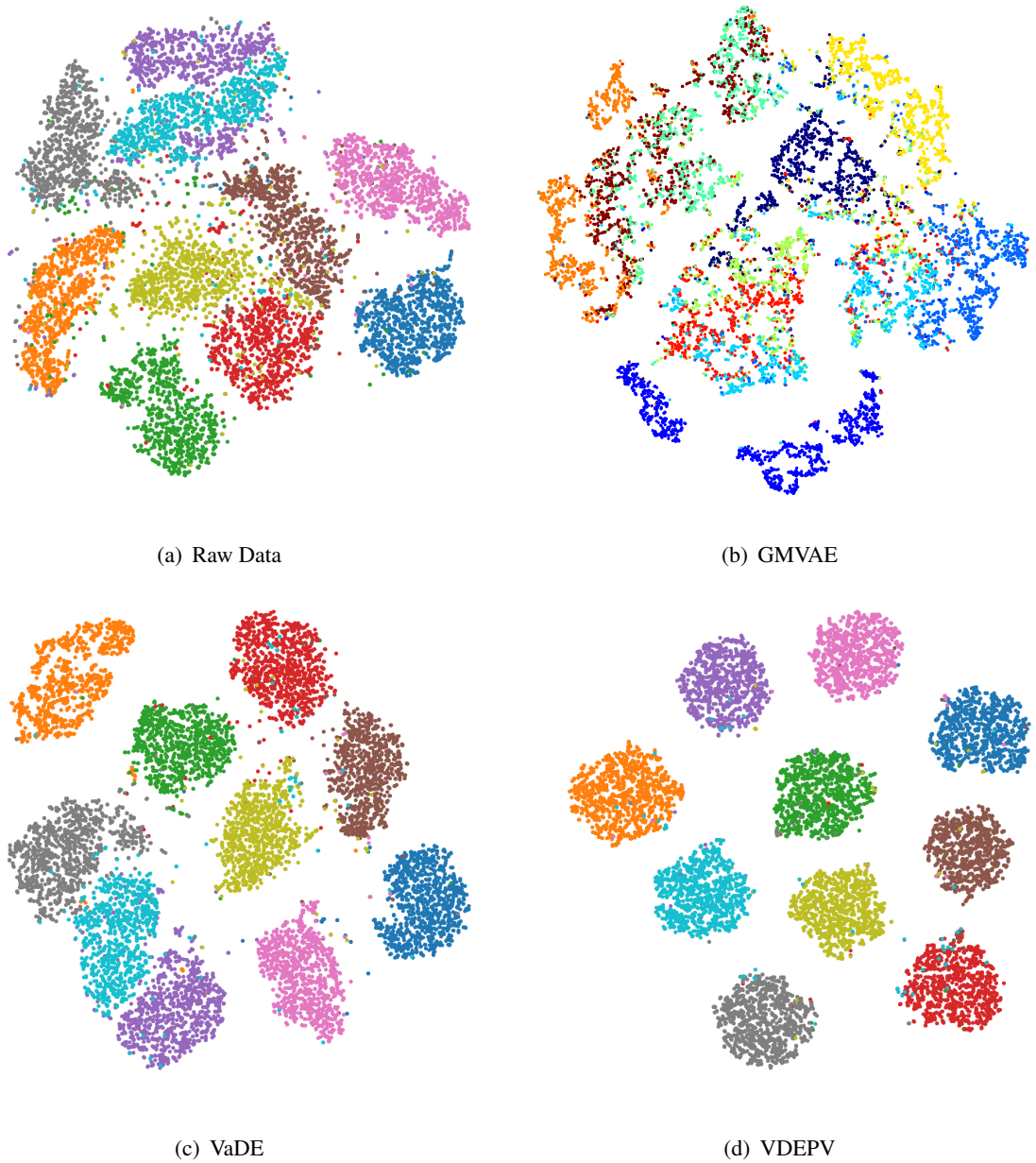
To assess the clustering performance of VDEPV, we employ standard unsupervised evaluation metrics, including Unsupervised Clustering Accuracy (ACC), Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). We compare VDEPV with several baseline methods and state-of-the-art approaches, as shown in Table 2. The table categorizes traditional clustering, multi-stage deep clustering, simultaneous deep clustering, generative deep clustering, constrained clustering, and VDEPV into six groups. For original papers providing experimental results, we directly adopt their best results. We conduct experiments using their code on the same database for papers without experimental results but with open-source code. The plots distinguish these results by an asterisk ($*$) in the upper right corner. We represent this absence with a dash ($-$) in the plots for papers that neither provided experimental results nor opened their code.

