Efficient Deep Embedded Subspace Clustering

Jinyu Cai[1,3], Jicong Fan[2,3]*, Wenzhong Guo[1], Shiping Wang[1], Yunhe Zhang[1], Zhao Zhang[4]
[1]College of Computer and Data Science, Fuzhou University, China
[2]School of Data Science, The Chinese University of Hong Kong (Shenzhen), China
[3]Shenzhen Research Institute of Big Data, China [4]Hefei University of Technology, China
{jinyucai1995,cszzhang}@gmail.com, fanjicong@cuhk.edu.cn
guowenzhong@fzu.edu.cn,{shipingwangphd,zhangyhannie}@163.com

Abstract

Recently deep learning methods have shown significant progress in data clustering tasks. Deep clustering methods (including distance-based methods and subspace-based methods) integrate clustering and feature learning into a unified framework, where there is a mutual promotion between clustering and representation. However, deep subspace clustering methods are usually in the framework of self-expressive model and hence have quadratic time and space complexities, which prevents their applications in large-scale clustering and real-time clustering. In this paper, we propose a new mechanism for deep clustering. We aim to learn the subspace bases from deep representation in an iterative refining manner while the refined subspace bases help learning the representation of the deep neural networks in return. The proposed method is out of the self-expressive framework, scales to the sample size linearly, and is applicable to arbitrarily large datasets and online clustering scenarios. More importantly, the clustering accuracy of the proposed method is much higher than its competitors. Extensive comparison studies with state-of-the-art clustering approaches on benchmark datasets demonstrate the superiority of the proposed method.

1. Introduction

Clustering is a fundamental issue in machine learning, which aims to separate samples into classes in the absence of label information, under the requirement of high intra-class similarity and low inter-class similarity. Many classical clustering algorithms such as k-means [29] and spectral clustering (SC) [30] have showed great success in real applications. However, they are not effective in handling data with complicated structures or/and high-dimensionality, which can be improved by using refined features of the data. Indeed, some previous works [14, 37, 38, 47] utilized the feature learning techniques such as non-negative matrix factorization [2], auto-encoder (AE) [1] and its variants [24, 31, 36] to learn low-dimensional embeddings for clustering, which increased the clustering accuracy. Nevertheless, since these methods are two-stage clustering and the feature learning is not specific to clustering, it is not guaranteed that the learned representations are appropriate for clustering.

Recently, a few researchers [3, 9, 26, 43, 46] have proposed end-to-end clustering methods, such as deep embedded clustering (DEC) [40], joint unsupervised learning (JULE) [41], deep adaptive clustering (DAC) [6], and deep comprehensive correlation mining (DCCM) [39]. In these methods, the clustering objectives are integrated with the network optimization process, which provides an approach to learning clustering-oriented embedded representations. However, most deep clustering methods use the Euclidean distance-based measure in identifying clusters, whereas Euclidean distance is not always valid or reasonable for different types of data structures.

Subspace clustering assumes that data lie in different subspaces [11]. A category of classical subspace clustering methods such as sparse subspace clustering (SSC) [11] and low-rank representation (LRR) [27] are mainly based on spectral clustering [30] and outperformed k-means and classical spectral clustering in many tasks such as face image clustering. Recently, a few researchers [8, 21, 25, 44] showed that joint subspace clustering and deep learning have promising performance on benchmark datasets. However, these approaches can hardly be extended to large-scale datasets because they need to learn a self-expressive matrix leading to quadratic time and space complexities. Consequently, some latest works [12, 48, 49] dedicate to improving the efficiency of subspace clustering.

In this paper, we aim to provide an approach to efficient and accurate deep subspace clustering. We propose to learn a set of subspace bases from the latent features extracted

*Jicong Fan is the corresponding author.

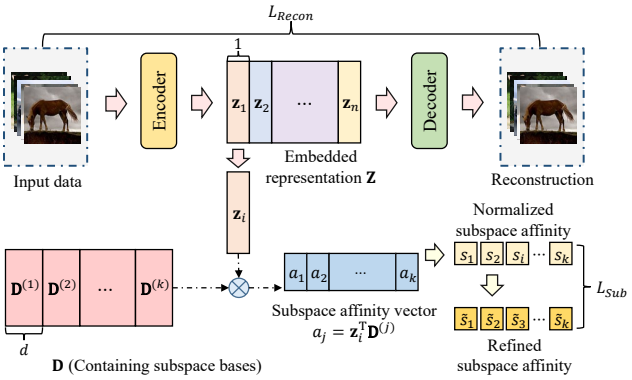


Figure 1. Illustration of the proposed method. The auto-encoder network is used to learn embedded representation $\mathbf{Z}$ for input data, then $\mathbf{Z}$ combines with subspace $\mathbf{D}$ to construct the subspace affinity vector which in turn yields the normalized subspace affinity S. Subsequently, the refined subspace affinity $\widetilde{S}$ is computed from S to provide self-supervised information. Note that d is the dimension of subspace, L_{Recon} and L_{Sub} represent the reconstruction loss and the discrepancy between $\widetilde{S}$ and S, and the network is trained by jointly optimizing them.

by deep auto-encoder, where the bases and network parameters are iteratively refined. The network structure of the proposed method is illustrated in Fig. 1. Our contributions are as follows.

- We present a novel deep subspace clustering method that is out of the conventional self-expressive framework.
- Our method has linear time and space complexity and hence is applicable to large-scale subspace clustering.
- We generalize the method to online clustering such that we can handle arbitrarily large datasets and streaming datasets effectively.
- We analyze the feasibility of using deep neural network to convert distance-based clustering and subspace clustering.

Numerical results on many benchmark datasets (*e.g.* Fashion-MNIST, STL-10, and REUTERS-10K) showed that our method is more effective than its competitors.

2. Related Work and A Brief Discussion

2.1. Deep Clustering

Earlier deep clustering methods aim at applying deep feature learning methods (*e.g.* auto-encoder [36], and variational auto-encoder (VAE) [24]) to extract features from complicated high-dimensional data for clustering. However, these solutions hardly learn the representations appropriate to clustering task. Current deep clustering methods focus on constructing end-to-end models. Xie *et al.* proposed DEC [40] that designs a clustering objective inspired by t-SNE [35]. It provided a clustering model that achieves simultaneous optimization of cluster centers and embedded features. Chang *et al.* [5] proposed deep self-evolution clustering (DSEC), which is a self-evolving-based algorithm to train the network alternatively with chosen pairs of patterns. In [39], Wu *et al.* presented a method called DCCM that uses pseudo-labels for self-supervision and uses mutual information to capture more discriminative representations for clustering. The partition confidence maximisation (PICA) proposed by Huang *et al.* [20] minimizes a partition uncertainty index and learns the most confident clustering allocation. Note that these deep clustering approaches assign clusters using Euclidean distance, which may not useful when the clusters do not concentrate on the mean values.

