Deep Structural Contrastive Subspace Clustering

Bo Peng PENGBOCV@126.COM
College of Information Engineering, China Jiliang University, Hangzhou, China
The University of Queensland, Brisbane, Australia

Wenjie Zhu* ZHWJ@CJLU.EDU.CN
College of Information Engineering, China Jiliang University, Hangzhou, China

Editors: Vineeth N Balasubramanian and Ivor Tsang

Abstract

Deep subspace clustering based on data self-expression is devoted to learning pairwise affinities in the latent feature space. Existing methods tend to rely on an autoencoder framework to learn representations for an affinity matrix. However, the representation learning driven largely by pixel-level data reconstruction is somewhat incompatible with the subspace clustering task. With the unavailability of ground truth, can structural representations, which is exactly what subspace clustering favors, be achieved by simply exploiting the supervision information in the data itself? In this paper, we formulate this intuition as a structural contrastive prediction task and propose an end-to-end trainable framework referred as Deep Structural Contrastive Subspace Clustering (DSCSC). Specifically, DSCSC makes use of data augmentation technique to mine positive pairs and constructs a data similarity graph in the embedding feature space to search negative pairs. A novel structural contrastive loss is proposed on the latent representations to achieve positive-concentrated and negative-separated property for subspace preserving. Extensive experiments on the benchmark datasets demonstrate that our method outperforms the state-of-the-art deep subspace clustering methods and imply the necessity of the proposed structural contrastive loss.

Keywords: Deep subspace clustering; Convolutional autoencoder; Contrastive representations.

1. Introduction

Subspace clustering focuses on partitioning the high-dimensional data into disjoint groups corresponding to the subspaces in an unsupervised manner. Motivated by the assumption that each data point can be represented using a linear or affine combination of the rest of data points on the dataset, subspace clustering methods based on data self-expression follow a two-step process by first constructing an affinity matrix between data points and then applying spectral clustering to the affinity matrix. As an excellent affinity matrix can indicate a weighted graph of data points, the subspace clustering technique has been applied successfully in motion segmentation Li et al. (2015a); Xia et al. (2018), face clustering Abavisani and Patel (2018), bioinformatics Zhao et al. (2008), and movie recommendation Li et al. (2015b), to name a few.

* Corresponding author.

The early endeavors in subspace clustering assume that data resides in a union of linear subspaces. Note that the self-expression scheme can be devised by replacing the dictionary Aharon et al. (2006) with the dataset itself, i.e.,

$$\boldsymbol{X} = \boldsymbol{X}\boldsymbol{C}, \quad \text{and} \quad \text{diag}(\boldsymbol{C}) = 0, \tag{1}$$

where $\boldsymbol{X} \in \mathbb{R}^{D\times N}$ is a collection of data points, $\boldsymbol{C} \in \mathbb{R}^{N\times N}$ is the coefficient matrix. It can be formulated as a optimization problem with an elaborate regularization $\mathcal{R}(\boldsymbol{C})$:

$$\begin{aligned} &\hat{\boldsymbol{C}} = \arg\min_{\boldsymbol{C}} ||\boldsymbol{X} - \boldsymbol{X}\boldsymbol{C}||_{\text{F}}^2 + \mathcal{R}(\boldsymbol{C}), \\ &\text{s.t.} \quad \text{diag}(\boldsymbol{C}) = 0. \end{aligned} \tag{2}$$

To avoid the trivial solution, the diagonal elements are enforced to be zero. Finally, affinities of data points can be induced by $(|\hat{\boldsymbol{C}}| + |\hat{\boldsymbol{C}}|^{\text{T}})/2$.

Traditional subspace clustering. The prominent sparse subspace clustering Elhamifar and Vidal (2013); Peng et al. (2013); Chen et al. (2020b); Li et al. (2018, 2017) deems that one data point is linearly related to a few data points in the dataset and encourages coefficients to be sparse. By contrast, the low-rank subspace clustering works Vidal and Favaro (2014); Chen and Yang (2014); Zhu et al. (2018) aim at learning the coefficients with a structured arrangement. Overall, the regularization $\mathcal{R}(\mathbf{C})$ has been investigated with ℓ_0, ℓ_1, nuclear norm, or the combination of them to pursue a more exactly block-diagonal affinity matrix which suits spectral clustering towards the distinct partitions. However, the real-world data does not necessarily conform with the linear subspace model. Therefore, several researchers have proposed kernel-based subspace clustering works Patel and Vidal (2014); Yin et al. (2016) to utilize a series of kernels to replace the inner product of the data matrix. Nevertheless, there is no theoretical guarantee that the empirically defined kernels could correspond to feature spaces that are well-suited for subspace clustering.

Deep subspace clustering. Past decade has witnessed the successful improvement in various applications by extending the traditional learning model to deep neural networks. With the original data Ji et al. (2017); Zhang et al. (2019); Zhou et al. (2018); Zhang et al. (2019); Seo et al. (2019); Kheirandishfard et al. (2020) or hand-crafted features Peng et al. (2016, 2018, 2020), deep subspace clustering methods design an explicit non-linear mapping to exploit complex underlying patterns of data and learn suitable representations that satisfy the self-expressiveness property. To build an end-to-end framework, deep subspace clustering (DSC) network, proposed in Ji et al. (2017), employs a convolutional autoencoder with a plug-in self-expression layer between the encoder and decoder to learn the intermediate deep features. Following the footsteps of DSC, several recent work introduces the clustering-guided loss Zhou et al. (2018); Zhang et al. (2019) to improve the deep representations and achieves promising clustering results.

