Neural Computing and Applications
https://doi.org/10.1007/s00521-022-07122-7

ORIGINAL ARTICLE

Dual-stream encoder neural networks with spectral constraint for clustering functional brain connectivity data

Hu Lu[1,2] · Tingting Jin[1]

Received: 3 September 2021 / Accepted: 21 February 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract
Functional brain connectivity data extracted from functional magnetic resonance imaging (fMRI), characterized by high dimensionality and nonlinear structure, has been widely used to mine the organizational structure for different brain diseases. It is difficult to achieve effective performance by directly using these data for unsupervised clustering analysis of brain diseases. To tackle this problem, in this paper, we propose a dual-stream encoder neural networks with spectral constraint framework for clustering the functional brain connectivity data. Specifically, we consider two different information while encoding the input data: (1) the information between the neighboring nodes, (2) the discriminative features, then design a spectral constraint module to guide the clustering of embedded nodes. The framework contains four modules, Graph Convolutional Encoder, Hard Assignment Optimization Network, Decoder module, and Spectral Constraint module. We train four modules jointly and implement a deep clustering network framework. We conducted experimental analysis on different public functional brain connectivity datasets for evaluating the proposed deep learning clustering model. Compared with the existing unsupervised clustering analysis methods for the brain connectivity data and related deep learning clustering methods, experiments on seven real brain connectivity datasets demonstrate the effectiveness and advantages of our proposed method. The source code is available at https://github.com/hulu88/DENs-SCC.

Keywords Functional brain connectivity data · Graph convolution network · Autoencoder · Clustering

1 Introduction

The functional brain network extracted from functional Magnetic Resonance Imaging (fMRI) [3] as a tool has been widely used in the diagnosis of brain diseases [27], becoming a popular topic in recent years. In many brain network studies [1, 5], the researchers classified the functional brain connectivity data [7, 13] from resting-state fMRI data for aiding clinical diagnostics. In actual application scenarios, many data are unlabeled. Moreover, some traditional clustering methods, such as K-means [16], and spectral clustering [19], are difficult to achieve the ideal clustering effect. Inspired by the popularity of deep learning [24], some workers have continued to explore how to combine the deep neural network with clustering for brain disease research.

Although the clustering algorithms of deep learning have made impressive signs of progress, many methods neglected the relationship between data samples [28]. In particular, Graph Convolutional Networks (GCN) [11], a powerful tool for learning good feature representations, using the topology of the data to aggregate node information, has attracted widespread attention. Some recent works have used GCN to mine the data [12, 26, 32]. We find that the research on diseases of GCN using structural information for convolution operation [8, 22] is valuable. However, GCN is highly dependent on learning a good feature representation. For the functional brain connectivity data, there is no primitive topology, and we cannot guarantee that the constructed topological structures can fully express the brain network relationship. It is difficult to

✉ Hu Lu
Luhu@ujs.edu.cn

[1] School of Computer Science and Communication Engineering, Jiangsu University, No.301 Xuefu Road, Jingkou District, Zhenjiang, Jiangsu, China

[2] Jiangsu Province Big Data Ubiquitous Perception and Intelligent Agricultural Application Engineering Research Center, Zhenjiang, Jiangsu, China

Published online: 19 March 2022

obtain a good node representation that can represent functional brain connectivity data only with the assistance of GCN.

To solve the problem of the high dimensionality and nonlinear structure of the functional brain connectivity data itself, inspired by [4, 29], we propose a new deep framework for clustering the data. Firstly, we introduce GCN to generate node-level information. However, the underlying structure has a limited ability to capture features with GCN. We propose dual-stream encoders to capture different forms features. Secondly, we introduce the autoencoder to encode input data. We propose a hard assignment network to guide the encoder network to generate the discriminative features to capture more abundant information. Moreover, the proposed network can alleviate the issue of over-smooth caused by GCN. Finally, we introduce clustering loss to guide the embedded nodes to learn more feasible representations for clustering. Therefore, we propose spectral constraint as an auxiliary module to assist the embedded nodes with self-optimization for clustering.

The significant contributions of this paper can be summarized as follows.

- We propose a new dual-stream encoder networks with the spectral constraint clustering framework for clustering functional brain connectivity data.
- We propose a hard assignment optimization module that a hard assignment network can optimize the encoder to generate discriminative nodes.
- We propose a spectral constraint module to assist the embedded nodes with self-optimization for clustering to obtain better clustering results.
- Many experiments show that the algorithm proposed in this paper performs well on seven public functional brain connectivity datasets.

2 Related work

2.1 Traditional brain network diseases study

Machine learning methods such as Support Vector Machine (SVM) [20] are used to diagnose brain diseases. However, these methods that require labels will consume many costs. To this end, many scholars have implemented various clustering methods on fMRI data. Among the traditional clustering methods, there are K-means clustering based on distance metric [16], graph-based spectral clustering [23], fuzzy clustering [2], consensus clustering [18], etc. In addition, Zhao et al. [30] discuss hierarchical clustering, Ordering Points To Identify the Clustering Structure (OPTICS), and Density Peak Clustering (DPC). However, the traditional clustering method has many problems. For example, K-means is very sensitive to outliers, and spectral clustering consumes much memory for large-dimensional data. These conventional clustering methods are difficult to achieve ideal results for high-dimensional and nonlinear structured data like functional brain connectivity data.

2.2 Deep learning and deep clustering model

Compared with traditional learning clustering methods, clustering methods based on deep learning, called deep clustering, have more advantages in processing complex data. Deep clustering is an unsupervised learning method that uses deep neural networks for clustering to learn better feature representations. The existing deep clustering can be analyzed from the following perspectives. From the perspective of the network framework, deep clustering based on autoencoders will add some mechanisms to focus on optimizing certain aspects to facilitate clustering. AAE [17] and ARGE [21] force the data to match the prior distribution based on adversarial learning. DAEGC [26] adds attention layers to learn node representation. VaDE [9] is a clustering method that generates highly realistic samples for any specified clustering under the framework of variational autoencoders. AdaGAE [15] proposed a graph autoencoder for locally preserving the structure to construct graphs adaptively for data without pre-constructed graph relations. Moreover, AGE [6] designed a nonparametric Laplacian smoothing filter to eliminate high-frequency noise. From the perspective of clustering loss, Yang et al. [29] embed data points into the feature space of their related graph Laplacian matrix for spectral clustering optimization guidance. DEC [28] learned a mapping from the data space to a lower-dimensional feature space in which it iteratively optimizes a clustering objective.

