Pattern Recognition 140 (2023) 109530

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Cloud-VAE: Variational autoencoder with concepts embedded

Yue Liu[a,b,*], Zitu Liu[a], Shuang Li[a], Zhenyao Yu[a], Yike Guo[c], Qun Liu[d], Guoyin Wang[d]

[a] *School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*
[b] *Shanghai Engineering Research Center of Intelligent Computing System, Shanghai 200444, China*
[c] *The Department of Computing, Imperial College, London SW7 2AZ, U.K*
[d] *The Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

ARTICLE INFO

Article history:
Received 10 August 2022
Revised 4 February 2023
Accepted 11 March 2023
Available online 17 March 2023

Keywords:
Variational autoencoder
Disentangled representation
Concept embedded
Cloud Model
Deep Learning Interpretability

ABSTRACT

Variational Autoencoder (VAE) has been widely and successfully used in learning coherent latent representation of data. However, the lack of interpretability in the latent space constructed by the VAE under the prior distribution is still an urgent problem. This paper proposes a VAE with understandable concept embedding named Cloud-VAE, which constructs interpretable latent space by disentangling the latent variables and considering their uncertainty based on cloud model. Firstly, cloud model-based clustering algorithm cast initial constraint of latent space into a prior distribution of concept which can be embedded into the latent space of the VAE to disentangle the latent variables. Secondly, reparameterization trick based on forward cloud transformation algorithm is designed to estimate the latent space concept by increasing the randomness of latent variables. Furthermore, variational lower bound of Cloud-VAE is derived to guide the training process to construct concepts of latent space, realizing the mutual mapping between latent space and concept space. Finally, experimental results on 6 benchmark datasets show that Cloud-VAE has good clustering and reconstruction performance, which can explicitly explain the aggregation process of the model and discover more interpretable disentangled representations.

1. Introduction

As a deep generative model, Variational Autoencoder (VAE) [1] is favored due to its good data representation and image generation capabilities, which is widely used in various research fields such as anomaly detection [2], pattern recognition [3], data generation [4–6], image classification [7–8]. VAE model learns the feature mapping from data to latent space, and then reconstructs the data to obtain data representation and image generation. Researchers have tried to optimize the structure of VAE [9–15] and improve its generation ability [16–20]. However, due to its unsupervised training process, VAE usually fails to learn the interpretable latent space, resulting in the fact that the distribution of latent space is not corresponding to the prior distribution and the posterior distribution. The lack of semantics and interpretability of the latent space will induce that VAE cannot meet human needs for reliability, controllability and credibility of the models well. Therefore, how to improve the interpretability of the VAE latent space has become an urgent problem to be solved.

Concretely, the unsupervised training process of VAE causes latent variables in latent space to show an implicitly uninterpretable cluster-like distribution [21], without semantics or even multimodal [22]. Existing methods usually introduce known prior distribution such as Gaussian mixture distribution [23], to guide the latent space to learn interpretable latent variables [23–25]. For example, Guo et al. [23] combined Gaussian mixture distribution to construct the prior and posterior distribution, which describes the original distribution information contained in data and can explicitly explain the aggregation phenomenon of VAE latent space. However, Gaussian mixture distribution is difficult to describe the randomness of concepts in the data, which leads to the inability of the prior distribution of the data to correspond to the latent space distribution of concepts. In detail, the reparameterization trick expands the range of concepts representation through random sampling to increase the randomness of concepts, e.g., the distribution of the sampled concepts is not consistent with the prior distribution. Additionally, the latent space based on a prior distribution may be hard to obtain a specific disentanglement latent distribution that is sensitive to a single concept. As an important component of VAE, the prior distribution needs to describe the concept in data and helps to disentangle the latent concepts for improving the interpretability of the latent space. Therefore, it is challenging to describe concepts in data using a prior distribution and introduce the randomness of concepts into the latent space.

[*] Corresponding author at: School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China.

https://doi.org/10.1016/j.patcog.2023.109530
0031-3203/© 2023 Elsevier Ltd. All rights reserved.

In order to obtain more interpretable latent representations, the posterior distribution generation of VAE is constrained by a series of regularized posterior distribution methods including β-VAE [26], β-TCVAE [27], Guided-VAE[28]. For instance, β-VAE [26] optimizes the posterior distribution by introducing adjustable hyperparameters to achieve disentanglement of the latent space with some interpretable feature representation. However, such models are trained towards feature representations, resulting in their poor reconstruction ability. The above methods default to the standard normal prior distribution and learn all latent variables in latent space by regularizing the posterior distribution. The standard normal prior distribution fails to contain the latent distribution information of data, which leads to a lack of interpretability in the process of aggregation and the latent space of the model. The concept embedded in the latent space is beneficial to solving the problem of separating data and knowledge, and enables a better study of the interpretability of VAE.

Therefore, in order to construct interpretable latent space to improve the interpretability of model, this paper is proposed a variational autoencoder with concept embedded named Cloud-VAE. The main idea of Cloud-VAE is to construct interpretable concepts by considering concept information and its uncertainty, which can be embedded into the latent space of the VAE model to disentangle the latent variables, then the learning objective is achieved under the guide of the prior concept distribution. According to the prior concept distribution, Cloud-VAE utilizes variational lower bound to learn the mutual mapping between latent space and concept space. Hence, the interpretability of the VAE latent space is improved. In summary, the main contributions of this work are summarized as follows:

- To describe the initial concepts in latent space, cloud model-based clustering algorithm construct the latent space of VAE by introducing the prior distribution of concept.
- To capture the randomness of latent variables, reparameterization trick based on forward cloud transformation algorithm is designed to constrain the representations range of latent space, making the latent variables interpretable and controllable.
- To obtain the optimal parameter estimates, variational lower bound of Cloud-VAE is derived to guide the training process and reconstruct concepts of latent space, so that the mutual mapping between latent space and concept space is established.
- Extensive experiments are executed on MNIST, Fashion-MNSIT, USPS, EMNIST, CIFAR-10, and COIL20 datasets. Experiments show that Cloud-VAE improves NMI metric by 22.9% and 19.9% respectively compared to deep clustering methods VaDE and GMVAE. Meanwhile, Cloud-VAE can explicitly explain the aggregation process of the model, and other interpretable latent representations are found on top of the existed.

The remainder of the paper is as follows: Section 2 introduces the related work. Section 3 presents the proposed variational autoencoder with concept embedded (Cloud-VAE). The extensive experiments of Cloud-VAE are analyzed in Section 4. Finally, Section 5 summarizes this work.

2. Related works

Deep clustering models utilize neural network to learn latent representations from high-dimensional data, and use these representations to perform clustering task. Deep clustering is divided into deep clustering models based on autoencoders [29,30,31], deep clustering models based on generative models [32], and deep clustering models based on graph convolutional neural networks [33]. As a kind of generative model which is widely used, VAE's latent space can automatically present the effect of aggregation when learning data representation. VAE fits the approximate distribution by optimizing the prior distribution and the posterior distribution of the model to construct a latent space for learning data latent variable representation, but the interpretability of VAE latent space is still a problem. Here we discuss the existing works that optimize the prior distribution or posterior distribution for improving the interpretability of VAE.

The prior distribution can describe the data distribution, which is important for the construction of latent space. Some researchers attempt to use Gaussian mixture distribution or Dirichlet prior as the prior distribution. For example, Jiang et al. [24] proposed Variational Deep Embedding (VaDE) to use Gaussian mixture model distribution as a prior distribution in which latent variables contain category information, thus explaining the reason for latent space aggregation. Dilokthanakul et al. [25] use Gaussian mixture model as the prior of latent space latent variable z to optimize variational lower bound and infer the category of z. DVAE# [21] relaxes Boltzmann machines to continuous distributions resulting distributions which can be trained as priors in DVAEs using an importance-weighted bound. Dirichlet Variational Autoencoders (DirVAE) [19] introduce Dirichlet prior for continuous latent variables, which leads to better latent representation and performance. VAE with prior distribution can explicitly explain the aggregation process of the latent space. However, prior distribution of existing methods is unable to represent the randomness of the latent variables of the VAE, which cause the latent variables entangling and difficult to explain. In this work, the prior distribution of model introduces uncertainty, which helps to describe the interaction and disentanglement of latent variables.

The posterior distribution is constrained to progressively fit the prior distribution to construct the latent space, which is of great significance to improve the interpretability of the model. By introducing a hyperparameter, β-TCVAE [27] strengthens the independent constraints to posterior distribution and encourages the separation of latent variables learned by model, but brings the disadvantage of poor quality. To solve this problem, FactorVAE [34], instead of using hyperparameters, adds a penalty term to encourage the representation of the marginal distribution factor without affecting the reconstruction quality. Ding et al. [28] proposed a controllable generative model Guided-VAE which is guided by a lightweight decoder with latent geometric transformations and principal component analysis. The model is transparent, simple and has better disentangle ability. JointVAE [35] derives from β-VAE and introduces discrete latent variables, which are combined with Gumbel-Softmax to effectively obtain class information. The above methods still retain the assumption of standard normal distribution in the setting of latent space prior distribution. However, the standard normal distribution is relatively simple, which makes the latent space difficult to reflect the data information correctly. Different from the above works, to better characterize the data information and study the interpretability of VAE, Cloud-VAE attempts to embed the concept in the latent space, constructing the mapping relationship between data and latent space concept.