Most of previous deep subspace clustering approaches are developed based on the standard autoencoder architecture where a decoder network is trained to preserve information in the intermediate representations. However, the current popular reconstruction loss itself, which merely makes use of pixel-level information in the input data, is insufficient to contribute to favorable representations to subspace clustering, which actually prefers the structural ones instead. Different from supervised tasks, the unavailability of ground truth makes it not easy to capture relations within data instances in the subspace clustering

task. In view of this, we propose a novel deep subspace clustering framework called Deep Structural Contrastive Subspace Clustering (DSCSC) by designing an elaborate structural contrastive prediction task which simply exploits the potential supervision information in the dataset. Particularly, given input data, DSCSC applies random data transformation on each of them to constitute positive pairs and construct a similarity graph on the embedding feature space to mine negative pairs. With the help of the proposed structural contrastive loss, the positive-concentrated and negative-separated property can be approximated so that the structure information in the dataset is preserved in the learned intermediate representations.

The contributions of this paper can be summarized as follows:

1. We propose an end-to-end trainable framework which, to the best of our knowledge, is the first one to incorporate contrastive learning and self-expression learning into a Siamese network for subspace clustering.

2. We design a novel structural contrastive loss which approximates the positive-concentrated and negative-separated property to facilitate structural representations which are more favored to the subspace clustering task.

3. Experiment results on benchmark datasets show that the proposed DSCSC outperforms other state-of-the-art subspace clustering methods and demonstrate the superiority of structural contrastive representations.

2. Related Work

In this paper, we aim to design a subspace-clustering-specific structural contrastive representation learning algorithm. Therefore, we briefly introduce the traditional contrastive learning methods and previous deep subspace clustering methods in this section.

2.1. Contrastive learning

Recently, contrastive learning has achieved state-of-the-art performance in unsupervised representation learning. The key motivation behind contrastive learning is to attract the positive sample pairs and repulse the negative sample pairs. In practice, contrastive learning methods profit from a large number of negative samples. Specifically, NCE Wu et al. (2018) introduces a memory bank to preserve negative samples. Based on a Siamese network, MoCo proposed in He et al. (2020) maintains a queue of negative samples and turns one branch into a momentum encoder to improve consistency of the queue. With the help of data augmentation, SimCLR Chen et al. (2020) directly uses negative samples coexisting in the current batch, which requires a large batch size to work well. Different from the current approaches contrastive learning which treat every training sample as an independent class, our framework proposes a structured contrastive loss with the consideration of the underlying supervision across the dataset to learn structural representations which are more favorable for subspace clustering.

2.2. Deep subspace clustering networks

The deep subspace clustering with sparse prior firstly computes the sparse regularized self-expression coefficients in the original space and then updates the network to preserve the coefficients for deep representations clustering Peng et al. (2016, 2018). The participation of pre-defined self-expression coefficients facilitates the deep representations complementing global structure information by learning to progressively transform input data into nonlinear latent space. However, the subspace clustering results hinge on the input hand-crafted features Peng et al. (2016, 2018).

Considering the unit sphere distribution assumption for the deep features, Peng et al. extend their work Peng et al. (2016, 2018) into a dynamical sparse representation searching task Peng et al. (2020). Ji et al. propose a convolutional autoencoder network which treats the original image as the input of networks and performs the deep representation learning and self-expression coefficient learning simultaneously Ji et al. (2017). Subsequently, some works introduce regularization on the intermediate feature for more powerful representation ability Seo et al. (2019); Kheirandishfard et al. (2020). In addition, the clustering-guided methods have been proposed in Zhou et al. (2018); Zhang et al. (2019). Deep adversarial subspace clustering (DASC), introduced in Zhou et al. (2018), consists of a subspace clustering generator and a quality-verifying discriminator, which learns against each other. In the GAN-alike architecture of DASC, the generator runs subspace estimation and sample clustering while the discriminator evaluates current clustering performance by inspecting whether the re-sampled data from the estimated subspaces is equipped with the corresponding subspace properties. DASC proposes a subspace estimation criterion to extract more proper deep feature for real subspace generation. Turning to the self-supervised deep subspace clustering method Zhang et al. (2019), it introduces a dual self-supervision to supervise representation learning and self-expression coefficient learning. Different from DASC, the self-supervised deep subspace clustering method Zhang et al. (2019) introduces the spectral clustering loss for improving the deep representations.

Nevertheless, the networks aforementioned ignore the supervision knowledge in the data itself. Inspired by the contrastive learning in unsupervised representation learning Chen et al. (2020a), we propose a novel network which aims to learn positive concentrated and negative separated representations by discovering the negative neighborhood for subspace clustering. Based on the autoencoder structure, the proposed DSCSC maps the encoded representation into another latent space where 1) the data point is close to its augmented counterpart and 2) the data point stays away from its negative neighbors. The combination of the autoencoder loss and structural contrastive loss helps to strengthen the discriminative ability of the intermediate representations for subspace clustering.

3. Methodology

Previous deep subspace clustering works pursue a minimum reconstruction of autoencoders Ji et al. (2017). However, it seems to be suboptimal to subspace clustering works. For one thing, subspace clustering requires a robust geometric structure of data points, rather than the pixel-level reconstructions. For another, the optimization of autoencoder-based reconstruction can decrease the structural information existing in the data points. Therefore, a relatively more cluster-friendly representation is necessary in the subspace clustering task.