2.3 Deep learning brain network study

As we all know, the brain structure is highly complex. In recent years, many studies have used fMRI neuroimaging data to diagnose mental diseases. The method for assisting brain disease diagnosis through functional brain connectivity data has gained the attention of many scholars. It is popular to mine functional brain connectivity data based on deep learning methods to yield better information. Some methods use labeled methods [13] to assist disease classification, and some methods [10, 22] use semi-supervised learning to predict diseases. In recent years, there are some clustering of brain diseases through unsupervised deep learning. For example, Cui et al. [8] cluster brain diseases by integrating structural similarity and structural similarity, and Wang et al. [27] propose a new method to find highly nonlinear structures from brain network data used to

predict brain diseases. Fully mining the functional brain connectivity data extracted from fMRI images and designing high-performance unsupervised algorithms are significant in diagnosing brain diseases.

3 Proposed method

Functional brain connectivity data has the characteristics of high dimensionality and nonlinear structure. It is a challenging task to learn a good feature representation from the brain network. To solve this problem, we propose dual-stream encoder neural networks with spectral constraint, where the comprehensive framework is shown in Fig. 1. The overall framework comprises four modules: Graph Convolutional Encoder (GCE), Hard Assignment Optimization Network (HAON), Decoder module, and Spectral Constraint module.

3.1 Notation

In this paper, uppercase letters and lowercase letters denote matrices and vectors, respectively. The graph can be expressed as $G = < V, E, W >$, where V represents the collection of vertices, each vertex is a sample, E represents the collection of edges, and W_{ij} represents the weight between edges, which is the similarity between node i and node j. Let the trace of matrix M be tr(M), and transpose it as M^T. For every node $v_i \in V$, can be represented by the vector $x_i \in R^d$, that is, $V = [x_1, x_2 . . . x_N]^T \in R^{N \times d}$, where X is the sample feature matrix, N is the number of sample nodes, and c denotes the number of cluster classes.

3.2 The structure of the preprocessed graph

The proposed framework uses the GCE based on the structure and characteristic information of the original samples to capture the node-level information. If a structure is not provided, the method shown in Fig. 2 can artificially construct an undirected structure. In the beginning, through the KNN method: calculate the distance between the current node i and other nodes, sort according to the distance, and select the k nodes with the smallest distance from the current node i. These nodes are regarded as the neighbors of node i, and they are considered to be related. Then, a directed graph G_k will be generated. Finally, in order to get the undirected graph A, if node i is related to node j, then j and i are also considered to be related. A must satisfy the following conditions:

$$A_{ij} = A_{ji} = \begin{cases} 0, x_i \notin KNN(x_j) \ and \ x_j \notin KNN(x_i) \\ 1, x_i \in KNN(x_j) \ or \ x_j \in KNN(x_i), \end{cases} \quad (1)$$

where $KNN(\bullet)$ denotes the method of constructing a topology in Fig. 2.

In other words, when V_{ij} has a connection relationship, it is considered that V_{ji} also has a connection relationship, that is, when the node pair (v_i, v_j) has a relationship, $A_{ij} = 1$, otherwise $A_{ij} = 0$.

3.3 Dual-stream encoder networks

The proposed dual-stream encoder networks are composed of the GCE and the HAON. Specifically, the GCE can learn node-level information through Laplacian smoothing. Otherwise, the HAON includes a Stack Encoder (SE) and a hard assignment network. The hard assignment network optimizes the SE to learn more discriminative information.

GCN has become a popular graph analysis method due to its excellent performance and interpretability. To extract the structure information and node information of the graph, we use the connection relationship between the nodes to aggregate the node information based on the GCE to form a new node representation. In addition, when using additional rules for aggregation, in order to avoid the features of nodes with higher degrees will become larger and larger while nodes with lower degrees will become smaller and smaller, and the GCE normalizes A as $\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}$. The forward propagation law of GCN is:

$$G^{(l+1)} = Relu(\hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}G^{(l)}W_g^{(l)} + b_g^{(l)}), \quad l = 0, 1, 2. . . mid_1 - 2, \quad (2)$$

$$G^{(l)} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}G^{(l-1)}W_g^{(l-1)} + b_g^{(l-1)}, l = mid_1, \quad (3)$$

where $G^{(l)}$ is the representation learned by the $l - th$ layer in the GCE and $G^{(0)} = X$. A denotes the adjacency matrix, and $\hat{A} = A + I$ is the adjacency matrix with self-connection added, $\hat{D} = \sum_j \hat{A}_{ij}$ is the degree matrix of A. mid_1 is the number of network layers of graph convolution, and $W_g^{(l)}$, $b_g^{(l)}$, respectively, represent the weight and bias of the $l - th$ layer in the GCE.