3. Concept embedding variational autoencoder

Variational autoencoder with concept embedded (Cloud-VAE) is divided into three parts: Concept space Initialization, Reparameterization trick based on Forward Cloud Transformation, and Variational Lower Bound with Concept embedded, which are shown in Fig. 1. For a given data set, pre-trained autoencoder model to obtain data representations in the low-dimensional latent space first and use cloud model-based clustering method to find concepts with uncertainty information to form the concept space (Fig. 1a). Afterwards, the encoded concept parameter is introduced into the reparameterization trick to increase the randomness of latent vari-

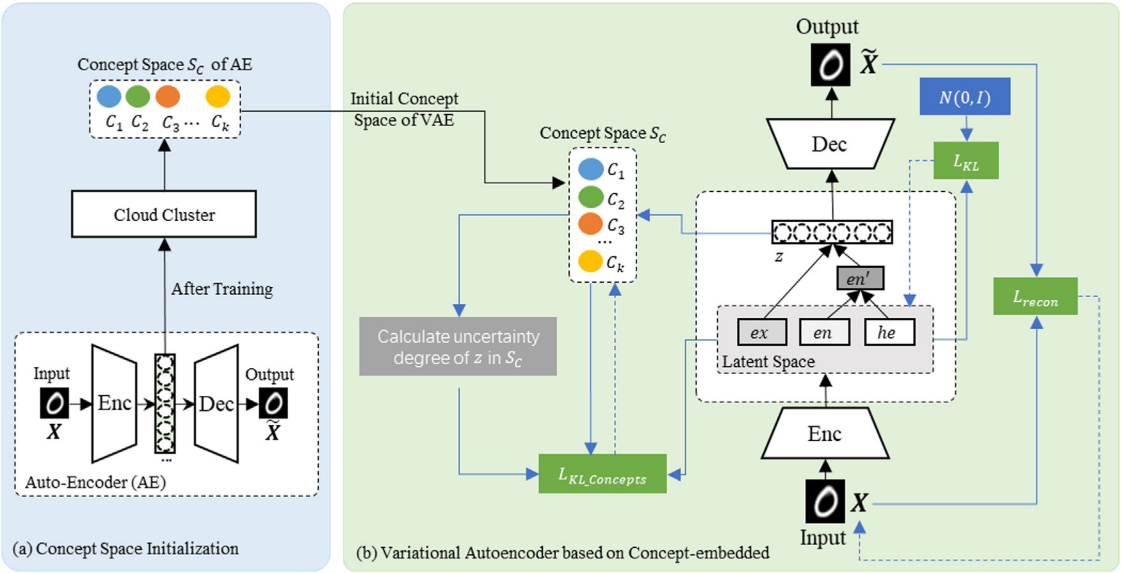


Fig. 1. Framework of Cloud-VAE.

able, and variational lower bound with concept embedded is proposed to guide the model training process to obtain the optimal parameter estimates, which enables the coarse concept parameter update and the mutual mapping between latent space and concept space (Fig. 1b).

3.1. Concept space initialization

Variational autoencoder is a generative model, and its training process is an unsupervised process. And clustering is the process of quantitative classification in latent space to help humans analyze and describe the real world. In order to describe latent space, clustering algorithms are often used. In order to obtain a better latent space of VAE, this paper learns the initial concept space from the latent variables in pre-training Autoencoder (AE).

The concept describes the general characteristics of latent variables, which has the characteristic of uncertainty. Randomness and fuzziness, as the two most basic uncertainties, have a strong correlation. Thereinto, randomness means that the definition of an event is deterministic, such as the probability of one impending event. Note that data inevitably contains uncertain information. Without considering randomness, the partition probability of data may have deterministic partition results. In order to capture the uncertain information in data, it is necessary to characterize the randomness of the object and preserve the uncertain information during optimizing the model.

Cloud model theory proposed by Li reflects the uncertainty inherent in qualitative concepts, and reveals the correlation between the fuzziness and randomness of objective things [36]. As shown in Fig. 2., the model uses numerical characteristics (expectation *Ex*, entropy *En*, hyper entropy *He*) to represent the mathematical properties of concepts, where entropy *En*, serves as the basic deterministic measure of concept granularity and hyper entropy *He* serves as the uncertainty measure of granularity.

Now, cloud model is widely used in algorithm improvement, image processing and performance evaluation [36], which indicates that Cloud model can describe the distribution and randomness of data. Therefore, in order to describe the information about the distribution of the data and the randomness contained in the latent variables, we consider cloud model as a concept representation of the prior distribution. Thus, Cloud-VAE is combines Cloud-Cluster [37] algorithm to construct an effective concept space to obtain latent feature information.

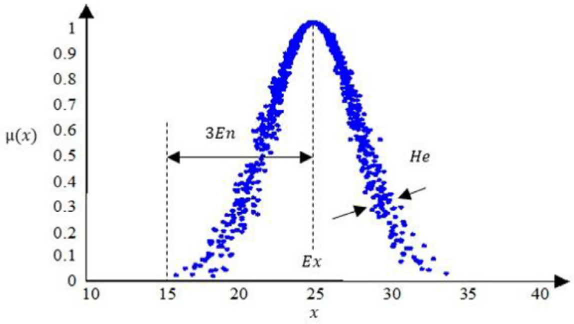


Fig. 2. Numerical characteristics of Cloud model [36].

Suppose X is an input dataset of N examples, enabling the mapping of data from a high-dimensional space to a low-dimensional latent feature space $f : X \rightarrow Z$, where Z is the latent feature space. Typically, the dimensionality of Z is much smaller than X. In order to learn concept representations of latent space, this paper considers the inclusion of randomness in uncertainty, as an important condition for modeling concepts. Then, Cloud-Cluster is used to extract concepts of latent space. According to the algorithm, the concept obtained after aggregation is represented as a concept of triples, which is represented as follows:

$$C = \{C_1, C_2, \dots, C_c\}$$
$$C_k \ : \ \{Ex_k, En_k, He_k\}, k = 1, 2, \dots, c \tag{1}$$

where *Ex* represents the most representative point in the concept, i.e., the center of concept; *En* and *He* are entropy and hyper entropy, respectively. These concepts constitute the concept space S_C.

3.2. Reparameterization trick based on forward cloud transformation

During model training, variational autoencoder not only considers the differences between the reconstructed data and the original input, but also adds variability to the latent variables through the variance parameter of the distribution function, which allows latent space to reconstruct latent variables with sensitivity based on the variance of the variables.

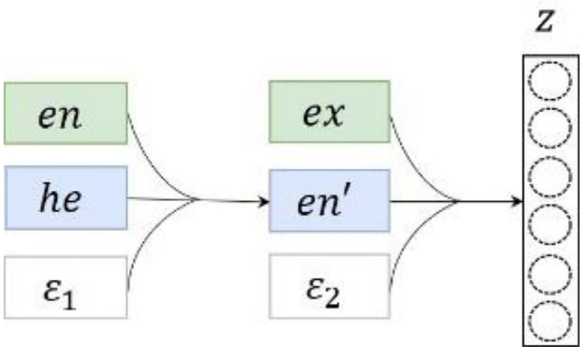


Fig. 3. Reparameterization trick of Cloud-VAE.

This paper hopes to expand the representation range of latent variables on this basis, i.e., increase the variability of latent variables, in which latent space is more likely to be similar to the representation range in the dataset. In order to expand the range, this paper proposes a reparameterization trick based on Forward Cloud Transformation to expand the boundary of points into a fuzzy boundary, hoping that VAE learns the information of latent space as much as possible. Inspired by the Forward Cloud Transformation method of cloud model, we try to make encoder of VAE calculate three parameters: expectation *Ex*, entropy *En* and hyper entropy *He*, and the three parameters constitute concept representation $C:\{Ex, En, He\}$ of latent variables concept representation. *Ex* represents the basic deterministic measure of the qualitative concept, namely the mathematical expectation in the distribution. *En* represents the uncertainty measure of the qualitative concept, which is jointly determined by randomness and fuzziness, and represents the discrete degree of the latent variables. *He* represents the measure of uncertainty of the entropy *En*, namely, hyper entropy. The reparameterization trick based on Forward Cloud Transformation is shown as formula (2) and formula (3).

$$En' = En + He \times \varepsilon_1 \tag{2}$$

$$z_{Cloud-VAE} = Ex + En' \times \varepsilon_2 \tag{3}$$

among them, $z_{Cloud-VAE}$ is the result of two combined random sampling from concept $C:\{Ex, En, He\}$ of latent variable z, ε_1 and ε_2 both obey the standard normal distribution.

The concept C computed from the encoder of Cloud-VAE cannot be back-propagated. Therefore, Cloud-VAE transforms the two-stage random sampling results into variables that conform to two standard normal distributions $N(0,1)$, in which the intermediate nodes that could not be derived or gradient propagation can be transformed into derivable, that is, $z = Ex + En \times \varepsilon_2 + He \times \varepsilon_1 \times \varepsilon_2$, as shown in Fig. 3. With two-layer reparameterization, the variational autoencoder learns parameters through gradient descent. Cloud-VAE increases randomness by introducing hyper entropy, thereby extending the possibility of sampling space, widening the representation range of latent variables and increasing the data variability.

