Clustering by Directly Disentangling Latent Space

Fei Ding Feng Luo
School of Computing, Clemson University
feid@clemson.edu luofeng@clemson.edu

Abstract

To overcome the high dimensionality of data, learning latent feature representations for clustering has been widely studied recently. However, it is still challenging to learn “cluster-friendly” latent representations due to the unsupervised fashion of clustering. In this paper, we propose Disentangling Latent Space Clustering (DLS-Clustering), a new clustering mechanism that directly learning cluster assignment during the disentanglement of latent spacing without constructing the “cluster-friendly” latent representation and additional clustering methods. We achieve the bidirectional mapping by enforcing an inference network (i.e., encoder) and the generator of GAN to form a deterministic encoder-decoder pair with a maximum mean discrepancy (MMD)-based regularization. We utilize a weight-sharing procedure to disentangle latent space into the one-hot discrete latent variables and the continuous latent variables. The disentangling process is actually performing the clustering operation. Eventually the one-hot discrete latent variables can be directly expressed as clusters, and the continuous latent variables represent remaining unspecified factors. Experiments on six benchmark datasets of different types demonstrate that our method outperforms existing state-of-the-art methods. We further show that the latent representations from DLS-Clustering also maintain the ability to generate diverse and high-quality images, which can support more promising application scenarios.

1. Introduction

As an important unsupervised learning method, clustering has been widely used in many computer vision applications, such as image segmentation [7], visual features learning [3], and 3D object recognition [43]. Clustering becomes difficult when processing large amounts of high-semantic and high-dimensional data samples [10]. In order to overcome these challenges, many latent space clustering approaches such as DEC [46], DCN [47] and ClusterGAN [34], have been proposed. In these latent space clustering methods, the original high-dimensional data is first projected to low-dimensional latent space, then clustering algorithms, such as K-means [30], are performed on the latent space.

Most existing latent space clustering methods focus on learning the “cluster-friendly” latent representations. To avoid learning the random discriminative representations, their training objectives are usually coupled with data reconstruction loss or data generation constraints, which allow to rebuild or generate the input samples from the latent space. These objectives force the latent space to capture all key factors of variations and similarities, which are essential for reconstruction or generation. Therefore, these learned low-dimensional representations are not just related to clusters, and not the optimal latent representations for clustering.

Furthermore, current latent space clustering methods depend on additional clustering methods (e.g., K-means) to output the final clustering result based on learned latent representations. It’s difficult to effectively integrate low-dimensional representation learning and clustering algorithm together. The performance of distance-based clustering algorithms, such as K-means [30], is highly dependent on the selection of proper similarity and distance measures. Although constructing latent space can alleviate the problem of computing the distance between high dimensional data, defining a proper distance in latent space to obtain best clustering performance is still a challenge.

In this paper, we propose disentangle latent space clustering (DLS-Clustering), a new type of clustering algorithm that directly obtains the cluster information during the disentanglement of latent space. The disentangling process partitions the latent space into two parts: the one-hot discrete latent variables directly related to categorical cluster information, and the continuous latent variables related to other factors of variations. The disentanglement of latent space is actually performing the clustering operation, and no further clustering method is needed. Unlike existing distance-based clustering methods, our method does not need any explicit clustering objective and distance/similarity calculation in the latent space.

To separate the latent space into two completely inde-
dependent parts and directly obtain clusters, we first couple the inference network and the generator of GAN to form a deterministic encoder-decoder pair under the maximum mean discrepancy (MMD) regularization [18]. Then, we utilize the weight sharing strategy, which involves the bidirectional mapping between latent space and data space, to separate the latent space into one-hot discrete variables and continuous variables of other factors. Our method integrates the GAN and deterministic Autoencoder together, to achieve the disentanglement of the latent space. It includes three different types of regularizations: an adversarial density-ratio loss in data space, MMD loss in the continuous latent code and cross-entropy loss in discrete latent code. We choose adversarial density-ratio estimation for modeling the data space because it can handle complex distributions. MMD-based regularizer is stable to optimize and works well with multivariate normal distributions [41]. Our code and models are publicly available at this link.

