International Journal of Machine Learning and Cybernetics (2024) 15:4575–4588
https://doi.org/10.1007/s13042-024-02173-9

ORIGINAL ARTICLE

Deep generative clustering methods based on disentangled representations and augmented data

Kunxiong Xu[1] · **Wentao Fan**[2] · **Xin Liu**[1]

Received: 20 June 2023 / Accepted: 8 April 2024 / Published online: 28 April 2024

Abstract

This paper presents a novel clustering approach that utilizes variational autoencoders (VAEs) with disentangled representations, enhancing the efficiency and effectiveness of clustering. Traditional VAE-based clustering models often conflate generative and clustering information, leading to suboptimal clustering performance. To overcome this, our model distinctly separates latent representations into two modules: one for clustering and another for generation. This separation significantly improves clustering performance. Additionally, we employ augmented data to maximize mutual information between cluster assignment variables and the optimized latent variables. This strategy not only enhances clustering effectiveness but also allows the construction of latent variables that synergistically combine clustering information from original data with generative information from augmented data. Through extensive experiments, our model demonstrates superior clustering performance without the need for pre-training, outperforming existing deep generative clustering models. Moreover, it achieves state-of-the-art clustering accuracy on certain datasets, surpassing models that require pre-training.

Keywords Clustering · Variational autoencoder (VAE) · Disentangling module · Mutual information · Augmented data

1 Introduction

The task of unsupervised clustering aims to partition data into distinct categories using unsupervised methods. This approach provides a solution to the challenge of relying on a large amount of labeled data for classification tasks [33]. Traditional techniques, such as K-means [40] and probabilistic mixture models [10, 13–15, 44], have been developed for unsupervised clustering. However, directly clustering the original data often leads to suboptimal results due to the presence of numerous irrelevant factors that impact the clustering process [63]. Additionally, as the dataset size increases, the training time of the model also escalates rapidly.

Deep generative models, including autoencoders (AEs), variational autoencoders (VAE) [31], and generative adversarial networks (GANs) [21], have demonstrated remarkable success across diverse domains by effectively extracting meaningful information from raw data using compact latent spaces. As a result, these models have received significant attention and have been applied in various files [4, 25, 27, 37, 39, 50, 51]. In the realm of unsupervised clustering, VAEs, in particular, have been extensively employed. A VAE is a sophisticated type of deep generative model that combines variational inference [9, 11, 12] with deep neural networks. However, previous VAE-based clustering approaches optimized both the generation and clustering components within the same framework, leading to entanglement of various information in the latent variables, where clustering and generating factors interacted. Moreover, several powerful clustering models required pre-training with stacked autoencoders, imposing significant demands on computing resources and storage space. To address the aforementioned challenges, FSVAE [58] proposed a disentanglement strategy for the latent variable in D dimensions. This strategy involves dividing the latent variable into two parts. The initial D_1 dimensions are assigned to the generation module, which follows a Gaussian prior distribution.

✉ Wentao Fan
wentaofan@uic.edu.cn

[1] Department of Computer Science and Technology, Huaqiao University, Xiamen, China

[2] Guangdong Provincial Key Laboratory IRADS and Department of Computer Science, Beijing Normal University-Hong Kong Baptist University United International College, Zhuhai, China

The remaining D_2 dimensions form the clustering module, which follows a Student's t mixture model (STMM) prior [62]. It is important to note that $D = D_1 + D_2$, and the clustering module benefits from the use of augmented data to enhance its effectiveness.

The strategic use of augmented data can markedly improve clustering accuracy. Prior models [29, 57] have demonstrated a simplistic or constrained approach to leveraging augmented data, thereby limiting their effectiveness in guiding clustering tasks. In contrast, FSVAE utilizes augmented data more robustly, applying basic constraints to the latent space, cluster assignment variables, and encoders. However, these models predominantly depend on the mean square error loss function, which restricts latent and cluster assignment variables to numerical similarities. Our proposed method advances this by employing mutual information maximization. It uniquely uses cluster assignment variables to shape latent variables, thereby generating innovative latent constructs. This technique enriches the information used during training for both cluster assignment and latent variables, potentially enhancing clustering performance.

The limitations of previous models, specifically their underutilization of augmented data, have led to suboptimal clustering effectiveness. To address this, our research introduces a novel unsupervised clustering model, focused on image clustering, leveraging the FSVAE framework. Our primary innovation involves the application of mutual information maximization, as outlined in Ji et al [28]. Through a streamlined objective function, we aim to enhance clustering effectiveness by maximizing mutual information between cluster assignment variables in both original and augmented data. Further, by leveraging augmented data and cluster assignment variables, our model optimizes the latent variables, as suggested in Haeusser et al [24]. This optimization not only extracts more informative features in the latent space but also significantly improves clustering performance. Finally, our approach introduces a unique method that merges the clustering module of the original data with the generation module of the augmented data. This synthesis results in a new type of latent variable, enabling the model to ignore extraneous information not pertinent to clustering, thereby further refining clustering effectiveness.

The main contributions of this work can be summarized as follows:

- We introduce the mutual information maximization technique to optimize the model's cluster assignment variables by the augmented data, thereby improving the model's robustness and clustering effectiveness.
- We constrain latent variables by using augmented data and clustering assigning variables, thereby improving the robustness of the model and the effectiveness of clustering.
- We propose a new latent variable construction method to help the model ignore irrelevant information, thereby improving the robustness of the model and the effectiveness of clustering.

The remainder of this paper is organized as follows. Section 2 presents an overview of existing work that utilizes the proposed model. Section 3 provides the necessary background knowledge related to our model. In Sect. 4, we introduce our deep generative clustering method for VAE, which includes disentangling modules and the utilization of augmented data. The experimental results obtained using our proposed model are presented in Sect. 5. Finally, Sect. 6 concludes the paper, summarizing the main findings and contributions.

2 Related works

Unsupervised deep clustering models have exhibited impressive performance in various clustering tasks. Among these models, deep unsupervised generative clustering approaches, such as AE, VAE, and GAN, have achieved notable success in both clustering and data reconstruction tasks.

One widely recognized AE-based deep generative clustering model is Deep Embedding Clustering (DEC) [54]. DEC employs the K-means algorithm in the pre-training stage to generate cluster centers for each cluster. These cluster centers are then iteratively optimized using an auxiliary distribution. However, DEC only utilizes a complete AE model during the pre-training stage, discarding the decoder in the subsequent training stage, and solely utilizing the encoder to optimize the cluster centers. Improved Deep Embedding Clustering (IDEC) [22] addresses the limitations of DEC by simultaneously training the reconstruction loss and clustering loss in the AE framework. This joint optimization enhances the assignment of clustering labels, enables learning of discriminative clustering features, and preserves the local structure. Adding the decoder back to the model during training leads to improved clustering results. Another DEC-based model, Deep Convolutional Embedding Clustering (DCEC) [23], incorporates convolutional neural networks (CNNs) to enhance clustering performance through improved feature extraction capabilities. The DSSEC model, as detailed in Cai et al [1], innovatively combines a sparse autoencoder with DEC for enhanced cluster analysis. Concurrently, the DCN algorithm, introduced in Yang et al [56], represents an autoencoder (AE)-based clustering approach that principally utilizes the K-means algorithm for

cluster formation. A notable development in this field is the DEPICT model [20], which marks a significant advancement by employing a softmax layer, derived from stacked AE pre-training, for predicting cluster assignments, thereby yielding considerable improvements in performance. Furthermore, the DECCA framework, as proposed in Diallo et al [6], adopts a contractive learning methodology to cultivate more effective latent variables, ACe/DeC model [46] ventures into categorizing the information gleaned by latent variables, effectively distinguishing between cluster-specific and shared information spaces, TSAE model [16] integrates Teacher-Student models and Autoencoders for cluster analysis. Lastly, the IMDGC [61] integrates hierarchical generative adversarial networks and mutual information maximization to improve clustering effectiveness.