3.3. Variational lower bound with concept embedded

Cloud-VAE is a generative model of deep clustering that models the clustering process through the data generation process, and takes the input data X, latent variables z and concept $C = \{C_1, C_2, \ldots, C_k\}$ as a joint generation distribution, which is expressed as $p_\theta(x, z, C)$, and the joint generation distribution is equivalently written as (4).

$$\begin{aligned} p(x) &= \int_z \sum_{k=1}^{c} p_\theta(x, z, C_k)dz \\ p_\theta(x, z, C_k) &= p_\theta(x|z, C_k)p_\theta(z, C_k) \\ p_\theta(x|z, C_k) &= p_\theta(x|z) = f(x|Ex_x, En_x, He_x) \\ p_\theta(z, C_k) &= \xi\, p_\theta(z|C_k) \end{aligned} \tag{4}$$

where ξ is a constant ${}^{1}/_{C}$, representing the likelihood weight of latent variable z to concept C_k. $p_\theta(x|z, C_k)$ indicates that after the model performs the re-parameterized sampling variable z, and then the reconstructed data through the decoder network. The whole process is independent of the concept C_k, thus $p_\theta(x|z, C_k) = p_\theta(x|z)$. Theoretically, the parameter estimates for the model is solved using maximized log-likelihood, as shown in Eq. (5).

$$\log p(x) = \log \int_z \sum_{k=1}^{c} p_\theta(x, z, C_k)dz \tag{5}$$

However, it is difficult to directly solve $\log p(x)$, thus Cloud-VAE uses the approximate posterior distribution $q_\phi(z, C|x_i)$ constructed by the encoder network to approximate the real posterior like other VAEs the probability distribution $p_\theta(z, C|x_i)$ can therefore be transformed into an optimization problem as shown in Eq. (6).

$$\begin{aligned} &KL\Big(q_\phi(z, C_k|x)\,\|\,p_\theta(z, C_k|x)\Big) \\ &= \int_z \sum_{k=1}^{c} q_\phi(z, C_k|x) \log \frac{q_\phi(z, C_k|x)}{p_\theta(z, C_k|x)}dz \\ &= \sum_{k=1}^{c} \mathbb{E}_{q_\phi(z, C_k|x)} \log \frac{q_\phi(z, C_k|x)p_\theta(x)}{p_\theta(z, C_k|x)p_\theta(x)} \\ &= \sum_{k=1}^{c} \mathbb{E}_{q_\phi(z, C_k|x)} \log \frac{q_\phi(z, C_k|x)p_\theta(x)}{p_\theta(x, z, C_k)} \\ &= \sum_{k=1}^{c} \mathbb{E}_{q_\phi(z, C_k|x)} \log \frac{q_\phi(z, C_k|x)}{p_\theta(x, z, C_k)} + \log p(x) \end{aligned} \tag{6}$$

In probability theory or information theory, KL divergence is a method to describe the difference between two probability distributions, which satisfies the property of an upward convex function and has non-negativity. Therefore, $KL(q_\phi(z, C_k|x)\,\|\,p_\theta(z, C_k|x))$ has non-negativity, and formula (6) is converted to solve the maximum evidence lower bound (Evidence Lower Bound, ELBO) as shown in formula (7).

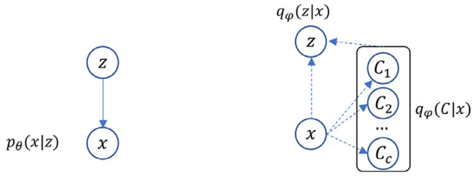
$$\begin{aligned} \log p(x) &\geq -\sum_{k=1}^{c} \mathbb{E}_{q_\phi(z, C_k|x)} \log \frac{q_\phi(z, C_k|x)}{p_\theta(x, z, C_k)} \\ &\geq \sum_{k=1}^{c} \mathbb{E}_{q_\phi(z, C_k|x)} \left[\log \frac{p_\theta(x, z, C_k)}{q_\phi(z, C_k|x)} \right] = \mathcal{L}_{ELBO}(x) \end{aligned} \tag{7}$$

In addition, according to the mean field theory, the latent variable z and the concept representation $C = \{C_1, C_2, .., C_c\}$ are independent of each other. The approximate posterior probability distribution can be written as Eq. (8).

$$q_\phi(z, C_k|x) = q_\phi(z|x)q_\phi(C_k|x) \tag{8}$$

where $q_\phi(z|x))$ is the encoder network distribution of VAE. Similar to VAE, Cloud-VAE uses a neural network to build a model, which is represented as Eq. (9).

$$q_\phi(z|x) = f(z; Ex_x, En_x, He_x) \tag{9}$$



（a） The generative process of Cloud-VAE $p_\theta(x|z,C_k)$=$p_\theta(x|z)$ （b） The inference process of Cloud-VAE $q_\phi(z,C_k|x)=q_\phi(z|x)q_\phi(C_k|x)$

Fig. 4. Generation process and inference process for Cloud-VAE.

$$g(x;\phi)\rightarrow[Ex_x,En_x,He_x]$$

where ϕ is the parameter of the neural network g, Ex_x, En_x, He_x are concept representation parameters of x. The entire generation process and variational process of Cloud-VAE are shown in Fig. 4. Fig. 4(a) is the generation process of Cloud-VAE, which uses neural network to estimate the probability distribution $p_\theta(x|z,C_k)$, the input of the generation network is z, and the output is the probability distribution $p_\theta(x|z,C_k)$. The variational inference process of VAE is shown in Fig. 4(b). The neural network is used to estimate the variational distribution $q_\phi(z,C_k|x)$. In theory, $q_\phi(z,C_k)$ is independent of x. However, since the goal of $q_\phi(z,C_k)$ is to approximate the posterior distribution $p_\theta(z,C_k|x)$ related to x, the variational density function is generally written as $q_\phi(z,C_k|x)$, where the input of the inference network is x, and the output is the variational distribution $q_\phi(z,C_k|x)$. Through the variational process and generation process of the model, latent variable z and the concept representation C_k are optimized, that is, a more flexible concept prior is updated.

$\mathcal{L}_{ELBO}(x)$ is the variational evidence lower bound of the model. In order to deduce the details of each part, without changing the original framework, we further decompose the evidence lower bound $\mathcal{L}_{ELBO}(x)$ into formula (10).

$$\begin{aligned}\mathcal{L}_{ELBO}(x)\ &=\sum_{k=1}^{c}\mathbb{E}_{q_\phi(z,C_k|x)}\left[\log\frac{p_\theta(x,z,C_k)}{q_\phi(z,C_k|x)}\right] \qquad (10)\\ &=\sum_{k=1}^{c}\mathbb{E}_{q_\phi(z,C_k|x)}\left[\log\frac{\xi\,p_\theta(x|z)p_\theta(z|C_k)}{q_\phi(z|x)q_\phi(C_k|x)}\right]\\ &=\sum_{k=1}^{c}\mathbb{E}_{q_\phi(z,C_k|x)}[\log p_\theta(x|z)+\log p_\theta(z|C_k)\\ &\quad-\log q_\phi(z|x)-\log q_\phi(C_k|x)\big]+c\log\xi\end{aligned}$$

In formula (10), the first term $p_\theta(x|z)$ is the data representation of the model under the reconstruction of the decoder network, and the second term $p_\theta(z|C_k)$ is the certainty of the latent variable z under the concept C_k, reflects the degree of uncertainty between concepts and latent variables, as shown in Eq. (11).

$$p_\theta(z|C_k)=e^{-\frac{(z-Ex_k)^2}{2\left(En_k'\right)^2}} \qquad (11)$$

where $En_k'=En_k+\varepsilon\times He_k$ represents the uncertainty degree of the concept and the fuzziness of the concept, and ε obeys the standard normal distribution $N(0,1)$. By characterizing the latent variable z to each concept, it reflects its ambiguity and randomness. The same latent variable fluctuates around its original variable after random sampling, and its probability to concept has the possibility to change. After the variational autoencoder is embedded in the concept, the range represented by the concept is also random in each training process, which increases the possibility of capturing latent variables based on the original Gaussian distribution. According to $p_\theta(z|C_k)$, the latent variable z in latent space of the current model can also be conceptually divided, that is, the prediction result can be obtained through the principle of maximizing the membership degree under the concept C.

In formula (11), the third term $q_\phi(z|x)$ is probability distribution result learned by VAE encoder network. The fourth term $q_\phi(C_k|x)$ is an approximate representation of $q_\phi(C_k|z)$, and the specific derivation is as shown in formula (12).

$$\begin{aligned}\mathcal{L}_{ELBO}(x)\ &=\sum_{k=1}^{c}\mathbb{E}_{q_\phi(z,C_k|x)}\left[\log\frac{p_\theta(x,z,C_k)}{q_\phi(z,C_k|x)}\right] \qquad (12)\\ &=\int_z\sum_{k=1}^{c}q_\phi(z|x)q_\phi(C_k|x)\log\frac{p_\theta(x|z)p_\theta(z)p_\theta(C_k|z)}{q_\phi(z|x)q_\phi(C_k|x)}dz\\ &=\int_z q_\phi(z|x)\log\frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}dz+\int_z\sum_{k=1}^{c}q_\phi(z|x)q_\phi(C_k|x)\log\frac{p_\theta(C_k|z)}{q_\phi(C_k|x)}dz\\ &=\int_z q_\phi(z|x)\log\frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}dz+\int_z\sum_{k=1}^{c}q_\phi(z|x)KL\big(q_\phi(C_k|x)\|p_\theta(C_k|z)\big)dz\end{aligned}$$

In formula (12), the term $\int_z q_\phi(z|x)\log\frac{p_\theta(x|z)p_\theta(z)}{q_\phi(z|x)}dz$ has nothing to do with the concept C, And the second term is non-negative. To maximize $\mathcal{L}_{ELBO}(x)$, $KL(q_\phi(C_k|x)\|p_\theta(C_k|z))\equiv 0$ should be satisfied. Therefore, the fourth term $q_\phi(C_k|x)$ in Eq. (10) needs to be the same $q_\phi(C_k|x)$ as much as possible. In summary, by maximizing $\mathcal{L}_{ELBO}(x)$ according to Eq. (9), training network parameters $\{\theta,\phi\}$ and concept representation $C=\{C_1,C_2,..,C_c\}$, the latent representation z can be obtained for each sample x as well as category results. The entire training process of Cloud-VAE includes two parts: reparameterization based on forward cloud transformation and variational inference with concept embedded, and its pseudocode is shown in Algorithm 1. Cloud-VAE consists of three main parts: obtaining the concept representations of latent variables, calculating the KL loss between the latent variable z, and calculating the total loss. The steps of obtaining the concept representations of latent variables (Lines 1–2) are decided by the number of concepts which is $O(n)$. The step of training the encoder can be carried out in $O(n)$ operations. We observe that the steps of calculating the KL loss between the latent variable z (Lines 6–8) are decided by the time complexity of reparameterization trick which is $O(cn)$. The steps of calculating reconstruction loss L_{recon} and KL loss L_{KL} requires $O(n)$ operations. The steps of calculate total loss is not at all time-consuming and can be carried out in $O(1)$ operations. There are b epochs from the start to convergence. The steps of calculating the KL loss between the latent variable z and calculating the total loss require $O(ncb)$ operations. Thus, both the time complexity and the space cost of Cloud VAE are $O(n)$.