In summary, our contributions are as follows:
(1) We propose a new clustering approach called DLS-Clustering, which directly obtain clusters in a completely unsupervised manner through disentangling latent space.
(2) We introduce a MMD-based regularization to enforce the inference network and the generator of standard GAN to form a deterministic encoder-decoder pair.
(3) We define a disentanglement training procedure based on the standard GAN and the inference network without increasing model parameters and requiring extra inputs. This procedure is also suitable for disentangling other factors of variation.
(4) We evaluate DLS-Clustering using six different types of benchmark datasets. DLS-Clustering achieves superior clustering performance on five of six datasets and close to best result on the other one.

2. Related works

Latent space clustering. Recently, many latent space clustering methods that leverage the advance of deep neural network based unsupervised representation learning [42, 4] have been developed. Several pioneering works propose to utilize an encoding architecture [48, 4, 23, 3] to learn the low-dimensional representations. In these methods, the pseudo-labels that are created based on some hypothetical similarities are used during optimization process. Because pseudo-labels usually underfit the semanticity of real-world datasets, they often suffer the Feature Randomness problem [33]. Most of recent latent space clustering methods are based on Autoencoders [46, 8, 20, 47, 49], which enables to reconstruct data sample from a low-dimensional representation. For example, Deep Embedded Clustering (DEC) [46] proposes to pretrain an Autoencoder with the reconstruction objective to learn low-dimensional embedded representations. Then, it discards the decoder and continues to train the encoder for clustering objective through a well-designed regularizer. IDEC [20] combines the reconstruction objective and clustering objective to jointly learn suitable representations with preserving local structure. DCN [47] proposes a joint dimensionality reduction and K-means clustering approach, in which the low-dimensional representation is obtained via the Autoencoder. Because the learned latent representations are closely related to the reconstruction objective, these methods still do not achieve the desired clustering results.

Recently, ClusterGAN [34] integrated GAN with an encoder network for clustering by creating a non-smooth latent space with the mixture of one-hot encoded discrete variables and continuous latent variables. However, the one-hot encoded discrete variables and continuous latent variables are not completely disentangled in ClusterGAN. Thus, the one-hot encoded discrete variable cannot effectively represent cluster. To obtain clustering assignment, ClusterGAN still need to perform additional clustering on entire dimensions of latent space under the discrete-continuous prior distribution.

Disentanglement of latent space. Learning disentangled representations enables us to reveal the factors of variation in the data [2], and provides interpretable semantic latent codes for generative models. Generally, existing disentangling methods can be mainly divided into two different types according to the disentanglement level. The first type of disentanglement involves separating the latent representations into two [32, 21, 52, 36] or three [16] parts. This type of method can be achieved in one step. For example, Mathieu et al. [32] introduce a conditional VAE with adversarial training to disentangle the latent representations into label relevant and the remaining unspecified factors. Y-AE [36] focuses on the standard Autoencoder to achieve the disentanglement of implicit and explicit representations. Meanwhile, two-step disentanglement methods based on Autoencoder [21] or VAE [52] are also proposed. In those methods, the first step is to extract the label relevant representations by training a classifier. Then, they obtain label irrelevant representations mainly via the reconstruction loss. All of these methods improve the disentanglement results by leveraging (partial) label information to minimize the cross-entropy loss. The second type of disentanglement, such as β-VAE [22], FactorVAE [24] and β-TCVAE [5], learns to separate each dimension in latent space without supervision. These VAE-based frameworks choose the standard Gaussian distribution as the prior distribution. And they aim to balance the reconstruction quality and the latent code regularization through a stochastic encoder-decoder pair.

Considering that the real-world data usually contains several discrete factors (e.g., categories), which are diffi-

1 after the paper is accepted
Figure 1. The architecture of DLS-Clustering (G: generator, E: encoder, D: discriminator). The latent representations are separated into cluster-relevant latent variables $z_c$ and other factors of variation $z_n$. The $z_c$ and $z_n$ are concatenated and fed into the $G_\theta$ for generation and the $E_\phi$ maps the samples ($x_g$, $x_r$ and $x_g'$) back into latent space. The $D_\psi$ is adopted for the adversarial training in the data space. Note that all generators share same parameters and all encoders share same parameters.
of objects would be reasonably represented by discrete variables, while other continuous factors, such as style and scale information, can be represented by the continuous variables. In this work, we split the latent representations into \( z_c \) and \( z_n \) based on the discrete-continuous prior.