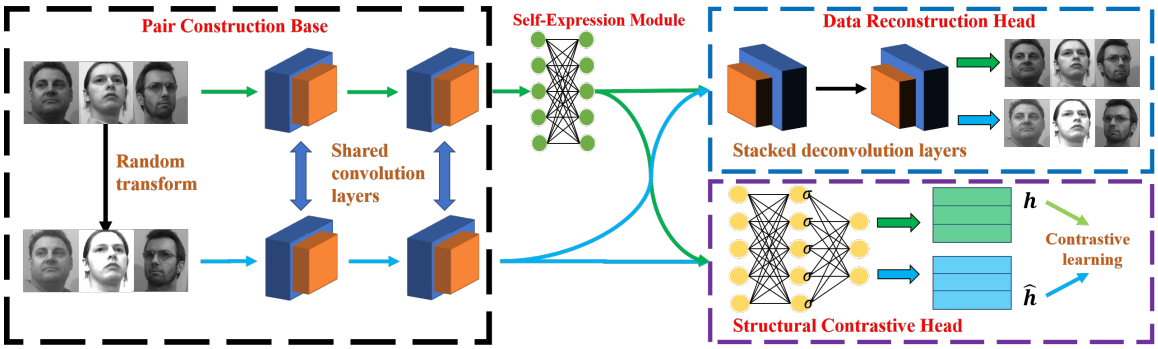


Figure 1: Deep subspace clustering network with structural contrastive representations. Along the data stream, the images and augmented ones are fed into a shared convolution layers to obtain the intermediate deep features. Then, the output features from self-expression layers and augmented features are decoded with deconvolution layers in DRH. Meanwhile, they are projected by fully-connected layers into low-dimensional representations for structural contrastive learning.

In this paper, the pixel-level reconstruction and structural contrastive representation are integrated into the deep subspace clustering framework to cluster multiple data points into multiple groups in an end-to-end way where each group includes data points from the same subspace with the help of our DSCSC. To this end, as illustrated in Figure 1, the proposed DSCSC network is designed with four principle modules:

- Pair Construction Base (PCB) that includes a random data augmentation module $T(\cdot)$ to stochastically transform data and a convolution encoder $f(\cdot)$ to extract features of both given data points and the corresponding augmented ones.
- Self-Expression Module (SEM) that enables the network to learn the intermediate features and affinities simultaneously.
- Structural Contrastive Head (SCH) $\mathcal{G}_1(\cdot)$ projects the encoded representations into another latent feature space where the proposed structural contrastive loss is applied.
- Data Reconstruction Head (DRH) $\mathcal{G}_2(\cdot)$ that projects the encoded representations into the original data space for data reconstruction.

As follows, we briefly introduce the architecture of the proposed network in more detail at first. Afterwards, a simple yet effective training strategy would be described to make it easy to train our network.

3.1. Pair construction base

Given one data point $\boldsymbol{x}_i$, we firstly apply stochastic data transformation on it to obtain another view of the data denoted as $\hat{\boldsymbol{x}}_i = T(\boldsymbol{x}_i)$. In this way, we can build a positive pair which comprises the image itself and its augmented one. Subsequently, a convolution

encoder is adopted to extract features from original images and the augmented samples respectively, resulting in $\boldsymbol{z}_i = f(\boldsymbol{x}_i)$ and $\hat{\boldsymbol{z}}_i = f(\hat{\boldsymbol{x}}_i)$. Note that our method does not rely on a specially designed encoder. Here, we simply exploit the same network architecture in Ji et al. (2017); Zhou et al. (2018); Zhang et al. (2019) as the encoder $f(\cdot)$ for a fair comparison.

3.2. Self-expression module

Following Ji et al. (2017); Zhou et al. (2018); Zhang et al. (2019), we also make use of the self-expression property Elhamifar and Vidal (2013) that interprets each point in a subspace as a linear combination of other points in the same subspace for subspace clustering in this paper. Besides, considering the significance that the learned representations are well-suitable for subspace clustering, we integrate a self-expression module into our network to enable affinity estimation and representation learning to be performed at the same time. Essentially, SEM works as a simple fully-connected layer without any biases and non-linear activation functions. Therefore, SEM imposes the following loss function in our network:

$$\begin{aligned} &\mathcal{L}_{\text{SEM}} = \frac{\alpha}{2}||\boldsymbol{Z} - \boldsymbol{Z}\boldsymbol{C}||_{\text{F}}^2 + ||\boldsymbol{C}||_{\text{F}}^2, \\ &\text{s.t.} \quad \text{diag}(\boldsymbol{C}) = 0, \end{aligned} \tag{3}$$

where $\boldsymbol{Z} = \{\boldsymbol{z}_i\}_{i=1,2,\cdots,N} \in \mathbb{R}^{d \times N}$ indicates the intermediate deep representations obtained by the encoders, $\boldsymbol{C} \in \mathbb{R}^{N \times N}$ represents the coefficient matrix, the diagonal constraint on $\boldsymbol{C}$ is designed to avoid the trivial solution that $\boldsymbol{C} = \boldsymbol{I}$ and α denotes to a tradeoff parameter.