4. Experiment

In this section, we perform method validation and analysis in terms of model performance and interpretability respectively, comparing Cloud-VAE with other deep clustering (e.g., DEC, VaDE, etc.) and latent variable feature learning methods (FactorVAE, JointVAE, etc.).

4.1. Experimental datasets

We here use MNIST, Fashion-MNIST, USPS, EMNIST, CIFAR-10, and COIL20 datasets to evaluate the performance of Cloud-VAE in latent space. Except COIL20 dataset, the different clusters of MNIST, Fashion MNIST, USPS, EMNIST and CIFAR-10 datasets fail to have significant boundaries, and exist randomness from each other. More details are listed in Table 1.

Table 1
Summary of the datasets.

No.	Dataset	Dimension	Points	Image Pixels	Classes	cluster boundaries
1	MNIST	784	70,000	28×28	10	unclear
2	Fashion-MNIST	784	70,000	28×28	10	unclear
3	USPS	256	9298	16×16	10	unclear
4	EMNIST	784	145,600	28×28	26	unclear
5	CIFAR-10	3072	60,000	32×32×3	10	unclear
6	COIL20	1024	1440	32×32	20	clear

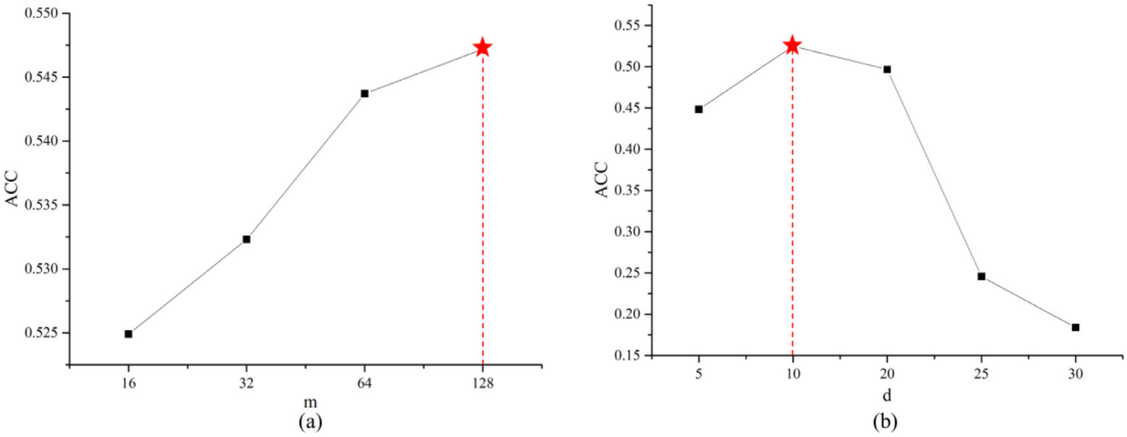


Fig. 5. Parameter sensitivity analysis of batch size (m) and dimension of latent space (d).

4.2. Experimental setup

For fair comparison, Cloud-VAE uses the same network architecture as the other models. Specifically, the architectures of f and g in Eq. (1) and Eq. (10) are 10–2000–500–500-D and D-500–500–2000–10, respectively, where D is the input dimension. ReLU is applied to all fully connected layers as the activation function, except for the input, output and embedding layers. The last layer of decoder network is two settings: Sigmoid and Linear corresponding to different datasets. The Epoch numbers were all set to 50 in pretraining and 300 during training. The Adam optimizer is used to maximize the lower bound ELBO on the evidence of Eq. (10) with a batch size of 128, a dimensionality of 10 for the learning representation and a learning rate of 0.001.

Good clustering algorithms produce high-quality clusters, that is, high similarity within clusters and low similarity between clusters. In the clustering performance analysis, the clustering validity evaluation index is used to evaluate the quality of the clustering, involving Clustering Accuracy (*ACC*), Adjusted Rand Index (*ARI*), and Normalized Mutual Information (*NMI*). The overall performance of the candidate model for each cluster is the average of the fitness values over all 10 iterations.

The variational lower bound of the VAE model includes the reconstruction error and the relative entropy between the distributions. In order to prove the effectiveness of the model, Fréchet Inception Distance (FID) is employed to evaluate its reconstruction performance. The experiments were conducted on a computer with an Intel Core i7–9700F 3.00 GHz CPU and NVIDIA RTX 3070Ti GPU.

4.3. Experimental results and discussion

4.3.1. Parameter determination

We utilize Adam optimizer to maximize the lower bound ELBO on the evidence of Cloud-VAE. We also executed sensitivity analysis and test of the other parameters such as batch size m and dimension of latent space d. The *ACC*, the prediction accuracy of Cloud-VAE on EMNIST dataset, was used as metric, Fig. 5 shows the results of the parameter sensitivity analysis. From Fig. 5(a), the *ACC* increases gradually with m increases. When m is 128, the *ACC* is increased by 0.547. Therefore, the batch size m is set as 128 in the following experiments. From Fig. 5 (b), the *ACC* increases with the growth of the demention d at first, until it reaches the maximum value when d equals 10. The results show that the dimension of latent space has a remarkable impact on the preservation of concept information. So, d is set as 10 in the following experiments.

4.3.2. Clustering performance validation

To verify the effectiveness of Cloud-VAE in clustering performance, we compared three classes of methods, including clustering methods related to the autoencoder AE model (DEC, IDEC), variational autoencoders with prior distribution optimization (GMVAE and VaDE), and unoptimized priori VAE variant (FactorVAE, JointVAE, and Guided-VAE). Cloud-VAE embeds prior optimization based on latent space of AE and modifies the variational lower bound of VAE, therefore, two related clustering methods based on AE are selected to compare with Cloud-VAE. In addition, Cloud-VAE introduces a prior distribution with data latent distribution information, so VaDE and GMVAE are employed for comparison, which learn to optimize priors to have data latent distribution information. To introduce prior distribution, Cloud-VAE also performs latent representation learning in its latent space, thus 4 VAE variants that learn latent space latent representation without prior optimization are selected for comparison. Note that, these methods are all implicitly learning the aggregated information of latent space.We conducted clustering performance experiments on the above 8 methods on 6 datasets respectively, and used the cluster accuracy (*ACC*), normalized mutual information (*NMI*), and adjusted rand index(*ARI*) as evaluation metrics. The experimental results are shown in Table 2 and Table 3.

It can be seen from Table 2 that Cloud-VAE obtains the highest *ACC*, *NMI*, and *ARI* on MNIST, Fashion-MNIST and USPS datasets. Specifically, the *ACC* of Cloud-VAE on MNIST, Fashion MNIST, and

Table 2
Clustering performance of models on MNIST, Fashion MNIST, USPS datasets.

Model	MNIST			Fashion-MNIST			USPS		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
DEC	0.867	0.786	0.754	0.576	0.631	0.461	0.712	0.696	0.602
IDEC	0.890	0.795	0.777	0.574	0.634	0.461	0.762	0.787	0.708
GMVAE	0.755	0.703	0.624	0.529	0.529	0.375	0.626	0.635	0.548
VaDE	0.936	0.875	0.869	0.542	0.561	0.409	0.722	0.791	0.698
VAE	–	–	–	–	–	–	–	–	–
VAE+GMM	0.558	0.387	0.509	0.349	0.215	0.357	0.393	0.304	0.383
VAE+KMeans	0.670	0.454	0.525	0.255	0.110	0.199	0.348	0.205	0.248
FactorVAE	–	–	–	–	–	–	–	–	
FactorVAE+GMM	0.699	0.549	0.636	0.286	0.106	0.212	0.442	0.212	0.401
FactorVAE+KMeans	0.710	0.541	0.612	0.596	0.423	0.550	0.732	0.634	0.704
JointVAE	0.881	0.793	0.829	0.450	0.367	0.556	0.403	0.216	0.360
JointVAE+GMM	0.300	0.105	0.183	0.250	0.074	0.149	0.388	0.201	0.338
JointVAE+KMeans	0.188	0.029	0.063	0.224	0.062	0.126	0.361	0.236	0.404
Guided-VAE	–	–	–	–	–	–	–	–	–
Guided-VAE+GMM	0.734	0.584	0.665	0.493	0.388	0.576	0.459	0.341	0.515
Guided-VAE+KMeans	0.654	0.535	0.636	0.545	0.409	0.591	0.475	0.358	0.537
Cloud-VAE (ours)	0.956	0.894	0.907	0.650	0.665	0.620	0.785	0.760	0.719

Table 3
Clustering performance of models on EMNIST, CIFAR-10 and COIL20 datasets.