The HAON includes a SE and a hard assignment network. The hard assignment network is used to instruct the SE to generate discriminative features. The autoencoder is an unsupervised neural network model, which can learn the hidden features of the input data after training. For deep

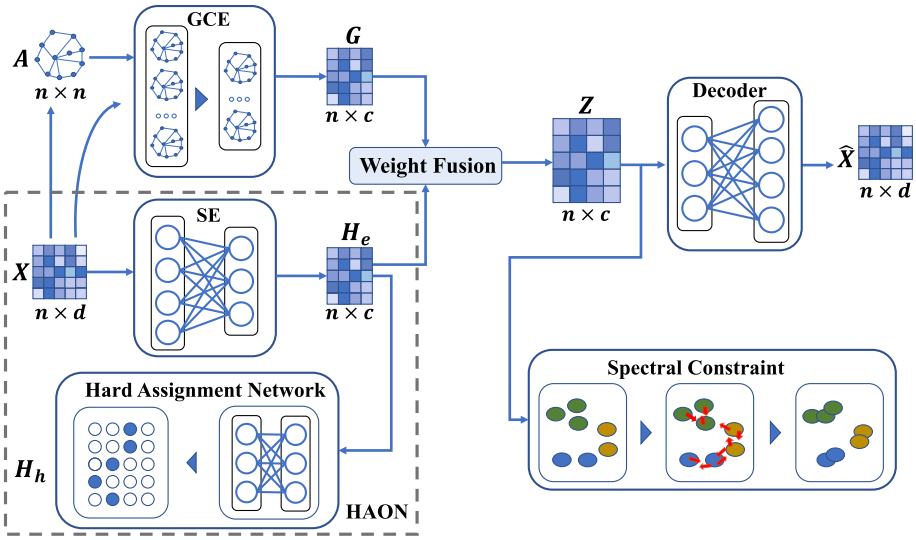

Fig. 1 The framework of DENs-SCC. The raw data X and the topology A are the input data. G and H_e are feature matrices generated by the GCE and the HAON, respectively. The embedded feature matrix Z in low-dimensional space is obtained after G and H_e are computed by weight fusion. The decoder generates $\hat{X}$ to reconstruct X. Further, the use of spectral constraint to assist the embedded nodes with self-optimization for clustering

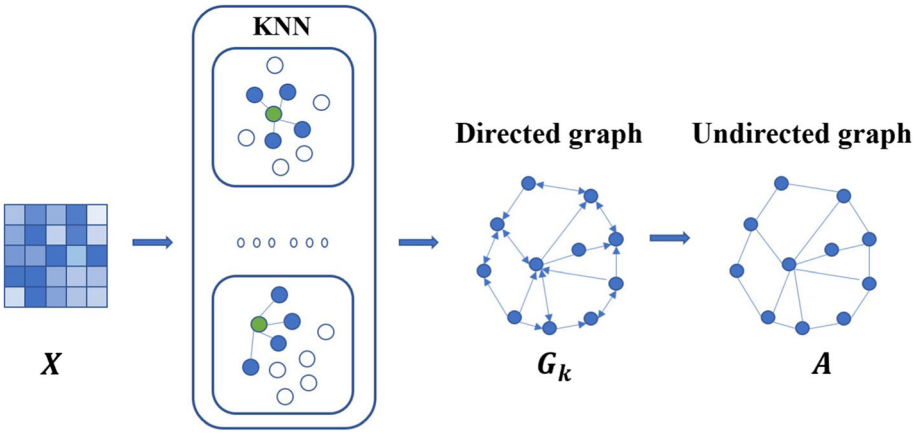

Fig. 2 A method to construct a graph. KNN denotes the K-Nearest Neighbor algorithm. X is node features, G_k represents a directed graph, and A represents an undirected graph

clustering, it is meaningful to learn effective implicit feature representations. The coding rules of the SE are:

$$H_e^{(l+1)} = Relu(W_e^{(l)} H_e^{(l)} + b_e^{(l)}), \quad l = 0, 1, 2 \ldots mid_2 - 2, \tag{4}$$

$$H_e^{(l)} = W_e^{(l-1)} H_e^{(l-1)} + b_e^{(l-1)}, \; l = mid_2 \tag{5}$$

where $H_e^{(l)}$ is the representation learned by the $l-th$ layer in the SE and $H_e^{(0)} = X$, mid_2 is the number of layers of the stack encoder, $W_e^{(l)}$, $b_e^{(l)}$, respectively, represent the weight and bias of the $l-th$ layer in the SE.

For making the SE capture the discriminative nodes, we use a hard assignment network to guide the SE optimization. Specifically, the hard assignment network is also composed of a simple neural network, and the coding rules are:

$$H_h^{(l+1)} = Relu(W_h^{(l)} H_h^{(l)} + b_h^{(l)}), \quad l = l_0, l_1 \ldots l_h - 2, \tag{6}$$

$$H_h^{(l)} = Softmax(W_h^{(l-1)} H_h^{(l-1)} + b_h^{(l-1)}), \quad l = l_h, \tag{7}$$

where $H_h^{(l)}$ is the representation learned by the $l-th$ layer in the hard assignment network and $H_h^{(l_0)} = H_e^{(mid_2)}$, l_h is the number of layers of the Hard Assignment Network, $W_h^{(l)}$, $b_h^{(l)}$, respectively, represent the weight and bias of the $l-th$ layer in the hard assignment network.

It's worth saying that the feature matrix $H_h^{(l_h)}$ generated by the hard assignment network is added with a constraint which forces the feature to attribute the probability of a specific attribute to close to 1 and the probability of other attributes to close to 0, similar to one-hot encoding, the specific form is:

$$H_h^{(l_h)} = \begin{pmatrix} p_{11} & \cdots & p_{1c} \\ \cdots & \cdots & \cdots \\ p_{N1} & \cdots & p_{Nc} \end{pmatrix}, \quad s.t. \sum_{j=1}^{c} p_{ij} = 1 \ and \ p_{ij} \in \{0, 1\}, \tag{8}$$

where $H_h^{(l_h)} \in R^{N \times c}$, p_{ij} is the $j-th$ attribute of the $i-th$ node of $H_h^{(l_h)}$.

To obtain the feature matrix Z in low-dimensional space, we perform softmax normalization on the feature matrices $G^{(mid_1)}$, $H_e^{(mid_2)}$ obtained by the dual-stream encoder networks, then use the weight fusion to map the features. We define the weight fusion as

$$\begin{aligned} Z = F_{wf}[G^{(mid_1)}, H_e^{(mid_2)}] = \lambda Softmax(G^{(mid_1)} \\ + (1-\lambda) Softmax(H_e^{(mid_2)}), \end{aligned} \tag{9}$$

where F_{wf} is the weight fusion function, and λ is the coefficient of the weight fusion function.