The standard generative adversarial networks \([17, 19]\) consist of two components: the generator \( G_\theta \) and the discriminator \( D_\psi \). \( G_\theta \) defines a mapping from the latent space \( Z \) to the data space \( X \) and \( D_\psi \) can be considered as a mapping from the data space \( X \) to a real value in \([0, 1]\), which represents the probability of one sample being real. Eq. \( \text{(1)} \) defines the minimax objective of the standard GANs:

\[
\begin{align*}
\min_G \max_D & \mathbb{E}_{x \sim p_r}[q(D_\psi(x))] + \mathbb{E}_{z \sim p(z)}[q(1 - D_\psi(G_\theta(z)))] ,
\end{align*}
\]

where \( p_r \) is the real data distribution, \( p(z) \) is the prior distribution on the latent space, and \( p_z \) is the model distribution of the generated sample \( x_G = G_\theta(z) \). For the original GAN \([17]\), the function \( q \) is chosen as \( q(t) = \log t \), and the Wasserstein GAN \([19]\) applies \( q(t) = t \). This adversarial density-ratio estimation \([41]\) enforces \( p_z \) to match \( p_r \).

### 3.3. Deterministic encoder-generator pair

Many previous works, such as ALL \([11]\), BiGAN \([9]\), combined the inference network (i.e., encoder) and GAN together to form a bidirectional mapping. However, due to the lack of consistent mapping between data samples and latent variables, it usually obtains poor reconstruction results. To turn the generator \( G_\theta \) in DLS-Clustering into a good decoder, we need to apply several constraints between the posterior distribution \( p(q(z|x)) \) and the prior distribution \( p(z) \). Because the latent variable \( z = (z_c, z_n) \), for the prior \( p(z_c, z_n) = p(z_c)p(z_n) \), these constraints can be added by simply penalizing the discrete variable part and the continuous variable part separately.

The constraint of discrete variables can be computed through the inverse network, which involves first generating the data sample \( x_G \) from \((z_c, z_n)\) and then encoding it back to the latent variable \((\hat{z}_c, \hat{z}_n)\), as shown in Figure \( \text{1} \). Therefore, the penalty of discrete variables can be defined by the cross-entropy loss between the original input \( z_c \) and the recalculated discrete variable \( \hat{z}_c \)

\[
L_{CE} = \mathbb{E}_{z \sim p(z)}[H(z_c, E_\phi(G_\theta((z_c, z_n))))].
\]

The constraint of continuous variables can be considered in the standard Autoencoder model. As shown in Figure \( \text{1} \), the encoder \( E_\phi \) encodes the real data sample \( x_0 \) to the latent variables \( z_c^\prime \) and \( z_n^\prime \). To ensure that the generator \( G_\theta \) can reconstruct the original data from these latent variables, we apply an additional regularizer to encourage the encoded posterior distribution to match the prior distribution like AAE \([31]\) and WAE \([40]\). The former uses the GAN-based density-ratio trick to estimate the KL-divergence \([41]\), and the latter minimizes the distance between distributions based on Maximum mean discrepancy (MMD) \([18]\). For the sake of optimization stability, we choose MMD to quantify the distance between the prior distribution \( p(z_n) \) and the posterior \( q_\phi(z_n|x) \). And the regularizer based on MMD can be expressed as

\[
\begin{align*}
L_{MMD} &= \frac{1}{N(N-1)} \sum_{\ell \neq j} k(z_\ell^n, z_j^n) + \frac{1}{N(N-1)} \sum_{\ell \neq j} k(\hat{z}_\ell^n, \hat{z}_j^n),
\end{align*}
\]

where \( k(\cdot, \cdot) \) can be any positive definite kernel, \( \{z_1^n, \ldots, z_N^n\} \) are sampled from the prior distribution \( p(z_n) \), \( \hat{z}_n \) is sampled from the posterior \( q_\phi(z_n|x^i) \) and \( x^i \) is sampled from the real data samples for \( i = 1, 2, \ldots, N \).