Model	EMNIST			CIFAR-10			COIL20		
	ACC	NMI	ARI	ACC	NMI	ARI	ACC	NMI	ARI
DEC	0.308	0.487	0.246	0.214	0.098	0.055	0.325	0.507	0.268
IDEC	0.194	0.275	0.086	0.195	0.066	0.038	0.588	0.696	0.467
GMVAE	0.515	0.574	0.382	0.174	0.131	0.057	0.612	0.788	0.560
VaDE	0.430	0.494	0.265	0.142	0.124	0.048	0.210	0.392	0.158
VAE	–	–	–	–	–	–	–	–	–
VAE+GMM	0.349	0.410	0.200	0.158	0.105	0.044	0.602	0.742	0.521
VAE+KMeans	0.263	0.278	0.129	0.136	0.099	0.040	0.316	0.435	0.204
FactorVAE	–	–	–	–	–	–	–	–	–
FactorVAE+GMM	0.395	0.433	0.235	0.203	0.079	0.045	0.631	0.727	0.493
FactorVAE+KMeans	0.316	0.359	0.163	0.196	0.066	0.037	0.640	0.734	0.509
JointVAE	0.418	0.487	0.295	0.202	0.087	0.043	0.379	0.548	0.232
JointVAE+GMM	0.262	0.333	0.168	0.181	0.067	0.035	0.195	0.295	0.014
JointVAE+KMeans	0.081	0.048	0.012	0.148	0.017	0.009	0.198	0.289	0.012
Guided-VAE	–	–	–	–	–	–	–	–	–
Guided-VAE+GMM	0.482	0.558	0.333	0.244	0.121	0.073	0.587	0.727	0.475
Guided-VAE+KMeans	0.412	0.455	0.244	0.223	0.105	0.056	0.628	0.739	0.507
Cloud-VAE (ours)	0.545	0.612	0.416	0.251	0.121	0.079	0.540	0.649	0.457

USPS datasets is 0.956, 0.65, and 0.785, respectively. All of them are much higher than those of DEC and IDEC. DEC and IDEC models choose to perform clustering learning on AE, and the loss function of AE only considers the similarity between the original data and the reconstructed data which lacks description of the amount of information contained in the data. While clustering is the aggregation of data information, which results in model clustering that IDEC is worse than Cloud-VAE. Take the ACC as an example, Cloud -VAE outperforms GMVAE by 26.6%, 22.9%, and 25.4% on MNIST, Fashion-MNIST and USPS datasets, respectively. Compared with VaDE, Cloud-VAE also has 2.1%, 19.9%, and 8.7% improvement on MNIST, Fashion-MNIST, and USPS datasets, respectively, which proves that introducing uncertainty into the prior distribution can guide the model to explicitly express the aggregation of latent space.

Table 3 shows that the clustering performance of models for EMNIST, CIFAR-10 and COIL20. Cloud-VAE achieves the highest *ACC*, *NMI*, and *ARI* performance on EMNIST and CIFAR-10 datasets. Especially, Cloud-VAE outperforms the strongest baseline *ACC* by 2.96% and 8.06% on EMNIST and CIFAR-10 datasets, respectively. In the reparameterization trick, Cloud-VAE expands the representation range of latent variables based on the original distribution, which increase the representation power of latent variables. We can also observe that Cloud-VAE fails to obtain the best result in COIL20 dataset, which is because the cluster boundaries of COIL20 dataset are clear. Cloud-VAE introduces the randomness into the clustering process, which leads to a fuzzy representation of the distribution range of the clear clustering data. One possible solution is to constrain the sampling range of reparameterization trick techniques to reduce the impact of randomness. It can be concluded that Cloud-VAE can preserve the intrinsic distribution of data and hence help concept to embed latent space appropriately.

We also compare Cloud-VAE with the other models in terms of computational runtime on different datasets, of which results are shown in Fig 6. For fair comparison, Cloud-VAE and the compared models use the same network architecture and the training iterations are all 300 epochs. It can be observed that VAE costs the highest runtime on most datasets. Cloud-VAE spends the least time on CIFAR-10 dataset. On the other 4 datasets except COIL20 dataset, the computational costs of Cloud-VAE ranked in the middle and are close to most models. When datasets have relatively high dimensions, such as CIFAR-10, VAE, FactorVAE, and Guided-VAE datasets do a poor job of finding the distribution of clusters, which results in high runtime on these datasets. Cloud-VAE has a greater advantage over Guided-VAE in dataset with high dimensions.

4.3.3. Statistical significance analysis

In order to give a thoroughly comparative study, we analyze the statistical significance using the Friedman test and post-hoc Nemenyi test [38]. The Friedman test determines whether there

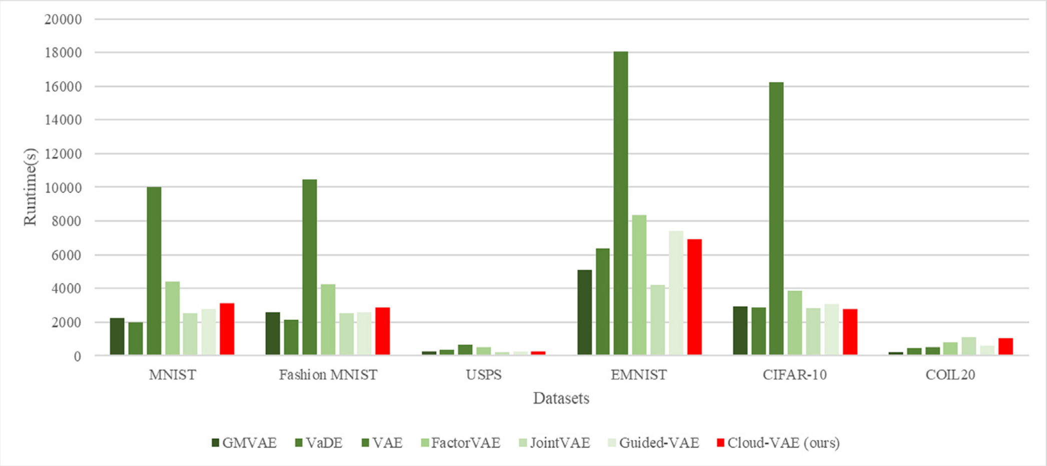


Fig. 6. Runtime of models on different datasets.

Table 4
The p-values in different evaluation index under Friedman test.

	ACC	*NMI*	*ARI*
p-value	0.00033	0.00002	0.0003

is a difference in performance among 14 models over 6 datasets. According to Friedman test and post-hoc Nemenyi test, we utilize the ranking of different models for statistical significance. In statistical analysis, p-value (the smallest level of significance) is compared with pre-specified significance level $\alpha = 0.05$ to test the significance of the results. The p-value provides information about whether a statistical hypothesis test is significant or not. The smaller the p-value, the stronger the evidence against the null hypothesis. The results are shown in Table 4 and Fig. 7.

Table 4 shows that the p-value under Friedman test in different evaluation index, such as *ACC*, *NMI*, and *ARI*. All p-values are far less than $\alpha = 0.05$, which rejects the null hypothesis of equivalent performance and confirms the existence of significant differences among the performance of all the models. Fig. 7 shows the results of Nemenyi post hoc tests. Cloud-VAE achieves the highest average rank under different evaluation metrics. With 95% confidence level, the performance of GMVAE cannot be distinguished from the models developed using DEC, IDEC, and VaDE. Cloud-VAE obtains the best ranking (each average rank on different evaluation index is approximately 2), although the differences between the top 5 algorithms are not statistically significant. Obviously, compared to other baseline algorithms, Cloud-VAE achieves the best ranking of performance with relative stability.

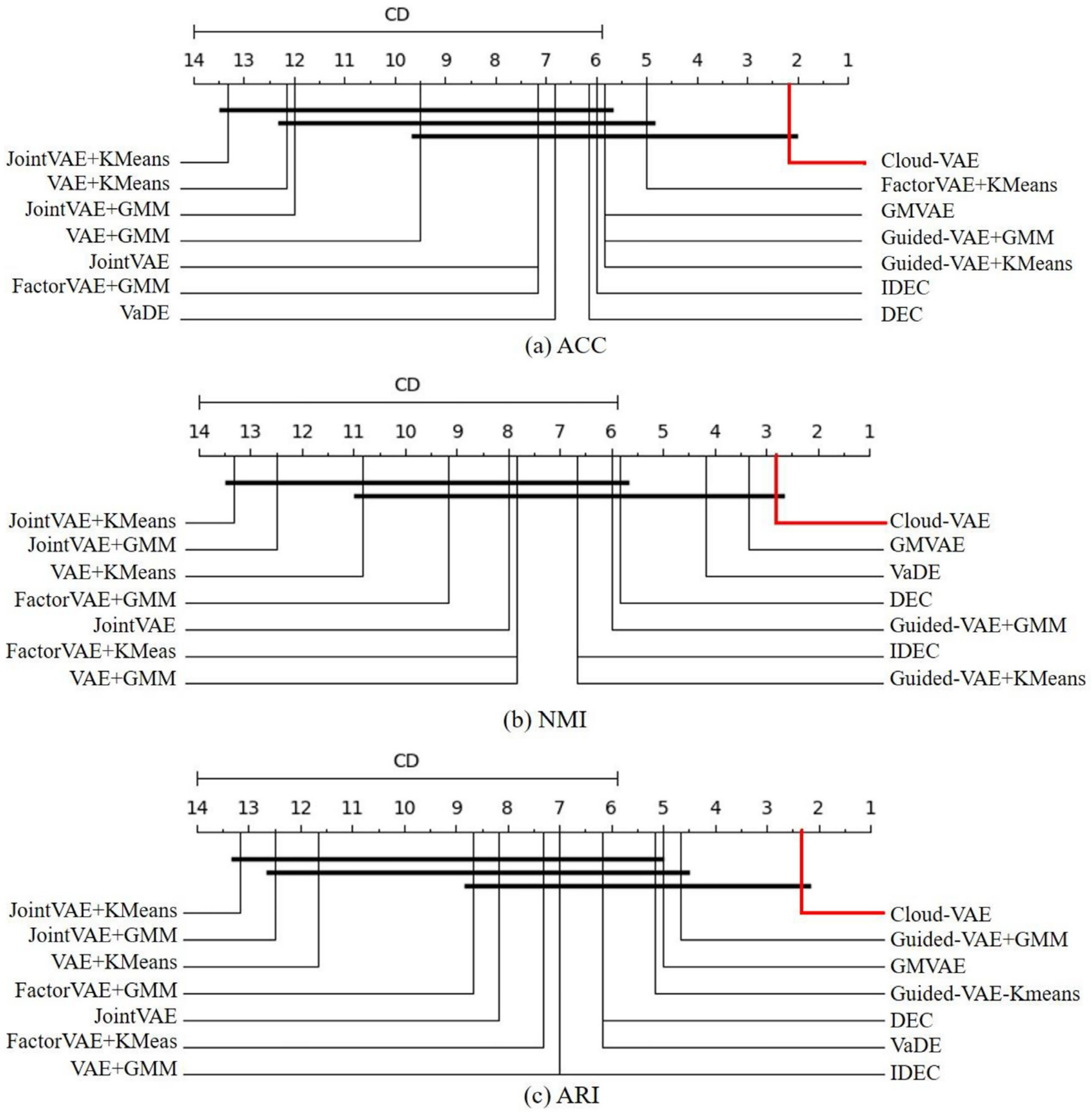
4.3.4. Reconstruction performance validation