Variational Deep Embedding (VaDE) [29] is a deep generative clustering model based on the VAE framework. Unlike VAE, which utilizes a standard Gaussian distribution as the prior, VaDE employs a Gaussian mixture model as the prior distribution. S3VDC [2] is a generative clustering model that builds upon VaDE. It incorporates initial γ training to optimize the pre-training of VaDE. Drawing inspiration from β-VAE [25], S3VDC introduces periodic β annealing to promote disentanglement in the VaDE model, allowing it to capture more informative representations. S3VDC also adopts mini-batch Gaussian Mixture Model (GMM) initialization to enhance scalability and employs inverse min-max transformation to mitigate NaN (Not-a-Number) issues during training. The vMF-VaDE model, as elucidated by Yang et al. [59] has demonstrated exceptional performance on clustering across various datasets. FSVAE [58] extends the VAE model by introducing a distinct treatment of clustering and generation information in the data. It utilizes the Student's t-Mixture model as the prior for the clustering module. FSVAE incorporates a bi-augmentation module to enhance training stability and notably achieves optimal performance without the need for pre-training. Other VAE-based generative clustering models include GMVAE [7] and DSCDAN [60]. In addition to the previously mentioned models, the field of deep generative clustering encompasses GAN-based approaches such as clusterGAN [47] and Va-GAN [57]. These models integrate a GAN component to enhance clustering effectiveness. Another notable model is Dual-AAE [19], which extends the Adversarial Autoencoder (AAE) [42] framework.

Autoencoder-based models including DEC, IDEC, DCEC, DCN, DEPICT, DECCA, and ACe/DeC, have shown notable success in clustering, with a primary emphasis on feature extraction. Unlike these models, VAE-based clustering approaches, such as VaDE, S3VDC, and GMVAE, generally incorporate Gaussian mixture models as priors, thereby enhancing clustering effectiveness. However, a limitation arises in these models' susceptibility to collapse, particularly when not utilizing the Student's t-distribution. For instance, the vMF-VaDE model, despite achieving optimal results, necessitates pre-training and is especially prone to collapse due to its reliance on the vMF distribution. In contrast, GAN-based models like clusterGAN, Va-GAN, and Dual-AAE leverage adversarial networks and employ strategies such as WGAN-GP to mitigate model collapse.

The primary limitation of the aforementioned models is their methodology of performing cluster analysis on the entire latent variable space. Differing from this approach, FSVAE introduces an innovative disentanglement strategy, segregating latent variables into two distinct categories: clustering variables and generated variables. Furthermore, FSVAE employs a Student's t-mixture model to prevent model collapse, a technique that has proven to significantly improve clustering effectiveness. Despite these advancements, FSVAE relies primarily on a simple mean square error loss function to direct the clustering of latent variables using augmented data, which could be seen as a constraint on its potential. In contrast, our model expands the utility of augmented data by implementing mutual information maximization. This enhancement not only constrains latent variables but also facilitates the construction of new latent variables, thereby allowing augmented data to more effectively guide the clustering process within the model.

Motivated by the works of Ji et al [28]; Haeusser et al [24], we incorporate the technique of maximizing mutual information in our generative model and introduce constraints on the latent variables. Furthermore, building upon the FSVAE framework mentioned earlier, we propose a novel method for constructing latent variables that allows the model to disregard irrelevant information for clustering. These three approaches collectively enhance the robustness and effectiveness of our model.

3 Preliminary

3.1 Variational autoencoder

The VAE is a neural network architecture that employs a Gaussian distribution as the prior for the latent space. This choice provides VAE with enhanced capabilities compared to the AE, such as the ability to estimate prediction uncertainties [17]. The primary objective of optimizing the VAE model can be formulated as follows:

$$\mathcal{L}(\theta,\phi;x) = E_{q\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p_\theta(z)), \tag{1}$$

where $\mathbb{E}[\cdot]$ denotes the expectation evaluation. In Eq. 1, the first term is commonly referred to as the reconstruction error, while the second term represents the regularization term, defined by the Kullback–Leibler (KL) divergence.

3.2 VAE with Gaussian mixture prior

The Gaussian mixture model has found applications in various fields [32, 38, 49] and can be utilized as a prior for the VAE.

For a given dataset X, we assume that the data x is generated by a random process. Moreover, for any potential embedding z of the data x, we consider it to follow a Gaussian Mixture Model (GMM) with K clusters. This GMM serves as the prior for the VAE model [29]. In the VAE's generation process with a GMM prior, we can generate z from the GMM distribution, which is defined by

$$p(z|y) = \mathcal{N}(z|\mu_y, \sigma_y^2), \tag{2}$$

where μ_y and σ_y^2 represent the parameters of the Gaussian distribution for cluster y. The variable y follows a categorical distribution $Cat(\pi)$, where $p(y) = Cat(\pi)$, and $Cat(\pi)$ denotes the categorical distribution parameterized by π.

By employing the GMM as the prior for the VAE, the final optimization objective can be expressed as:

$$\begin{aligned}\mathcal{L}(x) = &\frac{1}{L}\sum_{l=1}^{L}\sum_{d=1}^{D} x_d \log(\mu_{xd}^{(l)} + \sigma_{xd}^{2(l)}\varepsilon_d^{(l)}) \\ &+ (1-x_d)\log(1-(\mu_{xd}^{(l)} + \sigma_{xd}^{2(l)}\varepsilon_d^{(l)})) \\ &- \frac{1}{2}\sum_{y=1}^{K} q(y|x) \\ &\sum_{j=1}^{J}\left(\log\sigma_{yj}^2 + \frac{\vec{\sigma}_j^2}{\sigma_{yj}^2} + \frac{(\vec{\mu}_j - \mu_{yj})^2}{\sigma_{yj}^2}\right) \\ &+ \sum_{y=1}^{K} q(y|x)\log\frac{\pi_y}{q(y|x)} \\ &+ \frac{1}{2}\sum_{j=1}^{J}(1+\log\vec{\sigma}_j^2),\end{aligned} \tag{3}$$

where L represents the number of Monte Carlo samples. $\vec{\mu}$, $\vec{\sigma}^2$ are parameters of the Gaussian distribution modeled by the encoder. D denotes the dimensionality of x, $\mu_x^{(l)}$ and $\sigma_x^{(l)}$. J is the dimensionality of the parameters μ_y, σ_y, $\vec{\mu}$, and $\vec{\sigma}^2$. The cluster assignment variable $q(y|x)$ can be obtained using the SGVB estimator as:

$$q(y|x) = \frac{1}{L}\sum_{l=1}^{L}\frac{p(y)\mathcal{N}\left(z^{(l)}|\mu_y,\sigma_y^2\right)}{\sum_{y'=1}^{K} p(y')\mathcal{N}\left(z^{(l)}|\mu_{y'},\sigma_{y'}^2\right)}. \tag{4}$$

where $\mathcal{N}(\cdot)$ represents the Gaussian distribution, and $p(y)$ is the prior probability of cluster y.

3.3 VAE with student's t mixture prior

The Student's t mixture model is widely employed across diverse domains [5, 18, 45] and offers a robust alternative to the Gaussian mixture model, particularly when handling outliers.

In Sect. 3.2, we introduced a deep generative clustering model that employed a GMM as the prior for the VAE. While this approach demonstrated significant clustering capabilities, it suffered from performance degradation in the presence of outliers. To overcome this limitation, we utilize the Student's t distribution as a more robust alternative, owing to its heavy-tailed characteristics. By incorporating the Student's t mixture model (STMM) into the VAE framework, we can enhance the model's robustness [62]. The probability density function (PDF) of the Student's t distribution is defined by

$$\begin{aligned}\mathcal{S}(x|\mu,\sigma^2,v) = &\frac{\Gamma(\frac{v+1}{2})}{\Gamma(\frac{v}{2})\sqrt{\pi v\sigma}} \\ &\left(1+\frac{(z-\mu)^T\sigma^{-2}(z-\mu)}{v}\right)^{-\frac{v+1}{2}},\end{aligned} \tag{5}$$

where v, μ, and σ are the distribution parameters, and Γ denotes the gamma function. We adopt the reparameterization trick for the Student's t distribution [43, 48],which is similar to that of the Gaussian distribution:

$$z = \mu + \sqrt{\frac{v}{2\widetilde{z}}}\sigma\cdot\varepsilon, \tag{6}$$

where $\varepsilon \sim \mathcal{N}(0,1)$, and $\widetilde{z} \sim \mathcal{G}(\frac{v}{2},1)$. Here, $\widetilde{z}$ is obtained using the reparameterization trick for the Gamma distribution as

$$\widetilde{z} = \left(\frac{v}{2}-\frac{1}{3}\right)\left(1+\frac{\varepsilon}{\sqrt{\frac{9v}{2}-3}}\right)^3, \tag{7}$$

where $\varepsilon \sim \mathcal{N}(0,1)$.