Table 2. Clustering results of different algorithms on various datasets.

Methods	MNIST			QMNIST			EMNIST		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
K-means	49.67*	42.68*	32.42*	49.62*	42.64*	32.38*	46.16*	39.71*	29.66*
GMM	46.44*	47.79*	32.28*	54.19*	50.52*	35.97*	45.11*	41.35*	26.75*
BisectingKMeans	47.98*	40.24*	29.90*	48.14*	40.33*	30.39*	39.84*	30.47*	20.22*
AE+SC	68.37*	79.73*	63.42*	69.69*	79.12*	65.84*	68.33*	80.64*	64.58*
AE+VMFMM	75.88*	76.65*	65.29*	75.70*	77.34*	63.31*	53.84*	68.89*	45.76*
AE+GMM	80.81*	82.27*	74.94*	79.73*	84.09*	76.20*	82.84*	80.56*	73.26*
GAN-K	80.43*	75.72*	61.25*	73.38*	71.64*	56.87*	89.27*	83.11*	76.65*
VAE-K	75.13*	64.18*	57.33*	74.09*	62.99*	55.24*	67.68*	60.43*	50.34*
SENet [8]	96.8	91.8	93.1	–	–	–	72.1	79.8	76.6
DEC [13]	84.30	83.31*	79.20*	87.69*	83.36*	79.15*	93.64*	88.01*	87.07*
EDESC [16]	91.3	86.2	–	–	–	–	–	–	–
GMVAE [31]	93.22	68.26*	62.25*	64.96*	62.44*	52.37*	68.74*	63.09*	51.98*
VaDE [34]	94.06	84.51*	81.98*	91.64*	84.46*	83.14*	91.48*	85.24*	83.78*
DC-GMM [42]	96.7	91.7	93.0	96.02*	90.18*	91.52*	96.51*	91.41*	92.44*
PCKmeans [39]	49.62*	42.66*	32.38*	47.33*	42.18*	30.83*	47.11*	39.40*	28.10*
COPKmeans [40]	48.41*	42.55*	31.92*	47.22*	42.28*	30.99*	47.43*	39.97*	29.67*
PSSC [44]	84.30	76.76	54.92*	76.46*	78.35*	69.62*	76.53*	74.74*	66.59*
VDEPV	**98.25**	**95.26**	**96.17**	**96.49**	**91.42**	**92.41**	**98.62**	**96.08**	**96.97**

According to the results in the table, apart from traditional clustering methods, SENet, EDESC, DC-GMM, and PSSC yield superior results compared to other methods within the same group. Simultaneous deep clustering and generative deep clustering outperformed multi-stage deep clustering (excluding SENet). SENet demonstrates the ability to learn data representations for self-expression, leading to higher clustering performance than multi-stage deep clustering, which only learns general deep representations. This indicates that simultaneously learning deep representations and clustering algorithms is an efficient approach for deep clustering, especially for generative deep clustering. The

main reason lies in the significant advantage brought by simultaneously optimizing deep models and cluster analysis.



