TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 08/09 pp855–868
DOI: 10.26599/TST.2021.9010066
Volume 27, Number 5, October 2022

Graph Convolutional Network Combined with Semantic Feature Guidance for Deep Clustering

Junfen Chen, Jie Han, Xiangjie Meng, Yan Li*, and Haifeng Li

Abstract: The performances of semisupervised clustering for unlabeled data are often superior to those of unsupervised learning, which indicates that semantic information attached to clusters can significantly improve feature representation capability. In a graph convolutional network (GCN), each node contains information about itself and its neighbors that is beneficial to common and unique features among samples. Combining these findings, we propose a deep clustering method based on GCN and semantic feature guidance (GFDC) in which a deep convolutional network is used as a feature generator, and a GCN with a softmax layer performs clustering assignment. First, the diversity and amount of input information are enhanced to generate highly useful representations for downstream tasks. Subsequently, the topological graph is constructed to express the spatial relationship of features. For a pair of datasets, feature correspondence constraints are used to regularize clustering loss, and clustering outputs are iteratively optimized. Three external evaluation indicators, i.e., clustering accuracy, normalized mutual information, and the adjusted Rand index, and an internal indicator, i.e., the Davidson-Bouldin index (DBI), are employed to evaluate clustering performances. Experimental results on eight public datasets show that the GFDC algorithm is significantly better than the majority of competitive clustering methods, i.e., its clustering accuracy is 20% higher than the best clustering method on the United States Postal Service dataset. The GFDC algorithm also has the highest accuracy on the smaller Amazon and Caltech datasets. Moreover, DBI indicates the dispersion of cluster distribution and compactness within the cluster.

Key words: self-supervised clustering; graph convolutional network; feature correspondence; semantic feature guidance; confusion matrix; evaluation indicator

1 Introduction

In image processing, clustering methods based on similarity measurement can group data into several clusters, where images with high similarity tend to be in one cluster, whereas those with low similarity form a different cluster[1]. Generally, directly clustering images without feature extraction can lead to incorrect classification because of the interference from irrelevant information. Thus, the most effective features of an image in unsupervised learning tasks (e.g., the invariability of intraclass samples, divergence of interclass samples, and robustness to noises) need to be identified. Deep learning technologies have made some remarkable achievements in machine learning and computer vision[2], which generate effective characteristics for finding relationships among images and are further conducive to downstream tasks. The

• Junfen Chen, Jie Han, and Xiangjie Meng are with the Key Laboratory of Machine Learning and Computational Intelligence of Hebei Province, the College of Mathematics and Information Science, Hebei University, Baoding 071002, China. E-mail: chenjunfen2010@126.com; lzyhj0124@163.com; mengxiangjie12138@163.com.
• Yan Li is with the School of Applied Mathematics, Beijing Normal University Zhuhai, Zhuhai 519087, China. E-mail: 39826980@qq.com.
• Haifeng Li is with the Department of Computer Teaching, Hebei University, Baoding 071002, China. E-mail: 22436423@qq.com.
∗ To whom correspondence should be addressed.
Manuscript received: 2021-07-26; revised: 2021-08-23; accepted: 2021-08-25

feature extraction method uses various methods to map data from a high-dimensional space into a low-dimensional space[3]. Initial deep clustering methods[4] mapped image data into low-dimensional embedding spaces, followed by clustering tasks.

Learning both the similarity and difference among intracluster images and completing feature representation simultaneously are a challenge for unsupervised clustering[5]. In the literature, the incorporation of some supervised information in clustering is an effective way to improve performance. As reported in Refs. [6, 7], an evident gap exists between the experimental performance of unsupervised clustering and supervised classification. Furthermore, invariant information clustering (IIC)[8], a state-of-the-art deep clustering method, yields an unsupervised clustering accuracy of 0.596 and a semisupervised clustering accuracy of 0.888 on the STL-10 dataset. Similarly, another semisupervised clustering method (i.e., FixMatch)[9] showed that semisupervised learning is a powerful method for training without a large number of labels. A small amount of labeled data can benefit the clustering objective, i.e., obtaining highly recognizable visual features that considerably improve the clustering results. Based on these findings, the label information of a dataset will be employed to train a self-supervised clustering model on another unlabeled dataset in the proposed method.

Generally, when deep convolutional networks collect the feature representations of images, the neurons of convolutional network (ConvNet) exhibit a limited expression of the topological relationship among input images. Using a graph to depict data is increasingly widespread, such as social networks[10], which can compensate for the aforementioned problem. In 2016, a graph convolutional network (GCN) that considers the topological relationship among samples was proposed for semisupervised classification[11], where a weight matrix at each layer was built to describe node degrees through a normalized adjacent matrix. The GCN also explores hidden representations to improve clustering performances. In other words, the GCN can aggregate the feature information of a node and its neighboring nodes to improve the clustering performance of the model.

Herein, the problem of deep clustering images without labels is investigated based on GCNs and semantic feature guidance. A convolutional network is used as a feature generator to perform preliminary feature generation, and a GCN with a softmax layer performs clustering assignment. Our method draws on the idea of transfer learning, which is mostly used for supervised classification problems and needs to use labels for paired datasets. Our method aims to solve unsupervised problems through self-supervised learning. End-to-end joint training is iteratively conducted until the network converges and clustering is completed. The main contributions of this study are as follows:

(1) The semantic information of a labeled dataset is exploited as an auxiliary constraint for the self-supervised clustering of an unlabeled dataset. The clustering results on the unlabeled images are effectively improved using a small number of labeled images.

(2) The similarity information and local topological relationship of neighbors are integrated using a GCN, which expands the amount of information contained in nodes to obtain the effective features beneficial to the clustering task.

(3) The correspondence losses of the global and local features are used to regularize the clustering loss to further improve the feature representation capability and reduce incorrect clustering.

2 Related Work

From clustering methods based on K-means to self-supervised models based on mutual information maximization to contrastive clustering based on contrastive learning, clustering models have been significantly improved. The performance of deep clustering has been significantly improved in recent years with the increase in network depth and loss constraints. Several representative studies of clustering history are shown in Fig. 1.

Unsupervised deep clustering methods can effectively utilize the representation capability of a deep neural network, and the traditional clustering method based on a stacked denoising autoencoder has been proposed[16]. Most existing methods are based on the assumption that the distribution of image noise is known or observable. However, real-time images do not meet this assumption[17]. For example, in 2014, Huang et al.[4] proposed a deep embedding network, which used a deep autoencoder to learn low-dimensional feature representations from raw data but separated feature extraction from clustering, i.e., whether the extracted features were effective for clustering had been ignored. In 2016, Xie et al.[13] proposed a deep embedded clustering (DEC) algorithm, which learned feature