In DLS-Clustering, the encoding distribution \( q_\phi(z|x) \) and the decoding distribution \( p_\phi(x|z) \) are taken to be deterministic, i.e., \( q_\phi(z|x) \) and \( p_\phi(x|z) \) can be replaced by \( E_\phi \) and \( G_\theta \), respectively. Therefore, we use a mean squared error (MSE) criterion as reconstruction loss, and write the standard Autoencoder loss \( \mathcal{L}_{AE} \) as

\[
\mathcal{L}_{AE} = \mathbb{E}_{x \sim p_r}||x - G_\theta(E_\phi(x))||_2^2.
\]

### 3.4. Disentangled representation

Although the above constraints are applied to enforce consistency between the distributions over \( x \) and \( z \), in order to avoid “posterior collapse” and obtain more promising representations, we impose an additional penalty to the objective to disentangle the latent variables. We utilize the weights sharing generator and encoder to enforce the disentanglement between discrete and continuous latent variables. In our architecture (Figure \( \text{1} \)), all encoders and generators share the same weights. Thus, it requires no more parameters to disentangle latent variables.

In practice, we sample the data sample \( x \) from the real data distribution, and sample the latent variable \( z = (z_c, z_n) \) from the discrete-continuous prior. The encoder \( E_\phi \) maps the data sample \( x \) to latent representations \( z_c^\prime \) and \( z_n^\prime \). To ensure that \( z_c^\prime \) and \( z_n^\prime \) are independent, we create the new latent variable \( z' = (z_c, z_n^\prime) \) by recombining the variables \((z_c, z_n)\) and \((z_c, z_n^\prime)\). Therefore, the generated data samples \( x' \) and \( x_n' \) will have identical discrete latent variable \( z_c \). Then \( x_n' \) is re-encoded to the latent variables \((\hat{z}_c, \hat{z}_n^\prime)\). The cross-entropy loss between \( z_c \) and \( \hat{z}_c \) can ensure that the discrete variable \( z_c \) isn’t modified when the continuous variable \( z_n \) changes.

\[
\mathcal{L}_c = \mathbb{E}_{x \sim p_r} \mathbb{E}_{z \sim p(z)} H(z_c, E_\phi(G_\theta((z_c, z_n)))).
\]
it is also necessary to use an additional regularizer to penalize the continuous latent variable. The generator \( G_\theta \) generates the data sample \( x'_d \) from new latent variable \( z' \), and the encoder \( E_\phi \) recovers the continuous latent variable \( \hat{z}'_n \) from \( x'_d \). Therefore, we penalize the deviation between \( z'_n \) and \( \hat{z}'_n \) by using the MSE loss:

\[
\mathcal{L}_n = \mathbb{E}_{x_n \sim p_d} E_{z_n \sim p(z)} [ ||z'_n - E_\phi(G_\theta(z_c, z'_n)))||^2]. \tag{6}
\]

### 3.5. Objective of DLS-Clustering

The objective function of our approach can be integrated into the following form:

\[
\mathcal{L} = \mathcal{L}_{\text{GAN}} + \mathcal{L}_{\text{AE}} + \beta_1 \mathcal{L}_{\text{MMD}} + \beta_2 \mathcal{L}_n + \beta_3 \mathcal{L}_{\text{CE}} + \beta_4 \mathcal{L}_c, \tag{7}
\]

where the corresponding regularization coefficients \( \beta_1 - \beta_4 \geq 0 \), controlling the relative contribution of different loss terms. Each term of Eq.\(7\) plays a different role for the three components: generator \( G_\theta \), discriminator \( D_\psi \) and encoder \( E_\phi \). Both of \( \mathcal{L}_{\text{GAN}} \) and \( \mathcal{L}_{\text{AE}} \) are related to \( G_\theta \) and \( E_\phi \), which constrain the whole latent variables. The \( \mathcal{L}_{\text{MMD}} \) term is also related to the \( D_\psi \), which focus on distinguishing the true data samples from the fake samples generated by \( G_\theta \). \( \mathcal{L}_{\text{MMD}} \) and \( \mathcal{L}_n \) are related to continuous latent variables, and \( \mathcal{L}_{\text{CE}} \) and \( \mathcal{L}_c \) are related to discrete latent variables. All these loss terms can ensure that our algorithm will disentangle the whole latent space into cluster information and remaining unspecified factors. The training procedure of DLS-Clustering involves jointly updating the parameters of \( G_\theta, D_\psi \) and \( E_\phi \), as described in Algorithm 1. In this work, we empirically set \( \beta_1 = \beta_2 \) and \( \beta_3 = \beta_4 \) to enable a reasonable adjustment of the relative importance of continuous and discrete parts.

### 4. Experiments

In this section, we perform a variety of experiments to evaluate the effectiveness of our proposed method.