(a) Raw Data (b) GMVAE

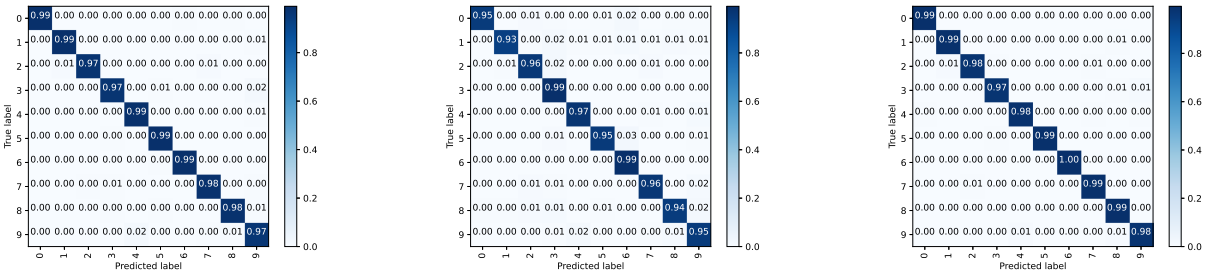
(c) VaDE (d) VDEPV

Figure 4. Visualization of embedding space on the MNIST dataset. (a) Space of original data. (b) Embedding space of GMVAE. (c) Embedding space of VaDE. (d) Embedding space of VDEPV.

For VAE, both DC-GMM and VaDE are based on GMM priors. It is noteworthy that DC-GMM belongs to the generative deep clustering group and the pairwise constraints clustering group. The superiority of DC-GMM over VaDE can be explained by introducing pairwise constraints, guiding deep models to learn representations more suitable for clustering in the latent space. The pairwise constraints used in DC-GMM require additional ground truth labels. However, under the same architecture

and unsupervised conditions, we achieved better performance than DC-GMM without using labeled data to tune any hyperparameters in VDEPV. This indicates that, for VAE models, the VMF mixture distribution prior may be more suitable for clustering tasks than a GMM prior.

To better understand the combined effects of VMF mixture distribution priors and pairwise constraints, we employed the t-SNE method to visualize some original image samples from the MNIST handwritten dataset and partial latent variables z from three deep generative models mapped to a 2D space, as shown in Figure 4. First, the initial state of the original data samples is displayed in the first image. Second, the second image illustrates the embedding space based on GMVAE, while the third image shows the embedding space based on VaDE. Both models primarily utilize GMM. Subsequently, the fourth image visualizes the embedding space obtained by our proposed method. As shown in the figures, by imposing pairwise constraints in the latent space, samples of the same category are brought closer in the latent space, while samples of different categories are pushed apart, thereby increasing the compactness within clusters and the compactness between clusters. Separability. This change in shape makes the original data easier to distinguish and cluster in the latent space, thus providing a clearer spatial structure for subsequent cluster analysis. Furthermore, since the latent space is reshaped into a more cluster-specific shape, clustering algorithms can exploit this structure more effectively, improving clustering performance and reducing errors. This process is equivalent to providing a representation of the feature space of the original data that is more suitable for clustering, making the clustering task simpler and feasible. Therefore, by changing the shape of the latent space, this method improves the clustering performance while also improving the understanding and interpretation of the data structure.



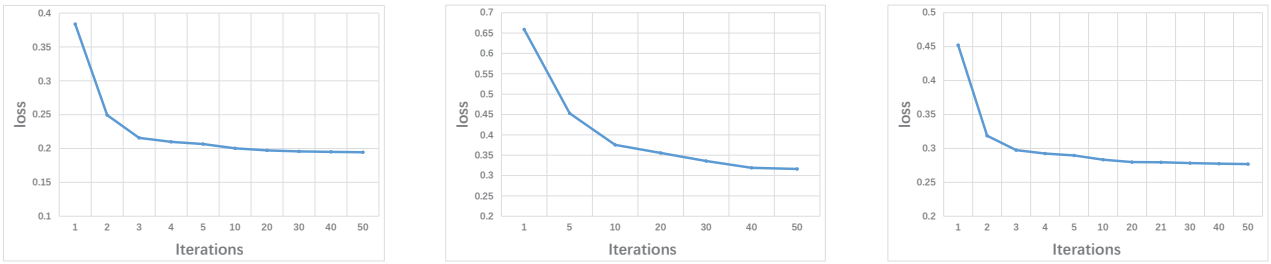
(a). MNIST **(b).** QMNIST **(c).** EMNIST

Figure 5. Confusion matrices on three datasets.