2.2. Subspace Clustering

Classical subspace clustering such as SSC [11], LRR [27], Kernel-SSC [32] aim to learn a self-expressive affinity matrix for spectral clustering. Ji *et al.* [21] proposed deep subspace clustering network (DSC-Net) that incorporated a self-expression module with auto-encoder network. DSC-Net showed significant improvement on several image datasets, compared to SSC and LRR. Zhou *et al.* [52] provided a method called deep adversarial subspace clustering (DASC) that utilized generative adversarial network [16] to provide an adversarial learning, which improved the performance of deep subspace clustering. Zhou *et al.* [51] proposed distribution preserving subspace clustering (DPSC) to retain the latent distribution in the subspace to improve the feature learning ability of the subspace clustering model. On the other hand, a few researchers tried to reduce the complexity of subspace clustering [7, 12, 13, 33, 49]. For example, Zhang *et al.* [49] proposed the k-subspace clustering network (k-SCN) to integrate the update of subspace into the learning of embedded space for addressing the drawback of learning the affinity matrix. Fan [12] proposed a method called k-factorization subspace clustering (k-FSC), which has linear time and space complexity and is able to handle missing data and streaming data.

2.3. A Brief Discussion

We analyze the time and space complexities of a few (due the space limitation) methods of classical subspace clustering, large-scale subspace clustering, and deep subspace clustering in Table 1. We see that these classical subspace clustering methods and deep subspace clustering methods have quadratic time and space complexities in terms of the number of samples. In contrast, our method has linear time and space complexity, which is comparable to the large-scale subspace clustering method of [12].

Method	Time complexity (per iter.)	Space complexity
SSC [11]	$O(mn^2)$	$O(mn+\rho n^2)$
LRR [27]	$O(mn^2+m^3)$	$O(mn+n^2)$
KSSC [32]	$O(n^3)$	$O(mn+n^2)$
SSSC [33]	$O(ms^3+k^2n_s)$	$O(mn+\rho n_s^2)$
S^{3}COMP-C [7]	$O(d\rho n^3(1-\delta))$	$O(mn+\rho n^2)$
k-FSC [12]	$O(kmrn+\vartheta mn)$	$O(mn+kmr+krn)$
DSC-Net [21]	$O(ln^2+\tilde{m}\tilde{l}n)$	$O(\tilde{m}n+n^2+\theta)$
DASC [52]	$O(ln^2+\tilde{m}\tilde{l}n)$	$O(\tilde{m}n+n^2+\theta)$
DPSC [51]	$O(ln^2+\tilde{m}\tilde{l}n)$	$O(\tilde{m}n+n^2+\theta)$
NCSC [50]	$O(ln^2+\tilde{m}\tilde{l}n)$	$O(\tilde{m}n+n^2+\theta)$
PSSC [28]	$O(mn^2+ln^2+\tilde{m}\tilde{l}n)$	$O(\tilde{m}n+n^2+\theta)$
EDESC(ours)	$O(kdpn+\tilde{m}\tilde{p}n)$	$O(\tilde{m}n+kn+kpd+\theta)$

Table 1. The time and space complexity of our method compared with some deep clustering and subspace clustering approaches in clustering n samples of dimension-m. To save space, we put the explanation for the parameters in the supplementary material.

3. Methodology

3.1. Proposed Model

In this paper, we aim at deep learning based subspace clustering and try to solve the following problem.

Problem 1 *Given a data matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$, where m denotes the number of features and n denotes the number of samples. Suppose $\mathbf{X} = \bar{\mathbf{X}}\mathbf{P}$, where $\bar{\mathbf{X}} = [\bar{\mathbf{X}}^{(1)}, \bar{\mathbf{X}}^{(2)}, \dots, \bar{\mathbf{X}}^{(k)}]$ and $\mathbf{P} \in \mathbb{R}^{n\times n}$ is an unknown permutation matrix. For $j = 1, \dots, k$, suppose the columns of $\bar{\mathbf{X}}^{(j)} \in \mathbb{R}^{m\times n_j}$ are generated by*

$$\mathbf{x} = \mathbf{f}_j(\mathbf{v}) + \varepsilon, \tag{1}$$

where $\mathbf{f}_j : \mathbb{R}^{r_k} \longrightarrow \mathbb{R}^m$ is an unknown nonlinear function, $r_j < m$, $\mathbf{v} \in \mathbb{R}^{r_j}$ is a random variable, and $\varepsilon \in \mathbb{R}^m$ denotes random Gaussian noise. Find the permutation matrix $\mathbf{P}$ (or $\bar{\mathbf{X}}$ equivalently) from $\mathbf{X}$.

The problem is exactly a clustering problem, for which we need to group the columns of $\mathbf{X}$ into k clusters corresponding to k different functions $\mathbf{f}_1, \dots, \mathbf{f}_k$. Figure 2 shows a simple example of Problem 1 when $m = 3$ and $r_1 = \cdots = r_5 = 1$. Note that when $\mathbf{f}_1, \dots, \mathbf{f}_k$ are all linear, the problem reduces to the classical subspace clustering. Hence, Problem 1 can be regarded as a nonlinear subspace clustering or manifold clustering [10,15,34] problem. A special case of Problem 1 is

Problem 2 *In Problem 1, for $j = 1, \dots, k$, suppose $\mathbf{f}_j(\mathbf{v}) = \mathbf{g}(\mathbf{B}^{(j)}\mathbf{v})$, where $\mathbf{B}^{(j)} \in \mathbb{R}^{p\times r_j}$ and $\mathbf{g} : \mathbb{R}^p \longrightarrow \mathbb{R}^m$. In addition, $\frac{\|\mathbf{B}^{(i)\top}\mathbf{B}^{(j)}\|_F}{\|\mathbf{B}^{(i)}\|_F\|\mathbf{B}^{(j)}\|_F}$ $(i \neq j)$ are small enough. Find the permutation matrix $\mathbf{P}$ (or $\bar{\mathbf{X}}$ equivalently) from $\mathbf{X}$.*