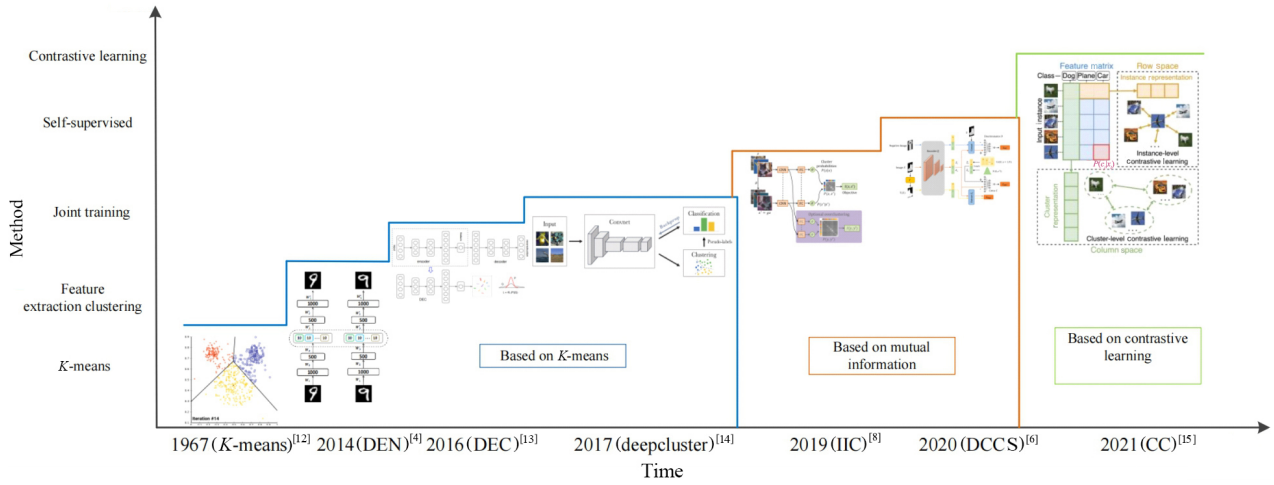


Fig. 1 Milestones in clustering. DEC indicates deep embedded clustering, DCCS indicates deep image clustering with category-style, and CC indicates contrastive learning.

representations and cluster assignments simultaneously. The framework is a deep fully-connected autoencoder so that the calculation is slow and easily overfitted. In 2017, Dizaji et al.[18] proposed a deep embedded regularized clustering (DEPICT) model, which was improved based on DEC. The DEPICT model used clean and noise data as inputs, and the network parameters were shared in two types of data. The target and prediction distributions were obtained using a convolutional denoising autoencoder. Therefore, the clustering loss function Kullback-Leibler (KL) forced the model to have invariant features for noise. The DEPICT model also used the softmax function to calculate the similarity between the features and cluster center points obtained using the clustering layer and added the clustering loss as a regularization term to the loss function. In 2018, Li et al.[19] proposed a unified clustering method called discriminatively boosted image clustering (DBC), which transformed the fully-connected autoencoder in the DEC into a fully convolutional encoder-decoder network. However, the generalization performance of the DBC network was poor, and different network structures were designed for different datasets. The aforementioned methods[13, 18–21] are based on the joint clustering method of autoencoders. However, the result of autoencoder initialization will seriously affect the final joint training clustering results. Therefore, only one-stage clustering methods with joint training, such as improved deep embedded clustering (IDEC)[22], can avoid the distortion of the embedding space during fine-tuning by preserving the local structure. After pretraining, the IDEC algorithm fine-tunes the overall loss, which contains reconstruction and clustering losses. Under the premise of ensuring that the embedding space is undistorted, the running time is considerably shortened, and the clustering accuracy is improved.

The contrastive learning method proposed by Chen et al.[23] (Hinton's team) is the SimCLR[23], which has made a major breakthrough in unsupervised learning. SimCLR used two types of data augmentation for each image in a batch to form two new pictures. The distance between similar images in the potential feature space was close to each other, and the distance between different images in the potential feature space was far from each other. In 2021, Li et al.[15] applied contrastive learning to clustering tasks for the first time and proposed a contrastive clustering (CC) method called the contrastive learning model. The instance-level and cluster-level contrastive learning were conducted in the row and column spaces of the feature matrix by collecting the positive pairs and scattering the negative pairs. Although it required only batch optimization and can be applied to large-scale and online scenarios, the CC model took a long time and required 160 GPU hours (eight threads) on the STL-10 dataset.

Unsupervised representation learning implies learning to map images into semantically meaningful features without the need for manual labels, which facilitates a variety of downstream tasks. Several new methods[8, 24–26] directly learn to map images into label-level features that are used as representation features during training. For example, in 2019, Ji et al.[8] proposed the IIC algorithm for unsupervised image classification and segmentation. The IIC algorithm expanded the data through data augmentation and used random transforms to obtain a pair from each image.

The IIC algorithm is easy to implement and rigorously grounded in information theory, which implies that it effortlessly avoids degenerate solutions to which other clustering methods are susceptible. The objective is simply to maximize the mutual information between the class assignments of each pair. However, the IIC algorithm is ineffective against unbalanced datasets. In 2020, Zhao et al.[6] proposed a new method called deep image clustering with category-style representation (DCCS) to learn a category latent representation in which the category information was disentangled from the image style and directly used as the cluster assignment. The loss function of DCCS comprised three parts to constrain the potential feature representation. First, to retain the essential information of each image and learn better discriminative latent representations, this model maximized the mutual information between the data and its feature representation. Second, an augmentation invariant regularization term was introduced based on the observation that certain augmentation should not change the category of images. Finally, a major breakthrough was made in clustering with the characteristics obtained through the adversarial loss constraint encoder and discrimination algorithm. In 2021, Zhang and Qian[7] proposed an unsupervised deep hashing method for large-scale data retrieval called autoencoder-based unsupervised clustering and hashing (AUCH). AUCH can unify unsupervised clustering and retrieval tasks into a single learning model. Moreover, the method can use a deep neural network to simultaneously learn feature representations, hashing functions, and cluster assignments.

In 2009, Scarselli et al.[27] first proposed a graph neural network (GNN) and used neural networks on graph data. GNN updates the state of its nodes by exchanging information among nodes to achieve a certain stable value. GCN is one of the most well-known GNNs. In 2014, Bruna et al.[28] proposed two constructions to process graph data; one was based on the hierarchical clustering of the spatial domain, and the other was based on the spectrum of the graph Laplacian to calculate feature vectors and feature matrices for smoothing purposes while reducing parameters. In 2017, Kipf and Welling[11] proposed a semisupervised classification algorithm with a GCN, which was derived from the Fourier transformations to prove the accuracy and validity of the graph convolutional formula. In 2018, Li et al.[29] showed that, in the semisupervised learning model based on the graph, the GCN model did not exceed three layers, i.e., the deeper the graph networks, the easier it was to cause overfitting, and required a large amount of label information. Based on the aforementioned problem, they proposed both co-training and self-training approaches to train GCNs. The approaches significantly improved GCNs in learning with only a few labels and exempted them from requiring additional labels for validation. In 2019, Wang et al.[30] proposed a new method of clustering faces using GCN. Based on the local information around a person's face in the feature space, including the rich information between the node itself and its neighbors, the experimental results show that it has a good effect on complex face clustering. GCNs can simultaneously learn the feature information of a node and the surrounding related nodes, further mining the relationship between data and have a wide range of applications.