3.3. Structural contrastive head

Since PCB has already performed data augmentation on the image data, we have $2N$ data samples $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \cdots, \boldsymbol{x}_N, \hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \hat{\boldsymbol{x}}_3, \cdots, \hat{\boldsymbol{x}}_N\}$. For each of image example $\boldsymbol{x}_i$, we only treat $\hat{\boldsymbol{x}}_i$ as its positive sample and leave the remaining $2(N-1)$ samples out. Afterwards, the contrastive learning is applied to learn structural contrastive representations of data points with the positive-concentrated and negative-seperated property. In order to escape from information loss caused by the contrastive learning, we adopt a structural contrastive head (SCH) to project the encoded representation into another latent space via $\boldsymbol{h}_i = \mathcal{G}_1(\boldsymbol{z}_i)$ and $\hat{\boldsymbol{h}}_i = \mathcal{G}_1(\hat{\boldsymbol{z}}_i)$. As a result, the intermediate features extracted from the encoder is more likely to preserve meaningful information for the subspace clustering task. We note that SCH is implemented as two stacked MLP layers, namely $\boldsymbol{h}_i = \mathcal{G}_1(\boldsymbol{z}_i)$ and $\hat{\boldsymbol{h}}_i = \mathcal{G}_1(\hat{\boldsymbol{z}}_i)$. The positive-concentrated property enables data points to be close to their augmented counterparts in the feature space while the negative-seperated property enforces different data points to be far away from their negative neighbors. Specifically, given a data point $\boldsymbol{x}_i$, we expect its positive sample $\hat{\boldsymbol{x}}_i$ to be recognized as $\boldsymbol{x}_i$ and its negative neighbors not to be classified into $\boldsymbol{x}_i$. To begin with, the possibility $P(i|\hat{\boldsymbol{x}}_i)$ that $\hat{\boldsymbol{x}}_i$ belongs to $\boldsymbol{x}_i$ can be defined as follows:

$$P(i|\hat{\boldsymbol{x}}_i) = \frac{\exp\left(\text{sim}(\boldsymbol{h}_i^T \hat{\boldsymbol{h}}_i)/\rho\right)}{\sum_{k=1}^{N} \exp\left(\text{sim}(\boldsymbol{h}_k^T \hat{\boldsymbol{h}}_i)/\rho\right)}, \tag{4}$$

where ρ represents the temperature factor that controls the concentration level of the distribution and $\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{v}/(||\boldsymbol{u}|| \cdot ||\boldsymbol{v}||)$ denotes to the pair-wise similarity measured by the cosine distance.

Moreover, the probability of $\boldsymbol{x}_j$ being classified as $\boldsymbol{x}_i$ is defined by

$$P(i|\boldsymbol{x}_j) = \frac{\exp\left(\text{sim}(\boldsymbol{h}_i^T\boldsymbol{h}_j)/\rho\right)}{\sum_{k=1}^{N}\exp\left(\text{sim}(\boldsymbol{h}_k^T\boldsymbol{h}_j)/\rho\right)}. \tag{5}$$

Correspondingly, the probability of $\boldsymbol{x}_j$ being not recognized as $\boldsymbol{x}_i$ is referred as $1 - P(i|\boldsymbol{x}_j)$.

Considering the event that every data point is recognized as $\boldsymbol{x}_i$ happens mutually independently, the joint possibility P_i that $\hat{\boldsymbol{x}}_i$ belongs to $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ being not recognized as $\boldsymbol{x}_i$ is calculated in the following equation.

$$P_i = P(i|\hat{\boldsymbol{x}}_i)\prod_{j\neq i}^{N}(1 - P(i|\boldsymbol{x}_j)). \tag{6}$$

As a result, the loss function of SCH can be expressed as the average of the negative log likelihood over all data points, leading to:

$$\mathcal{L}_{\text{SCH}} = -\frac{1}{N}\sum_i\left(\log P(i|\hat{\boldsymbol{x}}_i) + \sum_{j\neq i}\log(1 - P(i|\boldsymbol{x}_j))\right). \tag{7}$$

Obviously, the optimization problem of Eq. (7) can be solved by maximizing Eq. (4) while minimizing Eq. (5). On the one hand, the maximization of Eq. (4) requires to increase the value of $\exp\left(\text{sim}(\boldsymbol{h}_i^T\hat{\boldsymbol{h}}_i)/\rho\right)$ and decreases that of $\exp\left(\text{sim}(\boldsymbol{h}_k^T\hat{\boldsymbol{h}}_i)/\rho\right)$, $k \neq i$. This not only enlarges the similarity of positive pairs and but also shrinks the similarity of negative pairs, which promotes the augmentation-invariance and the spread-out property of the learned representations. On the other hand, the minimization of Eq. (4) is to minimize $\exp\left(\text{sim}(\boldsymbol{h}_i^T\boldsymbol{h}_j)/\rho\right)$, which further makes $\boldsymbol{x}_i$ separated from $\boldsymbol{x}_j$ and enhances the spread-out property. Although the loss function defined in Eq. (7) is beneficial to ensure the positive-concentrated and negative-seperated properties, such an instance-level learning mechanism does not take instance-to-instance correlation into account, simply selecting all of other original data points as the negative neighbors of the given data point $\boldsymbol{x}_i$. This makes it difficult to represent the desired cluster memberships from the learned representations and thus poses considerable barriers on the subspace clustering task. To alleviate the aforementioned problem, we propose a structural contrastive learning strategy to facilitate feature learning in our network. In particular, observing that data points in the same cluster tend to be close to each other and far from those in the different clusters, we are motivated to take advantage of the encoded representations of data points for negative neighbors discover. As a result, the loss function of SCH preliminarily defined in Eq. (7) can be rewritten as follows:

$$\mathcal{L}_{\text{SCH}} = -\frac{1}{N}\sum_i\left(\log P(i|\hat{\boldsymbol{x}}_i) + \sum_{j\notin N(i,k)}\log(1 - P(i|\boldsymbol{x}_j))\right), \tag{8}$$

where $N(i,k)$ indicates the k-nearest original data points of $\boldsymbol{x}_i$ with regard to the encoded feature $\boldsymbol{z}_i$. The evolvement of loss function from (7) to (8) means that we incorporate structural contrastive loss into the optimization framework of our network and using the underlying structure across the dataset to supervise representation learning. The structural supervision information facilitates cluster relationship exploration and cluster boundary determination, contributing to better clustering performance.

3.4. Data reconstruction head

To equip the encoded feature with meaningful and useful information about the original data as much as possible, we introduce a data reconstruction head (DRH) that is made up of several stacked deconvolution layers to project the encoded representation into the original space for data reconstruction, i.e. $\boldsymbol{y}_i = \mathcal{G}_2(\boldsymbol{z}_i)$ and $\hat{\boldsymbol{y}}_i = \mathcal{G}_2(\hat{\boldsymbol{z}}_i)$, where $\boldsymbol{y}_i$ and $\hat{\boldsymbol{y}}_i$ indicates the reconstructed $\boldsymbol{x}_i$ and $\hat{\boldsymbol{y}}_i$ respectively. Thus, the loss function for DRH is the reconstruction error given as follows:

$$\mathcal{L}_{\text{DRH}} = -\frac{1}{2N}\sum_{i=1}^{N}(\frac{1}{2}||\boldsymbol{x}_i - \boldsymbol{y}_i||_2^2 + \frac{1}{2}||\hat{\boldsymbol{x}}_i - \hat{\boldsymbol{y}}_i||_2^2). \tag{9}$$

3.5. Training strategy

On the one hand, since the datasets adopted for subspace clustering in this paper are not large-scale (e.g., in the order of thousands of images), our network can still remain of a tractable size. On the other hand, this, in turn, would make it difficult to directly train a network with millions of parameters from scratch. As described later, we design a simple yet effective training strategy to facilitate network training. Overall, the whole training process is performed in two independent stages.

Stage I: Pre-training network without SEM

we firstly remove SEM and subsequently pre-train our network without the self-expression module. Therefore, we adopt the overall cost function given in Eq. (10) to train the network in this stage.

$$\mathcal{L} = \mathcal{L}_{\text{DRH}} + \beta\mathcal{L}_{\text{SCH}}, \tag{10}$$

where $\mathcal{L}_{\text{DRH}}$, $\mathcal{L}_{\text{SCH}}$ represent the loss functions designed for DRH and SCH that have been introduced above and β is a trade-off parameter for balancing.

Stage II: Fine-tuning the whole network with SEM

In this stage, we first initialize the network with the weights learned from the previous stage and fine-tune the whole network together with self-expression module afterwards. Consequently, the overall cost function we design for training in this stage is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{DRH}} + \beta\mathcal{L}_{\text{SCH}} + \gamma\mathcal{L}_{\text{SEM}}, \tag{11}$$

where $\mathcal{L}_{\text{DRH}}$, $\mathcal{L}_{\text{SCH}}$, and $\mathcal{L}_{\text{SEM}}$ represent the loss functions designed for DRH, SCH and SEM and β, γ indicate trade-off parameters to balance the effect of different modules.

After the training stage, we no longer need to estimate the affinity among data points. Instead, we directly use the learned weights in SEM to construct the affinity matrix as $\boldsymbol{A} = \frac{1}{2}(|\boldsymbol{C}| + |\boldsymbol{C}|^T)$. Lastly, spectral clustering is performed on the resulting affinity matrix

$\boldsymbol{A}$ to obtain a segmentation of the data. All details of the training procedure are specified in Algorithm 1.

Algorithm 1 Training procedure for DSCSC

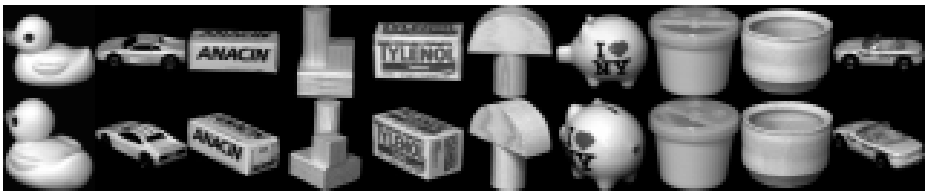
Input: original images $\mathbf{X}$, tradeoff parameters, neighborhood number k, temperature factor ρ, augmentation family $T(\cdot)$, and maximum iteration T_{max};
1. Pre-train the network without SEM via Eq. (10);
2. Randomly Initialize the self-expressive module;
for *epoch* = *1:*T_{max} **do**
    3. Run the encoder $f(\cdot)$ in PCB to extract features of images;
    4. Search negative neighbors of each image;
    5. Fine-tune the network via Eq. (11);
end
6. Run the spectral clustering on the affinity matrix;
Output: label assignment $\mathbf{Q}$.