Problem 2 is easier than Problem 1 because it is enough to identify the correct clusters when we obtain a good estimation[1] of $\{\mathbf{B}^{(1)}, \dots, \mathbf{B}^{(k)}\}$. Therefore, in this paper, first, we propose to estimate $\mathbf{B}^{(j)}$ via approximating $\mathbf{x}$ with a multilayer neural network, *i.e.*,

$$\mathbf{x}_i \approx h_{\mathcal{W}}(\hat{\mathbf{B}}^{(j)}\hat{\mathbf{v}}_i), \ \ \mathbf{x}_i \in \mathbb{C}_j, \tag{2}$$

where $h_{\mathcal{W}}$ denotes a multilayer neural network with parameter set $\mathcal{W}$ and $\mathbb{C}_j$ denotes the j-th cluster. It is difficult to obtain $\{\hat{\mathbf{B}}^{(1)}, \dots, \hat{\mathbf{B}}^{(k)}\}$ directly. Instead, we estimate $\mathbf{B}^{(j)}\mathbf{v}_i$, *i.e.*, $\mathbf{z}_i := \hat{\mathbf{B}}^{(j)}\hat{\mathbf{v}}_i$. Thus we propose to solve

$$\begin{aligned} \underset{\mathcal{W},\{\mathbf{z}_1,\dots,\mathbf{z}_n\}}{\text{minimize}} \ & \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{x}_i - h_{\mathcal{W}}(\mathbf{z}_i)\|^2, \\ \text{subject to} \ & \mathbf{z}_i \in \mathbb{S}_i, \ \ i = 1, \dots, n, \end{aligned} \tag{3}$$

where $\mathbb{S}_i$ denotes the true cluster $\mathbf{x}_i$ should belong to. Nevertheless, it is impossible to sovle (3) directly because $\mathbb{S}_i$ are unknown. Now we introduce a new variable $\mathbf{D} = [\mathbf{D}^{(1)}, \mathbf{D}^{(2)}, \dots, \mathbf{D}^{(k)}]$. It contains k blocks and $\mathbf{D}^{(j)} \in \mathbb{R}^{p\times d}$, $\|\mathbf{D}_u^{(j)}\| = 1$, $u = 1, \dots, d$, $j = 1, \dots, k$. Note that $d \geq r_j$ for all $j = 1, \dots, k$. We hope that $\mathbf{D}^{(j)}$ has the same column space as $\mathbf{B}^{(j)}$, $j = 1, \dots, k$. Then according to the assumption we made in Problem 2, for all $j \neq l$, $\|\mathbf{D}^{(j)\top}\mathbf{D}^{(l)}\|_F$ should be small enough, *i.e.*,

$$\|\mathbf{D}^{(j)\top}\mathbf{D}^{(l)}\|_F \leq \tau, \ \ j \neq l, \tag{4}$$

where τ is a small constant.

Denote $\alpha_i = \arg\max_j \|\mathbf{z}_i^\top\mathbf{D}^{(j)}\|$. We expect

$$\|\mathbf{z}_i^\top\mathbf{D}^{(\alpha_i)}\| \gg \max_{j\neq\alpha_i}\|\mathbf{z}_i^\top\mathbf{D}^{(j)}\|, \ \ i = 1, \dots, n. \tag{5}$$

In other words, $\mathbf{z}_i$ is only highly correlated with one block of $\mathbf{D}$. Now we use an encoder $h'_{\mathcal{W}'}$ with parameter set $\mathcal{W}'$ to represent $\mathbf{z}_i$, *i.e.*,

$$\mathbf{z}_i = h'_{\mathcal{W}'}(\mathbf{x}_i), \ i = 1, \dots, n. \tag{6}$$

For convenience, we let

$$\hat{\mathbf{x}}_i := h_{\mathcal{W}}(\mathbf{z}_i), \ i = 1, \dots, n. \tag{7}$$

and denote $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1, \dots, \hat{\mathbf{x}}_n]$. Now putting (3), (4), (5), (6), and (7) together, we solve

$$\begin{aligned} \underset{\mathcal{W},\mathcal{W}',\mathbf{D}}{\text{minimize}} \ & \frac{1}{2n}\left\|\mathbf{X} - \hat{\mathbf{X}}\right\|_F^2, \\ \text{subject to} \ & \|\mathbf{D}_u^{(j)}\| = 1, \ u = 1, \dots, d, \ j = 1, \dots, k, \\ & \|\mathbf{D}^{(j)\top}\mathbf{D}^{(l)}\|_F \leq \tau, \ j \neq l, \\ & \|\mathbf{z}_i^\top\mathbf{D}^{(\alpha_i)}\| \gg \max_{j\neq\alpha_i}\|\mathbf{z}_i^\top\mathbf{D}^{(j)}\|, \ \ i = 1, \dots, n. \end{aligned} \tag{8}$$

Note that in (8), $\mathbf{z}_i$ are just intermediate variables according to (6) and we do not need to explicitly optimize them. In

[1] The estimation is still useful if $\hat{\mathbf{B}}^{(j)}$ is a linear transformation of $\mathbf{B}^{(j)}$.

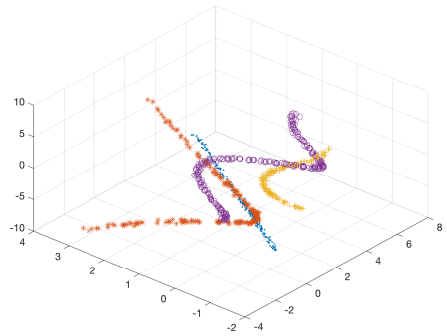


Figure 2. A toy example of Problem 1.

(8), the first constraint is to control the size of the columns of $\mathbf{D}$, otherwise $\|\mathbf{z}_i^\top \mathbf{D}^{(\alpha_i)}\|$ may become zero. The second constraint is to comply the assumption of dissimilarity between different subspaces made in Problem 1. The last constraint plays a role of subspace allocation. Note that our method (8) is still applicable to Problem 1, provided that the neural network is able to learn a $\mathbf{g}(\mathbf{B}^{(j)}\mathbf{v})$ to effectively approximate the $\mathbf{f}_j(\mathbf{v})$ in Problem 1.

3.2. Practical Solution

Now we show how to solve (8) approximately. For convenience, we let

$$L_{Recon} = \frac{1}{2n}\sum_{i=1}^{n}\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_F^2 . \tag{9}$$

We impose the first constraint in (8) by minimizing the following objective

$$D_{Cons1} := \frac{1}{2}\left\|\mathbf{D}^\top\mathbf{D}\odot\mathbf{I} - \mathbf{I}\right\|_F^2 , \tag{10}$$

where $\odot$ represents the Hadamard product, and $\mathbf{I}$ is an identity matrix of size kd by kd.