#### 4.1. Data sets

The clustering experiments first are carried out on five datasets: MNIST, Fashion-MNIST [45], YouTube-Face (YTF) [44], Pendigits and 10x_73k [51]. Both of the first two datasets contain 70k images with 10 categories, and each sample is a 28 \times 28 grayscale image. YTF contains 10k face images of size 55 \times 55, belonging to 41 categories. The Pendigits dataset contains a time series of \((x, y)\) coordinates of handwritten digits. It has 10 categories and contains 10992 samples, and each sample is represented as a 16-dimensional vector. The 10x_73k dataset contains 73233 data samples of single cell RNA-seq counts of 8 cell types, and the dimension of each sample is 720. We choose these datasets to demonstrate that our method can be effective for clustering different types of data.

#### 4.2. Implementation

We implement different neural network structures for \( G_\theta, D_\psi \) and \( E_\phi \) to handle different types of data. For the image datasets (MNIST, Fashion-MNIST and YTF), we employ the similar \( G_\theta \) and \( D_\psi \) of DCGAN [37] with conv-deconv layers, batch normalization and leaky ReLU activations with slope of 0.2. The \( E_\phi \) uses the same architecture as the \( D_\psi \) except the last layer. For the Pendigits and 10x_73k datasets, the \( G_\theta, D_\psi \) and \( E_\phi \) are the MLP with 2 hidden layers of 256 hidden units each. Table 1 summarizes the network structures for different datasets. The model parameters have been initialized following the random normal distribution. For the prior distribution of our model parameters have been initialized following the random normal distribution.
4.3. Evaluation of DLS-Clustering algorithm

To evaluate clustering results, we report two standard evaluation metrics: Clustering Purity (ACC) and Normalized Mutual Information (NMI). We compare DLS-Clustering with four clustering baselines: K-means, Non-negative matrix Factorization (NMF), Spectral Clustering (SC) and Agglomerative Clustering (AGGLO). We also compare our method with the state-of-the-art clustering approaches based on GAN and Autoencorder respectively. For GAN-based approaches, ClusterGAN is chosen as it achieves the superior clustering performance compared to other GAN models (e.g., InfoGAN and GAN with bp). For Autoencoder-based methods, DEC, DCN and DEPICT, especially, Dual Autoencoder Network (DualAE) are used for comparison. In addition, the deep spectral clustering (SpectralNet) and joint unsupervised learning (JULE) are also included in our comparison.

Table 2 reports the best clustering metrics of different models from 5 runs. Our method achieves significant performance improvement on Fashion-10, YTF, Pendigits and 10x_73k datasets than other methods. In particular, for the 16-dimensional Pendigit dataset, the methods all perform worse than K-means does, while our method significantly outperforms K-means in both ACC (0.847 vs. 0.793) and NMI (0.803 vs. 0.730). DLS-Clustering achieves the best ACC result on YTF dataset while maintaining comparable NMI value. For MNIST dataset, DLS-Clustering achieves close to best performance on both ACC and NMI metrics.

### Table 1. The structure summary of the generator (G), discriminator (D) and encoder (E) in DLS-Clustering for different datasets.

| Dataset    | Dimensions | Layer Type | G-1/D-4/E-4 | G-2/D-3/E-3 | G-3/D-2/E-2 | G-4/D-1/E-1 |
|------------|------------|------------|-------------|-------------|-------------|-------------|
| MNIST      | 28 × 28 × 1| Conv-Deconv| 4 × 4 × 64   | 4 × 4 × 128  | -           | -           |
| Fashion-10 | 28 × 28 × 1| Conv-Deconv| 4 × 4 × 64   | 4 × 4 × 128  | -           | -           |
| YTF        | 55 × 55 × 3| Conv-Deconv| 5 × 5 × 32   | 5 × 5 × 64   | 5 × 5 × 128 | 5 × 5 × 256 |
| Pendigits  | 16         | MLP        | 256         | 256         | -           | -           |
| 10x_73k    | 720        | MLP        | 256         | 256         | -           | -           |

### Table 2. The dimensions of one-hot discrete latent variables and continuous latent variables in DLS-Clustering for different datasets. Note that the dimension of one-hot discrete latent variables is equal to the number of clusters.

| Dataset    | Discrete Dim. | Continuous Dim. |
|------------|---------------|------------------|
| MNIST      | 10            | 25               |
| Fashion-10 | 10            | 40               |
| YTF        | 41            | 60               |
| Pendigits  | 10            | 5                |
| 10x_73k    | 8             | 30               |