Figure 5 clearly demonstrates the outstanding clustering performance of VDEPV on the MNIST, QMNIST, and EMNIST datasets. The diagonal elements of the confusion matrix represent the number of samples correctly clustered by the model, while the off-diagonal elements represent the number of samples incorrectly clustered by the model. The numerical percentages correspond to the respective clustering accuracy or error rate. Observing the results from (a) to (c), it can be seen that VDEPV achieves higher clustering accuracy on each category and exhibits good discriminative capability between different categories. This strong performance is attributed to VDEPV's specific model prior and clustering optimization. On the MNIST data set, VDEPV demonstrated excellent clustering capabilities for simple handwritten digits. For the QMNIST dataset, which contains more complex and real-world handwritten digit samples, VDEPV can accurately distinguish categories and exhibits a low error rate. On large-scale and diverse data sets such as the EMNIST data set, VDEPV can also effec-

tively handle handwritten digits of various styles and variations and achieve reliable clustering results.

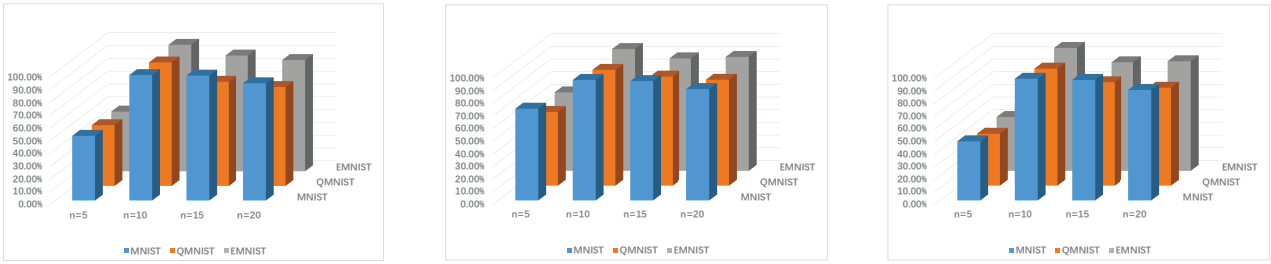
The convergence curves of VDEPV are plotted on the MNIST, QMNIST and EMNIST datasets as shown in Figure 6. By observing the convergence curve, it can be clearly seen that as the number of iterations increases, the loss of the VDEPV algorithm gradually stabilizes. This phenomenon shows that the algorithm has better convergence performance when processing data sets such as MNIST, QMNIST, and EMNIST. Of particular note is the fact that the VDEPV algorithm was able to reach a steady state after only 5 iterations when processing these datasets. This means that the algorithm converges quickly and effectively, and has potential applicability in practical applications.



(a). MNIST **(b).** QMNIST **(c).** EMNIST

Figure 6. Convergence curve of VDEPV.

In the above analysis, we follow the strategy of VaDE, treating the number of clusters as prior knowledge. In this regard, we changed the number of clusters to understand their impact on the performance of the MNIST, QMNIST, and EMNIST datasets. Since these datasets all have 10 classes, we set the number of clusters to 5, 10, 15, and 20, respectively, to see if there are no samples assigned to additional clusters or different variants within the class. As shown in Figure 7, the number of clusters is 10, and the clustering performance of VDEPV is highest in ACC, NMI, and ARI. When the number of clusters increases or decreases, the clustering performance decreases, indicating that data samples from one class are divided into different clusters.



(a). ACC **(b).** NMI **(c).** ARI

Figure 7. Comparison of clustering performance for different numbers of clusters on different datasets.