In the VAE model using STMM as the prior, a notable distinction from Sect. 3.2 is the assumption that z is generated by STMM, thus conforming to the Student's t distribution in the latent space. This assumption necessitates the encoder to provide an additional parameter v, which is unique to the Student's t distribution.

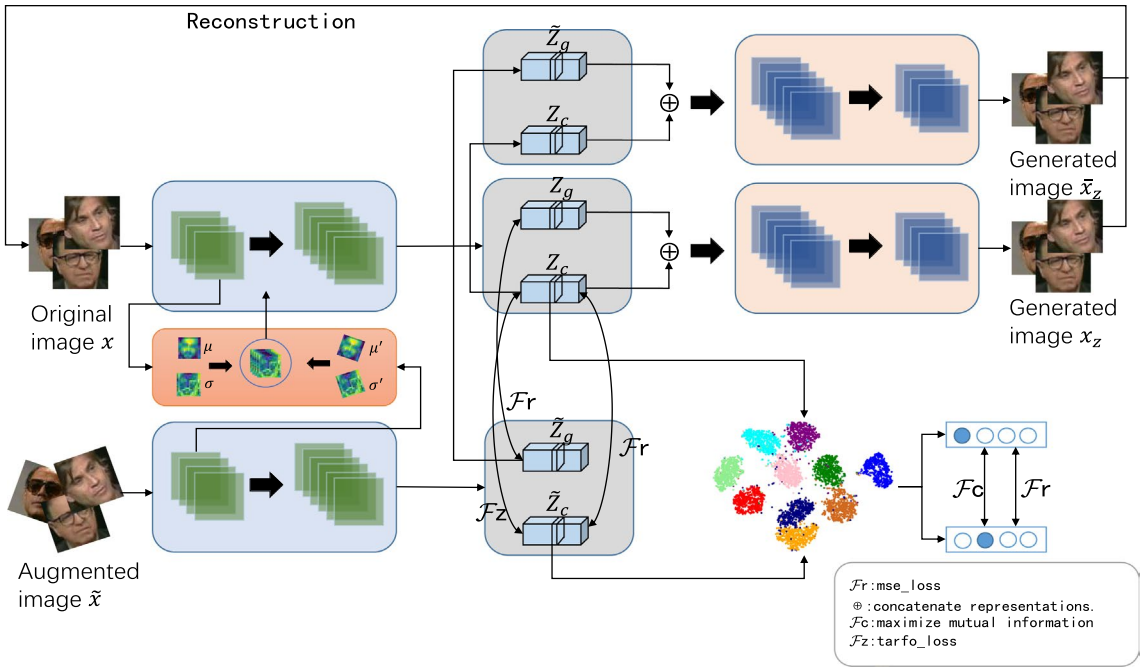


Fig. 1 The network architecture of the proposed model. Initially, we employ both feature augmentation and data augmentation techniques to enrich the training dataset. The model uniquely integrates a disentanglement strategy for latent variables, segregating them into two modules: Z_g for general features and Z_c for cluster-specific features. To enforce constraints, we utilize mean square error loss on both Z and $\tilde{Z}$, the latter representing augmented latent variables. Similar constraints are also applied to the cluster assignment variables. The function $\mathcal{F}_c$ is designated for maximizing mutual information between the original and augmented data, while $\mathcal{F}_z$ focuses on constraining the latent variables. Additionally, we introduce a novel latent variable, synthesized by combining the latent variables of both the original and augmented images, which is depicted at the top of the figure

4 The proposed model

This section outlines our clustering model, which integrates VAE with disentangled representations to enhance clustering performance. Figure 1 provides a visual representation of the model's architecture. The methodology is systematically detailed across several subsections: Sect. 4.1 delves into the process of disentangling latent variables within the model. Section 4.2 describes the bi-augmentation modules employed in FSVAE, enhancing its efficacy. In Sect. 4.3, we articulate our approach for maximizing mutual information, ensuring a robust correlation between original and augmented data. Section 4.4 explains how we impose constraints on the latent variables using cluster assignment variables and augmented data to refine the clustering process. Finally, Sect. 4.5 presents our innovative latent variables, offering both the mathematical formulation and the algorithmic structure of the model, underscoring its practical and theoretical contributions.

4.1 Disentanglement of latent representations

Various disentangling methods have been proposed in the past, aiming to ensure that a single latent variable corresponds to a single factor [3, 8, 25, 30, 64]. In this subsection, we provide a comprehensive elucidation of the methodology employed to disentangle the model's latent variables into two distinct modules: clustering and generative. Additionally, we present the training formulas specifically tailored for the disentangled clustering model.

In the VAE model, the presence of a decoder responsible for generating reconstructed data suggests that the latent variable z contains valuable information for the generation process. Expanding on this idea, we can consider the latent variable z as composed of two distinct modules: the clustering module and the generation module [58]. The generation module z_g follows a Gaussian distribution, while the clustering module z_c aligns with the description in Sect. 3.3, utilizing the Student's t Mixture Model (STMM) as its prior. The concatenation operation $\oplus$ combines the two modules as follows:

$$z = z_g \oplus z_c. \tag{8}$$

By allowing each module to fulfill its respective role, the clustering effectiveness and generation capabilities of the model can be enhanced.

Based on the discussions in Sects. 3.1, 3.2, and 3.3, the final optimization objective of the model can be expressed as

$$\begin{aligned}\mathcal{L}_{fs}(x) &= E_{q(z_g,z_c,y|x)}\left[\log \frac{p(x,z_g,z_c,y)}{q(z_g,z_c,y|x)}\right] \\ &= E_{q(z_g,z_c,y|x)}[\log p(x|z_g,z_c)] \\ &\quad - D_{KL}(q(z_g|x)||p(z_g)) \\ &\quad - D_{KL}(q(z_c,y|x)||p(z_c,y)),\end{aligned} \tag{9}$$

where the optimization objective consists of three parts: the reconstruction error, the loss of the generation module, and the loss of the clustering module with the STMM as the prior.

4.2 Bi-augmentation module

In this subsection, we present a thorough analysis of the bi-augmentation modules implemented in the FSVAE. These modules play a pivotal role in augmenting the clustering capabilities of the model.

The bi-augmentation module consists of two components: feature augmentation and data augmentation, both of which play a crucial role in the clustering process. Firstly, we introduce a transformed image, denoted as $\widetilde{x}$, into the model. Since x and $\widetilde{x}$ represent the same image, we assume that their corresponding latent variables, $\widetilde{z}$ and z, are similar or identical. To align z and $\widetilde{z}$, data augmentation is applied. Similarly, data augmentation is also performed on the cluster assignment variables $\gamma = q(y|x)$ and $\widetilde{\gamma} = q(y|\widetilde{x})$. The loss function for data augmentation can be formulated as follows:

$$\mathcal{L}_{aug} = \mathcal{C}(z_c,\widetilde{z}_c) + \mathcal{C}(z_g,\widetilde{z}_g) + \mathcal{C}(\gamma,\widetilde{\gamma}), \tag{10}$$

where $\mathcal{C}$ denotes the mean square error loss function.