3 Proposed GFDC Method

In this study, a deep clustering method based on GCN and semantic feature guidance (GFDC) is proposed. GFDC is a novel and effective clustering method. During training, the labeled data guide the unlabeled data to decrease the number of incorrect clustering. Formally, given an unlabeled dataset $X^u=\{x_1^u, x_2^u, \ldots, x_N^u\}$ and a labeled dataset from the same field $X^l=\{x_1^l, x_2^l, \ldots, x_M^l\}$, $Y^l = \{y_1^l, y_2^l, \ldots, y_M^l\}$, $\tilde{X}^u=\{\tilde{x}_1^u, \tilde{x}_2^u, \ldots, \tilde{x}_N^u\}$ is transformed through random rotation (0° to 360°) from X^u. Deep neural networks[31] can learn more useful features than traditional handpicked features, which have a better effect on downstream tasks. To reduce the difficulty in training a model, a mature network (e.g., VGG11) is used as a feature extractor to complete preliminary feature extraction. The nonlinear map $f:\{X^u, X^l, \tilde{X}^u\}\rightarrow\{Z^u, Z^l, \tilde{Z}^u\}$ projects high-dimensional images into low-dimensional features as $Z^u=\{z_1^u, z_2^u, \ldots, z_N^u\}$, $Z^l=\{z_1^l, z_2^l, \ldots, z_M^l\}$, and $\tilde{Z}^u=\{\tilde{z}_1^u, \tilde{z}_2^u, \ldots, \tilde{z}_N^u\}$. We formulate another nonlinear map g: $\{Z^u, Z^l, \tilde{Z}^u\} \rightarrow \{\hat{Y}^u, \hat{Y}^l, \hat{\tilde{Y}}^u\}$, where $\hat{Y}^u$, $\hat{Y}^l$, and $\hat{\tilde{Y}}^u$ denote the collections of cluster assignments.

The goal of the proposed model is to predict the clustering assignment. The overall framework of the GFDC method is shown in Fig. 2.

3.1 Graph convolutional clusterer

We establish a weighted undirected graph using the labeled and unlabeled features and input them into a clusterer. The weighted undirected graph utilizes the

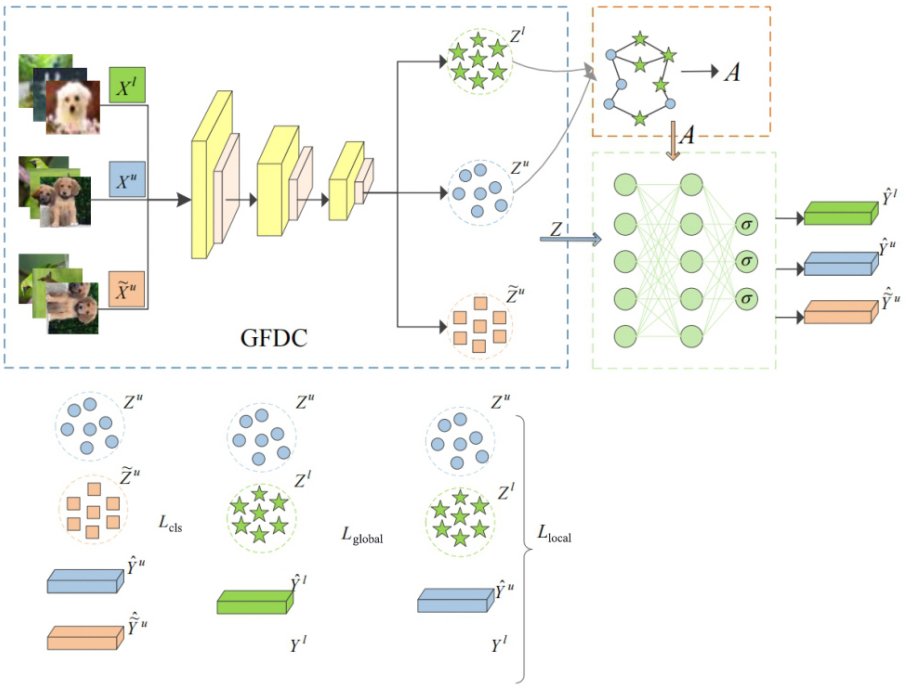


Fig. 2 Framework of the GFDC method.

information of the labeled data to guide the partitioning of the unlabeled data. Considering the features Z^u and Z^l as nodes, the similarity among nodes could be calculated. The connection between nodes is established when the similarity is greater than the threshold τ, and the weighted undirected graph denotes $G = (V, E)$. The similarity of the nodes is computed as follows:

$$s(z_h, z_r) = \exp(-\|z_h - z_r\|_2^2) \tag{1}$$

Assuming that the batch size of the unlabeled data is n_1 and that of the labeled data is n_2, we let $n = n_1 + n_2$. The adjacent matrix is expressed as follows:

$$A = (a_{ij})_{n\times n};\ a_{ij} = \begin{cases} s(z_i, z_j), & s(z_i, z_j) \geqslant \tau; \\ 0, & s(z_i, z_j) < \tau \end{cases} \tag{2}$$

where z_i and z_j are any two of the n features.

Adjacent matrix A is symmetrical and sparse. The sparseness depends on the value of parameter τ (e.g., $\tau = 0.7$), and the elements in matrix A represent the similarity among nodes. Each node is updated with the iteration optimization of the features until the network converges, and the higher the similarity is, the greater the influence of the nodes. The degree matrix D is computed by the summation of each row (or column) of the adjacency matrix, e.g., $D = \text{diag}(d_1, d_2, \ldots, d_n)$, where the diagonal element is defined as $d_i = \sum_{j=1}^{n} A_{ij}$. The feature Z and normalized adjacency matrix $\hat{A}$ are fed to a two-layer GCN. The output of the first layer is expressed as follows:

$$O = \text{ReLU}((\hat{A}Z)W_g),\ \hat{A} = D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \tag{3}$$

Because $\text{ReLU}(x) = \max(0, x)$ represents an unsaturated activation function and its derivative is 1, the network rapidly converges. W_g denotes a weight matrix of the clustering network to be trained.

The output of the clustering network is probability assignment $Q = \text{softmax}(O)$, which is calculated using the softmax function. For the input, the probability values assigned to different clusters are derived as follows:

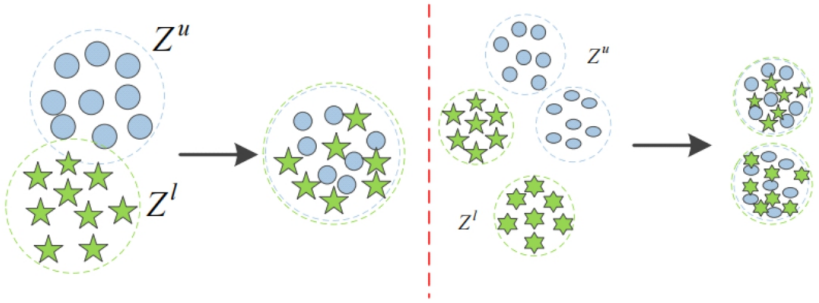
$$q_{ik} = \frac{\exp(O_{ik})}{\sum_{t=1}^{K}\exp(O_{it})} \tag{4}$$

where $O_{ik}(k = 1, 2, \ldots, K)$ is the output of the i-th sample on a clustering network, which is used to calculate q, i.e., the probability of the i-th sample assigned to class k.

3.2 Global and local feature correspondences

The global feature correspondence shown in Fig. 3a depicts the population distribution between two datasets. Conversely, the local feature correspondence shown in Fig. 3b depicts the fine-grained correspondence of cluster center features. Thus, accurate clusters with labeled data enhance the precision of clustering assignments for unlabeled data.

Given that the distributions of unlabeled and labeled



(a) Correspondence of global features (b) Correspondence of local features

Fig. 3 Global and local feature correspondences in two different datasets.

data tend to be consistent, label information guides the feature representation and cluster allocation of unlabeled data. According to Eq. (4), the probability value q_{ik}^l of the prediction label is calculated, and the embedding features are constrained so that the features generated from two similar datasets are also similar. By minimizing the global feature correspondence loss and feature similarity, network representations will be more beneficial to clustering. The global loss is expressed as follows:

$$L_{\text{global}} = -\frac{1}{n_2 K}\sum_{i=1}^{n_2}\sum_{k=1}^{K} y_{ik}^l \ln(q_{ik}^l) + \|E(Z^u) - E(Z^l)\|_2^2 \tag{5}$$

where y_{ik}^l represents the real label of the i-th sample in Category k in the labeled data.