4. Experiment and Discussion

4.1. Experiments setting

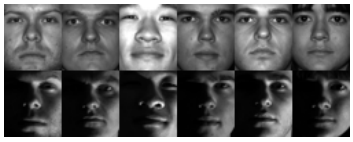
To evaluate the clustering performance of the proposed DSCSC, we conduct extensive experiments on five benchmark datasets, namely MNIST, COIL20, ORL, Extended Yale B and UMIST. Figure 2 illustrates the image examples of each dataset.

(a) MNIST



(b) COIL20

(c) ORL



(d) Extended Yale B

(e) UMIST

Figure 2: Examples of the five benchmark datasets.

We compare our network with a wide range of benchmark subspace clustering methods including sparse subspace clustering (SSC) Elhamifar and Vidal (2009, 2013), elastic net subspace clustering (ENSC) You et al. (2016a), kernel SSC (KSSC) Patel and Vidal (2014), SSC by orthogonal matching pursuit (SSC-OMP) You et al. (2016b), efficient dense subspace clustering (EDSC) Pan Ji et al. (2014), low-rank representation (LRR) Liu et al. (2010), low-rank subspace clustering (LRSC) Vidal and Favaro (2014), deep subspace clustering network (DSC-Net) Ji et al. (2017), SSC with pre-trained convolutional auto-encoder features (AE+SSC), Deep Adversarial Subspace Clustering (DASC) Zhou et al. (2018), Self-Supervised Convolutional Subspace Clustering Network (S^2ConSCN) Zhang et al. (2019), and Deep Low-rank Subspace Clustering (DLRSC) Kheirandishfard et al. (2020).

We search augmentation policies to find the best policy yields the highest mean Silhouette coefficient on different datasets via searching strategy proposed in Abavisani et al. (2020) using the default hyperparameter setting. We set the temperature $\tau = 0.08$ for all datasets. For the trade-off parameters α and γ, we set $\alpha = 1.0 \times 10^{\frac{n}{10}-3}$ and $\gamma = 1$ where n is the number of subspaces in each dataset. For the specific parameters for the proposed structural contrastive loss, we set $\beta = 10$. To better illustrate the effectiveness of our method, we introduce two kinds of criterion, accuracy (ACC) and normalized mutual information (NMI) Nguyen et al. (2010), to measure the clustering results. In all experiments, we fix the learning rate and temperature factor to be 1.0×10^{-3} and 0.1 respectively. Meanwhile, we adopt Adam and the Rectified Linear Unit (ReLU) Krizhevsky et al. (2012) as the optimizer and nonlinear activation function in our network. The output embedding feature dimension in SCH is defined to be 64 for ORL, MNIST and UMIST, 128 for Yale B and 256 for COIL20. The network architectures for different datasets are specified in Table 1.

Table 1: Network parameters ((kernel size) × #channels) for different datasets.

Layers	ORL	UMIST	Yale B	MNIST	COIL20
PCB-1	(3×3)×3	(3×3)×15	(5×5)×10	(5×5)×20	(3×3)×15
PCB-2	(3×3)×3	(3×3)×10	(3×3)×20	(3×3)×10	-
PCB-3	(3×3)×5	(3×3)× 5	(3×3)×30	(3×3)× 5	-
DRH-1	(3×3)×5	(3×3)× 5	(3×3)×30	(3×3)× 5	-
DRH-2	(3×3)×3	(3×3)×10	(3×3)×20	(3×3)×10	-
DRH-3	(3×3)×3	(3×3)×15	(5×5)×10	(5×5)×20	(3×3)×15

4.2. Experiments on ORL, Yale B and UMIST

In this section, we first test the clustering performance of our method on several fine-grained face datasets. In terms of ORL, it includes face images of 40 different subjects, each of which merely has 10 images taken under varying facial expressions and facial details. In the experiments on ORL, we down-sample images to 32×32 and adopt a three-layer convolutional network for the encoder in PCB and the corresponding a three-layer deconvolutional network for DRH. We set the kernel size to 3×3 and channels to 3-3-5 and 5-3-3 individually. When it comes to Yale B, the dataset consists of up to 2432 frontal face images for 38 individuals under different illumination. Following previous methods, we down-sample images

Table 2: Clustering results (%) on ORL, Yale B and UMIST datasets.

Dataset	ORL		Yale B		UMIST	
Metric	ACC	NMI	ACC	NMI	ACC	NMI
LRSC	67.75	80.82	67.43	77.19	67.29	74.98
SSC	67.00	82.06	62.54	86.04	69.04	74.89
EnSC	69.25	82.20	61.14	82.43	69.31	75.69
EDSC	70.38	77.99	84.99	86.36	69.37	75.22
AE+SSC	74.50	88.24	74.75	77.64	70.42	75.15
SSC-OMP	71.00	79.52	73.72	78.03	64.38	70.68
KSSC	71.43	80.70	69.21	73.59	65.31	73.77
DSCNet-L1	85.50	90.23	96.81	96.87	72.42	75.56
DSCNet-L2	86.00	90.34	97.33	97.03	73.12	76.62
DASC	88.25	93.15	98.56	98.01	76.88	80.42
S^2ConSCN-L1	89.50	—	98.48	—	—	—
S^2ConSCN-L2	88.75	—	98.44	—	—	—
DSCSC (ours)	**90.75**	**94.44**	**98.64**	**98.15**	**81.75**	**85.42**

to 48×42. The encoder in PCB comprises three stacked convolutional layers with 10, 20 and 20 channels. Turning to UMIST, it only contains 20 persons, each with 24 images being taken under very different poses. we use a three-layer convolutional structure as the encoder in the PCB and correspondingly three stacked deconvolutional layers to form DRH. The kernel sizes and channels of the encoder (DRH) are 3-3-3 and 15-10-5 (3-3-3 and 5-10-15).