For the second constraint in (8), we define

$$\begin{aligned} D_{Cons2} :=& \frac{1}{2}\left\|\mathbf{D}^{(j)\top}\mathbf{D}^{(l)}\right\|_F^2, \quad j \neq l, \\ =& \frac{1}{2}\left\|\mathbf{D}^\top\mathbf{D}\odot\mathbf{O}\right\|_F^2 . \end{aligned} \tag{11}$$

Here $\mathbf{O}$ is a matrix in which all d-size diagonal block elements are 0 and all others are 1. Now we can put (10) and (11) together to get the regularization term on $\mathbf{D}$

$$D_{Cons} = \xi(D_{Cons1} + D_{Cons2}), \tag{12}$$

where ξ is a tuning parameter fixed at 10^{-3} in this work.

For the last constraint in (8), we propose a novel subspace affinity S in this paper to measure the relationship between the embedded representation $\mathbf{Z}$ and the subspace bases proxy $\mathbf{D}$

$$s_{ij} = \frac{\left\|\mathbf{z}_i^\top\mathbf{D}^{(j)}\right\|_F^2 + \eta d}{\sum_j\left(\left\|\mathbf{z}_i^\top\mathbf{D}^{(j)}\right\|_F^2 + \eta d\right)}, \tag{13}$$

Algorithm 1 Work flows of the proposed method

Input: Data matrix $\mathbf{X}$, embedding dimension p, subspace dimension d, number of clusters k, hyper-parameters η and β, total training epochs T.
Output: Cluster labels $\mathcal{C}$.

```
Initialize the network by the pre-trained weights.
Initialize the subspace D with k-means clustering.
for t = 1 to T do
    Learn embedded representation Z.
    Compute the subspace affinity S by (13).
    Compute the refined subspace affinity S̃ by (14).
    Compute the loss terms L_Recon and L_Sub by (9) and (15).
    Compute the regularization term D_Cons by (12).
    Update the network parameters and the subspace D by minimizing the objective function (16).
end for
Use (17) to obtain the final updated cluster labels.
return Cluster labels C.
```

where η is a parameter controlling the smoothness. Thus s_{ij} represents the probability that the embedded representation $\mathbf{z}_i$ belongs to the j-th subspace represented by $\mathbf{D}^{(j)}$. We further introduce a refined subspace affinity $\widetilde{S}$ defined by

$$\widetilde{s}_{ij} = \frac{s_{ij}^2/\sum_i s_{ij}}{\sum_j(s_{ij}^2/\sum_i s_{ij})}. \tag{14}$$

$\widetilde{S}$ aims to emphasize those assignments with high confidence in S. In other words, $\widetilde{S}$ can be employed as a self-supervised information, that yields the following subspace clustering objective

$$L_{Sub} = KL(\widetilde{S}\,||\,S) = \sum_i\sum_j \widetilde{s}_{ij}\log\frac{\widetilde{s}_{ij}}{s_{ij}}. \tag{15}$$

Now we define an unconstrained relaxation of (8) as

$$L = L_{Recon} + D_{Cons} + \beta L_{Sub}. \tag{16}$$

The training flows of the proposed method is presented in Algorithm 1. The proposed method achieves a joint optimization of subspace clustering and embedded representation learning. The initialization of $\mathbf{D}$ is given by the column space of the clusters generated by k-means on the $\mathbf{Z}$ of the pre-trained model. When the training of the network is finished, the final clustering results can be obtained by

$$\mathcal{C}_i = \arg\max_j s_{ij}. \tag{17}$$

3.3. Universal Approximation and Converting Problems

One may argue that neural networks have universal approximation ability such that the subspace clustering prob-

lem can be transformed to a distance-based clustering problem such that k-means and DEC [40] apply, or a distance based clustering problem can be converted to a subspace clustering problem. Here we generate two synthetic datasets to show how the converting performs. The first dataset is for distance-based clustering and is generated by

$$\begin{aligned} \mathbf{x}_i^{(j)} \sim \mathcal{N}(\mu_j, \mathbf{I}), \;\; i = 1, \ldots, 1000, \\ \mu_j \in \mathbb{R}^m, \; \mu_j \sim \mathcal{U}(-1, 1), \end{aligned} \tag{18}$$

and followed by $\mathbf{x}_i^{(j)} \longleftarrow \sin(\mathbf{x}_i^{(j)})$, where $m = 100$ and $j = 1, \ldots, 10$. The second dataset is for subspace clustering and is generated by

$$\begin{aligned} \mathbf{x}_i^{(j)} = \sin(\mathbf{B}^{(j)}\mathbf{v}_i), \;\; \mathbf{v}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \;\; i = 1, \ldots, 1000, \\ \mathbf{B}^{(j)} \in \mathbb{R}^{m \times p}, \;\; \mathbf{B}^{(j)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \end{aligned} \tag{19}$$

where $m = 100$, $p = 2$, and $j = 1, \ldots, 10$. We also add Gaussian noise to the datasets, *i.e.*, $\mathbf{X} \longleftarrow \mathbf{X} + \mathbf{N}$, where the standard error of the noise is 0.2 times of the standard error of the clean $\mathbf{X}$.

The performance of DEC [40], IDEC [18], and our method is shown in Fig. 3. The first plot indicates that it is relatively easy to convert a distance-based clustering problem to a subspace clustering problem as the accuracy of our method is quite high. The second plot indicates that it is very difficult to convert a subspace clustering problem to a distance-based clustering problem since DEC and IDEC failed. One possible reason is that it is easier to convert a Euclidean distance (*e.g.* $\|\mathbf{u}_1 - \mathbf{u}_2\|$) to a subspace affinity (*e.g.* $\mathbf{v}_1^\top \mathbf{v}_2$). For example, let $\mathbf{v}_1 = \phi(\mathbf{u}_1)$ and $\mathbf{v}_2 = \phi(\mathbf{u}_2)$, where ϕ is the feature map of a radial basis function, *e.g.* a Gaussian kernel. Then we have

$$\mathbf{v}_1^\top \mathbf{v}_2 = \exp(-\gamma \|\mathbf{u}_1 - \mathbf{u}_2\|^2),$$

where $\gamma > 0$ is a hyperparameter. Thus the neural network only need to learn an approximation for ϕ, which is not difficult. If we exchange the roles of $\mathbf{u}$ and $\mathbf{v}$, the network needs to learn a function h such that $\|h(\mathbf{v}_1) - h(\mathbf{v}_2)\|$ is a monotonic (roughly) transformation of $\|\mathbf{v}_1^\top \mathbf{v}_2\|$, which is quite difficult.

The above result and analysis verified it is necessary to provide an efficient and accurate deep subspace clustering method to handle Problem 2 or Problem 1 more generally.