The first part of Eq. (5) is the cross-entropy loss between the prediction label of the labeled data and its real label, and the second part of Eq. (5) is the mean squared error of the global feature correspondence loss, where $E(\cdot)$ represents the expectation value of the low-dimensional embedding features in a minibatch. Global loss can ensure that the features of two similar datasets are more similar.

However, Eq. (5) is limited to push closer to the similar global features from the two datasets. To this end, the relationship between corresponding clusters in the two datasets, which can improve the precision of local feature generation, is utilized. The local correspondence loss between identically semantic clusters in the two datasets is expressed as follows:

$$L_{\text{local}} = \gamma \frac{1}{K}\sum_{k=1}^{K}\left\| \sum_{z_i^u \in Z^u} a_{ik}^u z_i^u - \sum_{z_j^l \in Z^l} b_{jk}^l z_j^l \right\|_2^2 \tag{6}$$

where $a_{ik}^u = \dfrac{\hat{y}_{ik}^u}{\sum_{t=1}^{N}\hat{y}_{tk}^u}$, $b_{jk}^l = \dfrac{\hat{y}_{jk}^l}{\sum_{t=1}^{M}\hat{y}_{tk}^l}$, and hyperparameter $\gamma = \dfrac{2}{1+\mathrm{e}^{-\frac{10ep_i}{ep}}} - 1$; ep_i denote the i-th epoch; ep denotes the number of epochs; and γ increases with the number of iterations, preventing the binding force of the local corresponding losses from decreasing. We let $\sum_{t=1}^{N}\hat{y}_{tk}^u = N_k$, $\sum_{t=1}^{M}\hat{y}_{tk}^l = M_k$; thus, Eq. (6) can be rewritten as follows:

$$L_{\text{local}} = \frac{\gamma}{K}\sum_{k=1}^{K}\left\| \frac{1}{N_k}\sum_{z_i^u \in Z^u}\hat{y}_{ik}^u z_i^u - \frac{1}{M_k}\sum_{z_j^l \in Z^l}\hat{y}_{jk}^l z_j^l \right\|_2^2 \tag{7}$$

where $\hat{y}_i$ denotes a K-dimensional one-hot column vector when the i-th feature belongs to the k-th cluster and $\hat{y}_{ik}$ is 1; otherwise, it is 0. Equation (7) measures the L^2 distance between the k-th cluster center of the unlabeled dataset and the corresponding cluster center of the labeled dataset to align the cluster centers of the identical category of different datasets and achieves the purpose of the local feature constraint.

3.3 Clustering loss

Our GFDC method is used for self-supervised clustering tasks. When the original and augmented images using random rotation are regarded as a pair of positive samples, the GFDC method draws them to the same cluster. The low-dimensional features denoting Z^u and $\tilde{Z}^u$ are obtained using the preliminary feature extractor, and the probability distributions Q and P are predicted through the clusterer. The KL divergence measures the proximity of two probability distributions; further, the features are constrained by the contrastive loss. The clustering loss is expressed as follows:

$$L_{\text{cls}} = \frac{1}{NK}\sum_{i=1}^{N}\sum_{k}^{K} p_{ik}\ln\frac{p_{ik}}{q_{ik}} + \frac{1}{N}\left\|Z^u - \tilde{Z}^u\right\|_2^2 \tag{8}$$

where N indicates the amount of unlabeled data, and K indicates the number of clusters.

The clustering loss is constrained by the global and local feature correspondence losses. The overall loss function is expressed as follows:

$$L = L_{\text{cls}} + \lambda_1 L_{\text{global}} + \lambda_2 L_{\text{local}} \tag{9}$$

where the balance-off parameter $\lambda_i > 0$, e.g., $\lambda_1 = 1$, $\lambda_2 = 0.4$ on several datasets. We iteratively optimize the network parameters and collect the clustering results, i.e., all of the unlabeled data are clustered into K clusters through self-supervised learning. The pseudo-code of the aforementioned description is shown in Algorithm 1.

Algorithm 1 GFDC method for clustering

Inputs: Datasets X^u, X^l; training epochs E; batchsize n_1, n_2; hyperparameter τ; cluster number K, Batch B;
Outputs: Clusters assignment; clustering evaluation indicators
① //initialization
Initialize network parameters; initialize adjacency matrix to zero matrix; rotate randomly dataset $\tilde{X}^u$.
② //training
for *epoch* = 1 to E do
  for $b \in B$ do
    Step 1
    Selecting n_1 samples as χ^u from x^u; n_2 samples as χ^l from X^l;
    Mapping features $Z^u = f(x^u)$, $Z^l = f(x^l)$, $\tilde{Z}^u = f(\tilde{x}^u)$;
    Step 2
    Calculating adjacency matrix A and degree matrix D;
    computing cluster probability using Eq. (4);
    Step 3
    Computing global and local features correspondence loss, clustering loss using Eqs. (6)–(8);
    Minimizing overall loss L in Eq. (9) to update f, g network parameters;
    Saving GFDC model;
    end
  end
  //testing using GFDC model
  for x in X^u
    Extracting features by $z = f(x)$;
    Computing cluster probability using Eq. (4);
  Allocating clusters;
  Calculating clustering evaluation values
end

4 Experimental Results

This section presents the exhaustive experiments conducted to verify the effectiveness of the proposed GFDC method. The datasets are briefly described in Section 4.1. The evaluation metrics are presented in Section 4.2. The implementation details of the experiments are discussed in Section 4.3. Finally, the comparison results and ablation experiments are described in detail in Section 4.4.

4.1 Datasets

Here, we briefly present the eight image datasets generally used in clustering and classification, including Modified National Institute of Standards and Technology (MNIST)[32], United States Postal Service (USPS)[33], CIFAR-10[34], STL-10[35], and Office_caltech_10, which comprises four subdatasets, i.e., Amazon, Caltech, Dslr, and Webcam. The detailed information of the datasets is summarized in Table 1, where "/" indicates that the image size is not identical and "Remark" is the division of the training and testing sets. The Office_caltech_10 dataset is small; thus, all images are used for training and testing. To unify the network structure, all pictures are cropped to 64 × 64. Several images from the eight datasets are shown in Fig. 4.

4.2 Evaluation metrics

Three external evaluation indicators, i.e., clustering accuracy (ACC)[36], normalized mutual information (NMI)[37], and the adjusted Rand index (ARI)[38], and an internal indicator, i.e., the Davidson-Bouldin index (DBI)[39], are exploited to measure clustering performances. The DBI is calculated as follows:

$$\text{DBI} = \frac{1}{K}\sum_{i=1}^{K}\max_{j\neq i}\left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)}\right) \tag{10}$$

where K denotes the number of clusters; c_i denotes the i-th cluster center; $d(\cdot)$ denotes the distance between two clustering centers; and σ_i denotes the average distance

Table 1 Detailed information of the datasets used for the clustering evaluation.

Dataset	Number of samples	Class	Dimension	Remark
MNIST	70 000	10	28×28×1	{60 000, 10 000}
USPS	20 000	10	16×16×1	{18 000, 2000}
CIFAR-10	60 000	10	32×32×3	{50 000, 10 000}
STL-10	13 000	10	96×96×3	{5000, 8000}
Amazon	958	10	300×300×3	{958, 958}
Caltech	1123	10	/	{1123, 1123}
Dslr	157	10	1000×1000×3	{157, 157}
Webcam	195	10	/	{195, 195}

Note: "/" indicates the image size is not identical and "Remark" is the division of the training and testing sets.