Secondly, we enhance the model by splitting the CNN into two parts and incorporating feature normalization [35, 36] on the output of the preceding CNN segment. The feature normalization operation is defined as:

$$\overline{h} = \widehat{\mu} + \widehat{\sigma}\frac{\vec{h} - \vec{\mu}}{\vec{\sigma}}, \tag{11}$$

where $\overline{h}$ represents the input to the subsequent CNN segment, while $\vec{h}$ denotes the output of the image from the preceding CNN segment. The parameters $\vec{\mu}$ and $\vec{\sigma}$ correspond to the mean and variance of $\vec{h}$, respectively. Similarly, the augmented image undergoes the same processing to obtain $\widehat{\mu}$ and $\widehat{\sigma}$ for feature normalization.

4.3 Augmented mutual information

In this subsection, we delve into a comprehensive explanation of the techniques employed to maximize mutual information between the original and augmented data. By emphasizing the synergy between original and augmented data, we aim to elucidate how this strategy significantly enhances the model's performance.

While the augmentation techniques discussed in Sect. 4.2 provide some assistance for clustering, there remain several deficiencies that require improvement. In Sect. 4.2, the data augmentation module is employed to enforce a mean square error loss between the cluster assignment variables of the original and augmented data. This aligns with the intuitive perception that the original and augmented data represent the same underlying information, and therefore, their cluster assignment variables should be consistent. Inspired by the work of Ji et al. [28], we posit that maintaining consistency in the information of cluster assignment variables between original and augmented data is crucial. Consequently, our research further investigates the maximization of mutual information in this context. Specifically, our focus lies in enhancing the mutual information of the cluster assignment variables. This approach is designed to facilitate the discovery of more refined and effective cluster assignment representations, thereby improving the overall clustering performance of our model.

In our model, for each data point x_i in the dataset $X = \{x_1, x_2, ..., x_N\}$, we obtain the probability of data x_i belonging to each category using Equation (4). The augmentation of mutual information aims to maximize the alignment of cluster assignment variables between the original and augmented data. The augmented mutual information loss is defined by

$$\mathcal{L}_I = \max I(\gamma, \gamma'), \tag{12}$$

where γ represents the cluster assignment variable for the original data and γ' represents the cluster assignment variable for the augmented data. Since γ can be treated as a discrete random variable distributed across K categories, we can directly compute $I(\gamma, \gamma')$ as

$$I(\gamma,\gamma') = \sum_{y=1}^{K}\sum_{y'=1}^{K} P_{yy'} \cdot \log \frac{P_{yy'}}{P_y \cdot P_{y'}}. \tag{13}$$

For any two pairs of samples (x, x'), the conditional joint distribution is given by $P(\gamma = y, \gamma' = y'|x, x') = \gamma^y \gamma^{y'}$. By marginalizing over the dataset, we can obtain the joint distribution P of the cluster assignment variables using the following equation

$$P = \frac{1}{n}\sum_{i=1}^{n} \gamma_i \cdot \gamma_i^{'T}, \tag{14}$$

where n denotes the number of sample pairs.

We considered the order of sample pairs as (x, x'). However, it is also valid to consider the order as (x', x). Consequently, we obtain the following joint distribution:

$$P_{yy'} = (P + P^T)/2, \tag{15}$$

where the marginal distributions P_y and $P_{y'}$ can be obtained by summing the rows and columns of $P_{yy'}$. Maximizing the mutual information of the cluster assignment variables helps to identify the similarities between samples, thereby improving the accuracy of clustering.

4.4 Augmenting latent variables

In Sect. 4.3, we discussed the constraints imposed on the cluster assignment variables. While acknowledging the utility of this approach, we recognize that the application of mean square error loss to latent variables, in isolation, may not yield substantial improvements in clustering performance. Inspired by Haeusser et al [24], this subsection introduces a strategy to apply more stringent constraints on the latent variables of the data. This enhanced approach is aimed at further refining the model's clustering capabilities, drawing upon advanced methodologies to achieve more significant improvements.

For a sample pair (x_i, x_j), we construct the loss as follows:

$$\mathcal{L}_{trafo} = |1 - z_{c,i}^T z_{c,j}^{'} - \ell_2(\gamma_i, \gamma_j^{'})|, \tag{16}$$

where ℓ_2 represents the cross-entropy loss function. Since we only constrain the clustering modules in the latent variables, we refer to them as z_c in the equation. The latent variable z_c is derived from the original data, while z_c' is obtained from the augmented data.

Unlike in Sect. 4.3, where the sample pair consists of the original data and its augmented counterpart, $\mathcal{L}_{trafo}$ has two optimized forms. In the first case, we set x_j to be the augmented data of x_i. In this scenario, when the cluster assignment variables are similar, the latent variables z_i and z_j of the image become more alike, thereby imposing a stronger constraint on the latent variables. In the second case, we set x_i and x_j to correspond to different data and optimize $\mathcal{L}_{trafo}$. By doing so, we encourage the latent variables z_i and z_j to be similar when their cluster assignment variables are similar. However, if the cluster assignment variables are dissimilar, the latent variables z_i and z_j will be different. Since we believe that cluster assignment variables belonging to the same category should be similar, this approach encourages the latent variables of the same category to be more compact, while keeping the latent variables of different categories more separated.

Table 1 Experimental results of two different tarfo loss on the MNIST dataset

	ACC(↑)	NMI(↑)
Baseline	97.0	92.5
Case1	97.4	93.2
Case2	97.0	92.4

We conducted a comparison of clustering accuracy and NMI (Normalized Mutual Information) for the two cases on the MNIST dataset, and the results are presented in Table 1. In the table, "Baseline" refers to the experimental results without utilizing the trafo loss, "Case1" represents the experimental results of the first case where trafo loss is applied to the baseline model, and "Case2" corresponds to the experimental results of the second case where trafo loss is applied to the baseline model. The table clearly demonstrates that the trafo loss in the first case yields significantly better clustering performance compared to the second case. For the sake of brevity, we rewrite equation (16) in the form of the first case as

$$\mathcal{L}_t = |1 - z_c^T z_c^{'} - \ell_2(\gamma, \gamma^{'})|. \tag{17}$$

4.5 Latent variable construction

In this subsection, we provide a comprehensive description of our method for constructing new latent variables for training purposes. We also briefly discuss the rationale behind this approach and present a figure illustrating the process of creating these new latent variables. Additionally, the final formulation and algorithm of the model are outlined.

In the previous section, we employed the latent variable z, derived from the original data via the encoder, for reconstruction in the decoder. This method operates under the assumption that z contains ample information for effective

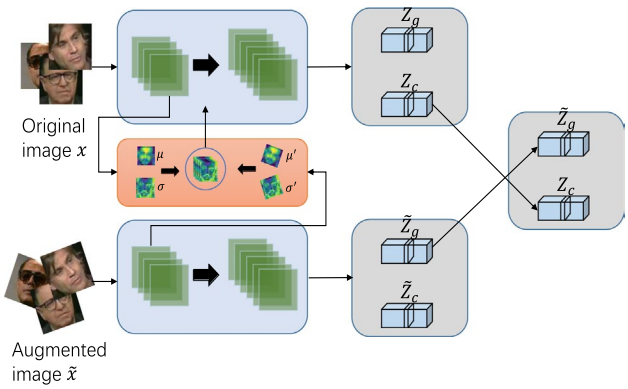


Fig. 2 New latent variables constructed on all datasets

clustering, while it overlooks the potential contributions of augmented data. However, using augmented data for reconstruction might inadvertently incorporate irrelevant information into z, which could impede clustering performance. For example, in cases where simple rotations are used for data augmentation, the latent variable may inadvertently encode rotation-related information. We hypothesize, though, that if the latent variables can assimilate a modest amount of additional information, the model could, during its training phase, learn to autonomously disregard this extraneous data along with other irrelevant factors, thus enhancing clustering accuracy. As illustrated in Fig. 2, we suggest substituting the generation component associated with the original data's latent variable with that of the augmented data's latent variable for reconstruction. This alteration enables the latent variable to encompass a broader range of information, facilitating its ability to eliminate non-essential elements for clustering and consequently improving the overall efficiency of the clustering model.