The experiment results on the two datasets is provided in Table 2. It is clear that the proposed DSCSC consistently outperforms other the state-of-the-art deep subspace clustering methods, which further demonstrates the significance in utilizing contrastive learning for subspace. As will illustrated later, the success of our DASC can be attributed to the participation of neighborhood-level contrastive learning.

4.3. Experiments on MNIST and COIL20

We further conduct experiments on the MNIST dataset for handwritten digit recognition. MNIST consists of 70000 hand-written digit images of size 28×28. Due to computation limitation, we randomly select 100 images of each category for clustering, resulting in a subset of 1,000 images. In this experiment, we use a three-layer convolutional structure as the encoder in the PCB and correspondingly three stacked deconvolutional layers to form DRH. The kernel sizes and channels of the encoder (DRH) are 5-3-3 and 20-10-5 (3-3-5 and 5-10-20). COIL20 consists of 1440 gray-scale image samples, distributed over 20 objects. Object images in the dataset tend to more diverse since they are taken with poses varying at an interval of 5 degrees. Following most of previous methods, We down-sample each of images to 32×32. For such a special dataset, the encoder in PCB have only one layer with 15 channels and a 3×3 kernel.

The experiment results on the two datasets are reported in Table 3. It is noticeable that traditional subspace clustering techniques produce relatively unsatisfying clustering results, which means that deep features can assist in subspace clustering. Among deep clustering

Table 3: Clustering results (%) on MNIST and COIL20 datasets.

Dataset	MNIST		COIL20	
Metric	ACC	NMI	ACC	NMI
LRSC	51.40	55.76	74.16	84.52
SSC	45.30	47.09	86.31	88.92
EnSC	49.83	54.95	87.60	89.52
EDSC	56.50	57.52	83.71	88.28
AE+SSC	48.40	53.37	87.11	89.90
SSC-OMP	34.00	32.72	64.10	74.12
KSSC	52.20	56.23	70.87	82.43
DSCNet-L1	72.80	72.17	93.05	93.53
DSCNet-L2	75.00	73.19	94.86	94.08
DASC	80.40	78.00	96.39	96.86
S^2ConSCN-L1	—	—	97.86	—
S^2ConSCN-L2	—	—	97.67	—
DSCSC (ours)	**85.12**	**82.12**	**97.88**	**97.42**

techniques, AC+SSC results in the worst clustering results, showing the superiority of end-to-end scheme for feature learning and affinity estimation. Lastly, DSCSC achieves the best clustering performance, which demonstrates the effectiveness of combing contrastive learning with subspace clustering.

4.4. Ablation Study

4.4.1. Module contribution.

Table 4 reports the contribution of modules in the proposed DSCSC with regard to the clustering performance on ORL. Note that, when the proposed DSCSC only has the encoder for feature extraction and DRH for data reconstruction, it works in the same way as DSCNet Ji et al. (2017). We note that data augmentation plays much difference in our method. When no data augmentation is introduced, every positive pair consists of totally same data and thus only negative neighbors take part in model optimization, which leads to pretty poor results. With help of DRH, the encoded representation are enforced to preserve the useful information of original input data as much as possible, which makes the learned features more meaningful and therefore facilitate contrastive learning in SCH. That is the reason why our DSCSC is able to significantly outperforms other state-of-the-art deep subspace clustering techniques with the joint contribution of data augmentation, DRH and SCH.

Table 4: Clustering results (%) on ORL dataset with different combinations of modules.

Metric	ACC	NMI
DRH	86.00	90.34
DRH+SCH	87.00	92.10
Augmentation+DRH+SCH	**90.75**	**94.44**

4.4.2. Effect of structural contrastive learning.

Figure 3 shows how the clustering performance varies with different values of k. Clearly, when $k = 0$, the adopted contrastive loss would turn to be instance-level. This would make it difficult to represent the desired cluster memberships from the learned intermediate features. As k grows, an increasing amount of valuable data could get rid of being treated as negative neighbors, which facilitates cluster-discriminative representation learning and subspace clustering. As could be seen from Figure 3, when k is set between 10 and 20, the clustering performance of DSCSC does not change largely and almost keeps better than that of S^2ConSCN (89.50%), implying that our method is not highly depend on a well-designed k.

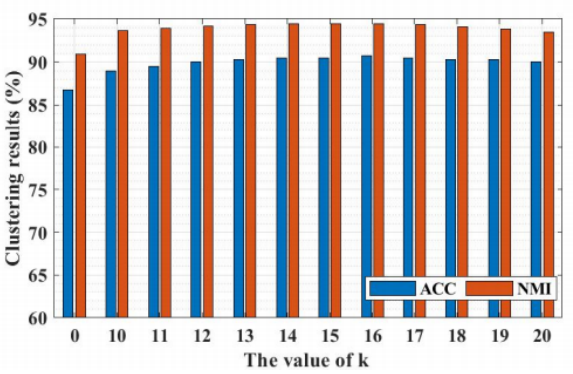


Figure 3: Clustering results (%) versus k on ORL dataset.