Fig. 4 Several sample images from the eight datasets.

of all points in cluster i to its center. A small DBI value is equal to a small intracluster distance and a large intercluster distance, which leads to a good clustering result of the algorithm.

4.3 Implementation details

The GFDC framework contains the VGG11 and two-layer GCN. VGG11 comprises the convolution and pooling layers, which have fewer parameters and lower complexity than other deep networks. Adaptive moment estimation[40] is used as the optimizer. The learning rate is initially 1×10^{-4} and subsequently decays to 0.1 times for every 50 epochs. The batch size of unlabeled images is 240 and that of labeled images is 60. Based on the experimental results, two superparameters in Eq. (9) are set as $\lambda_1 = 1$ and $\lambda_2 = 0.4$ on the MNIST, USPS, CIFAR-10, and Caltech datasets and 0.001 on the other datasets.

The images in a minibatch comprise labeled images from the training set of one dataset, unlabeled images from the testing set of another dataset, and their augmented variants; all are used to train the GFDC model. For example, we mix the MNIST testing set as the unlabeled image data with the USPS training set as the labeled image data to learn the parameters of the GFDC model. The reported clustering performance of the GFDC model is obtained from the MNIST testing set. Similarly, we mix the USPS testing set as the unlabeled image data with the MNIST training set as the labeled image data to train the GFDC model. The clustering performance of the GFDC model is verified on the USPS testing set. The six other datasets have similar constructions.

4.4 Results and analyses

Several groups of experiments are performed to test the GFDC algorithm and compare it with other clustering models. The contribution of each part is verified through ablation experiments. Finally, the clustering results on the CIFAR-10 and STL-10 datasets are investigated exhaustively.

4.4.1 Comparison with the state-of-the-art

The comparisons of GFDC with other clustering methods are shown in Table 2, where some accuracies are obtained by running the original codes in our experimental environment; the best results are emphasized in bold. The baseline methods are described as follows:

(1) K-means[12] ;

(2) unsupervised deep embedding for clustering analysis (DEC, 2016)[13];

(3) deep clustering via joint convolutional autoencoder

Table 2 Comparison of the accuracy of GFDC with that of several baseline methods.

Method	MNIST	USPS	CIFAR-10	STL-10	Amazon	Caltech	Dslr	Webcam
K-means[12]	0.392	0.315	0.206	0.210	0.400	0.219	0.350	0.3864
DEC[13]	0.843**	0.441	0.244	0.359**	0.460	0.257	0.426	0.4540
DEPICT[18]	0.917**	0.964**	0.212	0.224	0.455	0.243	0.324	0.4130
DAC[41]	0.977**	0.653	0.521**	0.469**	0.307	0.236	0.312	0.3250
Deepcluster[14]	–	–	0.376	0.332	–	–	–	–
IIC[8]	0.992**	–	0.617**	0.596**	–	–	–	–
DCCS[6]	0.989**	–	**0.656****	0.536**	–	–	–	–
AUCH[7]	0.960**	0.775**	0.318**	**0.734****	–	–	–	–
GFDC	**0.993**	**0.974**	0.615	0.720	**0.902**	**0.833**	**0.993**	**0.9520**

Note: "**" denotes the clustering accuracy provided by a previous study, "–" denotes no value available, and the best results are emphasized in bold.

embedding and relative entropy minimization (DEPTICT, 2017)[18];

(4) deep adaptive image clustering (DAC, 2017)[41];

(5) deep clustering for the unsupervised learning of visual features (deepcluster, 2018)[14];

(6) invariant information clustering for unsupervised image classification and segmentation (IIC, 2019)[8];

(7) deep image clustering with category-style representation (DCCS, 2020)[6];

(8) autoencoder-based unsupervised clustering and hashing (AUCH, 2021)[7].

As shown in Table 2, the GFDC algorithm significantly outperforms the state-of-the-art baselines by a large margin on six datasets, except for CIFAR-10 and STL-10. Specifically, GFDC surpasses DEC in terms of ACC by 15 percentage points on MNIST, 53.3 percentage points on USPS, 37.1 percentage points on CIFAR-10, and 36.1 percentage points on STL-10. Moreover, GFDC exhibits even better clustering performances than its best competitors, i.e., IIC, DCCS, and AUCH. The better results of the GFDC algorithm on different datasets indicate that it has a stronger generalization capability than other algorithms.

On STL-10, the unsupervised clustering accuracy of IIC[8] is 0.596, whereas the semisupervised clustering accuracy of IIC is 0.888, which reduces to 0.792 with the samples of each category decreasing by 10%. Graph structure data were used in related GCN methods, with the latest unsupervised clustering method GCC[42] reaching an accuracy of 0.788 on STL-10. Based on these findings, our accuracy of 0.720 is still comparable, which implies that a small number of labels can play a vital role in guiding clustering.

4.4.2 Clustering evaluation of the GFDC algorithm

Four measurement indicators are exploited to evaluate GFDC on the eight datasets. The results are shown in Table 3, where the epoch numbers of convergence are listed in the last column. Notably, the trends of ARI and NMI are consistent with ACC. However, DBI sometimes does not decrease with an increase in ACC, particularly on the four smaller datasets. For instance, ACC is 0.993 and DBI is 1.024 on the Dslr dataset, whereas ACC is 0.952 and DBI is 0.307 on the Webcam dataset, i.e., DBI does not increase but decreases. The t-SNE visualization[43] of the features extracted from the two datasets is shown in Fig. 5. Although the distinction between clusters is more distinguishable in Fig. 5a, the intracluster distance is more scattered than that shown in Fig. 5b, which explains why ACC is higher and the DBI value is larger on the Dslr dataset.

Table 3 Clustering performances of the GFDC algorithm on the eight datasets.

Dataset	ACC	ARI	NMI	DBI	Number of epochs
MNIST	0.993	0.982	0.977	0.076	44
USPS	0.974	0.944	0.938	0.230	52
CIFAR-10	0.615	0.417	0.495	0.529	34
STL-10	0.720	0.545	0.607	0.547	22
Amazon	0.902	0.794	0.812	0.557	8
Caltech	0.833	0.676	0.715	1.128	3
Dslr	0.993	0.986	0.989	1.024	3
Webcam	0.952	0.915	0.946	0.307	6

4.4.3 Ablation experiments

This section examines the influence of GCN, local feature correspondence loss, and guidance of label information on the clustering accuracy of the GFDC algorithm. The effectiveness of different components of GFDC is shown in Table 4. Moreover, Figs. 6 and 7 visualize the features of the MNIST and USPS datasets. From the Amazon and Caltech datasets, 100 images are selected randomly to visualize the clustering results of GFDC, as shown in Fig. 8. Finally, a set of experiments were performed to examine the clustering accuracy of GFDC with the decreasing proportion of labeled images in a minibatch.