3.4 Decoder module

The purpose of the decoder is to preserve the features of the input layer as much as possible with the features learned by the encoder. As for now, the autoencoder has two reconstruction methods, namely reconstruction features and reconstruction topology. We reconstruct the input feature matrix X for the following reasons. On the one hand, the features are full of more information. On the other hand, the topology is only used as the underlying structure, while reconstructing A will lose lots of information.

In order to preserve the critical features of the original nodes as much as possible with the features learned by the dual-stream encoder networks, feature matrix Z is used as the input of the decoder $H_d^{(mid)}$, namely

$$H_d^{(mid)} = Z. \tag{10}$$

The input data $H_d^{(mid)}$ is reconstructed into $\hat{X}$ with the same dimension as the original sample through formulas 11 and 12:

$$H_d^{(l+1)} = Relu(W_d^{(l)} H_d^{(l)} + b_d^{(l)}), \qquad l = mid...L-2, \tag{11}$$

$$\hat{X} = H_d^{(l+1)} = W_d^{(l)} H_d^{(l)} + b_d^{(l)}, \qquad l = L-1, \tag{12}$$

where $H_d^{(l)}$ is the representation learned by the $l-th$ layer in the decoder, $W_d^{(l)}$, $b_d^{(l)}$, respectively, are the weight and bias of the $l-th$ layer in the decoder.

3.5 Spectral constraint

The spectral constraint module ensures that the points in the same manifold within a short distance should be closely mapped. The nodes have enhanced clustering characteristics in the new representation space. Specifically, the embedded feature matrix $Z \in R^{N \times c}$ is embedded into the feature space of their related graph Laplacian matrix, and the optimization process with orthogonal constraints is used for training.

Firstly, use the embedded feature matrix Z to construct a nonnegative affinity matrix, that is, the weight matrix W, which is defined as

$$W_{ij} = e^{-\frac{\|z_i - z_j\|^2}{2\sigma^2}}, \tag{13}$$

where σ is the parameter that needs to be set.

Secondly, to prevent the nodes from being grouped into the same cluster, the QR decomposition of feature matrix Z is calculated to obtain a column-orthogonal matrix $Y \in R^{N \times c}$, namely

$$Y^T Y = I_{c \times c}, \tag{14}$$

Finally, the goal is to minimize the following formula

$$argmin \sum_{i,j=1}^{N} W_{ij} (y_i - y_j)^2. \tag{15}$$

3.6 Loss function

DENs-SCC uses a total of three-loss functions for model training. In the HAON, we use the information entropy loss function. The more uniform the probability distribution of different categories, the greater the information entropy, and the less likely it is to be predicted. To optimize the SE, the hard assignment network outputs features similar to one-hot encoding to constrain the SE to generate discriminative features. The error penalty is defined as follows,

$$L_1 = -\sum_{i}^{N} \sum_{j}^{c} p_{ij} log p_{ij}. \tag{16}$$

In the decoder, to reconstruct the raw input data, we use Mean Square Error (MSE) loss. MSE refers to the mean value of the sum of squared errors of the corresponding points between the predicted data and the original data, which is defined as follows:

$$L_2 = \frac{1}{N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|_2^2 = \frac{1}{N} \|X - \hat{X}\|_F^2. \tag{17}$$

And in the spectral constraint module, we perform spectral constraint on embedded features to obtain better clustering

results. L_3 is a loss function guided by clustering. Make weakly similar pairs of nodes weakly close, and make strongly similar pairs of nodes strongly similar. When the column space of Y is the subspace of the c eigenvectors corresponding to the smallest c eigenvalues of D-W, the minimum value is obtained. Moreover, the orthogonalization constraint is performed here to prevent all points mapping to the same cluster.

Algorithm 1: DENs-SCC

input : Features: X, Topology: A, Number of clusters: c, Maximum iterations: MI, Hyperparametr: σ, learning rate: lr, weight fusion coefficient: λ;
output: Clustering results R;

Initial: $W_g^{(l)}$, $b_g^{(l)}$, $W_e^{(l)}$, $b_e^{(l)}$, $W_h^{(l)}$, $b_h^{(l)}$, $W_d^{(l)}$, $b_d^{(l)}$;
for $I = 0$ **to** MI **do**
  Generate GCE representation G ;
  Generate SE representation H_e;
  Use H_e *to generate representation* H_h *by hard assignment network*;
  Use G, H_e *to generate representation* Z *by weight fusion*;
  Given feature matrix Z, *reconstruct raw data* X *by decoder*;
  Compute L_1, L_2, L_3, *respectively*;
  Compute function loss by Eq.(19);
  Back propagation, update parameter;
Perform K-means on Z, *and return clustering result* R.

$$
\begin{aligned}
L_3 &= \frac{1}{2}\sum_{i,j=1}^{N} W_{ij}(y_i - y_j)^2 \\
&= \sum_{i=1}^{N} y_i^2 d_i - \sum_{i,j=1}^{N} y_i y_j W_{ij} \\
&= Tr(Y^T DY) - Tr(Y^T WY) \\
&= Tr(Y^T LY), \qquad s.t. Y^T Y = I,
\end{aligned}
\tag{18}
$$

where $L = D - W$, L is the Laplacian matrix, $D_{ii} = \sum_j^N W_{ij}$ denotes the degree matrix, each element W_{ij} represents the weight between two nodes, the smaller the distance between the two nodes, the more similar the nodes, and the greater the weight.

In general, the total loss function L_{loss} is defined as follows:

$$
L_{loss} = L_1 + \alpha_1 L_2 + \alpha_2 L_3, \tag{19}
$$

where α_1, α_2 are the impact factors, L_1 is the loss caused by using the hard assignment network to optimize the encoder to generate the feature nodes, L_2 is the loss caused by reconstructing the raw features, and L_3 is the loss caused by performing spectral constraint on the embedded features. The algorithm of the proposed model is shown in Algorithm 1.