4.4. Analysis on continuous latent variables

The superior clustering performance of DLS-Clustering demonstrates that the one-hot discrete latent variables directly represent the category information in data. To understand the information contained in the continuous latent variables, we first use t-SNE to visualize the continuous latent variable \( z_n \) of MNIST and Fashion-MNIST datasets and compare them to the original data. As shown in Figure 2, we can clearly see category information in original MNIST (a(1)) and Fashion-MNIST (b(1))data. Meanwhile, there is no obvious category in the \( z_n \) of MNIST (a(2)) and Fashion-MNIST (b(2)) data. Samples in all categories are well mixed in both data sets. A small bulk of samples in the right part of a(2) is a group of “1” images. The reason that they are not distributed may be due to their low complexity.

Then, we fix the discrete latent variable and generate images belonging to the same clusters by sampling the continuous latent variables. As shown in Figure 3, the diversity of generated images indicates that the continuous latent variable contains a large number of generative factors, except the cluster information. To further understand the factors in continuous latent variable \( z_n \), we change the value of one single dimension from [-0.5, 0.5] in \( z_n \) while fixing other dimensions and the discrete latent variable \( z_c \). As shown in Figure 4, the value changing leads to semantic changes in the generated images. For the MNIST data, this changed dimension represents the width factor of variation in the digits. For the Fashion-MNIST data, it captures the shape factor of objects. All these informative continuous factors are independent of cluster categories.

These results demonstrate that the learned continuous latent representations from DLS-Clustering have captured
### Table 3. Comparison of clustering algorithms on five benchmark datasets. The results marked by (*) are from existing sklearn.cluster.KMeans package. The dash marks (-) mean that the source code is not available or that running released code is not practical, all other results are from [34] and [49].

| Method    | MNIST ACC | MNIST NMI | Fashion-10 ACC | Fashion-10 NMI | YTF ACC | YTF NMI | Pendigits ACC | Pendigits NMI | 10x,73k ACC | 10x,73k NMI |
|-----------|-----------|-----------|----------------|----------------|--------|--------|-------------|-------------|------------|-------------|
| K-means   | 0.532     | 0.500     | 0.474          | 0.512          | 0.601  | 0.776  | 0.793*      | 0.730*      | 0.623*     | 0.577*      |
| NMF [27]  | 0.560     | 0.450     | 0.500          | 0.510          | -      | -      | 0.670       | 0.580       | 0.710      | 0.690       |
| SC [39]   | 0.656     | 0.731     | 0.508          | 0.575          | 0.510  | 0.701  | 0.700       | 0.690       | 0.400      | 0.290       |
| AGGLO [50]| 0.640     | 0.650     | 0.550          | 0.570          | -      | -      | 0.700       | 0.690       | 0.630      | 0.580       |
| DEC [46]  | 0.863     | 0.834     | 0.518          | 0.546          | 0.371  | 0.446  | -           | -           | -          | -           |
| DCN [47]  | 0.830     | 0.810     | -              | -              | -      | 0.720  | 0.690       | -           | -          | -           |
| JULE [48] | 0.964     | 0.913     | 0.563          | 0.608          | 0.684  | 0.848  | -           | -           | -          | -           |
| DEPICT [14]| 0.965    | 0.917     | 0.392          | 0.392          | 0.621  | 0.802  | -           | -           | -          | -           |
| SpectralNet [38]| 0.800 | 0.814     | -              | -              | 0.685  | 0.798  | -           | -           | -          | -           |
| InfoGAN [6] | 0.890   | 0.860     | 0.610          | 0.590          | -      | -      | 0.720       | 0.730       | 0.620      | 0.580       |
| ClusterGAN [34]| 0.950 | 0.890     | 0.630          | 0.640          | -      | -      | 0.770       | 0.730       | 0.810      | 0.730       |
| DualAE [49]| 0.978    | 0.941     | 0.662          | 0.645          | 0.691  | 0.857  | -           | -           | -          | -           |
| Our Method | 0.975    | 0.936     | **0.693**      | **0.669**      | **0.721** | **0.790** | **0.847** | **0.803** | **0.905** | **0.820** |

Table 4. The scalability to large number of clusters (K=100) on CoIL-100 dataset.

| Method    | ACC      | NMI      | ARI      |
|-----------|----------|----------|----------|
| K-means   | 0.668    | 0.836    | 0.574    |
| Our method| **0.822**| **0.911**| **0.764**|

other meaningful generative factors that are not related to clusters. Therefore, the proposed method successfully performs the mapping from the data to the disentangled latent space. The one-hot discrete latent variable is directly related to clusters, and the continuous latent variable, which corresponds to the other unspecified generative factors, governs the diversity of generated samples.