4. Experiment

4.1. Datasets and Evaluation Metrics

To evaluate the clustering performance of our method, we consider six widely-used benchmark datasets, including two gray-scale image datasets (MNIST [2] and Fashion-MNIST [3]), three challenging real-world image datasets (CIFAR-10 [4], CIFAR-100, and STL-10 [5]), and one text dataset REUTERS-10K [6]. The detailed information is shown in Tab. 2.

[2]http://yann.lecun.com/exdb/mnist/

[3]https://github.com/zalandoresearch/fashion-mnist

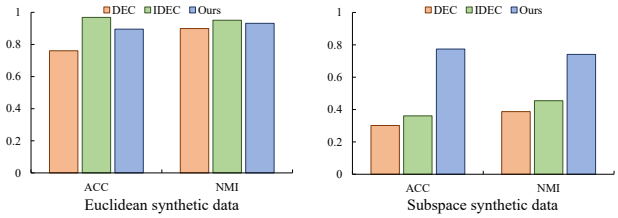


Figure 3. Comparison of clustering performance on synthetic data based on Euclidean and subspace principles.

Dataset name	# Total samples	# Classes	# Size
MNIST	70,000	10	28×28
Fashion-MNIST	70,000	10	28×28
CIFAR-10	60,000	10	32×32×3
CIFAR-100	60,000	20	32×32×3
STL-10	13,000	10	96×96×3
REUTERS-10K	10,000	4	2,000

Table 2. Detailed information of the six benchmark datasets.

We use two popular clustering metrics, Clustering Accuracy (ACC) and Normalized Mutual Information (NMI), to quantify the clustering performance. The two metrics take values in the range of $[0, 1]$, and higher score implies better clustering performance.

4.2. Experimental Settings

We construct our model with an encoder of architecture m-500-500-1,000-p fully-connected network and a decoder symmetric to it. We first pre-train 50 epochs by an auto-encoder with the same structure, then fit the pre-trained weight to initialize our model. The Adam [23] optimizer is used in our method. The learning rate is set as 0.001, the training epochs are set to 200, the batch-size is fixed as 512, and the clusters k are given by the categories of the corresponding dataset. In particular, for three real-world image datasets (STL-10, CIFAR-10, and CIFAR-100), we apply the ResNet50 [19] to extract their 2,048-dimensional features. As for the settings of hyper-parameter, η is fixed to the same value as d, and we further discuss about the impact of different values of d and β on clustering in Sec. 4.6.

4.3. Comparison with Stat-of-the-Art Approaches

In this section, we conduct comprehensive experiments in comparison with the SOTA approaches from three aspects, including classical approaches (k-means [29], SC [45], AC [17], and NMF [2]), deep clustering methods (AE [1], DAE [36], VAE [24], DEC [40], IDEC [18],

[4]http://www.cs.toronto.edu/ kriz/cifar.html

[5]https://cs.stanford.edu/ acoates/stl10/

[6]https://keras.io/api/datasets/reuters/

Method/Dataset	Fashion-MNIST		CIFAR-10		CIFAR-100		STL-10		REUTERS-10K	
	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI	ACC	NMI
k-means [29]	0.474	0.512	0.229	0.087	0.130	0.084	0.192	0.125	0.524	0.312
SC [45]	0.508	0.575	0.247	0.103	0.136	0.090	0.159	0.098	0.402	0.375
AC [17]	0.500	0.564	0.228	0.105	0.138	0.098	0.332	0.239	–	–
NMF [2]	0.434	0.425	0.190	0.081	0.118	0.079	0.180	0.096	–	–
AE [1]	0.567	0.553	0.314	0.239	0.165	0.100	0.303	0.250	0.597	0.323
DAE [36]	0.493	0.548	0.297	0.251	0.151	0.111	0.302	0.224	0.582	0.354
VAE [24]	0.607	0.575	0.291	0.245	0.152	0.108	0.282	0.200	0.625	0.329
DEC [40]	0.590	0.601	0.301	0.257	0.185	0.136	0.359	0.276	0.618	0.314
IDEC [18]	0.592	0.604	0.316	0.273	0.191	0.140	0.378	0.324	0.684	0.351
VaDE [22]	0.578	0.630	0.156	0.036	–	–	–	–	0.723	0.416
JULE [41]	0.563	0.608	0.272	0.192	0.137	0.103	0.277	0.182	0.626	0.405
DAC [6]	**0.615**	**0.632**	0.522	0.396	0.238	0.185	0.470	0.366	–	–
DCC [4]	–	–	0.524	0.424	–	–	0.489	0.371	–	–
DCCM [39]	–	–	**0.623**	**0.496**	**0.327**	**0.285**	**0.482**	0.376	–	–
VaGAN-GMM [42]	**0.638**	**0.633**	0.287	0.158	–	–	–	–	**0.801**	**0.536**
DSEC [5]	–	–	0.477	0.437	0.255	0.212	0.481	**0.403**	**0.783**	**0.708**
PICA [20]	–	–	**0.696**	**0.591**	**0.337**	**0.310**	**0.713**	**0.611**	–	–
EDESC (ours)	**0.631**	**0.670**	**0.627**	**0.464**	**0.385**	**0.370**	**0.745**	**0.687**	**0.825**	**0.611**

Table 3. Clustering performance compared with the baseline and state-of-the-art approaches on five experimental datasets. Note that the best three results are marked in **bold**.

VaDE [22], JULE [41], DAC [6], DCC [4], DCCM [39], VaGAN-GMM [42], DSEC [5], and PICA [20]), and subspace clustering methods (SSC [11], LRR [27], KSSC [32], DSC-Net [21], k-SCN [49], DASC [52], DPSC [51], PSSC [28]). Note that we directly report their experimental results from related papers.