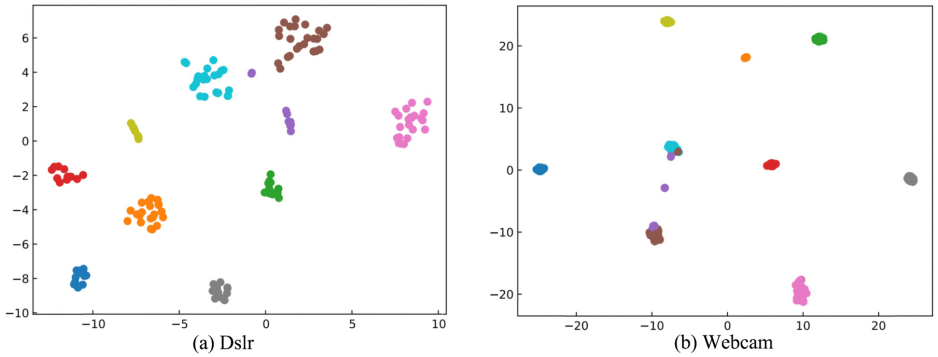
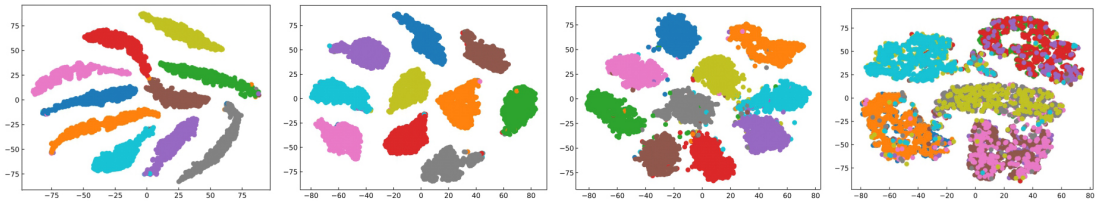


Fig. 5 *t*-SNE visualization of the Dslr and Webcam datasets, where the horizontal and vertical axes represent quantitative index without unit. The different colours are used to separate clusters.

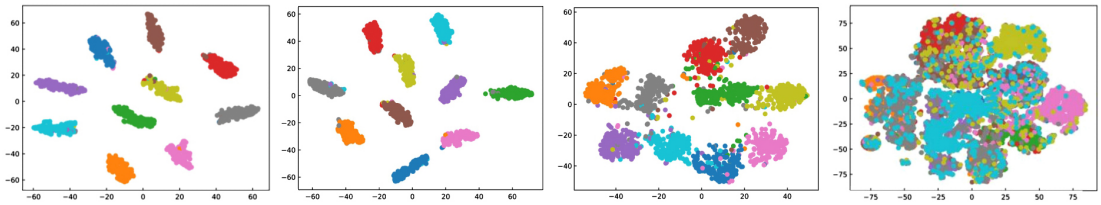
Table 4 Clustering accuracies of GFDC in ablation experiments.

Dataset	GFDC	W/o GCN	W/o local corresponding features	W/o labeled data
MNIST	**0.993**	0.992	0.715	0.456
USPS	**0.974**	0.970	0.705	0.579
CIFAR-10	**0.615**	0.600	0.518	0.416
STL-10	**0.720**	0.702	0.704	0.632
Amazon	**0.902**	0.902	0.864	0.398
Caltech	**0.833**	0.796	0.728	0.235
Dslr	**0.993**	0.955	0.974	0.350
Webcam	**0.952**	0.949	0.935	0.379



(a) (b) (c) (d)

Fig. 6 ***t*-SNE visualization of 2-dimensional features of the MNIST dataset, where the horizontal and vertical axes represent quantitative index without unit: (a) GFDC model, (b) GFDC model without GCN, (c) GFDC model without local feature correspondence, and (d) GFDC model without labeled data.**



(a) (b) (c) (d)

Fig. 7 ***t*-SNE visualization of 2-dimensional features of the USPS dataset, where the horizontal and vertical axes represent quantitative index without unit: (a) GFDC model, (b) GFDC model without GCN, (c) GFDC model without local feature correspondence, and (d) GFDC model without labeled data.**

(a) Amazon (b) Caltech

Fig. 8 Clustering results of the GFDC algorithm on the Amazon and Caltech datasets.

The GFDC model without GCN indicates that the features generated using VGG11 are directly used for clustering. Without local feature correspondence loss regularization, the total loss in Eq. (9) remains only in the first two parts. If only unlabeled data are used, Eq. (9) degenerates into clustering loss. Some observations are listed in Table 4: (1) Compared with the second column, clustering accuracies in the third column change slightly, which represents the topological information among images affecting the clustering to a certain extent. (2) Compared with the second column, clustering accuracies in the fourth column significantly decrease, e.g., the accuracy on MNIST decreases to 0.715. The local feature correspondence regularization reflects the guidance of the label information to make the features from the identically semantic clusters of the two datasets similar.

(3) Clustering accuracies in the last column are the worst.

The distribution of the features shown in Figs. 6b and 7b is quite similar to that shown in Figs. 6a and 7a. As shown in Figs. 6c and 7c, the region boundaries become blurred when removing the local feature loss constraints. The distribution of features learned cannot be distinguished well in Figs. 6d and 7d. Therefore, the GFDC method with the three parts has a good effect on feature representation, which significantly improves the clustering performance.

As shown in Fig. 8, although three clusters, such as keyboard, calculator, and laptop_computer, have button parts that are easier to group incorrectly, the majority of the results are correct, which indicates that GFDC can capture their respective unique features and further explains the effectiveness of the proposed GFDC algorithm and the robustness of the network to feature representation.

On STL-10, we set the batch size as 300 (excluding augmented images). The ratio varies from 1:1 to 1:5 between labeled and unlabeled images. The clustering accuracies are shown in Fig. 9. At a ratio of 1:1, the accuracy is the lowest at 0.71, and at a ratio of 1:4, the accuracy is the highest at 0.72. Notably, the appropriate number of labeled images can play a vital role in guiding clustering. Conversely, more label information will prompt features that are closer to the labeled data, leading to poor clustering results for unlabeled images.

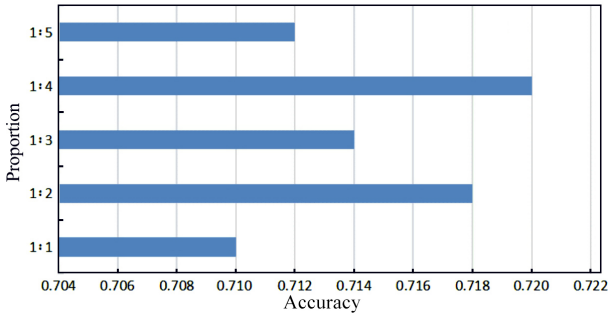


Fig. 9 Influence of clustering accuracy with varying proportions of labeled images.

4.4.4 Influence of semantic information

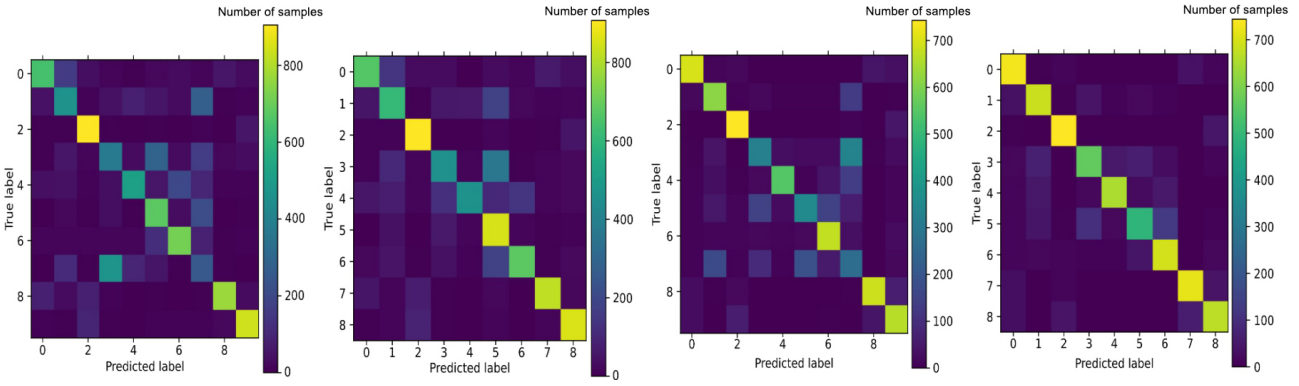
There are nine identical categories and one different category, i.e., frog on the CIFAR-10 dataset and monkey on the STL-10 dataset. By removing the different categories and their corresponding images, the CIFAR-10 and STL-10 datasets are transformed into the CIFAR-9 and STL-9 datasets. The clustering performances are shown in Table 5. According to the results obtained through GFDC, we calculate the confusion matrices shown in Fig. 10.