### 4.5. Scalability of large number of clusters

To further evaluate the scalability of DLS-Clustering to large numbers of clusters, we run the it on the multi-view object image dataset COIL-100 [35]. The COIL-100 dataset has 100 clusters and contains 7200 images of size $128 \times 128$. Here, we compare our clustering method with K-means on three standard evaluation metrics: ACC, NMI and Adjusted Rand Index (ARI). As shown in Table 4, DLS-Clustering achieves better performance on all three metrics by directly learning clusters and 100-dimensional continuous latent representations. Especially, DLS-Clustering gains an increase of 0.154 on ACC metric. We also perform image generation task on Coil-100 dataset, to further verify the generative performance, which involves mapping latent variables to the data space. Figure 5 shows the generated samples by fixing one-hot discrete latent variables, which are diverse and realistic. The continuous latent variables represent meaningful factors such as the pose, location and orientation information of objects. Therefore, the disentanglement of latent space not only provides the superior clustering performance, but also retains the remarkable ability of diverse and high-quality image generation.
Figure 3. Samples generated by fixing discrete latent code from the models trained on MNIST and Fashion-MNIST. Note that the discrete latent variables are directly related to the cluster assignment, the continuous latent variables correspond to other informative factors.

Figure 4. Samples generated on fixed discrete latent variables from the models trained on MNIST and Fashion-MNIST.

Figure 5. The samples generated on fixed discrete latent variables from the models trained on Coil-100 dataset. Each column corresponds to a specific cluster.

5. Conclusion

In this work, we present DLS-Clustering, a new type of clustering method that directly obtain the cluster assignments by disentangling the latent space in an unsupervised fashion. Unlike existing latent space clustering algorithms, our method does not build clustering friendly latent space explicitly and does not need extra clustering operation. Furthermore, our method does not disentangle class relevant features from class non-relevant features. The disentanglement in our method is targeted to extract “cluster information” from data. Moreover, unlike distance-based clustering algorithms, our method does not depend on any explicit distance calculation in the latent space. The distance between data may be implicitly defined in neural network.

Besides clustering, the generator in our method can also generate diverse and realistic samples. The proposed method can also support other applications, including conditional generation based on clusters, cluster-specific image transfer and cross-cluster retrieval. In the future, we will explore better priors for the latent space and more disentanglement of other generative factors.

References

[1] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: fine-grained image generation through asymmetric training. In Proceedings of the IEEE International Conference on Computer Vision, pages 2745–2754, 2017.

[2] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

[3] M. Caron, P. Bojanowski, A. Joulin, and M. Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision (ECCV), pages 132–149, 2018.

[4] J. Chang, L. Wang, G. Meng, S. Xiang, and C. Pan. Deep adaptive image clustering. In Proceedings of the IEEE International Conference on Computer Vision, pages 5879–5887, 2017.

[5] T. Q. Chen, X. Li, R. B. Grosse, and D. K. Duvenaud. Isolating sources of disentanglement in variational autoencoders. In Advances in Neural Information Processing Systems, pages 2610–2620, 2018.

[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In
Advances in neural information processing systems, pages 2172–2180, 2016.

[7] K.-S. Chuang, H.-L. Tzeng, S. Chen, J. Wu, and T.-J. Chen. Fuzzy c-means clustering with spatial information for image segmentation. *Computerized medical imaging and graphics*, 30(1):9–15, 2006.

[8] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan. Deep unsupervised clustering with gaussian mixture variational autoencoders. *arXiv preprint arXiv:1611.02648*, 2016.

[9] J. Donahue, P. Krähenbühl, and T. Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[10] D. L. Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[11] V. Dumoulin, I. Belghazi, B. Poole, O. Mastropietro, A. Lamb, M. Arjovsky, and A. Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[12] E. Dupont. Learning disentangled joint continuous and discrete representations. In *Advances in Neural Information Processing Systems*, pages 710–720, 2018.