We express our newly constructed latent variables as

$$\overline{z} = \widetilde{z}_g \oplus z_c. \tag{18}$$

Similar to the latent variables in the original model, we utilize $\overline{z}$ in the decoder to obtain $\overline{x}$, and then compute the cross-entropy loss between $\overline{x}$ and x. Our objective function is defined as

$$\mathcal{L}_r = \ell(x, \overline{x}), \tag{19}$$

where ℓ represents the cross-entropy loss function. Notably, our proposed method for constructing latent variables exclusively relies on augmented data, making it applicable to other models with disentangled latent variables.

The overall loss function of the model can be expressed as

$$\mathcal{L}_{total} = -\mathcal{L}_{fs}(x) + \mathcal{L}_{aug} + \lambda_I \mathcal{L}_I + \lambda_t \mathcal{L}_t + \lambda_r \mathcal{L}_r, \tag{20}$$

where the hyperparameter $\lambda_I, \lambda_t, \lambda_r$ is used to tune the contribution of the loss term $\mathcal{L}_I, \mathcal{L}_t, \mathcal{L}_r$.

The algorithm of our model through the SGVB estimator is shown in Algorithm 1.

Algorithm 1 Training steps

Input:
Training data: X; learning rate lr.
Output:
Parameters of decoder ϕ_d, encoder ϕ_e and SMM ϕ_m.

while not converged **do**
  Sample a data x from X
  Augmented data $\widetilde{x} = aug(x)$
  $z_g, z_c \leftarrow e(x, \phi_e)$
  $\widetilde{z}_g, \widetilde{z}_c \leftarrow e(\widetilde{x}, \phi_e)$
  Calculate $\mathcal{L}_{aug}$ using (10)
  Calculate $\mathcal{L}_I$ using (12)
  Calculate $\mathcal{L}_t$ using (17)
  Generate data $x_z \leftarrow d(z, \phi_d)$
  Generate data $\overline{x}_z \leftarrow d(\overline{z}, \phi_d)$
  Calculate $\mathcal{L}_{fs}(x)$ using (9)
  Calculate $\mathcal{L}_r$ using (19)
  $\phi_d, \phi_e, \phi_m \leftarrow \text{Adam}(\mathcal{L}_{total}, \text{lr})$
end while

Table 2 Summary of the benchmark datasets

Data Sets	#Samples	#Dimensions	#Clusters
MNIST	70000	1*28*28	10
USPS	9298	1*16*16	10
GTSRB	15540	3*32*32	10
YTF	4733	3*32*32	20
F-MNIST	70000	1*28*28	10

Table 3 Network settings for all datasets

Layers	Dncoder	Eecoder
1	Input z	Input $X \in R$
2	FC, ReLU	4×4 conv 64 stride 2, pad 1 BN, ReLU
3	FC, BN, ReLU	4×4 conv 128 stride 2, pad 1 BN, ReLU
4	4×4 conv 64 stride 2, pad 1 BN, ReLU	FC, BN, ReLU
5	4×4 upconv 1 stride 2, pad 1, Sigmoid	FC

Table 4 Latent dimension for all data sets

	MNIST	USPS	GTSRB	YTF	F-MNIST
z_c	7	7	25	25	10
z_g	3	3	5	5	5

5 Experimental results

In this section, we present the experimental results obtained from evaluating our approach on five distinct image datasets. We assess the performance using two metrics. The hardware setup used for the experiments includes an Intel i7-11700 CPU running at 2.50GHz, a GeForce RTX 3060 GPU, and 32GB of memory. The software environment consists of Python version 3.6 and PyTorch version 1.10.0.

5.1 Datasets

MNIST: LeCun et al [34] The MNIST dataset comprises 70,000 handwritten digit images, featuring 10 distinct classes ranging from '0' to '9'. The resolution is 28×28 pixels. To augment the dataset, we apply random rotations within the range of -25 to 25 degrees, adjust the image contrast, and use PIL.ImageChops.offset() to randomly shift the image by up to 0.35 times its size.

USPS: The USPS dataset comprises 9,298 handwritten digit images, covering the numbers '0' to '9'. The resolution is 16×16 pixels. Unlike the MNIST dataset, this dataset includes random rotations ranging from -35 to 35 degrees.

GTSRB: Houben et al [26] The GTSRB dataset comprises images of 43 different categories of traffic signs. For our experiments, we select 10 categories, resulting in a total of 15,540 images. The resolution is 28×28 pixels. Similar to the USPS dataset, the GTSRB dataset includes random rotations ranging from -180 to 180 degrees.

YTF: Wolf et al [52] The YTF dataset comprises 4,733 face images in 20 categories. The resolution is 32×32 pixels.

Table 5 The ACC results by different methods without pre-training

Methods	MNIST ACC(↑)	USPS ACC(↑)	GTSRB ACC(↑)	YTF ACC(↑)	F-MNIST ACC(↑)
K-means Lloyd [40]	53.2	66.8	30.2	34.3	51.1
GMM McLachlan et al [44]	55.3	53.0	33.1	34.8	52.3
AE + k-means	81.8	68.4	50.3	60.4	57.1
DEC Xie et al [54]	84.3	77.1	54.1	50.3	58.8
IDEC Guo et al [22]	88.1	77.3	55.4	52.2	59.2
GMVAE Dilokthanakul et al [7]	88.5	81.2	57.1	59.3	59.6
ClusterGAN Mukherjee et al [47]	95.2	90.1	61.2	66.0	63.3
S3VDC Cao et al [2]	95.3	87.1	58.0	63.6	61.1
DECCA Diallo et al [6]	96.4	77.3	55.9	59.7	60.1
TSAE Fei et al [16]	95.3	94.0	66.7	67.4	64.4
IMDGC Yang et al [61]	97.0	93.5	67.5	68.7	**66.5**
FSVAE Yang et al [58]	97.0	92.3	67.3	67.8	65.3
Ours	**98.1**	**96.9**	**68.9**	**69.8**	65.8

The augmented image method used in this dataset aligns with the approach employed for the GTSRB dataset.

Fashion-MNIST: Xiao et al [53] The Fashion-MNIST dataset comprises 70,000 images, encompassing 10 categories of fashion items such as T-shirts, coats, sandals, etc. The resolution is 28×28 pixels. The augmentation procedure employed for this dataset aligns with that of the MNIST dataset.

Detailed information about these datasets is shown in Table 2.

Table 6 The NMI results by different methods without pre-training

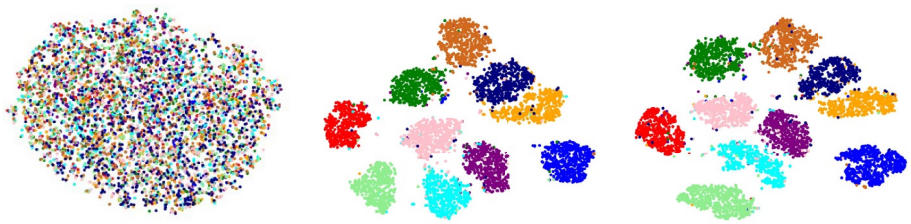
Methods	MNIST NMI(↑)	USPS NMI(↑)	GTSRB NMI(↑)	YTF NMI(↑)	F-MNIST NMI(↑)
K-means Lloyd [40]	51.6	45.0	34.1	41.0	39.2
GMM McLachlan et al [44]	50.1	62.1	36.0	41.1	40.9
AE + k-means	72.5	65.2	54.2	64.0	56.7
DEC Xie et al [54]	81.2	80.3	56.3	54.1	51.0
IDEC Guo et al [22]	82.7	79.9	56.0	58.3	51.0
GMVAE Dilokthanakul et al [7]	85.8	73.0	61.4	70.1	57.0
ClusterGAN Mukherjee et al [47]	89.0	87.2	63.0	74.3	64.0
S3VDC Cao et al [2]	90.2	84.1	63.3	71.8	61.0
DECCA Diallo et al [6]	90.7	80.5	60.1	68.7	63.9
TSAE Fei et al [16]	88.1	88.2	68.2	76.2	**65.0**
IMDGC Yang et al [61]	92.5	88.5	69.2	77.7	64.8
FSVAE Yang et al [58]	92.5	87.1	68.9	76.6	63.7
Ours	**94.8**	**92.5**	**70.9**	**78.5**	63.3