We compared the reconstruction performance experiments of GMVAE, VaDE, VAE, FactorVAE, JointVAE, and Guided-VAE on 6 datasets respectively, and used FID as the evaluation metric. The experimental results are shown in Table 5. Cloud-VAE has the best reconstruction performance on Fashion MNIST, USPS and COIL20. Cloud-VAE performs similarly to the optimal performance of VaDE on MNIST and GMVAE on EMNIST. Compared with 4 unoptimized priors, the reconstruction performance of our proposed model is greatly improved, especially on USPS and COIL20. Compared to GMVAE, Cloud-VAE enhances the learning ability by introducing randomness, which enables it to reconstruct data with complex features. In addition, although FactorVAE and JointVAE learn latent representations, their optimization methods lead to worse reconstruction ability than VAE. Compared with GMVAE, the reconstruction ability of Cloud-VAE is higher under five datasets. For GMVAE and VaDE, the performance on MNIST, USPS and EMNIST is higher than that of VAE, FactorVAE, JointVAE, and Guided-VAE, but on the Fashion-MNIST dataset, the reconstruction performance can only be equal to it. For Cloud-VAE, its reconstruction performance stands out on Fashion-MNIST, USPS, and COIL20, as the learning of randomness results in an increased sampling space for latent variables in latent space and an enhanced learning range, resulting in improved performance on complex datasets.

4.3.5. Interpretability analysis

The model training is guided by prior distribution of concept representations and embedding concepts to learn the mapping from feature space to concept space. The concept generation results on MNIST, Fashion MNIST, and USPS datasets after training are shown in Fig. 8., where the concepts have comprehensible meanings.

During the training process, the concept space interacts with the feature space for updating, as shown in Fig. 9. The figure gives a visualization of the reduced dimensionality of latent space and the generated images of the coarse concepts in the concept space for different training times (Epoch = 1, 50, 100, 150, 200, and 300) of the model on MNIST dataset. The latent space is gradually aggregated and the potential variables have corresponding category information. In addition, the coarse concepts are gradually updated during the training process of the model. For example, the number 4 is gradually learned as a 'horizontal' stroke during the training process. The variational autoencoder constructs the concept representation of latent space towards a better representation of the concept during the training process. To discover whether there is an intelligible latent representation feature for Cloud-VAE, we observe whether the model learns a different latent representation by varying the latent variable dimension value from [−2,2] under the data pair latent space. Whereas FactorVAE, Guided-VAE, and JointVAE learn latent variable features, there lacks concept result for dimensional perturbation changes. Therefore, existing methods often select any sample from the real data for the visual presentation of feature changes, as shown in Fig. 10. JointVAE, FactorVAE generate images that produce some feature changes but completely lose the detailed features of the data compared to the original image on the left. FactorVAE, Guided-VAE have many unintelligible changes (marked red circle). From the above analysis, the visualization of feature variations without a standard concept result shows limited interpretable potential representations.



(a) ACC

(b) NMI

(c) ARI

Fig. 7. Visualization of the Nemenyi post-hoc test under different evaluation indices with unclear datasets.

Table 5
Reconstruction performance of models (*FID*).

Model	MNIST	Fashion-MNIST	USPS	EMNIST	CIFAR-10	COIL20
GMVAE	94.249	125.157	51.160	19.933	183.642	305.048
VaDE	88.080	134.438	39.728	22.013	194.814	165.655
VAE	116.992	124.270	71.012	34.861	224.947	138.860
FactorVAE	108.143	132.822	67.859	124.097	217.800	122.078
JointVAE	100.515	150.552	101.386	45.238	270.775	188.486
Guided-VAE	96.276	125.149	71.544	22.547	201.547	125.177
Cloud-VAE(ours)	90.163	122.484	36.668	23.422	219.210	107.129

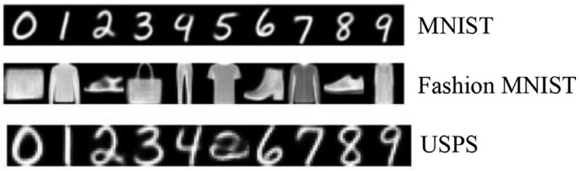


Fig. 8. Concept generation results on different datasets.

In contrast, Cloud-VAE represents the aggregated information of latent space explicitly and represents this aggregated information as a set of concepts $C = \{C_1, C_2, \ldots, C_c\}$. Therefore, in this paper, the learned concepts are used as a basic concept for performing feature changes to assist JointVAE, FactorVAE, and Guided-VAE in visualizing and analyzing latent space, and the experimental results are shown in Fig. 11. Cloud-VAE learns the basic 4 smooth changes of tilt degree, width and thickness (coarse to fine, fine to coarse) compared to the 3 comparison methods. On this basis, Cloud-VAE finds a latent representation which changes of the degree for graph curvature with the distribution. For FactorVAE,

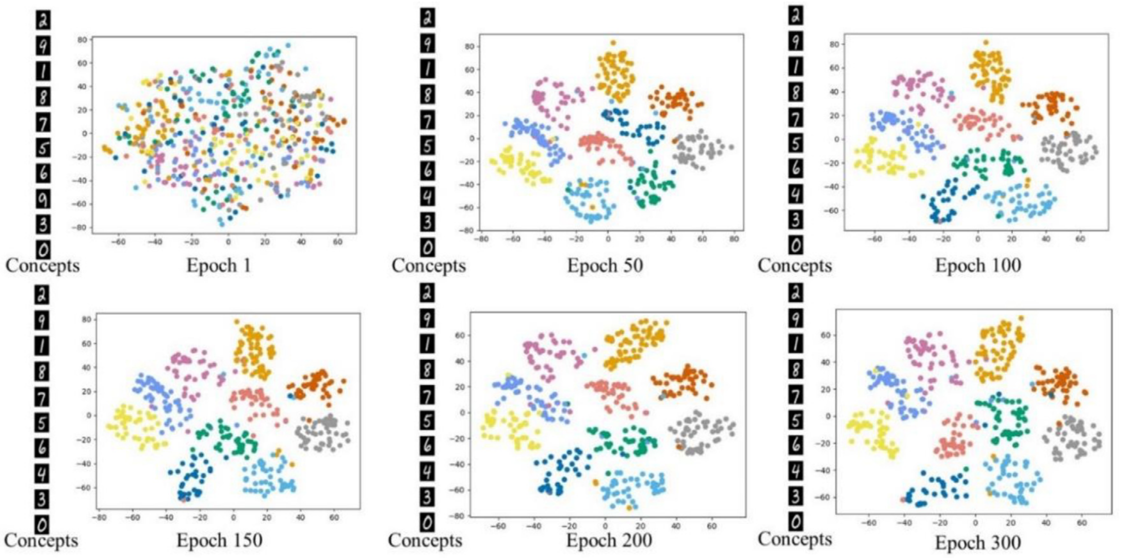


Fig. 9. The change of concept and latent space during model training.

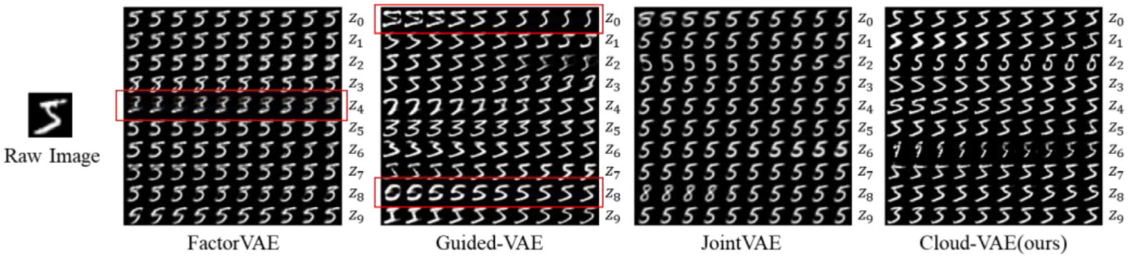


Fig. 10. Variation in features with real data.