[13] B. Esmaili, H. Wu, S. Jain, A. Bozkurt, N. Siddharth, B. Paige, D. H. Brooks, J. Dy, and J.-W. van de Meent. Structured disentangled representations. *arXiv preprint arXiv:1804.02086*, 2018.

[14] K. Ghasedi Dizaji, A. Herandi, C. Deng, W. Cai, and H. Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5736–5745, 2017.

[15] P. Ghosh, M. S. Sajjadi, A. Vergari, M. Black, and B. Schölkopf. From variational to deterministic autoencoders. *arXiv preprint arXiv:1903.12436*, 2019.

[16] A. Gonzalez-Garcia, J. van de Weijer, and Y. Bengio. Image-to-image translation for cross-domain disentanglement. In *Advances in Neural Information Processing Systems*, pages 1287–1298, 2018.

[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[18] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[19] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[20] X. Guo, L. Gao, X. Liu, and J. Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017.

[21] N. Hadad, L. Wolf, and M. Shahar. A two-step disentanglement method. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 772–780, 2018.

[22] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. *ICLR*, 2(5):6, 2017.

[23] W. Hu, T. Miyato, S. Tokui, E. Matsumoto, and M. Sugiyama. Learning discrete representations via information maximizing self-augmented training. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1558–1567. JMLR. org, 2017.

[24] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[25] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[26] A. B. L. Larsen, S. K. Sonderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

[27] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788, 1999.

[28] Y. Li, K. Swersky, and R. Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, pages 1718–1727, 2015.

[29] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[30] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[31] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05645*, 2015.

[32] M. F. Mathieu, J. J. Zhao, J. Zhao, A. Ramesh, P. Sprechmann, and Y. LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.

[33] N. Mrabah, M. Bougessaa, and R. Ksantini. Adversarial deep embedded clustering: on a better trade-off between feature randomness and feature drift. *arXiv preprint arXiv:1909.11832*, 2019.

[34] S. Mukherjee, H. Asnani, E. Lin, and S. Kannan. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4610–4617, 2019.

[35] S. A. Nene, S. K. Nayar, H. Murase, et al. Columbia object image library (coil-20).

[36] M. Patacka, P. Fox-Roberts, and E. Rosten. Yautoencoders: disentangling latent representations via sequential-encoding. *arXiv preprint arXiv:1907.10949*, 2019.

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[38] U. Shaham, K. Stanton, H. Li, B. Nadler, R. Basri, and Y. Kluger. Spectralnet: Spectral clustering using deep neural networks. *arXiv preprint arXiv:1801.01587*, 2018.

[39] J. Shi and J. Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.
[40] I. Tolstikhin, O. Bousquet, S. Gelly, and B. Schoelkopf. Wasserstein auto-encoders. arXiv preprint arXiv:1711.01558, 2017.

[41] M. Tschannen, O. Bachem, and M. Lucic. Recent advances in autoencoder-based representation learning. arXiv preprint arXiv:1812.05069, 2018.

[42] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. Journal of machine learning research, 11(Dec):3371–3408, 2010.

[43] C. Wang, M. Pelillo, and K. Siddiqi. Dominant set clustering and pooling for multi-view 3d object recognition. arXiv preprint arXiv:1906.01592, 2019.

[44] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. IEEE, 2011.

[45] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[46] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In International conference on machine learning, pages 478–487, 2016.

[47] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 3861–3870. JMLR. org, 2017.

[48] J. Yang, D. Parikh, and D. Batra. Joint unsupervised learning of deep representations and image clusters. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5147–5156, 2016.

[49] X. Yang, C. Deng, F. Zheng, J. Yan, and W. Liu. Deep spectral clustering using dual autoencoder network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4066–4075, 2019.

[50] W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph. In European Conference on Computer Vision, pages 428–441. Springer, 2012.

[51] G. X. Zheng, J. M. Terry, P. Belgrader, P. Ryvkin, Z. W. Bent, R. Wilson, S. B. Ziraldo, T. D. Wheeler, G. P. McDermott, J. Zhu, et al. Massively parallel digital transcrip- tion profiling of single cells. Nature communications, 8:14049, 2017.

[52] Z. Zheng and L. Sun. Disentangling latent space for vae by label relevant/irrelevant dimensions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 12192–12201, 2019.