In Table 5, ACC is significantly improved, i.e., increased by 0.085 and 0.105, after deleting the different categories, whereas DBI slightly changes. Inconsistent label information (frog versus monkey) misleads the clustering results. Moreover, our proposed local feature correspondence loss has good applicability. The four confusion matrices shown in Fig. 10 indicate that the color of diagonal elements in Fig. 10a is slightly dark, whereas the color of diagonal elements in Fig. 10b becomes bright, i.e., the number of correctly clustered images increases. A similar phenomenon is observed in Figs. 10c and 10d.

Ten pictures per class are randomly selected from the STL-10 dataset and then 10 and 9 clusters are tested. The visualizations of the results are shown in Fig. 11. In Fig. 11a, only one monkey picture is

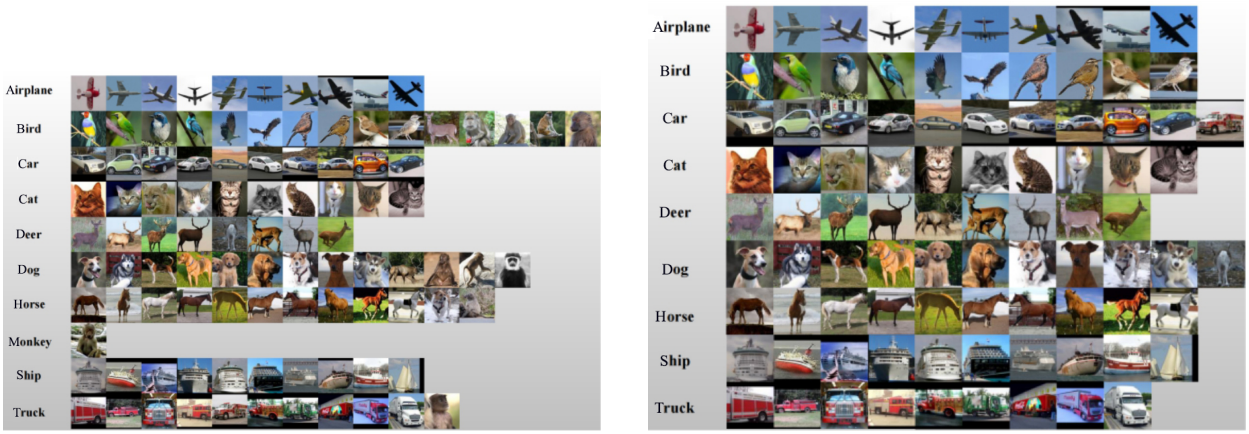
Table 5 Comparison of the clustering performances between 10 and 9 clusters.

Dataset	ACC	ARI	NMI	DBI
CIFAR-10	0.615	0.417	0.495	0.529
CIFAR-9	**0.700**	0.465	0.529	0.518
STL-10	0.720	0.545	0.607	0.547
STL-9	**0.825**	0.695	0.685	0.540



(a) CIFAR-10 (b) CIFAR-9 (c) STL-10 (d) STL-9

Fig. 10 Confusion matrices of 10 and 9 clusters on the CIFAR and STL datasets.



(a) 10 clusters (b) 9 clusters

Fig. 11 Clustering results of the GFDC method on the STL-10 dataset.

shown, whereas the nine other pictures are clustered into other animal clusters, which yields degenerate solutions. When the categories of the two datasets do not completely correspond, the clustering error will increase. At this point, the GFDC algorithm needs to be improved.

5 Conclusion

Several studies have shown that the performances of semisupervised clustering for unlabeled data considerably surpass those of unsupervised clustering, which indicates that the semantic information of clusters is important to enhance the feature representation capability of a network. Moreover, each node of the graph contains information about itself and its neighbors, which is beneficial to the common and unique features among samples. Based on these findings, a deep clustering method, i.e., GFDC, was proposed. The method introduces a part of the labeled data of different datasets in the same field to expand the diversity of input data. The method also utilizes GCN that integrates the topological information among inputs to improve the feature extraction capability of the network. The local corresponding loss constrains the identical semantic clusters of different datasets, which significantly improves the clustering results. The experimental results show that the GFDC provides better clustering performances on eight datasets and outperforms the competitive deep clustering methods involved in this study. The visualizations also illustrate that each component of the GFDC model contributes to the improvement in clustering.

However, when the semantic information of the two datasets does not entirely match, the GFDC method may yield the degenerate solutions of clustering. Future works will solve this problem by improving the feature extraction capability and exploring the independence of the network for image feature extraction.

Acknowledgment

This work was supported by the Hebei Province Introduction of Studying Abroad Talent Funded Project (No. C20200302), the Opening Fund of Hebei Key Laboratory of Machine Learning and Computational Intelligence (Nos. 2019-2021-A and ZZ201909-202109-1), the National Natural Science Foundation of China (No. 61976141), and the Social Science Foundation of Hebei Province (No. HB20TQ005).

References

[1] H. J. Zhang and I. Davidson, Deep descriptive clustering, arXiv preprint arXiv: 2105.11549, 2021.

[2] B. T. Li, D. C. Pi, Y. X. Lin, and L. Cui, DNC: A deep neural network-based clustering-oriented network embedding algorithm, *J. Netw. Comput. Appl.*, vol. 173, p. 102854, 2021.

[3] J. J. Gao, F. Z. Li, B. J. Wang, and H. L. Liang, Unsupervised nonlinear adaptive manifold learning for global and local information, *Tsinghua Science and Technology*, vol. 26. no. 2, pp. 163–171, 2021.

[4] P. H. Huang, Y. Huang, W. Wang, and L. Wang, Deep embedding network for clustering, presented at the 22nd Int. Conf. Pattern Recognition, Stockholm, Sweden, 2014, pp. 1532–1537.

[5] C. Niu and G. Wang, SPICE: Semantic pseudo-labeling for image clustering, arXiv preprint arXiv: 2103.09382, 2021.

[6] J. J. Zhao, D. H. Lu, K. Ma, Z. Zhang, and Y. F. Zheng, Deep image clustering with category-style representation, presented at the 16th European Conf. Computer Vision, Glasgow, UK, 2020, pp. 54–70.

[7] B. L. Zhang and J. B. Qian, Autoencoder-based unsupervised clustering and hashing, *Appl. Intell.*, vol. 51, no. 1, pp. 493–505, 2021.

[8] X. Ji, A. Vedaldi, and J. Henriques, Invariant information

clustering for unsupervised image classification and segmentation, presented at the 2019 IEEE/CVF Int. Conf. Computer Vision, Seoul, Republic of Korea, 2019, pp. 9864–9873.