5. Conclusion

In this paper, we propose a trainable end-to-end deep contrastive subspace clustering (DSCSC) network. Apart from the convolution auto-encoder framework, we also integrate structural contrastive learning into the proposed network to jointly learn pixel-level and structural contrastive representations for subspace-preserving affinity construction. Thanks to the introduced structural contrastive loss, the supervision information existing in the data itself helps to make the encoded feature positive-concentrated and negative-separated, leading to the state-of-the art clustering performance. Extensive experiments on benchmark datasets demonstrates the effectiveness of the proposed DSCSC.

Acknowledgments

This research was supported by Zhejiang Provincial Natural Science Foundation of China under Grant No. LQ20F030015 and the Key Laboratory of Electromagnetic Wave Information Technology and Metrology of Zhejiang Province.

References

M. Abavisani and V. M. Patel. Deep multimodal subspace clustering networks. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1601–1614, 2018.

M. Abavisani, A. Naghizadeh, D. N. Metaxas, and A. M. Patel. Deep subspace clustering with data augmentation. In *NeurIPS*, pages 1925–1931, 2020.

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11): 4311–4322, 2006.

J. Chen and J. Yang. Robust subspace segmentation via low-rank representation. *IEEE Transactions on Cybernetics*, 44(8):1432–1445, 2014.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. 2020.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. Unsupervised embedding learning via invariant and spreading instance feature. In *International Conference on Machine Learning*, 2020a.

Y. Chen, C. G. Li, and C. You. Stochastic sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4154–4163, 2020b.

E. Elhamifar and R. Vidal. Sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2790–2797, 2009.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.

K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

P. Ji, T. Zhang, H. D. Li, M. Salzmann, and I. Reid. Deep subspace clustering networks. In *Advanced Neural Information Processing Systems*, 2017.

M. Kheirandishfard, F. Zohrizadeh, and F. Kamangar. Deep low-rank subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 3776–3781, 2020.

Alex Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

C. Li, C. You, and R. Vidal. Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework. *IEEE Transactions on Image Processing*, 26(6):2988–3001, 2017.

C. Li, C. You, and R. Vidal. On geometric analysis of affine sparse subspace clustering. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1520–1533, 2018.

S. Li, K. Li, and Y. Fu. Temporal subspace clustering for human motion segmentation. In *IEEE International Conference on Computer Vision*, pages 4453–4461, 2015a.

Z. Li, J. Liu, J. Tang, and H. Lu. Robust structured subspace learning for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10): 2085–2098, 2015b.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

Xuan Vinh Nguyen, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary? *Journal of Machine Learning Research*, 11(1):2837–2854, 2010.

Pan Ji, M. Salzmann, and Hongdong Li. Efficient dense subspace clustering. In *IEEE Winter Conference on Applications of Computer Vision*, pages 461–468, 2014.

V. M. Patel and R. Vidal. Kernel sparse subspace clustering. In *IEEE International Conference on Image Processing*, pages 2849–2853, 2014.

X. Peng, L. Zhang, and Z. Yi. Scalable sparse subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 430–437, 2013.

X. Peng, S. Xiao, J. Feng, W. Yau, , and Z. Yi. Deep subspace clustering with sparsity prior. In *International Joint Conference on Artificial Intelligence*, 2016.

X. Peng, J. Feng, S. Xiao, W. Yau, J. T. Zhou, and S. Yang. Structured autoencoders for subspace clustering. *IEEE Transactions on Image Processing*, 27(10):5076–5086, 2018.

X. Peng, J. Feng, J. T. Zhou, Y. Lei, and S. Yan. Deep subspace clustering. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–13, 2020.

J. Seo, J. Koo, and T. Jeon. Deep closed-form subspace clustering. In *IEEE International Conference on Computer Vision Workshop*, pages 633–642, 2019.

R. Vidal and P. Favaro. Low rank subspace clustering. *Pattern Recognition Letters*, 43(1): 47–61, 2014.

Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

G. Xia, H. Sun, L. Feng, G. Zhang, and Y. Liu. Human motion segmentation via robust kernel sparse subspace clustering. *IEEE Transactions on Image Processing*, 27(1):135–150, 2018.

M. Yin, Y. Guo, J. Gao, Z. He, and S. Xie. Kernel sparse subspace clustering on symmetric positive definite manifolds. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5164, 2016.

C. You, C. Li, D. P. Robinson, and R. Vidal. Oracle based active set algorithm for scalable elastic net subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3928–3937, 2016a.

C. You, D. P. Robinson, and R. Vidal. Scalable sparse subspace clustering by orthogonal matching pursuit. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3918–3927, 2016b.

J. Zhang, C. Li, C. You, X. Qi, H. Zhang, J. Guo, and Z. Lin. Self-supervised convolutional subspace clustering network. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5468–5477, 2019.

Tong Zhang, Pan Ji, Mehrtash Harandi, Wenbing Huang, and Hongdong Li. Neural collaborative subspace clustering. In *International Conference on Machine Learning*, 2019.

Y. Zhao, J. Xu Yu, G. Wang, L. Chen, B. Wang, and G. Yu. Maximal subspace coregulated gene clustering. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):83–98, 2008.

P. Zhou, Y. Hou, and J. Feng. Deep adversarial subspace clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1596–1604, 2018.

X. F. Zhu, Zhang S. C., Y. G. Li, Zhang J. L., and Yang L. F. Low-rank sparse subspace for spectral clustering. *IEEE Transactions on Knowledge and Data Engineering*, 31(8): 1532–1543, 2018.