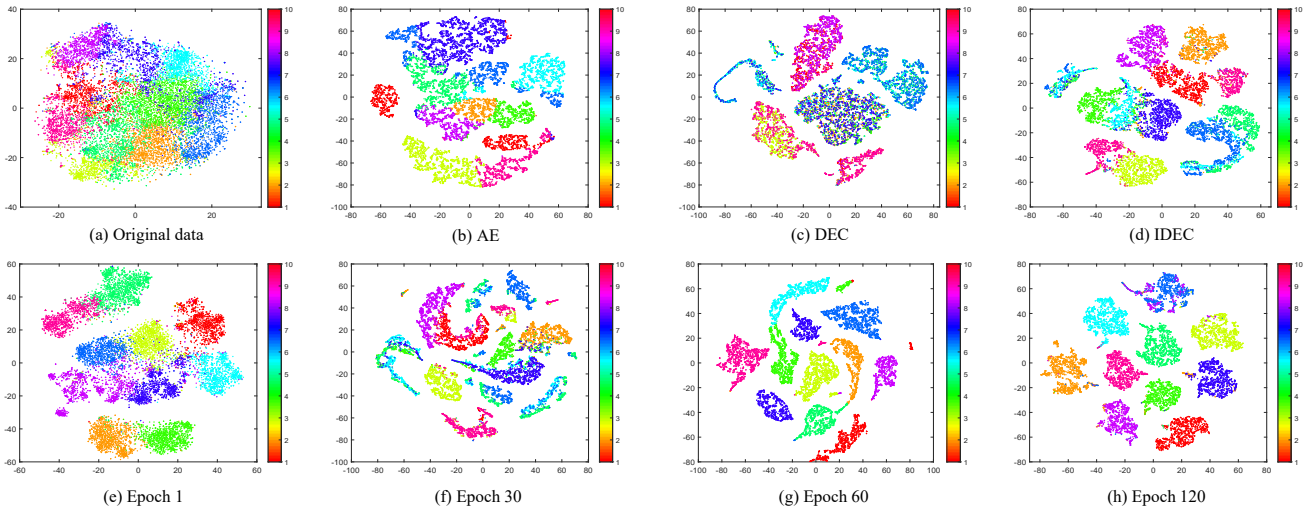
The clustering performance comparison with classical clustering and deep clustering approaches are shown in Tab. 3, where it is observed that our method achieves superior clustering performance on three different types of datasets. Especially on STL-10, the proposed method outperforms PICA by 3.2% and 7.6% in terms of ACC and NMI. Whereas, on the other four datasets, the proposed method also consistently maintains the top three clustering performance. Furthermore, since most of the deep clustering methods are based on the Euclidean distance measure, the observations also imply that the Euclidean distance-based measure is not always valid for all data structures, considering from different measures such as the angular relationship between data may help to reveal a better clustering structure. Table 4 illustrates the performance comparison with several subspace clustering approaches. Methods with SAE means that perform on the features learned from stacked auto-encoder. The proposed method prevails in both ACC and NMI on fashion-MNIST, while outperforming other comparative methods significantly on MNIST. It is worth noting that since subspace clustering methods are mostly based on spectral clustering, which requires the computation of an $n \times n$ affinity matrix and leads to a high time complexity. The comparison of time and space complexity with several clustering methods referred to Tab. 1 also demonstrates the advantage of the proposed method in terms of computational cost. Moreover, the comparison in running time with several clustering methods are also shown in Tab. 5, which further demonstrate the efficiency of our method. It should be mentioned that the proposed method can be implemented with mini-batch, *i.e.*, it has the potential to be extended to online clustering issue, which will be discussed in Sec. 4.8.

Method/Dataset	MNIST		Fashion-MNIST	
	ACC	NMI	ACC	NMI
SSC-SAE [11]	0.754	0.662	0.523	0.512
LRR-SAE [27]	0.740	0.669	0.580	0.591
KSSC-SAE [32]	0.815	0.845	0.571	0.604
DSC-Net [21]	0.532	0.479	0.558	0.548
k-SCN [49]	0.833	0.773	0.600	0.623
DASC [52]	–	–	0.617	0.647
DPSC [51]	–	–	0.624	0.645
PSSC [28]	0.843	0.843	–	–
EDESC (ours)	**0.913**	**0.862**	**0.631**	**0.670**

Table 4. Clustering performance compared with several subspace clustering approaches on MNIST and Fashion-MNIST. Note that the best results are marked in **bold**.

4.4. Qualitative Study

Visualization of Embedded Representation. Figure 4 presents the t-SNE [35] visualization of the learned embedded representation on the challenging dataset STL-10. In



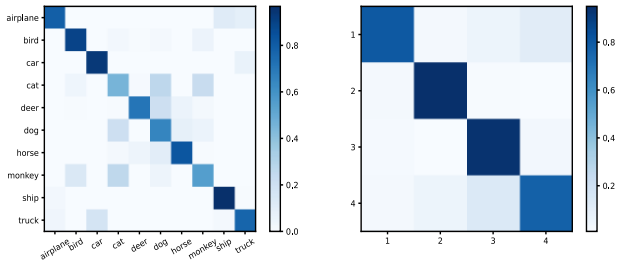
(a) Original data (b) AE (c) DEC (d) IDEC

(e) Epoch 1 (f) Epoch 30 (g) Epoch 60 (h) Epoch 120

Figure 4. Visualization of the embedded representations with t-SNE on STL-10. Note that the first row shows the visualization of several methods used as comparison, and the second row shows the visualization of the proposed method during training.

Dataset	MNIST	Fashion-MNIST	REUTERS-10K
SSC	OT	OT	13038.84
DSC-Net	5364.85	4225.53	N/A
DEC	383.49	346.57	65.92
EDESC (MB)	414.68	372.20	61.09
EDESC (w/o MB)	68.37	59.43	10.53

Table 5. Running time (second) comparison. MB, OT, and N/A denote mini-batch, out of memory, and results not available.



(a) STL-10 (b) REUTERS-10K

Figure 5. Confusion matrices obtained from the predicted clusters of our method and groundtruth on STL-10 and REUTERS-10K.

the first row, the competitors failed to find a good clustering structure. In particular, without the guidance of reconstruction loss, the representation learned from DEC cannot reflect the data structure well, leading to somewhat inferior visualization. The second row shows the visualization of the representation learned by the proposed method in different epochs, as it is important to understand how the representation evolves during training. We see that the embedded representation becomes more and more discriminative as the training epochs increase, and the proposed method finally reveals more significant clustering structures compared to other methods.

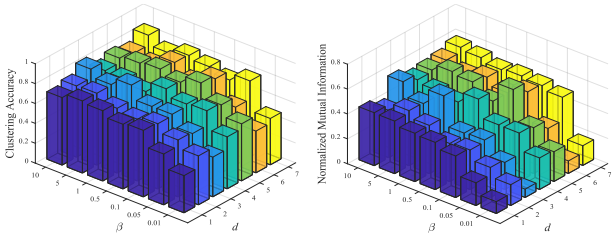
Confusion Matrices. The confusion matrices of the proposed method on STL-10 and REUTERS-10K are shown in Fig. 5, note that the predicted cluster labels as already processed to possess the best mapping to the groundtruth. It can be found a diagonal structure for both confusion matrices, *i.e.*, majority of the samples are correctly assigned to the corresponding classes. Moreover, taking the confusion matrix of STL-10 as an example, there are some interesting observations. The higher salience in the confusion matrix is always the more relevant category in the logic, *e.g.*, 'cat', 'dogs' and 'monkey' belong to 'animal', while 'airplane', 'ship', 'truck' and 'car' belong to 'transportation'. This is consistent with the human mind as these things can sometimes be confused in real-world scenarios.