Values in bold represent the best performance

Table 7 The performance comparison of different methods with or without the pre-training step, where the values in brackets denote the performance obtained without using pre-training

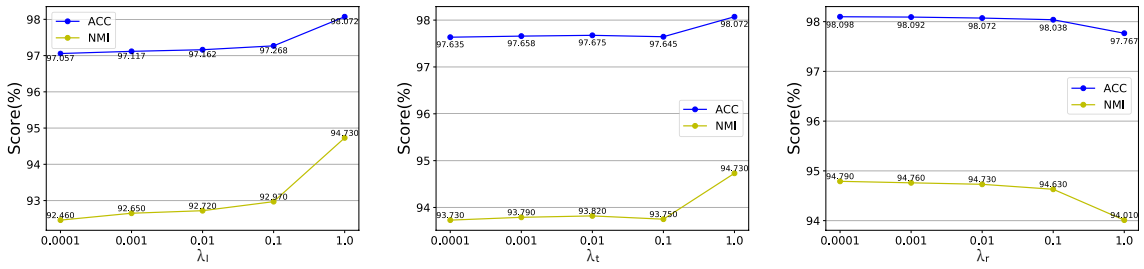
Methods	MNIST ACC(↑)	USPS ACC(↑)	GTSRB ACC(↑)	YTF ACC(↑)	F-MNIST ACC(↑)
DCEC	89.0(85.3)	79.0(73.1)	52.2(46.1)	57.3(53.2)	58.2(52.3)
DCN	91.2(83.0)	73.4(67.4)	51.3(46.2)	55.0(50.1)	57.3(50.5)
VaDE	94.5(84.2)	80.0(72.8)	55.0(49.3)	60.1(55.0)	62.9(58.3)
DEPICT	96.5(87.3)	96.4(88.4)	59.3(53.3)	62.1(55.3)	64.2(60.1)
DSCDAN	97.8(84.4)	87.2(80.3)	62.3(55.1)	69.1(62.3)	66.2(60.1)
Dual-AAE	97.6(88.4)	85.3(76.4)	60.0(53.3)	66.1(59.3)	67.1(61.2)
Va-GAN	95.8(89.0)	86.4(80.5)	59.4(55.0)	64.2(59.7)	67.0(60.3)
ACe/Dec	98.0(94.0)	88.0(73.0)	68.0(56.0)	67.3(64.1)	65.0(62.5)
vMF-VaDE	**98.7**(96.2)	93.2(90.1)	67.5(60.7)	**70.3**(66.4)	**67.2**(62.9)
Ours	98.1	**96.9**	**68.9**	69.8	65.8

Values in bold represent the best performance



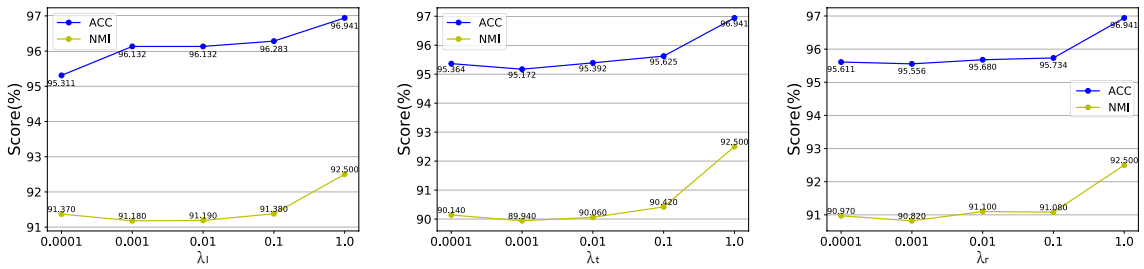
(a) Epoch #0 (b) Epoch #200 (c) Epoch #500

Fig. 3 Results of clustering visualization on the MNIST dataset. We selected epoch0, epoch200 and the final results for display



(a) MNIST (b) MNIST (c) MNIST

Fig. 4 This figure shows the ACC and NMI of the model under different hyperparameters on the MNIST dataset



(a) USPS (b) USPS (c) USPS

Fig. 5 This figure shows the ACC and NMI of the model under different hyperparameters on the USPS dataset

Table 8 Ablation experiments of different components on the MNIST and USPS dataset

Method	MNIST		USPS	
	ACC(↑)	NMI(↑)	ACC(↑)	NMI(↑)
Baseline	97.0	92.5	92.3	87.1
Ours w/o $\mathcal{L}_t$+$\mathcal{L}_r$	97.5	93.4	94.1	89.3
Ours w/o $\mathcal{L}_l$+$\mathcal{L}_r$	97.4	93.2	95.1	90.2
Ours w/o $\mathcal{L}_l$+$\mathcal{L}_t$	97.1	92.6	94.0	88.3
Ours w/o $\mathcal{L}_r$	97.8	94.1	95.4	90.7
Ours w/o $\mathcal{L}_t$	97.9	94.4	95.3	89.9
Ours w/o $\mathcal{L}_l$	97.5	93.5	96.4	91.5
Ours	98.1	94.8	96.9	92.5

5.2 Implementation details

We evaluate the performance of our approach using two metrics: clustering accuracy (ACC) and normalized mutual information (NMI) [55]. Our model is implemented in PyTorch and optimized using the Adam optimizer with hyperparameters $\beta_1 = 0.5$ and $\beta_2 = 0.99$. The learning rate for all datasets is set to 2e-3 and decays by 95% every 10 epochs to prevent overfitting. We use a batch size of 64 and train the model for 500 epochs. The network architecture of our model is summarized in Table 3, and the distribution of latent variables across different datasets is provided in Table 4.

5.3 Experimental results

In this experiment, we compare our method, which does not involve pre-training, to several deep clustering algorithms, including K-means [40], GMM [44], AE+K-means, DEC [54], IDEC [22], GMVAE [7], ClusterGAN [47], S3VDC [2], DECCA [2], TSAE [16] and IMDGC [61]. Additionally, we assess the effectiveness of our newly added loss by comparing it to the baseline model FSVAE [58]. The experimental results across multiple experiments are reported in Tables 5 and 6. With the exception of ACC and NMI on the F-MNIST dataset, our model achieves the highest performance in terms of ACC and NMI among the models without pre-training. Moreover, it outperforms the baseline model FSVAE, demonstrating the effectiveness of our newly added loss.

Moreover, we conduct a comprehensive comparison between our proposed model and nine advanced clustering methods that incorporate pre-training. These methods include DCEC [23], DCN [56], VaDE [29], Va-GAN

Table 9 The ACC performance with different outlier ratios on the MNIST-test dataset

Outlier ratios	5%	10%	15%
VaDE	86.1(77.4)	79.2(72.3)	74.3(68.1)
Dual-AAE	93.2(83.1)	83.3(76.4)	77.4(73.2)
VaGAN	94.4(86.1)	89.3(80.1)	87.3(79.8)
GMVAE	79.4	73.3	70.1
S-VaDE	80.3	75.2	73.4
S3VDC	88.1	80.4	77.3
FSVAE	94.7	87.3	86.8
Ours	95.6	88.3	88.0

Table 10 The Running time results by different methods on the MNIST dataset

Methods	Running time
DCEC (Guo et al [23])	68.3S/epoch
DCN (Yang et al [56])	68.4S/epoch
VaDE (Jiang et al [29])	72.7S/epoch
DEPICT (Ghasedi Dizaji et al [20])	70.8S/epoch
DSCDAN (Guo et al [22])	73.6S/epoch
Dual-AAE (Ge et al [19])	82.1S/epoch
Va-GAN (Yang et al [57])	75.0S/epoch
ACe/Dec (Miklautz et al [46])	70.2S/epoch
vMF-VaDE (Yang et al [59])	74.4S/epoch
Ours	**66.4S/epoch**

[57], DEPICT [20], Dual-AAE [19], DSCDAN [22], ACe/Dec [46], and vMF-VaDE [59]. The comparative results are detailed in Table 7. Notably, the performance metrics within parentheses correspond to results achieved without the use of pre-training. Our model not only matches but in some instances, surpasses the clustering accuracy of these pre-training based methods across various datasets. This underlines the robustness and effectiveness of our approach, even in the absence of pre-training.