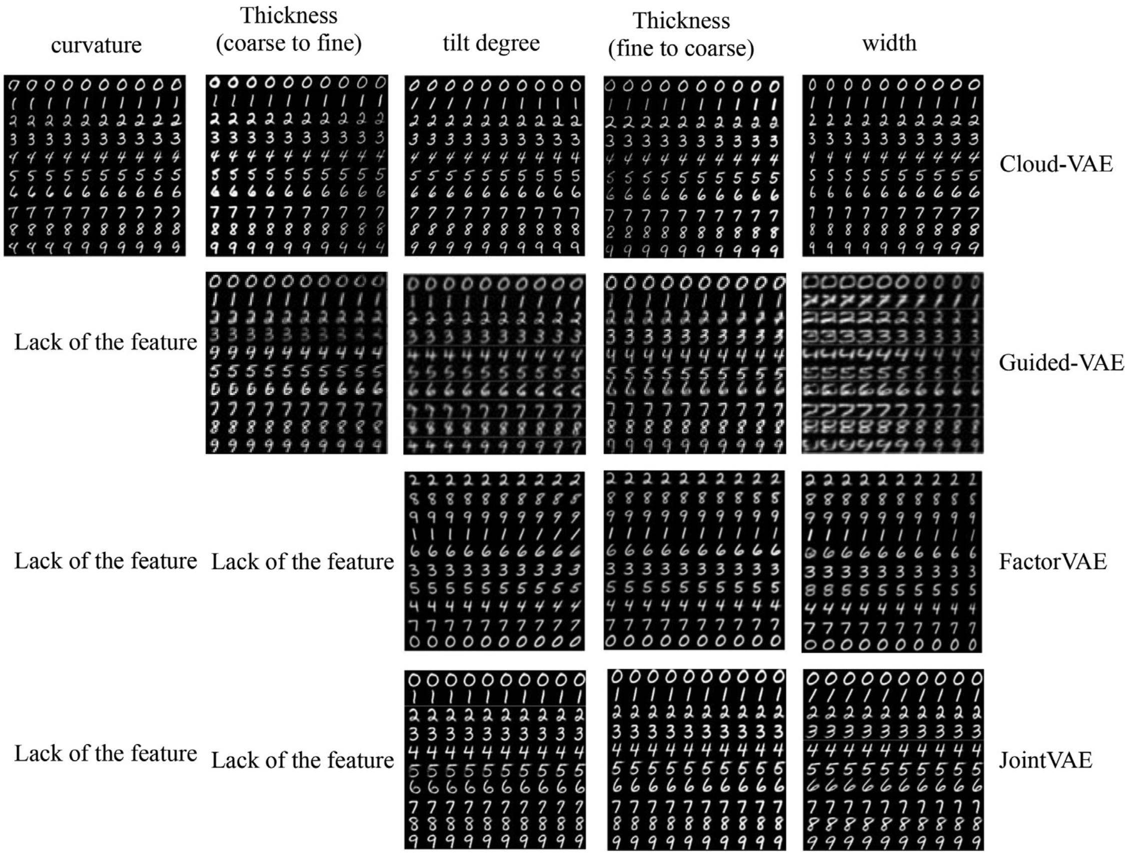
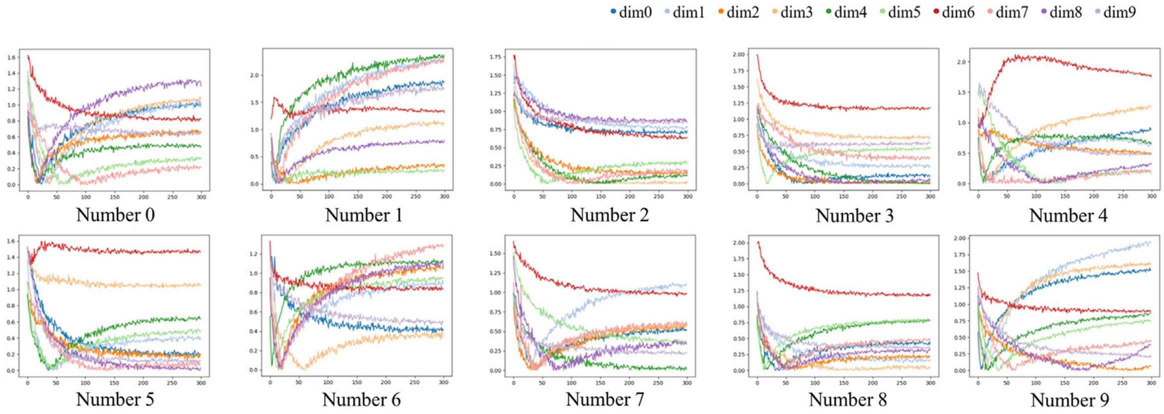


Fig. 11. Variation of features in images corresponding to different concept centers.

Algorithm 1
Variational autoencoder for concept embedding (Cloud-VAE).

```
Input: Dataset $X$.
Output: The trained model.
input dataset $X$ to train autoencoder model, and obtain latent variables $Z = \{z_1, z_2, \ldots, z_n\}$;
use Cloud-Cluster to obtain concept representations of latent variables, including $c$ concepts, each of which is represented as $\{Ex_k, En_k, He_k\}$;
for epoch:
    train encoder with data $X$ as input;
    use reparameterization trick based on the forward cloud transformation to obtain latent variable $z$;
    for $k \leftarrow 0$ to $c$ do:
        calculate KL loss $L_{KL\_concept_k}$between latent variable $z$, concept parameter corresponding to latent variable $z$ and concept $C_k$;
    end for
    train decoder with the latent variable $z$ as input;
    calculate reconstruction loss $L_{recon}$;
    calculate KL loss $L_{KL}$ between the concept parameter corresponding to latent variable $z$ and standard normal distribution;
    calculate total loss $L = \sum_{k=1}^{c} (L_{KL\_concept\ k}) + L_{KL} + L_{recon}$
end for
```



Number 0 Number 1 Number 2 Number 3 Number 4
Number 5 Number 6 Number 7 Number 8 Number 9

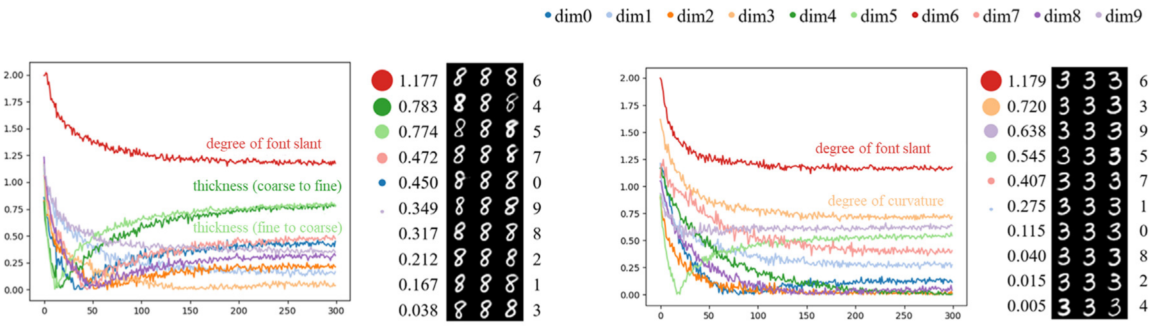
Fig. 12. KL scatter results for all categories on MNIST dataset.

JointVAE finds the numerical feature variations of degree of tilt, width and thickness (fine to coarse) in the potential representation of latent space under the MNIST dataset. FactorVAE guides the construction of latent space by adding a penalty term for feature independence, while JointVAE models the potential variables by dividing them into continuous and discrete variables. The prior distributions of both models only follow the standard normal distribution, and the constraints on latent space from this distribution fail to guide the model to map the full data information in latent space, thus resulting in a potential representation of the degree of curvature and thickness (coarse to fine) not being learned in latent space. For Cloud-VAE, the 4 basic latent representations are learned, but the representation under the feature of width appears to be unsmooth, especially as the feature changes from 0 to negative values. The method adds a lightweight decoder outside the VAE framework, but the geometric transformation with principal component analysis decodes in such a way that the latent variables lack clear physical meaning, leading to a loss of meaning during part of the latent representation change. In contrast, Cloud-VAE follows a priori distributions with potential distribution information, and its latent space construction process of potential variables can well represent the information contained in the data, so more interpretable potential representations are found on this basis.

Different from disentangled visualization methods such as Joint VAE, this paper attempts to analyze the importance of the fine-grained concepts contained under different concepts, i.e., the fine-grained feature visualization for some of the numbers, and the main important feature for numbers 3 and 8 is the degree of font slant which is greater than the other dimensional features. For the number 8, both provide similar information regardless of the font thickness. The KL scatter results are calculated and plotted as shown in Fig. 12. The degree of font slant (red line) is the more important feature in all categories. Fig. 13. shows the results of the fine-grained feature visualization for some of the numbers, and the main important feature for numbers 3 and 8 is the degree of font slant which is greater than the other-dimensional features. For the number 8, both provide similar information regardless of the font thickness.

5. Conclusion

In order to improve the interpretability of latent space of VAE, its variants optimize the prior distributions of model to constrain the construction of latent space. However, variational autoencoder and its variants still obey prior distributions that are complex in structure, artificially hypothetical and unable to describe the randomness of latent variables, making them unable to describe the concepts of latent space and hard to explain the aggregation process of latent space. Thus, this paper proposes a variational autoencoder with concept embedded named Cloud-VAE. Firstly, it uses encoder encodes concept parameters of latent variables and introduces the encoded hyper entropy into reparameterization process of model to increase the randomness of latent variable, which expands the sampling range of variables, improves the flexibility of data, and represents latent space information more accurately. Then, Cloud-VAE embeds multiple concepts into model as an effective prior distribution to constrain model, and a theoretical derivation of variational lower bound with concept embedded is combined with variation method to guide the model training to obtain the optimal parameter estimates. It realizes concept parameter updates and the mutual mapping between latent space and concept space. Finally, experimental results on the on MNIST, Fashion-



(a) Fine-grained concept of 8

(b) Fine-grained concept of 3

Fig. 13. Fine-grained results of 3 and 8 on MNIST.

MNSIT, USPS, EMNIST, CIFAR-10, and COIL20 datasets show that Cloud-VAE improves NMI metric by 22.9% and 19.9% respectively compared to deep clustering methods such as VaDE and GMVAE, showing better clustering and reconstruction performance. In addition, compared with FactorVAE, JointVAE and Guided-VAE, Cloud-VAE explicitly explains the model aggregation process, and other interpretable representations are found on top of the existing interpretable potential representations.

It can be concluded that the interpretability of VAE can be improved by embedding the uncertainty concepts into the latent space. This concept embedding approach can try to enhance the interpretability of more deep learning models (such as DNNs and CNNs) in the future.

Data & code availability

The data and source codes for the experiments are available from the corresponding author upon reasonable request. Email: yueliu@shu.edu.cn.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work is supported by the State Key Program of National Nature Science Foundation of China (Grant No. 61936001), the National Natural Science Foundation of China (No. 52073169), Training Program of the Major Research Plan of the National Natural Science Foundation of China (No. 92270124), the National Nature Science Foundations of China (No. 62221005), the Natural Science Foundation of Chongqing Nos. cstc2021ycjh-bgzxm0013 and cstc2019jcyj-cxttX0002, and the key cooperation project of Chongqing Municipal Education Commission (Grant No. HZ2021008). We appreciated the High Performance Computing Center of Shanghai University and Shanghai Engineering Research Center of Intelligent Computing System for providing the computing resources and technical support.