4.5. Ablation Study

In this section, we conduct an ablation study to explore the impact of each loss term in the proposed method on the clustering performance. Specifically, we construct three degradation models through removing the corresponding loss terms and conduct experiment on the STL-10 and REUTERS-10K datasets. Table 6 summarizes the experimental results of the ablation study, from which we can draw some conclusions. First, L_{Recon} is important to maintain the inherent data structure information during training, which has a great impact on the clustering performance. Second, the clustering objective is crucial in training, because a significant performance decreases can be observed after removing L_{Sub} from the training of both datasets. Third, the constraint on the subspace proxy $\mathbf{D}$ can help the model capture more discriminative embedded representa-

Methods	STL-10		REUTERS-10K	
	ACC	NMI	ACC	NMI
w/oL_{Recon}	0.512	0.565	0.727	0.466
w/oL_{Sub}	0.626	0.658	0.662	0.338
w/oD_{Cons}	0.550	0.618	0.799	0.590
Compete EDESC	**0.745**	**0.687**	**0.825**	**0.611**

Table 6. Ablation study results of the proposed method and its degradation models.



(a) ACC (b) NMI

Figure 6. Parameter sensitivity of d and β of the proposed method on REUTERS-10K.

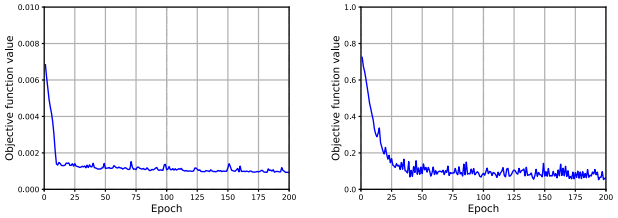
tions, thus improving the clustering performance.

4.6. Parameter Sensitivity

In this section, we analyse the impact of two main hyper-parameters d and β on the clustering performance. Specifically, we set the range of values of d to $[1, 2, \cdots, 6, 7]$ and of β to $[0.01, 0.05, \cdots, 5, 10]$, then conduct experiment on REUTERS-10K. The clustering performance under different parameter values is displayed in Fig. 6, from which we have the following observations. First, the clustering performance is seriously affected when the value of β is too low, especially on NMI, which illustrates that the proposed clustering objective is beneficial for clustering. Second, an excessive β also has a negative impact on the clustering performance. One plausible explanation is that the excessive value influences the learning of the inherent structure of original data, resulting in a perturbation of the embedding space. Third, it seems that NMI is more sensitive to the changes of d compared to ACC. Nevertheless, they maintain relatively good clustering performance at most parameter values. Overall, the recommended value of β ranges from $[0.1, 1]$, and d depends on the number of classes in the dataset, but empirically no more than 10.

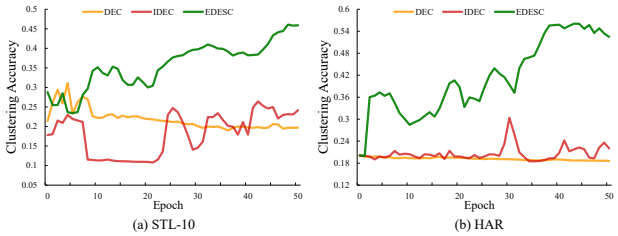
4.7. Convergence Analysis

To validate the convergence of the proposed method, we run 200 epochs on STL-10 and REUTERS-10K datasets, and then present the convergence curves in Fig. 7. It can be observed that the both curves nearly flatten out after 25 epochs, and basically reach convergence after 100 epochs, which demonstrates the convergence and the fast convergence speed of the proposed method.



(a) STL-10 (b) REUTERS-10K

Figure 7. Convergence curves on STL-10 and REUTERS-10K.



(a) STL-10 (b) HAR

Figure 8. Online clustering performance on STL-10 and HAR.

4.8. Online clustering

Online clustering aims to cluster streaming data and hence requires highly efficient algorithms, which is a challenge to existing subspace clustering methods but can be handled by our method. Here we apply our method, in an online manner (detailed in the supplement), to STL-10 and Human Activities Recognition (HAR) [7] datasets compared with DEC and IDEC. The clustering performance is shown in Fig. 8, which verified the feasibility and effectiveness of our method in online clustering.

5. Conclusion

In this paper, we have proposed a novel deep learning based subspace clustering method [8]. The method has linear time and space complexity and hence is applicable to large datasets. The experimental results on many benchmark datasets verified that the proposed method has higher clustering accuracy than its competitors. The main limitation of our work stems from the fully-connected network, which may be enhanced with more complicated network structures.

Acknowledgment

This work is in part supported by the National Natural Science Foundation of China (Grant No. U21A20472), the Natural Science Foundation of Fujian Province (Grant No. 2020J01130193) and Shenzhen Research Institute of Big Data (No.T00120210002).

[7] https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones

[8] Code available at https://github.com/JinyuCai95/EDESC-pytorch

References

[1] Yoshua Bengio, Pascal Lamblin, Dan Popovici, and Hugo Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 153–160, 2007. 1, 5, 6

[2] Deng Cai, Xiaofei He, Xuanhui Wang, Hujun Bao, and Jiawei Han. Locality preserving nonnegative matrix factorization. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1010–1015. International Joint Conferences on Artificial Intelligence, 2009. 1, 5, 6

[3] Jinyu Cai, Shiping Wang, Chaoyang Xu, and Wenzhong Guo. Unsupervised deep clustering via contractive feature representation and focal loss. *Pattern Recognition*, 123:108386, 2021. 1

[4] Jianlong Chang, Yiwen Guo, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep discriminative clustering analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–11, 2019. 6

[5] Jianlong Chang, Gaofeng Meng, Lingfeng Wang, Shiming Xiang, and Chunhong Pan. Deep self-evolution clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):809–823, 2020. 2, 6

[6] Jianlong Chang, Lingfeng Wang, Gaofeng Meng, Shiming Xiang, and Chunhong Pan. Deep adaptive image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5879–5887, 2017. 1, 6

[7] Ying Chen, Chun-Guang Li, and Chong You. Stochastic sparse subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4155–4164, 2020. 2, 3

[8] Zhiyuan Dang, Cheng Deng, Xu Yang, and Heng Huang. Multi-scale fusion subspace clustering using similarity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2020. 1