To visualize the separation of cluster assignments among latent variables during training, we utilize t-SNE [41] on the entire MNIST dataset, as shown in Fig. 3.

In Equation (20), our newly proposed loss function incorporates three hyperparameters: λ_I, λ_t, and λ_r. We assessed the impact of various hyperparameter settings on the model's Accuracy (ACC) and Normalized Mutual Information (NMI) using the MNIST and USPS datasets, as illustrated in Figs. 4 and 5.

In Fig. 4, our investigation into the MNIST dataset involved adjusting all three hyperparameters within the range of 0.0001 to 1. For λ_I and λ_t, we observed minimal variations in performance in the 0.0001 to 0.1 range, followed by a consistent increase, achieving optimal performance at a setting of 1. Based on these findings, we selected λ_I=1 and λ_t=1 for subsequent experiments. In contrast, λ_r displayed a decreasing trend in performance, leading us to choose λ_r=0.01 for further experimentation. Regarding the USPS dataset, as depicted in Fig. 5, we similarly adjusted the hyperparameters from 0.0001 to 1. Across the board, an increase in hyperparameter values corresponded with an upward trend in the model's performance. Consequently, we set all three hyperparameters to 1 for subsequent analyses.

To further analyze the impact of our two newly added losses on the model's clustering performance, we conduct an ablation experiment, and the results are presented in Table 8. In the table, "Baseline" refers to the original model, "Ours w/o $\mathcal{L}_t$+$\mathcal{L}_r$" indicates the absence of both augmented latent variables and the construction of new latent variables, "Ours w/o $\mathcal{L}_I$+$\mathcal{L}_r$" indicates the absence of both augmented mutual information and the construction of new latent variables, "Ours w/o $\mathcal{L}_I$+$\mathcal{L}_t$" indicates the absence of both augmented mutual information and augmented latent variables, "Ours w/o $\mathcal{L}_r$" indicates the absence of constructing new latent variables, "Ours w/o $\mathcal{L}_t$" indicates the absence of augmented latent variables, "Ours w/o $\mathcal{L}_I$" indicates the absence of augmented mutual information, and "Ours" corresponds to the final model.

Finally, we evaluated the model's robustness to outliers. Following the approach described in FSVAE [58], we conducted experiments by introducing outliers. We compared our model against several pre-trained deep generative clustering methods, including VaDE, Dual-AAE, and VaGAN, as well as a selection of non-pre-trained deep generative clustering methods, including VaDE, Dual-AAE, VaGAN, GMVAE, S-VaDE, and S3VDC. Additionally, we compared our results with the baseline FSVAE to highlight the enhanced robustness of our model. Table 9 presents the ACC values on the MNIST test dataset with 5%, 10%, and 15% outliers. We observed that our model exhibits superior robustness compared to all non-pre-trained models, surpasses certain pre-trained models, and outperforms the baseline model FSVAE in terms of robustness.

The computational complexity of our proposed method is $O(m \cdot n \cdot s)$, where m represents the number of epochs, n quantifies the ratio of the total data volume to the batch size, and s signifies the execution time of the encoder and decoder neural networks. As illustrated in Table 10, a comparative analysis of the epoch-wise runtime between our model and existing models is presented. To ensure a fair comparison, we standardized the encoder and decoder across all models. A distinctive advantage of our model is its elimination of the need for pre-training. This attribute significantly contributes to its efficiency, particularly in achieving a reduced runtime when compared with other methods.

6 Conclusion

In this work, we introduce a novel generative clustering approach based on FSVAE aimed at significantly enhancing clustering performance. Our methodology encompasses several key advancements. First, we optimize clustering efficiency by maximizing the cluster assignment variables for both original and augmented data. Second, by integrating augmented data and cluster assignment variables, we impose more rigorous constraints on the latent variables, thereby achieving improved clustering results. Finally, our approach includes the innovative creation of new latent variables, which are then utilized in the reconstruction process to further boost the clustering effect. Comparative experimental results affirm that our model outperforms existing methods, demonstrating its superior capabilities in clustering tasks.

Acknowledgements The completion of this work was supported by the National Natural Science Foundation of China (62276106), the Guangdong Provincial Key Laboratory IRADS (2022B1212010006, R0400001-22) and the UIC Start-up Research Fund (UICR0700056-23).

Data Availability The MNIST data sets is available at: [http://yann.lecun.com/exdb/mnist/]. The USPS data set is available at: [https://www.kaggle.com/datasets/bistaumanga/usps-dataset]. The GTSRB data set is available at: [https://benchmark.ini.rub.de/gtsrb_news.html] The YTF data set is available at: [https://www.cs.tau.ac.il/~wolf/ytfaces/] The F-MNIST data set is available at: [https://www.kaggle.com/datasets/zalando-research/fashionmnist].

References

1. Cai J, Wang S, Guo W (2021) Unsupervised embedded feature learning for deep clustering with stacked sparse auto-encoder. Expert Syst Appl 186(115):729
2. Cao L, Asadi S, Zhu W, Schmidli C, Sjöberg M (2020) Simple, scalable, and stable variational deep clustering. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp 108–124
3. Chen RT, Li X, Grosse RB, Duvenaud DK (2018) Isolating sources of disentanglement in variational autoencoders. vol 31
4. Chen X, Duan Y, Houthooft R, Schulman J, Sutskever I, Abbeel P (2016) Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in neural information processing systems, vol 29
5. Dai Q, Zhao C, Zhao S (2022) Variational bayesian student'st mixture model with closed-form missing value imputation for robust process monitoring of low-quality data. IEEE Trans Cybern pp 1–14
6. Diallo B, Hu J, Li T, Khan GA, Liang X, Zhao Y (2021) Deep embedding clustering based on contractive autoencoder. Neurocomputing 433:96–107
7. Dilokthanakul N, Mediano PA, Garnelo M, Lee MC, Salimbeni H, Arulkumaran K, Shanahan M (2016) Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648
8. Dupont E (2018) Learning disentangled joint continuous and discrete representations. vol 31
9. Fan W, Bouguila N (2014) Variational learning for dirichlet process mixtures of dirichlet distributions and applications. Multimed Tools Appl 70(3):1685–1702
10. Fan W, Hou W (2022) Unsupervised modeling and feature selection of sequential spherical data through nonparametric hidden markov models. Int J Mach Learn Cybern 13(10):3019–3029
11. Fan W, Sallay H, Bouguila N, Bourouis S (2016) Variational learning of hierarchical infinite generalized dirichlet mixture models and applications. Soft Comput 20(3):979–990
12. Fan W, Bouguila N, Bourouis S, Laalaoui Y (2018) Entropy-based variational bayes learning framework for data clustering. IET Image Proc 12(10):1762–1772
13. Fan W, Bouguila N, Du JX, Liu X (2019) Axially symmetric data clustering through dirichlet process mixture models of watson distributions. IEEE Trans Neural Netw Learn Syst 30(6):1683–1694
14. Fan W, Yang L, Bouguila N (2022) Unsupervised grouped axial data modeling via hierarchical bayesian nonparametric models with watson distributions. IEEE Trans Pattern Anal Mach Intell 44(12):9654–9668
15. Fan W, Zeng L, Wang T (2023) Uncertainty quantification in molecular property prediction through spherical mixture density networks. Eng Appl Artif Intell 123(106):180
16. Fei Z, Gong H, Guo J, Wang J, Jin W, Xiang X, Ding X, Zhang N (2023) Image clustering: Utilizing teacher-student model and autoencoder. IEEE Access 11:104,846-104,857
17. Feng K, Qin H, Wu S, Pan W, Liu G (2020) A sleep apnea detection method based on unsupervised feature learning and single-lead electrocardiogram. IEEE Trans Instrum Meas 70:1–12
18. Gao X, Huang W, Liu Y, Zhang Y, Zhang J, Li C, Bore JC, Wang Z, Si Y, Tian Y et al (2023) A novel robust student's t-based granger causality for eeg based brain network analysis. Biomed Signal Process Control 80(104):321
19. Ge P, Ren CX, Dai DQ, Feng J, Yan S (2019) Dual adversarial autoencoders for clustering. IEEE Trans Neural Netw Learn Syst 31(4):1417–1424
20. Ghasedi Dizaji K, Herandi A, Deng C, Cai W, Huang H (2017) Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In: Proceedings of the IEEE international conference on computer vision, pp 5736–5745
21. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville AC, Bengio Y (2014) Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol 27
22. Guo X, Gao L, Liu X, Yin J (2017a) Improved deep embedded clustering with local structure preservation. In: Ijcai, pp 1753–1759
23. Guo X, Liu X, Zhu E, Yin J (2017b) Deep clustering with convolutional autoencoders. In: International conference on neural information processing, pp 373–382
24. Haeusser P, Plapp J, Golkov V, Aljalbout E, Cremers D (2019) Associative deep clustering: Training a classification network with no labels. In: Pattern Recognition: 40th German Conference, GCPR 2018, Stuttgart, Germany, October 9-12, 2018, Proceedings 40, pp 18–32
25. Higgins I, Matthey L, Pal A, Burgess C, Glorot X, Botvinick M, Mohamed S, Lerchner A (2017) beta-vae: Learning basic visual concepts with a constrained variational framework. In: International conference on learning representations
26. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C (2013) Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In: The 2013 international joint conference on neural networks (IJCNN), pp 1–8
27. Hu Q, Zhang G, Qin Z, Cai Y, Yu G, Li GY (2023) Robust semantic communications with masked vq-vae enabled codebook. IEEE Transactions on Wireless Communications p 1