References

[1] D.P. Kingma, M Welling, Auto-encoding variational bayes, in: Proceedings of the International Conference on Learning Representations, 2014.
[2] A. Kusiak, Convolutional and generative adversarial neural networks in manufacturing, Int. J. Prod. Res. 58 (5) (2020) 1594–1604.
[3] T.M. Tran, T.N. Vu, N.D. Vo, et al., Anomaly analysis in images and videos: a comprehensive review, ACM Comput. Surv. CSUR, 2022.
[4] Y. Xiong, R. Zuo, Z. Luo, et al., A physically constrained variational autoencoder for geochemical pattern recognition, Math. Geosci. 54 (2022) 783–806.
[5] Y. Zhu, M.R. Min, A. Kadav, et al., S3vae: self-supervised sequential vae for representation disentanglement and data generation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6538–6547.
[6] G. Greco, A. Guzzo, G. Nardiello, FD-VAE: a feature driven VAE architecture for flexible synthetic data generation, in: Proceedings of the International Conference on Database and Expert Systems Applications, Springer, Cham, 2020, pp. 188–197.
[7] C. Zhang, P. Gao, Defending adversaries using unsupervised feature clustering VAE, in: Proceedings of the International Conference on Machine Learning 2021 Workshop on Adversarial Machine Learning, 2021.
[8] C. Liu, M.K.P. Ng, T. Zeng, Weighted variational model for selective image segmentation with application to medical images, Pattern Recognit. 76 (2018) 367–379.
[9] X. Chen, D.P. Kingma, T. Salimans, et al., Variational lossy autoencoder, in: Proceedings of the International Conference on Learning Representations, 2017.
[10] T.T.T. Nguyen, T.T. Nguyen, A.W.C. Liew, et al., Variational inference based bayes online classifiers with concept drift adaptation, Pattern Recognit. 81 (2018) 280–293.
[11] I. Gulrajani, K. Kumar, F. Ahmed, et al., Pixelvae: a latent variable model for natural images, in: Proceedings of the International Conference on Learning Representations, 2017.
[12] L. Guo, Q. Dai, Graph clustering via variational graph embedding, Pattern Recognit. 122 (2022) 108334.
[13] C. Louizos, K. Swersky, Y. Li, et al., The variational fair autoencoder, in: Proceedings of the International Conference on Learning Representations, 2016.
[14] L. Cai, H. Gao, S Ji, Multi-stage variational autoencoders for coarse-to-fine image generation, in: Proceedings of the SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2019, pp. 630–638.
[15] T.N. Kipf, M. Welling, Variational graph autoencoders, NIPS Workshop on Bayesian Deep Learning, 2016, pp. 1–3.
[16] Y. Burda, R. Grosse, R. Salakhutdinov, Importance weighted autoencoders, Proceedings of the International Conference on Learning Representations, 2016, pp. 1–14.
[17] Y.L. He, S.S. Xu, J.Z Huang, Creating synthetic minority class samples based on autoencoder extreme learning machine, Pattern Recognit. 121 (2022) 108191.
[18] R. Sabathé, E. Coutinho, B. Schuller, Deep recurrent music writer: memory-enhanced variational autoencoder-based musical score composition and an objective measure, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, 2017, pp. 3467–3474.
[19] W. Joo, W. Lee, S. Park, et al., Dirichlet variational autoencoder, Pattern Recognit. 107 (2020) 107514.
[20] S. Liu, J. Liu, Q. Zhao, et al., Discovering influential factors in variational autoencoders, Pattern Recognit. 100 (2020) 107166.
[21] A. Atanov, K. Struminsky, M. Welling, et al., The deep weight prior, in: Proceedings of the International Conference on Learning Representations, 2019.
[22] J. Zhou, Z. Lai, D. Miao, et al., Multigranulation rough-fuzzy clustering based on shadowed sets, Inf. Sci. 507 (2020) 553–573 Ny.
[23] C. Guo, J. Zhou, H. Chen, et al., Variational autoencoder with optimizing gaussian mixture model priors, IEEE Access 8 (2020) 43992–44400.
[24] Z. Jiang, Y. Zheng, H. Tan, et al., Variational deep embedding: an unsupervised and generative approach to clustering, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1965–1972.
[25] N. Dilokthanakul, P.A.M. Mediano, M. Garnelo, et al., Deep unsupervised clustering with gaussian mixture variational autoencoders, in: Proceedings of the International Conference on Learning Representations, 2017, pp. 1–12.
[26] C.P. Burgess, I. Higgins, A. Pal, et al., Understanding disentangling in β-VAE, in: Workshop on Learning Disentangled Representations at the 31st Conference on Neural Information Processing Systems, 2017, pp. 1–11.

[27] R.T.Q. Chen, X. Li, R. Grosse, et al., Isolating sources of disentanglement in VAEs, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 2615–2625.
[28] Z. Ding, Y. Xu, W. Xu, et al., Guided variational autoencoder for disentanglement learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7920–7929.
[29] J. Xie, R. Girshick, A. Farhadi, Unsupervised deep embedding for clustering analysis, in: Proceedings of the International conference on machine learning, PMLR, 2016, pp. 478–487.
[30] X. Guo, L. Gao, X. Liu, et al., in: Improved deep embedded clustering with local structure preservation, 2017, pp. 1753–1759.
[31] X. Yang, C. Deng, F. Zheng, et al., Deep spectral clustering using dual autoencoder network, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4066–4075.
[32] X. Yang, C. Deng, K. Wei, et al., Adversarial learning for robust deep clustering, Adv. Neural Inf. Process. Syst. 33 (2020) 9098–9108.
[33] X. Yang, C. Deng, T. Liu, et al., Heterogeneous graph attention network for unsupervised multiple-target domain adaptation, IEEE Trans. Pattern Anal. Mach. Intell. 44 (4) (2020) 1992–2003.
[34] H. Kim, A. Mnih, Disentangling by factorising, in: Proceedings of the International Conference on Machine Learning, PMLR, 2018, pp. 2649–2658.
[35] E. Dupont, Learning disentangled joint continuous and discrete representations, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 708–718.
[36] W. Li, Y. Zhou, G. Xun, Evaluation of rural landscape resources based on cloud model and probabilistic linguistic term set, Land 11 (1) (2022) 60 Basel.
[37] Y. Liu, Z. Liu, S. Li, et al., Cloud-cluster: an uncertainty clustering algorithm based on cloud model, Knowl. Based Syst. 263 (2023) 110261.
[38] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

Yue Liu obtained her B.S. and M.S. in computer science from Jiangxi Normal University in 1997 and 2000. She finished her Ph.D. in control theory and control engineering from Shanghai University (SHU) in 2005. She has been working with the School of Computer Engineering and Science of SHU since July 2000 and a visiting scholar at the University of Melbourne from Sep. 2012 to Sep. 2013. At present, she is a professor of SHU. Her current research interests focus on machine learning, data mining, and their applications.

Zitu Liu received the M.S. degree from the Compute Science, Heilongjiang University, Heilongjiang, China, in 2020. He is currently pursuing the Ph.D. degree in Shanghai University, Shanghai, China. His-main research interests include data mining and deep learning interpretability.

Shuang Li received the B.S. degree from the School of Information Engineering, Sichuan Agricultural University, Sichuan, China, in 2019 and the M.S. degree from the School of Computer Engineering and Science of Shanghai University, Shanghai, China, in 2022. Her research interests include clustering algorithm and deep learning interpretability.

Zhenyao Yu received the B.S. degree from the College of Communication and Information Technology, Xi'an University of Science and Technology, Shaanxi, China, in 2021. He is currently pursuing the M.S. degree in Shanghai University, Shanghai, China. His-research interests include data mining and deep learning interpretability.

Yike Guo is the vice president of Hong Kong University of Science and Technology and professor at Imperial College London. He is an IEEE fellow, fellow of the Royal Academy of Engineering (FREng), member of the Academia Europaea (MAE), fellow of the British Computer Society and a trustee of the Royal Institution of Great Britain. Professor Guo has published over 200 articles, papers, and reports. His-current research interests focus on data mining, machine learning, and dig data of science.

Qun Liu received her B.S. degree from Xi'An Jiaotong University in China in 1991, and the M.S. degree from Wuhan University, in China in 2002, and the Ph.D from Chongqing University in China in 2008. She is currently a Professor with Chongqing University of Posts and Telecommunications. Her current research interests include complex and intelligent systems, neural networks and intelligent information processing.

Guoyin Wang received the B.S., M.S., and Ph.D. degrees from Xi'an Jiaotong University, Xian, China, in 1992, 1994, and 1996, respectively. He was at the University of North Texas, and the University of Regina, Canada, as a visiting scholar during 1998–1999. Since 1996, he has been at the Chongqing University of Posts and Telecommunications, where he is currently a professor, the director of the Chongqing Key Laboratory of Computational Intelligence, the Vice-President of the University, and the dean of the School of Graduate. He was the director of the Institute of Electronic Information Technology, Chongqing Institute of Green and Intelligent Technology, CAS, China, 2011–2017. He is the author of over 10 books, the editor of dozens of proceedings of international and national conferences, and has more than 300 reviewed research publications. His-research interests include rough sets, granular computing, knowledge technology, data mining, neural network, and cognitive computing, etc. Dr. Wang was the President of International Rough Set Society (IRSS) 2014–2017. He is a Vice-President of the Chinese Association for Artificial Intelligence (CAAI), and a council member of the China Computer Federation (CCF). He is a Fellow of IRSS, CAAI and CCF.