[9] K. Sohn, D. Berthelot, C. L. Li, Z. Z. Zhang, N. Carlini, E. D. Cubuk, A. Kurakin, H. Zhang, and C. Raffel, FixMatch: Simplifying semi-supervised learning with consistency and confidence, arXiv preprint arXiv: 2001.07685, 2020.

[10] Z. H. Wu, S. R Pan, F. W. Chen, G. D. Long, C. Q. Zhang, and P. S. Yu, A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, 2021.

[11] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, in *Proc. 5th Int. Conf. Learning Representations*, Toulon, France, 2017.

[12] J. MacQueen, Some methods for classification and analysis of multivariate observations, in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, Berkeley, CA, USA, 1967, pp. 281–297.

[13] J. Y. Xie, R. Girshick, and A. Farhadi, Unsupervised deep embedding for clustering analysis, in *Proc. 33rd Int. Conf. Machine Learning*, New York, NY, USA, 2016, pp. 478–487.

[14] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, Deep clustering for unsupervised learning of visual features, presented at the 15th European Conf. Computer Vision, Munich, Germany, 2018, pp. 139–156.

[15] Y. F. Li, P. Hu, Z. T. Liu, D. Z. Peng, J. T. Zhou, and P. Xi, Contrastive clustering, presented at the 35th AAAI Conf. Artificial Intelligence, New York, USA, 2021, pp. 8547–8555.

[16] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion, *J. Mach. Learn. Res.*, vol. 11, no. 12, pp. 3371–3408, 2010.

[17] L. Zhang, J. C. Liu, F. X Shang, G. Li, J. M. Zhao, and Y. Q. Zhang, Robust segmentation method for noisy images based on an unsupervised denosing filter, *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 736–748, 2021.

[18] K. G. Dizaji, A. Herandi, C. Deng, W. D. Cai, and H. Huang, Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization, presented at the 2017 IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 5747–5756.

[19] F. F. Li, H. Qiao, and B. Zhang, Discriminatively boosted image clustering with fully convolutional auto-encoders, *Pattern Recogn.*, vol. 83, pp. 161–173, 2018.

[20] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, Deep unsupervised clustering with Gaussian mixture variational autoencoders, presented at 2017 Int. Conf. Learning Representations, Toulon, France, 2017.

[21] P. Zhou, Y. Q. Hou, and J. S. Feng, Deep adversarial subspace clustering, presented at the 2018 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 1596–1604.

[22] X. F. Guo, L. Gao, X. W. Liu, and J. P. Yin, Improved deep embedded clustering with local structure preservation, in *Proc. 26th Int. Joint Conf. Artificial Intelligence*, Melbourne, Australia, 2017, pp. 1753–1759.

[23] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, A simple framework for contrastive learning of visual representations, in *Proc. 37th Int. Conf. Machine Learning*, 2020, pp. 1597–1607.

[24] J. L. Wu, K. Y. Long, F. Wang, C. Qian, C. Li, Z. C. Lin, and H. B. Zha, Deep comprehensive correlation mining for image clustering, presented at the 2019 IEEE/CVF Int. Conf. Computer Vision, Seoul, Republic of Korea, 2019, pp. 8149–8158.

[25] J. B. Huang, S. G. Gong, and X. T. Zhu, Deep semantic clustering by partition confidence maximisation, presented at 2020 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Seattle, WA, USA, 2020, pp. 8846–8855.

[26] C. Niu, J. Zhang, G. Wang, and J. M. Liang, GATcluster: Self-supervised Gaussian-attention network for image clustering, presented at the 16th European Conf. Computer Vision, Glasgow, UK, 2020, pp. 735–751.

[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, The graph neural network model. *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2009.

[28] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, Spectral networks and locally connected networks on graph, in *Proc. 2nd Int. Conf. Learning Representations*, Banff, Canada, 2014.

[29] Q. M. Li, Z. C. Han, and X. M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in *Proc. 32nd AAAI Conf. Artificial Intelligence*, New Orleans, LA, USA, 2018, pp. 3538–3545.

[30] Z. D Wang, L. Zheng, Y. L. Li, and S. J. Wang, Linkage based face clustering via graph convolution network, presented at the 2019 IEEE/CVF Conf. Computer Vision and Pattern Recognition, Long Beach, CA, USA, 2019, pp. 1117–1125.

[31] Y. Bengio, A. Courville, and P. Vincent, Representation learning: A review and new perspectives, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, 2013.

[32] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[33] J. J. Hull, A database for handwritten text recognition research, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 5, pp. 550–554, 1994.

[34] A. Krizhevsky and G. Hinton, Learning multiple layers of features from tiny images, *Handbook of Systemic Autoimmune Diseases*, vol. 1, no. 4, pp. 1–58, 2009.

[35] A. Coates, A. Ng, and H. Lee, An analysis of single-layer networks in unsupervised feature learning, in *Proc. 14th Int. Conf. Artificial Intelligence and Statistics*, Fort Lauderdale, FL, USA, 2011, pp. 215–223.

[36] T. Li and C. Ding, The relationships among various nonnegative matrix factorization methods for clustering, presented at the 6th Int. Conf. Data Mining, Hong Kong, China, 2006, pp. 362–371.

[37] A. Strehl and J. Ghosh, Cluster ensembles—A knowledge

reuse framework for combining multiple partitions, *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, 2003.

[38] L. Hubert and P. Arabie, Comparing partitions. *J. Classif.*, vol. 2, no. 1, pp. 193–218, 1985.

[39] A. Fahad, N. Alshatri, Z. Tari, A. Alamri, I. Khalil, A. Y. Zomaya, S. Foufou, and A. Bouras, A survey of clustering algorithms for big data: Taxonomy and empirical analysis, *IEEE Trans. Emerg. Top. Comput.*, vol. 2, no. 3, pp. 267–279, 2014.

[40] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, in *Proc. 3rd Int. Conf. Learning Representations*, San Diego, CA, USA, 2015, pp. 1–15.

[41] J. L. Chang, L. F. Wang, G. F. Meng, S. M. Xiang, and C. H. Pan, Deep adaptive image clustering, presented at 2017 IEEE Int. Conf. Computer Vision, Venice, Italy, 2017, pp. 5880–5888.

[42] H. S. Zhong, J. L. Wu, C. Chen, J. Q. Huang, M. H. Deng, L. Q. Nie, Z. C. Lin, and X. S. Hua, Graph contrastive clustering. arXiv preprint arXiv: 2104.01429, 2021.

[43] L. van der Maaten and G. Hinton, Visualizing data using *t*-SNE, *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

Junfen Chen received the PhD degree in computer sciences from Universiti Sains Malaysia, Penang, Malaysia, in 2016. She is currently an associate professor and an MS supervisor at Hebei University, Baoding, China; and she is also a member of China Computer Federation (CCF, 77453M). Her research interests include data analysis, machine learning, and image processing.

Jie Han received the BS degree from Hebei University, Baoding, China, in 2019. She is now pursuing the MS degeree in software engineering at Hebei University, Baoding, China. Her research interests include image clustering and machine learning.

Xiangjie Meng is pursuing the BS degree in mathematics at Hebei University, Baoding, China. His research interests include software engineering and machine learning.

Yan Li received the PhD degree from Hong Kong Polytechnic University, Hong Kong, China, in 2006. She is currently a professor and an MS supervisor at Beijing Normal University Zhuhai, Zhuhai, China. Her research interests include rough sets and machine learning.

Haifeng Li received the MS degree in computer application technology from Hebei University, Baoding, China, in 2009. He is currently an associate professor and an MS supervisor at Hebei University, Baoding, China. His research interests include machine learning and natural language processing.