4 Experiments

In this chapter, firstly, we introduce the dataset shown in Table 1 and various parameters shown in Table 2, then compare with various clustering algorithms, and finally, show our ablation experiments and discuss the model through experiments.

4.1 Datasets

This paper uses seven datasets from different brain diseases to evaluate the proposed model. It is mainly divided into four types of disease data, namely Alzheimer's Disease (AD), Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorder (ASD), Post Traumatic Stress Disorder (PTSD). The three datasets (AD, ADHD, and ASD) are from [30, 31][1]. And the other four datasets(ADIBE, ADHD_2, ADNI, and PTSD) are obtained from [13, 14][2]. The data includes clinical diagnostic labels and resting-state functional magnetic resonance imaging connectivity features obtained from individuals. Further details about the data acquisition, fMRI preprocessing and fMRI connectivity construction pipeline are presented in the research paper associated with the datasets release [14, 30].

4.2 Clustering measurement

In order to evaluate the clustering performance of the model, three performances are used: accuracy (ACC),

[1] https://github.com/xinyuzhao/identification-of-brain-based-disorders.git.

[2] https://github.com/pradlanka/malini.

Table 1 Seven different datasets are presented, N represents the number of samples, d represents the number of features, and c represents the number of clusters

Datasets	Disease	N	d	c
AD	Alzheimer's Disease	96	888	4
ADHD	Attention Deficit Hyperactivity Disorder	487	672	3
ASD	Autism Spectrum Disorder	454	1722	3
ADIBE	Autism Spectrum Disorder	988	1357	3
ADHD_2	Attention Deficit Hyperactivity Disorder	930	1179	4
ADNI	Alzheimer's Disease	132	687	4
PTSD	Post Traumatic Stress Disorder	174	677	3

adjusted rand index (ARI), F-value (F). For all performances, the higher the value means the better.

$$ACC = \max_{m} \frac{\sum_{i=1}^{N} 1\{l_i = m(c_i)\}}{N}, \tag{20}$$

where l_i and c_i are the true label and predicted cluster of data point x_i, and m ranges over all possible one-to-one mappings between clusters and labels.

TP(group 2 similar nodes into the same cluster) and TN(group 2 dissimilar nodes into different clusters) are both correct decisions. FP(group 2 dissimilar nodes into the same clusters) and FN(group 2 similar nodes into different clusters) are both wrong decisions.

$$RI = \frac{TP + TN}{C_n^2}, \tag{21}$$

$$ARI = \frac{RI - E(RI)}{max(RI) - E(RI)}, \tag{22}$$

where ARI$\in [-1, 1]$, reflecting the degree of overlap between the two divisions. The larger the value, the more consistent the clustering result is with the real situation.

$$F = 2\frac{Pre * Recall)}{Pre + Recall} = 2\frac{TP}{2TP + FP + FN}. \tag{23}$$

F is a comprehensive analysis of whether TP is large enough from a subjective (predicted) and objective (actual) perspective.

4.3 Experiment setup

In our experiment, the learning rate of ADIBE is set to 5e-4, and the learning rate of other datasets is 1e-3. We train the DENs-SCC using all data points with 200 epochs. In order to construct the topology, we need to find the K, which is the best number of neighbors. When doing weight fusion, we need to set the coefficient λ, and when doing spectral constraint, we also need to set a parameter σ. Our GCE and CE are two-layer structure networks with 128 and 256 middle dimensions, respectively. Furthermore, the dataset decides the dimension c of the GCE and CE output feature.

4.4 Performance comparison

DENs-SCC is evaluated on seven brain disease datasets and compares with the current mainstream four traditional and eight deep learning clusters. The four traditional clusters are K-means, Spectral Clustering (SC), OPTICS, and Agglomerative Clustering (Agg). Traditional clustering comes directly from the Scikit-learn library [https://scikit-learn.org/stable]. The eight deep learning clusters are graphencoder [25], GAE [12], DEC [28], SDCN [4], DAEGC [26], AGE [6], AdaGAE [15], and ARGE [21].

As shown in Table 3, the proposed clustering model performs well on brain disease datasets. The datasets of AD and ADNI have not achieved the best accuracy. After analysis, the performance of the model is relatively stable on the large-dimensional dataset. On the small dataset, problems that initialization and too little sample data will affect the judgment of the model so that sometimes cannot play a corrective role. In general, compared with traditional clustering and most deep clustering methods, DENs-SCC achieves good results.

4.5 Ablation study

4.5.1 Loss function ablation study

The loss function can be used to express the gap between the prediction and the actual data. In deep learning, the model is backpropagated by calculating the error value of the loss function to optimize the model. In order to verify the rationality of the loss function design of the model, an

Table 2 Parameter settings

	AD	ADHD	ASD	ADIBE	ADHD_2	ADNI	PTSD
K	9	29	26	39	33	18	33
λ	0.8	0.5	0.85	0.5	0.5	0.5	0.5
σ	0.4	0.4	0.2	0.8	0.8	0.1	0.1

Table 3 Performance (%) comparison with proposed methods on seven different datasets