Furthermore, VDEPV exhibits excellent performance across all datasets. The validation of experimental results confirms its effectiveness in clustering tasks.

5. Conclusions

We introduce an innovative deep generative clustering model called VDEPV based on the von Mises-Fisher mixture model and pairwise constraints. The von Mises-Fisher mixture distribution prior enhances the robustness and modeling capability of VAE, particularly in modeling features mapped to the underlying hypersphere. Pairwise constraints strengthen intra-cluster samples' compactness and inter-cluster samples' separability in the spherical embedding space. Our model facilitates the joint optimization of deep representation learning and clustering through the proposed unified loss function. Compared to other advanced clustering methods, VDEPV demonstrates outstanding performance.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

We would like to express our sincere gratitude to the anonymous reviewers for their valuable and constructive suggestions on our paper.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. A. E. Ezugwu, A. M. Ikotun, O. O. Oyelade, L. Abualigah, J. O. Agushaka, C. I. Eke, et al., A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects, *Eng. Appl. Artif. Intell.*, **110** (2022), 73–89. https://doi.org/10.1016/j.engappai.2022.104743
2. S. Zhou, H. Xu, Z. Zheng, J. Chen, Z. li, J. Bu, et al., A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions, preprint, arXiv: 2206.07579. https://doi.org/10.48550/arXiv.2206.07579
3. K. A. István, F. Róbert, G. Péter, Unsupervised clustering for deep learning: A tutorial survey, *Acta Polytech. Hung.*, **15** (2018), 29–53. https://doi.org/10.12700/APH.15.8.2018.8.2
4. T. R. Davidson, L. Falorsi, N. D. Cao, T. Kipf, J. M. Tomczak, Hyperspherical variational auto-encoders, in *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, (2018), 856–865.
5. K. V. Mardia, P. E. Jupp, K. V. Mardia, *Directional Statistics*, John Wiley & Sons, 2000. https://doi.org/10.1002/9780470316979
6. J. Taghia, Z. Ma, A. Leijon, Bayesian estimation of the von-Mises Fisher mixture model with variational inference, *IEEE Trans. Pattern Anal. Mach. Intell.*, **36** (2014), 1701–1715. https://doi.org/10.1109/TPAMI.2014.2306426

7. F. Yuan, L. Zhang, J. She, X. Xia, G. Li, Theories and applications of auto-encoder neural networks: A literature survey, *Chin. J. Comput.*, **42** (2019), 203–230. https://doi.org/10.11897/SP.J.1016.2019.00203
8. S. Zhang, C. You, R. Vidal, C. Li , Learning a self-expressive network for subspace clustering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2021), 12393–12403. https://doi.org/10.1109/CVPR46437.2021.01221
9. Y. Tao, K. Takagi, K. Nakata, Clustering-friendly representation learning via instance discrimination and feature decorrelation, preprint, arXiv:2106.00131. https://doi.org/10.48550/arXiv.2106.00131
10. Z. Dang, C. Deng, X. Yang, K. Wei, H. Huang, Nearest neighbor matching for deep clustering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2021), 13693–13702. https://doi.org/10.1109/CVPR46437.2021.01348
11. M. Nasrazadani, A. Fatemi, M. Nematbakhsh, Sign prediction in sparse social networks using clustering and collaborative filtering, *J. Supercomput.*, **78** (2022), 596–615. https://doi.org/10.1007/s11227-021-03902-5
12. N. Alami, M. Meknassi, N. En-nahnahi, Y. E. Adlouni, O. Ammor, Unsupervised neural networks for automatic arabic text summarization using document clustering and topic modeling, *Expert Syst. Appl.*, **172** (2021). https://doi.org/10.1016/j.eswa.2021.114652
13. J. Xie, R. Girshick, A. Farhad, Unsupervised deep embedding for clustering analysis, in *International Conference on Machine Learning*, (2016), 478–487.
14. X. Ye, C. Wang, A. Imakura, T. Sakurai, Spectral clustering joint deep embedding learning by autoencoder, in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021. https://doi.org/10.1109/IJCNN52387.2021.9533825
15. K. Thirumoorthy, K. Muneeswaran, A hybrid approach for text document clustering using Jaya optimization algorithm, *Expert Syst. Appl.*, **178** (2021). https://doi.org/10.1016/j.eswa.2021.115040
16. J. Cai, J. Fan, W. Guo, S. Wang, Y. Zhang, Z. Zhang, Efficient deep embedded subspace clustering, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. https://doi.org/10.1109/CVPR52688.2022.00012
17. Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, X. Peng, Contrastive clustering, in *Proceedings of the AAAI Conference on Artificial Intelligence*, **35** (2021), 8547–8555. https://doi.org/10.1609/aaai.v35i10.17037
18. K. Do, T. Tran, S. Venkatesh, Clustering by maximizing mutual information across views, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2021), 9928–9938. https://doi.org/10.1109/ICCV48922.2021.00978
19. Y. Shen, Z. Shen, M. Wang, J. Qin, P. H. S. Torr, L. Shao, You never cluster alone, *Adv. Neural Inf. Process. Syst.*, **34** (2021), 27734–27746.
20. H. Zhong, J. Wu, C. Chen, J. Huang, M. Deng, L. Nie, et al., Graph contrastive clustering, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2021), 9224–9233. https://doi.org/10.1109/ICCV48922.2021.00909