[9] Zhiyuan Dang, Cheng Deng, Xu Yang, Kun Wei, and Heng Huang. Nearest neighbor matching for deep clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13693–13702, 2021. 1

[10] Ehsan Elhamifar and René Vidal. Sparse manifold clustering and embedding. *Advances in Neural Information Processing Systems*, 24:55–63, 2011. 3

[11] Ehsan Elhamifar and René Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013. 1, 2, 3, 6

[12] Jicong Fan. Large-scale subspace clustering via k-factorization. In *Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 342–352, 2021. 1, 2, 3

[13] Jicong Fan, Zhaoyang Tian, Mingbo Zhao, and Tommy W.S. Chow. Accelerated low-rank representation for subspace clustering and semi-supervised classification on large-scale data. *Neural Networks*, 100:39–48, 2018. 2

[14] Jicong Fan, Chengrun Yang, and Madeleine Udell. Robust non-linear matrix factorization for dictionary learning, denoising, and clustering. *IEEE Transactions on Signal Processing*, 69:1755–1770, 2021. 1

[15] Jicong Fan, Chengrun Yang, and Madeleine Udell. Robust non-linear matrix factorization for dictionary learning, denoising, and clustering. *IEEE Transactions on Signal Processing*, 69:1755–1770, 2021. 3

[16] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in Neural Information Processing Systems*, 27, 2014. 2

[17] K Chidananda Gowda and G Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern Recognition*, 10(2):105–112, 1978. 5, 6

[18] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1753–1759, 2017. 5, 6

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 5

[20] Jiabo Huang, Shaogang Gong, and Xiatian Zhu. Deep semantic clustering by partition confidence maximisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8849–8858, 2020. 2, 6

[21] Pan Ji, Tong Zhang, Hongdong Li, Mathieu Salzmann, and Ian Reid. Deep subspace clustering networks. *Advances in Neural Information Processing Systems*, 30:24–33, 2017. 1, 2, 3, 6

[22] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. Variational deep embedding: an unsupervised and generative approach to clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1965–1972, 2017. 6

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[24] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 2, 5, 6

[25] Changsheng Li, Chen Yang, Bo Liu, Ye Yuan, and Guoren Wang. Lrsc: Learning representations for subspace clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8340–8348, 2021. 1

[26] Yunfan Li, Peng Hu, Zitao Liu, Dezhong Peng, Joey Tianyi Zhou, and Xi Peng. Contrastive clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8547–8555, 2021. 1

[27] Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013. 1, 2, 3, 6

[28] Juncheng Lv, Zhao Kang, Xiao Lu, and Zenglin Xu. Pseudo-supervised deep subspace clustering. *IEEE Transactions on Image Processing*, 30:5252–5263, 2021. 3, 6

[29] James 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. 1, 5, 6

[30] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems*, pages 849–856, 2002. 1

[31] Gaurav Parmar, Dacheng Li, Kwonjoon Lee, and Zhuowen Tu. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 823–832, 2021. 1

[32] Vishal M Patel and René Vidal. Kernel sparse subspace clustering. In *Proceedings of the International Conference on Image Processing*, pages 2849–2853. IEEE, 2014. 2, 3, 6

[33] Xi Peng, Lei Zhang, and Zhang Yi. Scalable sparse subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 430–437, 2013. 2, 3

[34] R. Souvenir and R. Pless. Manifold clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, volume 1, pages 648–653 Vol. 1, 2005. 3

[35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 2, 6

[36] P Vincent Pascalvincent, Larochelle Larocheh, and H S-tacked Denoising Autoencoders. Learning useful representations in a deep network with a local denoising criterion pierre-antoine manzagol. *Journal of Machine Learning Research*, 11:3371–3408, 2010. 1, 2, 5, 6

[37] Shiping Wang, Jinyu Cai, Qihao Lin, and Wenzhong Guo. An overview of unsupervised deep feature representation for text categorization. *IEEE Transactions on Computational Social Systems*, 6(3):504–517, 2019. 1

[38] Xiumei Wang, Tianzhen Zhang, and Xinbo Gao. Multi-view clustering based on non-negative matrix factorization and pairwise measurements. *IEEE Transactions on Cybernetics*, 49(9):3333–3346, 2018. 1

[39] Jianlong Wu, Keyu Long, Fei Wang, Chen Qian, Cheng Li, Zhouchen Lin, and Hongbin Zha. Deep comprehensive correlation mining for image clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8150–8159, 2019. 1, 2, 6

[40] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *Proceedings of the International Conference on Machine Learning*, pages 478–487. PMLR, 2016. 1, 2, 5, 6

[41] Jianwei Yang, Devi Parikh, and Dhruv Batra. Joint unsupervised learning of deep representations and image clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5147–5156, 2016. 1, 6

[42] Lin Yang, Wentao Fan, and Nizar Bouguila. Clustering analysis via deep generative models with mixture models. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–11, 2020. 6

[43] Xu Yang, Cheng Deng, Kun Wei, Junchi Yan, and Wei Liu. Adversarial learning for robust deep clustering. *Advances in Neural Information Processing Systems*, 33, 2020. 1

[44] Xu Yang, Cheng Deng, Feng Zheng, Junchi Yan, and Wei Liu. Deep spectral clustering using dual autoencoder network. In *Proceedings of the IEEE/CVF cConference on Computer Vision and Pattern Recognition*, pages 4066–4075, 2019. 1

[45] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2005. 5, 6

[46] Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6688–6697, 2020. 1

[47] Changqing Zhang, Yeqing Liu, and Huazhu Fu. Ae2-nets: Autoencoder in autoencoder networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2577–2585, 2019. 1

[48] Shangzhi Zhang, Chong You, René Vidal, and Chun-Guang Li. Learning a self-expressive network for subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12393–12403, 2021. 1

[49] Tong Zhang, Pan Ji, Mehrtash Harandi, Richard Hartley, and Ian Reid. Scalable deep k-subspace clustering. In *Proceedings of the Asian Conference on Computer Vision*, pages 466–481. Springer, 2018. 1, 2, 6

[50] Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. In *Proceedings of the International Conference on Machine Learning*, pages 7384–7393. PMLR, 2019. 3

[51] Lei Zhou, Bai Xiao, Xianglong Liu, Jun Zhou, Edwin R Hancock, et al. Latent distribution preserving deep subspace clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 4440–4446. York, 2019. 2, 3, 6

[52] Pan Zhou, Yunqing Hou, and Jiashi Feng. Deep adversarial subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1596–1604, 2018. 2, 3, 6