Methods	Metric	AD	ADHD	ASD	ADIBE	ADHD_2	ANDI	PTSD
	ACC	42.71	57.7	51.76	56.58	61.51	28.79	48.28
SC	ARI	10.61	3.67	0.34	0.99	-0.08	0.64	0.56
	F	35.76	29.33	31.68	26.91	19.27	15.42	25.23
	ACC	48.96	54.83	53.52	47.77	46.99	**55.3**	70.69
k-means	ARI	17.93	16.86	6.62	3.6	7.76	15.52	**35.65**
	F	45.75	46.6	39.34	36.86	32.11	**55.9**	69.94
	ACC	31.25	55.44	56.61	56.07	61.29	26.52	48.85
OPTICS	ARI	− 0.79	− 0.41	0.3	− 0.17	− 0.39	− 0.1	0.41
	F	13.45	23.78	24.5	23.95	19	11.82	23.04
	ACC	44.79	57.49	53.96	48.68	50	45.45	59.77
Agg	ARI	15.61	17.54	8.06	0.74	8.97	12.41	15.2
	F	31.7	48.01	39.35	33.25	32.08	43.18	54.51
	ACC	41.67	52.57	49.12	46.86	48.92	39.39	42.53
graphencoder	ARI	8.22	7.17	0.15	1.67	3.26	5.05	1.63
	F	33.8	39.4	33.54	34.42	25.93	36.61	37.73
	ACC	57.29	54.62	50.88	50	48.06	53.03	56.9
GAE	ARI	32.84	16.88	6.65	2.02	7.9	**15.67**	20.76
	F	52.72	**51.26**	38.31	35.11	35.76	53.17	55.18
	ACC	44.79	55.44	56.61	48.99	61.29	41.67	69.54
DEC	ARI	13.04	− 0.62	0.3	8.17	− 0.39	6.47	39.03
	F	43.59	23.78	24.5	43.34	19	36.91	64.3
	ACC	38.54	45.79	52.2	45.65	38.17	46.97	41.95
SDCN	ARI	3.43	4.41	7	4.84	0.45	8.76	-2.08
	F	38.52	41.14	38.15	36.86	22.15	45.46	24.44
	ACC	54.17	58.52	49.78	49.78	42.04	46.97	68.97
DAEGC	ARI	33.69	17.56	10.57	10.57	4.69	14.91	32.4
	F	40.01	46.21	40.45	40.45	34.43	42.86	67.44
	ACC	53.12	58.73	51.32	52.63	53.23	47.73	64.94
AGE	ARI	29.47	**19.82**	10.34	**13.18**	13.09	14.04	26.54
	F	41.63	49.24	39.3	**43.96**	**38.59**	40.09	61.94
	ACC	31.25	55.44	56.61	57.27	61.29	26.52	48.85
AdaGAE	ARI	-0.79	− 0.41	0.3	0.99	-0.39	− 0.1	0.41
	F	13.45	23.78	24.5	26.42	19	11.82	23.04
	ACC	63.54	52.77	54.63	49.49	46.34	49.24	62.07
ARGE	ARI	40.25	15.46	9.01	1.4	10.81	11.84	20.32
	F	61.98	46.69	40.4	34.21	32.89	49.74	61.14
	ACC	**69.79**	**60.16**	**61.01**	**60.12**	**62.69**	50.76	**72.41**
DENs-SCC	ARI	**44.94**	13.26	**15.97**	8.6	**13.16**	12.74	34.03
	F	**65.38**	40.16	**44.19**	38.61	32.95	50.2	**71.02**

ablation experiment shown in Fig. 3 was carried out on the loss function.

Specifically, the performance of the combination of $J1$ and $J2$ is better than $J1$, which proves that based on reconstruction loss, using a hard assignment network to optimize SE to generate a discriminative node can complement each other with the features generated by the GCE. And the combination of $J1$, $J2$, and $J3$ is better than the combination of $J1$ and $J2$, which proves that after using the dual-stream encoder networks to capture the characteristics of node-level information and distinguishing information, applying the spectral constraint to the embedded nodes is also helpful to improve the performance of the model. Experiments prove that the designed loss function is reasonable.

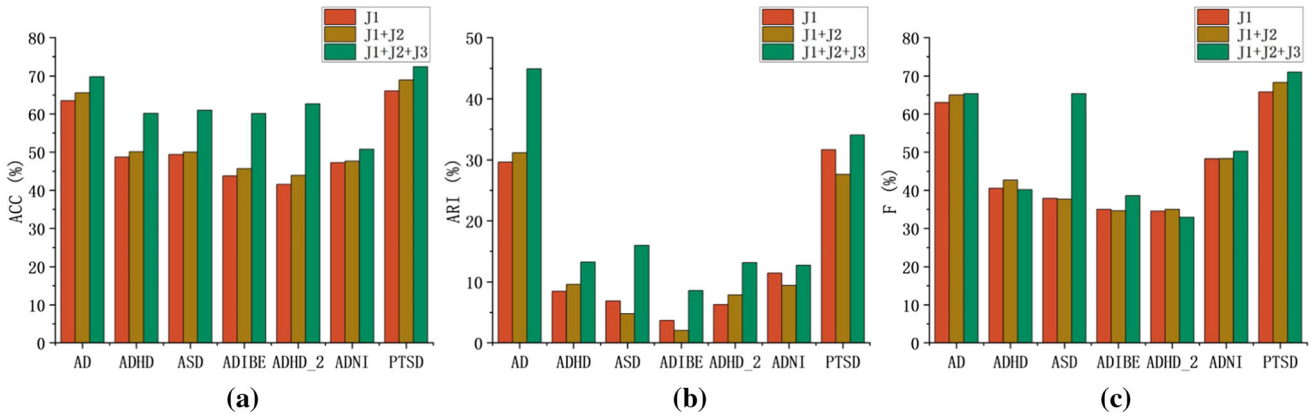

Fig. 3 Ablation study of the loss function. *J*1 represents the function of reconstructing, *J*2 represents the function of the hard assignment network, and *J*3 represents the function of spectral constraint. (**a**), (**b**), (**c**), respectively, represent the comparison of each dataset on the performance ACC, ARI, F on different combinations of loss functions

4.5.2 Model ablation study

In this part, ablation experiments are performed on the model to verify the rationality of the model design. The baseline of the model in this paper is based on AE and GCN, so the model covered by the GCE is regarded as *Model*_1, the model covered by the HAON is regarded as *Model*_2, and the overall model is regarded as *Model*.

In order to prove the feasibility of the proposed dual-stream encoder networks to capture the characteristics of node-level information and discriminative information at the same time, we conducted a model ablation experiment. The performance of *Model* in Fig.4 is better than that of *Model*_1 and *Model*_2, which proves that the features generated by the GCE and the HAON can significantly improve performance. Experiments demonstrate that the design of the model is reasonable.