21. Q. Ji, Y. Sun, J. Gao, Y. Hu, B. Yin, A decoder-free variational deep embedding for unsupervised clustering, *IEEE Trans. Neural Networks Learn. Syst.*, **33** (2021), 5681–5693. https://doi.org/10.1109/TNNLS.2021.3071275

22. W. Wang, J. Bao, S. Guo, Neural generative model for clustering by separating particularity and commonality, *Inf. Sci.*, **589** (2022), 813–826. https://doi.org/10.1016/j.ins.2021.12.037

23. J. Mirecka, M. Famili, A. Kota'nska, N. Juraschko, B. Costa-Gomes, C. Palmer, et al., Affinity-VAE for disentanglement, clustering and classification of objects in multidimensional image data, preprint, arXiv: 2209.04517. https://doi.org/10.48550/arXiv.2209.04517

24. J. Xu, Y. Ren, H. Tang, X. Pu, X. Zhu, M. Zeng, et al., Multi-VAE: Learning disentangled view-common and view-peculiar visual representations for multi-view clustering, in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, (2021), 9234–9243. https://doi.org/10.1109/ICCV48922.2021.00910

25. G. Chen, S. Long, Z. Yuan, W. Zhu, Q. Chen, Y. Wu, Ising granularity image analysis on VAE–GAN, *Mach. Vision Appl.*, **33** (2022). https://doi.org/10.1007/s00138-022-01338-2

26. E. Palumbo, S. Laguna, D. Chopard, J. E. Vog, Deep generative clustering with multimodal variational autoencoders, in *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2023.

27. L. Yang, C. Cheung, J. Li, J. Fang, Deep clustering by gaussian mixture variational autoencoders with graph embedding, in *Proceedings of the AAAI Conference on Artificial Intelligence*, (2019), 6440–6449.

28. Y. Liang, Z. Lin, F. Yuan, H. Zhang, L. Wang, W. Wang, Towards polymorphic adversarial examples generation for short text, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. https://doi.org/10.1109/ICASSP49357.2023.10095612

29. K. Yonekura, Quantitative analysis of latent space in airfoil shape generation using variational autoencoders, *Trans. JSME*, **87** (2021). https://doi.org/10.1299/transjsme.21-00212

30. T. Nishida, T. Endo, Y. Kawaguchi, Zero-Shot domain adaptation of anomalous samples for semi-supervised anomaly detection, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023. https://doi.org/10.1109/ICASSP49357.2023.10095897