28. Ji X, Henriques JF, Vedaldi A (2019) Invariant information clustering for unsupervised image classification and segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 9865–9874
29. Jiang Z, Zheng Y, Tan H, Tang B, Zhou H (2016) Variational deep embedding: An unsupervised and generative approach to clustering. arXiv preprint arXiv:1611.05148
30. Kim H, Mnih A (2018) Disentangling by factorising. In: International Conference on Machine Learning, pp 2649–2658
31. Kingma DP, Welling M (2013) Auto-encoding variational bayes. In: International Conference on Learning Representations
32. Külah E, Çetinkaya YM, Özer AG, Alemdar H (2023) Covid-19 forecasting using shifted gaussian mixture model with similarity-based estimation. Expert Syst Appl 214(119):034
33. Le Guennec A, Malinowski S, Tavenard R (2016) Data augmentation for time series classification using convolutional neural networks. In: ECML/PKDD workshop on advanced analytics and learning on temporal data, pp 3558–3565
34. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
35. Li B, Wu F, Weinberger KQ, Belongie S (2019) Positional normalization. vol 32
36. Li B, Wu F, Lim SN, Belongie S, Weinberger KQ (2021a) On feature normalization and data augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12,383–12,392
37. Li X, Kou K, Zhao B (2021b) Weather gan: Multi-domain weather translation using generative adversarial networks. arXiv preprint arXiv:2103.05422
38. Liu T, Yuan Q, Ding X, Wang Y, Zhang D (2023) Multi-objective optimization for greenhouse light environment using gaussian mixture model and an improved nsga-ii algorithm. Comput Electron Agric 205(107):612
39. Liu X, Hu Z, Ling H, Cheung YM (2021) Mtfh: A matrix tri-factorization hashing framework for efficient cross-modal retrieval. IEEE Trans Pattern Anal Mach Intell 43(3):964–981
40. Lloyd S (1982) Least squares quantization in pcm. IEEE Trans Inf Theory 28(2):129–137
41. Van der Maaten L, Hinton G (2008) Visualizing data using t-sne. J Mach Learn Res 9(11):2579–2605
42. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B (2015) Adversarial autoencoders. arXiv preprint arXiv:1511.05644
43. Marsaglia G, Tsang WW (2000) A simple method for generating gamma variables. ACM Trans Math Softw (TOMS) 26(3):363–372
44. McLachlan GJ, Lee SX, Rathnayake SI (2019) Finite mixture models. Ann Rev Stat Appl 6:355–378
45. Meitz M, Preve D, Saikkonen P (2023) A mixture autoregressive model based on student's t-distribution. Commun Stat Theory Methods 52(2):499–515
46. Miklautz L, Bauer LG, Mautz D, Tschiatschek S, Böhm C, Plant C (2021) Details (don't) matter: Isolating cluster information in deep embedded spaces. In: IJCAI, pp 2826–2832
47. Mukherjee S, Asnani H, Lin E, Kannan S (2019) Clustergan: Latent space clustering in generative adversarial networks. Proc AAAI Conf Artif Intell 33:4610–4617
48. Naesseth C, Ruiz F, Linderman S, Blei D (2017) Reparameterization gradients through acceptance-rejection sampling algorithms. In: Artificial Intelligence and Statistics, pp 489–498
49. Niknam G, Molaei S, Zare H, Clifton D, Pan S (2023) Graph representation learning based on deep generative gaussian mixture models. Neurocomputing 523:157–169
50. Satheesh C, Kamal S, Mujeeb A, Supriya M (2021) Passive sonar target classification using deep generative β-vae. IEEE Signal Process Lett 28:808–812
51. Sevgen E, Moller J, Lange A, Parker J, Quigley S, Mayer J, Srivastava P, Gayatri S, Hosfield D, Korshunova M, et al (2023) Prot-vae: Protein transformer variational autoencoder for functional protein design. bioRxiv pp 2023–01
52. Wolf L, Hassner T, Maoz I (2011) Face recognition in unconstrained videos with matched background similarity. CVPR 2011:529–534
53. Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747
54. Xie J, Girshick R, Farhadi A (2016) Unsupervised deep embedding for clustering analysis. In: International conference on machine learning, pp 478–487
55. Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, pp 267–273
56. Yang B, Fu X, Sidiropoulos ND, Hong M (2017) Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In: International conference on machine learning, pp 3861–3870
57. Yang L, Fan W, Bouguila N (2022) Clustering analysis via deep generative models with mixture models. IEEE Trans Neural Netw Learn Syst 33(1):340–350
58. Yang L, Fan W, Bouguila N (2022) Robust unsupervised image categorization based on variational autoencoder with disentangled latent representations. Knowl-Based Syst 246(108):671
59. Yang L, Fan W, Bouguila N (2023) Deep clustering analysis via dual variational autoencoder with spherical latent embeddings. IEEE Trans Neural Netw Learn Syst 34(9):6303–6312
60. Yang X, Deng C, Zheng F, Yan J, Liu W (2019) Deep spectral clustering using dual autoencoder network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 4066–4075
61. Yang X, Yan J, Cheng Y, Zhang Y (2023) Learning deep generative clustering via mutual information maximization. IEEE Trans Neural Netw Learn Syst 34(9):6263–6275
62. Zhang Y, Fan W, Bouguila N (2019) Unsupervised image categorization based on variational autoencoder and student'st mixture model. In: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), pp 2403–2409
63. Zhu X, Zhu Y, Zheng W (2020) Spectral rotation for deep one-step clustering. Pattern Recogn 105(107):175
64. Zhu X, Xu C, Tao D (2021) Commutative lie group vae for disentanglement learning. In: International Conference on Machine Learning, pp 12,924–12,934