4.6 Model discussion

The proposed model is built based on the GCN, which utilizes the characteristics and structure of nodes to obtain node-level representations. For making up for the loss of information caused by the GCE, the HAON is designed to capture more details. The hard assignment may be more sensitive to outliers and boundary points. Therefore, we only use the hard assignment network to optimize the SE instead of forcing the encoder's outputs directly in a hard assignment subspace. Furthermore, because the artificially constructed topology may introduce some noise and cannot capture enough information, the proposed model tries to reconstruct the raw data to learn enough information. In addition, we apply the spectral constraint to assist embedded features with self-optimization for clustering.

on the one hand, the hidden problem of GCN is maybe over-smooth, and autoencoders can alleviate this problem. On the other hand, hard assignment is more sensitive to outliers and boundary points, but the GCE mitigates this

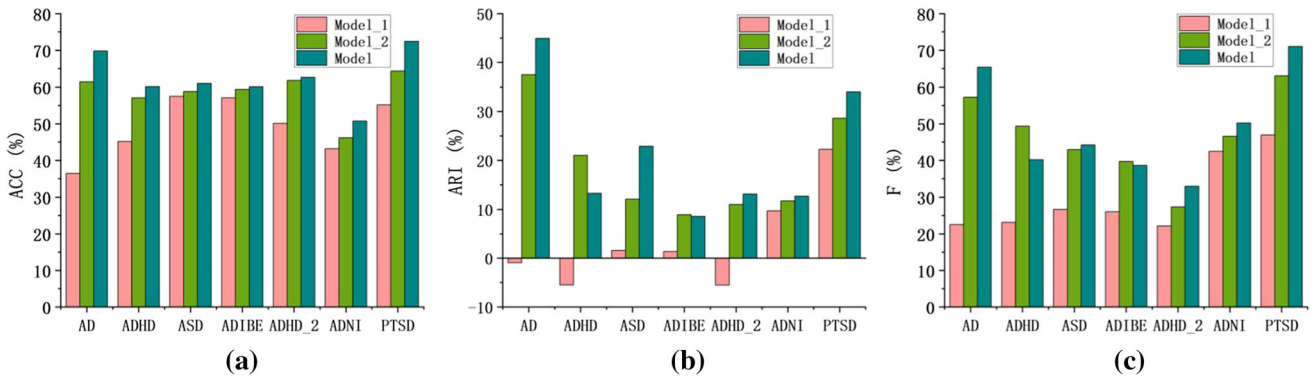

Fig. 4 Ablation study of the model. (**a**), (**b**), (**c**), respectively, represent the comparison of each dataset on the performance ACC, ARI, F on different models

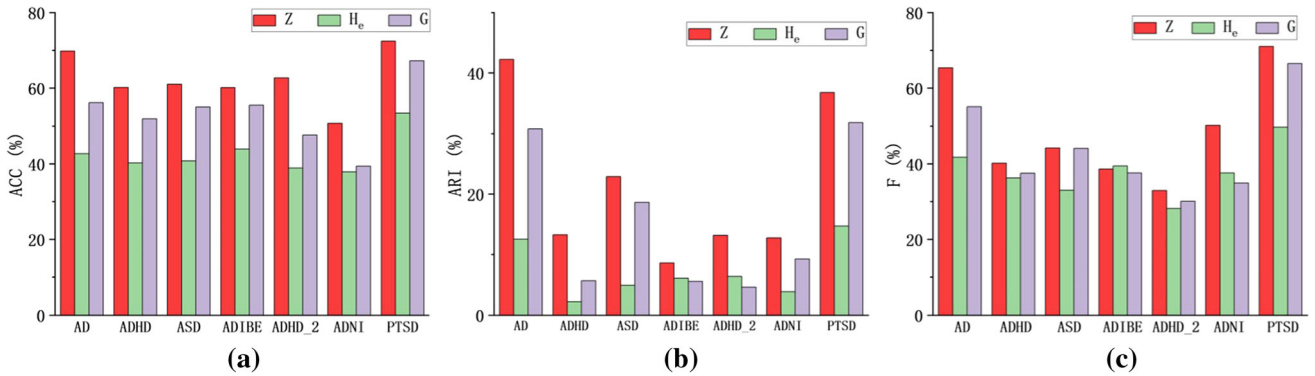

Fig. 5 Performance comparison of the three feature subspaces. G is the feature matrix captured by the *GCE*, H_e is the feature matrix captured by the HAON, Z is the feature matrix generated by embedded space. (**a**), (**b**), (**c**), respectively, represent the comparison of each dataset on the performance ACC, ARI, F on different feature spaces

effect. From a theoretical point of view, the designed dual-stream encoder networks are complementary to each other.

The proposed model uses dual-stream encoder networks to capture the node-level information G and discriminative information H_e. In the embedded feature space, G and H_e are mapped to generate embedded feature matrix Z by weight fusion. To prove that it is meaningful to use G and H_e to obtain Z, clustering the feature matrices G, H_e, and Z is performed, respectively. Figure 5 shows that the performance of embedded space is better than the other feature spaces, which proves that our idea of using dual-stream encoder networks is feasible.

5 Conclusions

This paper proposes a dual-stream encoder networks clustering framework named DENs-SCC for functional brain connectivity data. In DENs-SCC, the dual-stream encoder networks can capture the characteristics of node information and discriminative information simultaneously. The decoder reconstructs the input data to preserve the original features to a great extent. Further, the embedded nodes are self-optimized for clustering by applying the spectral constraint to make them more suitable for clustering. To evaluate our clustering framework, we conducted clustering experiments on seven real functional brain connectivity datasets. Experimental results show that DENs-SCC can find meaningful feature representations. Compared with various existing clustering methods, the performance of the proposed model on various brain diseases has achieved good results.