31. D. Nat, A. M. M. Pedro, G. Marta, C. H. L. Matthew, H. Salimbeni, A. Kai, et al., Deep unsupervised clustering with gaussian mixture variational autoencoders, preprint, arXiv:1611.02648. https://doi.org/10.48550/arXiv.1611.02648

32. W. Wu, Y. Liu, M. Guo, Constructing training distribution by minimizing variance of risk criterion for visual category learning, in *2012 19th IEEE International Conference on Image Processing*, (2012), 101–104. https//doi.org/10.1109/ICIP.2012.6466805

33. W. Wu, Y. Liu, W. Zeng, M. Guo, C. Wang, X. Liu, Effective constructing training sets for object detection, in *2013 IEEE International Conference on Image Processing*, (2013), 3377–3380. https//doi.org/10.1109/ICIP.2013.6738696

34. Z. Jiang, Y. Zheng, H. Tan, B. Tang, H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, preprint, arXiv:1611.05148. https://doi.org/10.48550/arXiv.1611.05148

35. W. Liu, Y. Zhang, X. Li, Z. Liu, B. Dai, T. Zhao, et al., Deep hyperspherical learning, *Adv. Neural Inf. Process. Syst.*, **30** (2017).

36. Q. Li, W. Fan, Mixture density hyperspherical generative adversarial networks, in *Proceedings of the 2022 6th International Conference on Innovation in Artificial Intelligence*, (2022), 31–37. https://doi.org/10.1145/3529466.3529475

37. L. Yang, W. Fan, N. Bouguila, Deep clustering analysis via dual variational autoencoder with spherical latent embeddings, *IEEE Trans. Neural Networks Learn. Syst.*, **34** (2021), 6303–6312. https://doi.org/10.1109/TNNLS.2021.3135460

38. W. Fan, H. Huang, C. Liang, X. Liu, S. Peng, Unsupervised meta-learning via spherical latent representations and dual VAE-GAN, *Appl. Intell.*, **53** (2023), 22775–22788. https://doi.org/10.1007/s10489-023-04760-9

39. S. Basu, A. Banerjee, R. Mooney, Active semi-supervision for pairwise constrained clustering, in *Proceedings of the 2004 SIAM International Conference on Data Mining*, (2004), 333–344. https://doi.org/10.1137/1.9781611972740.31

40. K. Wagstaff, C. Cardie, S. Rogers, S. Schroedl, Constrained k-means clustering with background knowledge, in *Proceedings of the Eighteenth International Conference on Machine Learning*, **1** (2001), 577–584.

41. J. Goschenhofer, B. Bischl, Z. Kira, ConstraintMatch for semi-constrained clustering, in *2023 International Joint Conference on Neural Networks (IJCNN)*, (2023). https://doi.org/10.1109/IJCNN54540.2023.10191186

42. L. Manduchi, K. Chin-Cheong, H. Michel, S. Wellmann, J. E. Vogt, Deep conditional gaussian mixture model for constrained clustering, *Neural Inf. Process. Syst.*, **34** (2021), 11303–11314.

43. S. E. Hajjar, F. Dornaika, F. Abdallah, Multi-view spectral clustering via constrained nonnegative embedding, *Inf. Fusion*, **78** (2021), 209–217. https://doi.org/10.1016/j.inffus.2021.09.009

44. J. Lv, Z. Kang, X. Lu, Z. Xu, Pseudo-Supervised deep subspace clustering, *IEEE Trans. Image Process.*, **30** (2021), 5252–5263. https://doi.org/10.1109/TIP.2021.3079800

45. L. Bai, J. Liang, Y. Zhao, Self-Constrained spectral clustering, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 5126–5138. https://doi.org/10.1109/TPAMI.2022.3188160

46. C. Hinojosa, E. Vera, H. Arguello, A fast and accurate similarity-constrained subspace clustering algorithm for hyperspectral image, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **14** (2021), 10773–10783. https//doi.org/10.1109/JSTARS.2021.3120071

AIMS Press