Acknowledgements This work is in part supported by the National Science Foundation of China (No. 62006098) and the Postgraduate Research & Practical Innovation Program of Jiangsu Province (SJCX21_1696).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Allen EA, Damaraju E, Plis SM, Erhardt EB, Eichele T, Calhoun VD (2014) Tracking whole-brain connectivity dynamics in the resting state. Cereb Cortex 24(3):663–676
2. Baumgartner R, Scarth G, Teichtmeister C, Somorjai R, Moser E (1997) Fuzzy clustering of gradient-echo functional mri in the human visual cortex. part i: reproducibility. J Magn Reson Imaging 7(6):1094–1101
3. Biswal B, Zerrin Yetkin F, Haughton VM, Hyde JS (1995) Functional connectivity in the motor cortex of resting human brain using echo-planar mri. Magn Reson Med 34(4):537–541
4. Bo D, Wang X, Shi C, Zhu M, Lu E, Cui P(2020) Structural deep clustering network. In: Proceedings of The Web Conference pp 1400–1410
5. Craddock RC, James GA, Holtzheimer PE III, Hu XP, Mayberg HS (2012) A whole brain fmri atlas generated via spatially constrained spectral clustering. Hum Brain Mapp 33(8):1914–1928
6. Cui G, Zhou J, Yang C, Liu Z Adaptive graph encoder for attributed graph embedding. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp 976–985
7. Cui X, Xiang J, Guo H, Yin G, Zhang H, Lan F, Chen J (2018) Classification of alzheimer's disease, mild cognitive impairment, and normal controls with subnetwork selection and graph kernel principal component analysis based on minimum spanning tree brain functional network. Front Comput Neurosci 12:31
8. Cui X, Xiao J, Guo H, Wang B, Li D, Niu Y, Xiang J, Chen J (2020) Clustering of brain function network based on attribute and structural information and its application in brain diseases. Front Neuroinform 13:79
9. Jiang Z, Zheng Y, Tan H, Tang B, Zhou H (2016) Variational deep embedding: a generative approach to clustering. CoRR, abs/1611.05148 **1**
10. Kazi A, Shekarforoush S, Krishna S.A, Burwinkel H, Vivar G, Kortüm K, Ahmadi S.A, Albarqouni S, Navab N (2019) Inceptiongcn: receptive field aware graph convolutional network for

disease prediction. In: International Conference on Information Processing in Medical Imaging, pp 73–85. Springer
11. Kipf TN, Welling M (2016) Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907
12. Kipf TN, Welling M (2016) Variational graph auto-encoders. arXiv preprint arXiv:1611.07308
13. Lanka P, Rangaprakash D, Dretsch MN, Katz JS, Denney TS, Deshpande G (2020) Supervised machine learning for diagnostic classification from large-scale neuroimaging datasets. Brain Imaging Behav 14(6):2378–2416
14. Lanka P, Rangaprakash D, Gotoor SSR, Dretsch MN, Katz JS, Denney TS Jr, Deshpande G (2020) Malini (machine learning in neuroimaging): a matlab toolbox for aiding clinical diagnostics using resting-state fmri data. Data Brief 29:105213
15. Li X, Zhang H, Zhang R (2020) Adaptive graph auto-encoder for general data clustering. arXiv preprint arXiv:2002.08648
16. MacQueen J, et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, vol. 1, pp 281–297. Oakland, CA, USA
17. Makhzani A, Shlens J, Jaitly N, Goodfellow I, Frey B (2015) Adversarial autoencoders. arXiv preprint arXiv:1511.05644
18. Moretti P, Muñoz MA (2013) Griffiths phases and the stretching of criticality in brain networks. Nat Commun 4(1):1–10
19. Ng A, Jordan M, Weiss Y (2001) On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 14:849–856
20. Ota K, Oishi N, Ito K, Fukuyama H, Group SJS et al (2014) A comparison of three brain atlases for mci prediction. J Neurosci Methods 221:139–150
21. Pan S, Hu R, Long G, Jiang J, Yao L, Zhang C (2018) Adversarially regularized graph autoencoder for graph embedding. arXiv preprint arXiv:1802.04407
22. Parisot S, Ktena SI, Ferrante E, Lee M, Guerrero R, Glocker B, Rueckert D (2018) Disease prediction using graph convolutional networks: application to autism spectrum disorder and alzheimer's disease. Med Image Anal 48:117–130
23. Snyder LH, Batista A, Andersen RA (1997) Coding of intention in the posterior parietal cortex. Nature 386(6621):167–170
24. Song C, Liu F, Huang Y, Wang L, Tan T (2013) Auto-encoder based data clustering. In: Iberoamerican congress on pattern recognition, pp 117–124. Springer
25. Tian F, Gao B, Cui Q, Chen E, Liu TY (2014) Learning deep representations for graph clustering. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28
26. Wang C, Pan S, Hu R, Long G, Jiang J, Zhang C (2019) Attributed graph clustering: a deep attentional embedding approach. arXiv preprint arXiv:1906.06532
27. Wang S, He L, Cao B, Lu C.T, Yu PS, Ragin AB (2017) Structural deep brain network mining. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp 475–484
28. Xie J, Girshick R, Farhadi A Unsupervised deep embedding for clustering analysis. In: International conference on machine learning, pp 478–487
29. Yang X, Deng C, Zheng F, Yan J, Liu W Deep spectral clustering using dual autoencoder network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 4066–4075
30. Zhao X, Rangaprakash D, Denney TS Jr, Katz JS, Dretsch MN, Deshpande G (2019) Identifying neuropsychiatric disorders using unsupervised clustering methods: data and code. Data Brief 22:570–573
31. Zhao X, Rangaprakash D, Yuan B, Denney TS Jr, Katz JS, Dretsch MN, Deshpande G (2018) Investigating the correspondence of clinical diagnostic grouping with underlying neurobiological and phenotypic clusters using unsupervised machine learning. Front Appl Math Stat 4:25
32. Zhuang C, Ma Q (2018) Dual graph convolutional networks for graph-based semi-supervised classification. In: Proceedings of the 2018 World Wide Web Conference, pp 499–508

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.